CRIN: Rotation-Invariant Point Cloud Analysis and Rotation Estimation via Centrifugal Reference Frame

Yujing Lou[1], Zelin Ye[1], Yang You[1],
Nianjuan Jiang[2], Jiangbo Lu[2], Weiming Wang[1], Lizhuang Ma[1*], Cewu Lu[1*]

[1] Shanghai Jiao Tong University
[2] SmartMore
{louyujing, h_e_r_o, qq456cvb, wangweiming, lucewu}@sjtu.edu.cn, ma-lz@cs.sjtu.edu.cn,
{jnianjuan, jiangbo.lu}@gmail.com

Abstract

Various recent methods attempt to implement rotation-invariant 3D deep learning by replacing the input coordinates of points with relative distances and angles. Due to the incompleteness of these low-level features, they have to undertake the expense of losing global information. In this paper, we propose the CRIN, namely Centrifugal Rotation-Invariant Network. CRIN directly takes the coordinates of points as input and transforms local points into rotation-invariant representations via centrifugal reference frames. Aided by centrifugal reference frames, each point corresponds to a discrete rotation so that the information of rotations can be implicitly stored in point features. Unfortunately, discrete points are far from describing the whole rotation space. We further introduce a continuous distribution for 3D rotations based on points. Furthermore, we propose an attention-based downsampling strategy to sample points invariant to rotations. A relation module is adopted at last for reinforcing the long-range dependencies between sampled points and predicts the anchor point for unsupervised rotation estimation. Extensive experiments show that our method achieves rotation invariance, accurately estimates the object rotation, and obtains state-of-the-art results on rotation-augmented classification and part segmentation. Ablation studies validate the effectiveness of the network design.

Introduction

Deep learning on point cloud has achieved tremendous development in recent years. Methods like PointNet(Qi et al. 2017a), PointNet++ (Qi et al. 2017b) and PointCNN (Li et al. 2018) are the pioneers for processing point cloud and obtain several achievements. However, most of these methods heavily rely on the alignment of input data. They always fail to generalize to unseen rotations, as the 3D object datasets (Chang et al. 2015; Mo et al. 2019; Yi et al. 2016) for training always keep the objects aligned canonically. Besides, a performance decline exists after applying data augmentation since rotations are impossible to enumerate. An excess of rotation augmentation also brings unbearable computation consumption. In contrast, a rotation-invariant representation of an object is much more convenient and efficient. Therefore, seeking a novel representation for 3D objects invariant to 3D rotations is necessary.

Traditional methods (Rusu, Blodow, and Beetz 2009; Tombari, Salti, and Di Stefano 2010) develop hand-crafted descriptors to represent local geometries. They utilize local geometric measurements invariant to rotations, e.g., relative distances and angles used in Point Pair Features (Drost et al. 2010; Deng, Birdal, and Ilic 2018b,a). However, they only focus on local geometries and ignore the global relationship. Spherical CNNs (Cohen et al. 2018; Esteves et al. 2018) first explore rotation-equivariant feature extraction. They discretize the 3D rotation group SO(3) on a unit sphere, extract the features for independent rotations and fuse them for global rotation-invariant features. Nevertheless, insufficient computation resources limit the resolution on the sphere, leading to an inaccurate rotation division and a considerable performance decline. Deep learning methods further explore the rotation invariance of point clouds. RINet (Zhang et al. 2019), ClusterNet(Chen et al. 2019), PR-invNet (Yu et al. 2020) and SGMNet (Xu et al. 2021) exploit various combinations of low-level geometric measurements to replace the original coordinates of input points. Though invariant to rotations, these measurements lose essential geometric relationships from the original data. Besides, some methods (Gojcic et al. 2019; Zhang et al. 2019) build local reference frames (LRFs) by the geometric center, barycenter, etc. RI-GCN (Kim, Park, and Han 2020) builds LRFs by PCA of local points to transform point clouds into rotation-invariant representations. Unfortunately, these LRFs are sensitive to the distribution and density of points, which is not robust enough. Li et al. (Li et al. 2021) achieve rotation invariance by removing the ambiguity of PCA-based canonical poses, while still requiring 24 known rotations as the prior.

To address the issues mentioned before, we propose Centrifugal Rotation-Invariant Network. Concretely, we first introduce the Centrifugal Reference Frame (CRF), a rotation-invariant representation of point clouds. A CRF is based on two polar CRFs (PCRFs). The PCRF has two essential properties. First, it turns a rotation in SO(3) to a basic rotation

*Lizhuang Ma and Cewu Lu are the corresponding authors. Cewu Lu is the member of Shanghai Jiao Tong University, China and Shanghai Qi Zhi Institute.

about the z-axis. Second, a PCRF transforms points into a representation invariant to basic rotations about the z-axis of the original space. With the help of PCRF, a 3D rotation invariance goal is firstly simplified to a single-axis rotation invariance problem. And the remaining degree of freedom can be further eliminated by applying a PCRF again, due to its basic rotation invariance. CRF relies less on local geometry than LRFs mentioned before. A CRF constructs one possible rotation-invariant representation. However, finding a global representation is challenging, as selecting one specific CRF from input points is difficult. To avoid this issue, instead of finding a global representation, we strive for local rotation invariance for local points in different CRFs.

Actually, each point corresponds to a discrete rotation, as its CRF is an orthogonal matrix in SO(3). We endeavor to cover all possible rotations so that we build a continuous distribution of rotations based on input points. Compared with the discretization of SO(3) by predefined resolutions (Cohen et al. 2018), sampling rotation from a continuous distribution is more efficient and accurate. CRIN also introduces an attention-based down-sampling strategy, which is robust to rotations. After down-sampling, a relation module is adopted to reinforce the long-range dependencies of points in feature space and predicts an anchor point for rotation estimation. We can estimate the rotation in an unsupervised manner via the CRF of the anchor point. Extensive experiments show that CRIN ensures both global and local rotation invariance and achieves state-of-the-art results on rotation-augmented classification and part segmentation tasks. Besides, CRIN can estimate the rotation of an object without rotation supervision.

Finally, the main contributions of this paper are summarized as follows:

- We propose a Centrifugal Reference Frame to represent the point cloud in a rotation-invariant way.
- We build a continuous distribution for 3D rotations based on points and introduce a rotation-invariant down-sampling strategy. CRIN can predict an anchor point for unsupervised rotation estimation.
- CRIN achieves state-of-the-art results on rotated object classification and part segmentation tasks and estimates rotations without supervision.

Related Work

Rotation Equivariance Methods With the development of group convolutions (Cohen and Welling 2016), Spherical CNNs (Cohen et al. 2018; Esteves et al. 2018) first propose rotation-equivariant feature extraction. Spherical CNNs discretize the 3D rotation group SO(3) on a sphere and propose a spherical convolution to extract features for each discrete rotation. These features are rotation-equivariant, and spherical CNNs fuse them to get the global rotation-invariant features. However, the resolution of the rotation group is too coarse to cover all rotations, which becomes the main obstacle for these methods. Based on Spherical CNNs, PRIN (You et al. 2020b) and SPRIN (You et al. 2021) make an extension to get point-wise rotation-invariant features. Equivariant Point Network (Chen et al. 2021) design a group attentive pooling to fuse equivariant features into invariant counterparts, which improves the performance. Nevertheless, these methods are still constrained by the resolution of discretization.

Rotation Invariance Methods Traditional methods design local rotation-invariant descriptors by exploiting low-level geometries. Hand-crafted (Rusu, Blodow, and Beetz 2009; Tombari, Salti, and Di Stefano 2010) and learning-based descriptors(Deng, Birdal, and Ilic 2018b; Gojcic et al. 2019; Khoury, Zhou, and Koltun 2017) both integrate local geometries into rotation-invariant features, as the local structures are invariant to the global rigid transformations. However, these descriptors lack long-range dependencies between different parts of an object. PointNet (Qi et al. 2017a), SpiderCNN (Xu et al. 2018), etc. implement a spatial transformer network (STN) to implicitly learn a transformation that aligns the input point cloud to a canonical pose. RSCNN (Liu et al. 2019), PointConv (Wu, Qi, and Fuxin 2019), KPConv (Thomas et al. 2019), etc., improve the rotation robustness with the help of STN. But STN has a drawback in that it needs numerous data augmentation to enhance the performance on rotated data. STN with data augmentation is not rigorous rotation-invariant. Besides, RI-Conv (Zhang et al. 2019), ClusterNet (Chen et al. 2019), and SGMNet (Xu et al. 2021) replace input coordinates with low-level measurements (e.g., distances, relative angles), since the coordinate is sensitive to rotations. However, these measurements lose essential geometric information, which is deficient in recovering the original data structures. PR-invNet (Yu et al. 2020) constructs a pose space with 120 known rotations to remove the PCA ambiguity and introduces a pose selector to find the canonical pose of an object. RI-GCN (Kim, Park, and Han 2020) proposes a local reference frame built by the stochastic dilated k-NN algorithm. Such an LRF is easily influenced by the sampling strategies and distribution of points. Another branch of methods (Fang et al. 2020; Spezialetti et al. 2020; Li et al. 2021) tries to recover the canonical pose of the input object before feature extraction. They all need rotation augmentation or supervision during training time.

Method

In this section, we first propose the Centrifugal Reference Frame. Then a continuous distribution is built for 3D rotations. We further propose an attention-based down-sampling method invariant to rotations and predict an anchor point to estimate the rotation of an object. Finally, we introduce the architecture of CRIN. A 2D illustration of the pipeline is demonstrated in Fig. 1.

Centrifugal Reference Frame

Our goal is to design a rotation-invariant representation $r : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ for a point cloud $\mathcal{P} = \{p_i\}_{i=1}^N$, which satisfies $r(p) = r(Rp)$ for any $p \in \mathcal{P}$ and $R \in SO(3)$. Therefore, we propose the *Centrifugal Reference Frame (CRF)* where points are invariant to 3D rotations. A CRF is based on an orthogonal basis $B \in \mathbb{R}^{3\times3}$, so that the representation is formulated as $r(p) = B^T p$. A CRF is composed of two polar CRFs (PCRFs). We first define the PCRF.

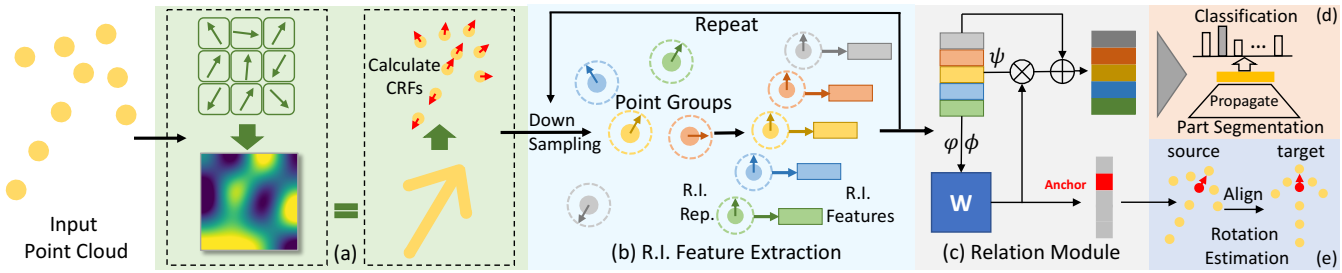

Figure 1: A 2D illustration of CRIN pipeline. (a) Build a continuous distribution for rotations and calculate the CRFs of points. (b) Down-sample and group points, then transform them into CRFs. Extract local rotation-invariant features. (c) Relation module. (d) Max-pool the global features for classification and part segmentation. (e) Rotation estimation via the anchor point.

Polar Centrifugal Reference Frame Given a query point $q \in \mathcal{P}$, a PCRF is based on an orthogonal basis, denoted as $B_p = [u, v, w] \in \mathbb{R}^{3\times3}$. Specifically, $w = \frac{q}{\|q\|}$ is defined as the centrifugal vector, $u = \frac{z\times w}{\|z\times w\|}$ and $v = w \times u$, where $[x, y, z] \in \mathbb{R}^{3\times3}$ is the basis of the original space.

The left image in Fig. 2 illustrates a PCRF. The name of PCRF is inspired by the characteristic that its v-axis is tangent to the longitude of q and always points to the "North Pole" of the sphere. Actually, a PCRF builds a representation $g(p) = B_p^T p$, with two essential properties: 1) PCRF builds a representation invariant to the basic rotation about the z-axis; 2) PCRF simplifies a SO(3) rotation in original space to a basic rotation about the z-axis. Formally, we have

$$g(R_z p) = g(p), \quad g(Rp) = R_z g(p), \tag{1}$$

where R_z is the 3×3 basic rotation matrix about the z-axis. The proof of Eq. 1 is in the appendix. The left two columns of Fig. 3 demonstrate the properties. Fig. 3(a) shows that points in the PCRF keep static when the airplane is rotated about the z-axis. In Fig. 3(b), the points revolve around the centrifugal vector in their PCRFs when the airplane is rotated by arbitrary rotations in SO(3). The inspiration for PCRF is from Euler's rotation theorem (Alperin 1989; Palais and Palais 2007; Taylor 2014) that every 3D rotation can be specified by one axis and an angle. Cohen et al. (Cohen 2013; Cohen and Welling 2014) give that an orthogonal matrix with determinant +1 can be factorized as $R = WR_zW^T$, where W is an orthogonal matrix, and R_z is a basic rotation matrix about the z-axis. The PCRF is one specific solution of factorization.

Benefiting from the properties of PCRF, the procedure of building the rotation invariance representation can be split into two steps by applying PCRF twice. We first restrict the random rotation to one degree of freedom, i.e., the basic rotation about the centrifugal vector. Then, we eliminate the variance about the remaining axis by another PCRF. So we introduce the definition of the CRF.

Centrifugal Reference Frame Given a query point $q \in \mathcal{P}$, the basis of its CRF is formulated as $B = B_p^1 B_p^2$, where B_p^1 is determined by q and B_p^2 by taking the normal vector of q as the centrifugal vector of the second PCRF, shown in the right of Fig. 2. The CRF builds a rotation-invariant representation $r(p)$, formulated as $r(Rp) = r(p)$, where $r(p) = [g^2 \circ g^1](p) = B^T p$, $g^1(p) = B_p^{1T} p$ and $g^2(p) = B_p^{2T} p$. The proof is in the appendix. Fig. 3(c) shows that the points in the CRF are invariant to rotations in the original space.

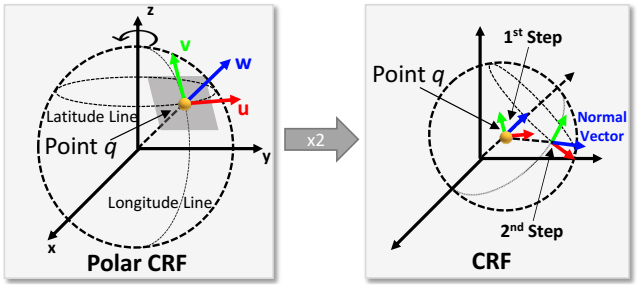

Figure 2: Illustrations of CRFs.

Some methods (Chen et al. 2019; Yu et al. 2020) replace the input of the neural network with low-level measurements (distances, angles, etc.) to make the output rotation-invariant. However, these methods lose essential geometry information from the original data. CRFs build the rotation-invariant representation only by changing the basis of the point cloud. The inherent geometries between points are completely reserved. Besides, our two-step CRF relies less on local geometries and is more robust than previously mentioned LRFs, which is compared in robustness analysis.

Each point defines a CRF to represent the point cloud in one possible rotation-invariant way. Empirically, it is hard to select one specific CRF as the canonical representation directly. To avoid finding a specific CRF representing the whole point cloud, we turn the global rotation invariance problem into a local one. We can extract pointwise rotation-invariant features through their neighborhoods in CRFs.

Continuous Distribution of Rotations

Previous methods (Chen et al. 2021; Cohen et al. 2018; You et al. 2020b) try to discretize the rotation space with grids on a sphere and extract features for each rotation. Spherical CNNs (Cohen et al. 2018) split a sphere into grids and get rotation-equivariant features at each grid coordinate for discrete rotations. However, the discretization is inaccurate. Besides, the limited resolution leads to a vast performance decline due to its high memory occupation (see Tab. 1). Ac-

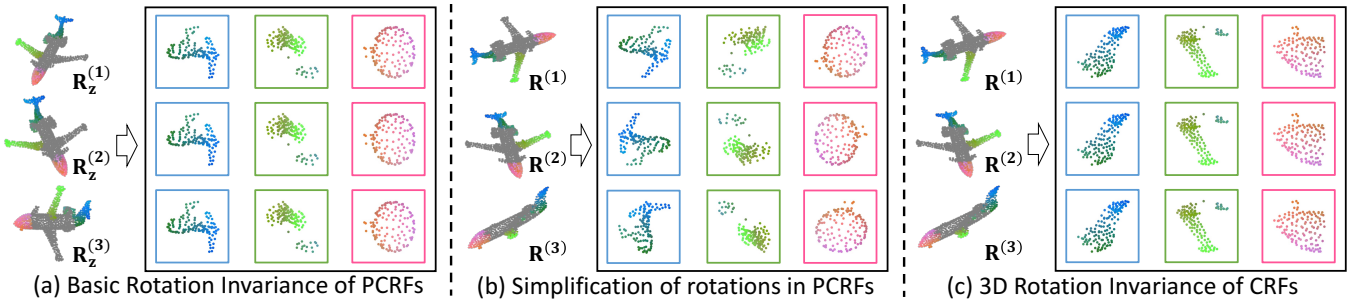

Figure 3: Illustrations of CRF properties. Projection images of three local point groups (red, green, and blue) along the w-axis in different CRFs, are given in boxes. (a) The basic rotation invariance of PCRFs. (b) Simplification from 3D rotations to basic rotations. (c) The 3D rotation invariance of CRFs.

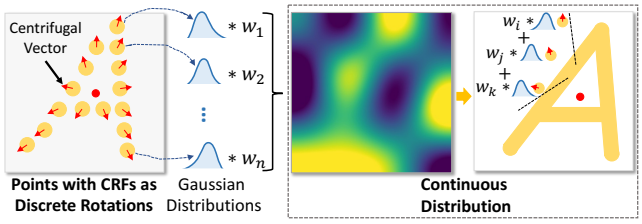

Figure 4: A 2D continuous distribution illustration.

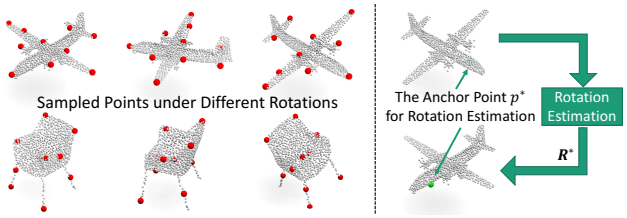

Figure 5: Left: Points sampled by our method are robust to rotations. Right: Rotation estimation with the anchor point.

tually, each CRF can be regarded as a discrete rotation since the basis of a CRF is an orthogonal matrix with the determinant $+1$. The basis is equivalent to an element in SO(3). Therefore, we map each point to a discrete rotation, as illustrated in the leftmost image of Fig. 4. We can further extract features for discrete points, i.e., rotations.

Considering finite discrete points are far from describing the rotation space thoroughly, we introduce a continuous distribution of rotations. Suppose that the point cloud constructs a continuous distribution with the probability density function $f(p), p \in \mathbb{R}^3$. We model it as a mixture distribution. For each input point $p_i \in \mathcal{P}$, we build a sub-distribution with the density function $f_i(p)$ using the three-variate Gaussian distribution, $\mathcal{N}(p_i, \Sigma_i)$. The covariance matrix is decided by the average closest distance between points. Therefore, the continuous distribution can be derived by a weighted sum, $f(p) = \sum_{i=1}^{N} w_i f_i(p)$, where w_i is a learnable weight.

Drawing point samples from the distribution is a two-step process. First, we sample a point $\hat{p}_i$ from $\mathcal{N}(p_i, \mathbf{\Sigma}_i)$ for each sub-distribution. Second, we sum all samples with learnable weights to get an element of the distribution, formulated as $\hat{p} = \sum_{i=1}^{N} w_i \hat{p}_i$. The procedure is demonstrated in Fig. 4. We build the distribution and draw point samples at each entry into CRIN. With the help of continuous distribution, we only need to sample points from the distribution instead of regressing specific rotations. In contrast to discretizing the rotation group, our continuous distribution covers all possible rotations. Besides, sampling rotations from distribution is more accurate and easier to implement.

Unsupervised Rotation Estimation

Each CRF can be used to represent one object's orientation. We want to select an anchor point invariant to rotations from the sampled points to estimate the rotation of the point cloud. The hierarchical down-sampling structure is often adopted to increase the receptive fields for 3D deep learning methods. Farthest Point Sampling (FPS) (Qi et al. 2017b) is always chosen because the sampled points are uniformly distributed. However, these points are not fixed because of the initial randomness, which is not qualified for rotation-invariant sampling. Although the FPS can be deterministic if we fix the initial searching point, the points sampled from the distribution vary each time, making FPS unstable. Therefore, we adopt an attention-based sampling strategy using FPS as the supervision to improve stability. Some keypoint datasets (You et al. 2020a; Lou et al. 2020) can be used to train a keypoint detector at the category level, which is unsuitable because their categories are limited.

Suppose the point set in layer l is $\mathcal{P}_l$. The sampled point set $\mathcal{P}_{l+1}$ is calculated by:

$$\mathcal{P}_{l+1} = softmax(\mathcal{M}(\mathcal{D})) \cdot \mathcal{P}_l, \tag{2}$$

where $\mathcal{D}$ is an $N_l \times N_l$ matrix representing distances between points in $\mathcal{P}_l$ and N_l is point number. $\mathcal{M}$ is a multi-layer perceptron (MLP) mapping $\mathcal{D}$ to N_{l+1} dimensions. The Chamfer Distance (Fan, Su, and Guibas 2017) is selected for loss calculation by measuring the similarity between $\mathcal{P}_{l+1}$ and target points $\hat{\mathcal{P}}_{l+1}$ from FPS. Fig. 5 (left) shows that points sampled in a layer by our method are stable after rotations.

After sampling points and increasing their receptive field layer-by-layer, we need to choose the anchor point at the last layer. Inspired by (Wang et al. 2018), we first develop a *relation module* to reinforce the dependencies between sampled points in the feature space, formulated as:

$$\hat{F} = W\psi(F) + F, \quad W = softmax(\varphi(F)\phi(F)^T), \quad (3)$$

where $F \in \mathbb{R}^{N_l \times C_l}$ is the feature matrix in layer l, C_l is the channel number and φ, ϕ, ψ are three MLPs. W is a weight matrix to describe the dependency between points. We average matrix W along the first dimension and get a feature map depicting the points' representativeness. The point p^* with the highest value in the feature map is selected. The basis of the anchor CRF B^* can be used to calculate the rotation. Given a point cloud with a target pose and a rotated one as the source, the estimated rotation is:

$$\mathcal{R}^* = B_t^* B_s^{*T}, \quad (4)$$

where B_t^* and B_s^* are bases of target and source objects. The key to our rotation estimation is finding the anchor point, where rotation supervision is not required. As shown in Fig. 5 (right), we can align the source point cloud with $\mathcal{R}^*$ to the target pose since p^* is invariant to rotations.

Network Architecture

Fig. 1 shows a 2D illustration of the CRIN architecture. It is mainly composed of five parts. (a) CRIN first builds the continuous distribution, samples points from the distribution, and calculates the CRFs. (b) Then we down-sample points, group local points centered at the sampled points by k-NN algorithm (Fix and Hodges 1989), and transform each group of points into corresponding CRF to get local rotation-invariant representations. Each group of points with features uses modified EdgeConv (Wang et al. 2019b) with relative angles to extract local rotation-invariant features. Part (b) is repeated twice for down-sampling and increasing the receptive fields. (c) The relation module is applied at the last layer to reinforce the relationship between points and choose the anchor point. (d) The global rotation-invariant features are fused by max-pooling. The global features are further used for classification and part segmentation with a feature propagation module used in PointNet++ (Qi et al. 2017b). (e) CRIN utilizes the anchor point to estimate the rotation.

Experiments

In this section, we evaluate CRIN on several 3D object datasets and conduct the ablation study. The robustness of our CRF is also validated. The experiments are conducted on a single GeForce RTX 2080Ti GPU and an Intel(R) Core(TM) i9-7900X @ 3.30GHz CPU.

Object Classification

We evaluate CRIN on ModelNet40 dataset (Wu et al. 2015) for object classification. We follow (Qi et al. 2017a) to split the dataset into 9843 and 2468 point clouds for training and testing, respectively. Each point cloud includes 1024 points uniformly sampled from the object face and is rescaled to fit into a unit sphere. We use *accuracy (%)* over instances as the evaluation metric. We use Adam (Kingma and Ba 2014) optimizer during training and set the initial learning rate as 0.001. The batch size is 32, with about 2 minutes per training epoch on one GPU. CRIN has 1.72M parameters.

Method	z/z	z/SO(3)	SO(3)/SO(3)
PointNet (Qi et al. 2017a)	89.2	16.4	75.5
PointNet++ (Qi et al. 2017b)	91.8	18.4	77.4
SO-Net (Li, Chen, and Lee 2018)	**92.6**	21.1	80.2
DGCNN (Wang et al. 2019b)	92.2	20.6	81.1
PointCNN (Li et al. 2018)	91.3	41.2	84.5
Spherical CNNs (Cohen et al. 2018)	88.9	76.9	86.9
PRIN (You et al. 2020b)	80.1	70.4	-
RIConv (Zhang et al. 2019)	86.5	86.4	86.4
ClusterNet (Chen et al. 2019)	87.1	87.1	87.1
EPN (Chen et al. 2021)	88.3	88.1	88.3
PR-invNet (Yu et al. 2020)	89.2	89.2	89.2
RI-GCN (Kim, Park, and Han 2020)	91.0	91.0	91.0
AECNN (Zhang et al. 2020)	91.0	91.0	91.0
SGMNet (Xu et al. 2021)	90.0	90.0	90.0
LGR-Net (Zhao et al. 2022)	90.9	90.9	91.1
Li et al. (w/o TTA) (Li et al. 2021)	90.2	90.2	90.2
Li et al. (w/ TTA) (Li et al. 2021)	91.6	91.6	91.6
CRIN (ours)	91.8	**91.8**	**91.8**

Table 1: Classification results on ModelNet40.

The evaluation is conducted in three different train/test settings: 1) training and testing with basic rotations about the z-axis (z/z); 2) training with basic rotations and testing with arbitrary rotations (z/SO(3)); 3) training and testing with arbitrary rotations (SO(3)/SO(3)).

The results are reported in Tab. 1. It shows that our CRIN ensures rotation invariance and outperforms other networks testing in SO(3). Despite networks like SO-Net performing better in the z/z setting, they hardly approach rotation invariance. Besides, CRIN also has a better performance than other rotation-invariant methods. The results of Li et al. (w/ TTA) are close because they provide 24 rotations for augmenting the poses of PCA preprocessed objects during training and testing. Thanks to our CRF and continuous distribution, CRIN reserves the original geometries and considers all rotations. Note that there is a small branch of equivariant methods (Cohen et al. 2018) using NR/AR (train: no rotation / test: arbitrary rotation) setting for evaluation. CRIN also gets the same results under NR/AR setting as in Tab. 1, as CRIN is independent of the rotation augmentation.

Object Part Segmentation

We use the ShapeNet part dataset (Yi et al. 2016) for 3D part segmentation, where 16681 point clouds from 16 categories are provided. We uniformly sample 2048 points from each object. The train/test splitting is according to (Qi et al. 2017a). The first part of the architecture for part segmentation, which is used to extract global features, is same as the classification task. We follow PointNet++ (Qi et al. 2017b) to propagate the global feature to input points via inverse distance weighted interpolation for per-point prediction. The evaluation metric is *mean IoU scores across classes (%)* (Qi et al. 2017b). The results are reported in Tab. 2 and validate that CRIN ensures local rotation invariance. The performances in part segmentation also support the statements

Method	z/z	z/SO(3)	SO(3)/SO(3)
PointNet (Qi et al. 2017a)	76.2	37.8	74.4
PointNet++ (Qi et al. 2017b)	80.7	48.2	76.7
PointCNN (Li et al. 2018)	**81.5**	34.7	71.4
DGCNN (Wang et al. 2019b)	78.8	37.4	73.3
PRIN (You et al. 2020b)	70.3	54.2	-
RIConv (Zhang et al. 2019)	75.6	75.3	75.5
PR-invNet (Yu et al. 2020)	79.4	79.4	79.4
RI-GCN (Kim, Park, and Han 2020)	-	77.2	77.3
AECNN (Zhang et al. 2020)	80.2	80.2	80.2
SGMNet (Xu et al. 2021)	79.3	79.3	79.3
LGR-Net (Zhao et al. 2022)	80.0	80.0	80.1
Li et al. (Li et al. 2021)	75.9	75.9	75.9
CRIN (ours)	80.5	**80.5**	**80.5**

Table 2: Part segmentation results on ShapeNet part dataset.

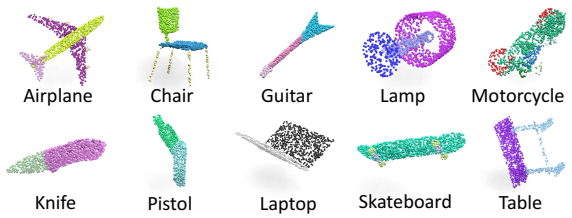

Figure 6: Visualizations of part segmentation.

in classification. Fig. 6 visualizes the part segmentation results on rotated objects.

Classification on Real-World Object

We also test CRIN on the real-world dataset ScanObjectNN (Uy et al. 2019). We follow Li et al. (Li et al. 2021), using the OBJ_BG split of ScanObjetNN, which includes 15 categories and 2890 indoor objects, where 2312 for training and 578 for testing. And z* denotes objects without any rotations. As shown in Tab. 3, CRIN also performs well on real-world objects. The performance of Li et al. (w/ TTA) is better, benefiting from 24 rotations augmentation during both training and testing. CRIN, without any rotation augmentation, even outperforms Li et al. (w/o TTA) with augmentation during training.

Object Rotation Estimation

We conduct rotation estimation on the ModelNet40. For the evaluation metric, we adopt the *Average Distance (AD)* metric used in 6D pose estimation (Xiang et al. 2018). Given an aligned point cloud $\mathcal{P}$ with N points, we randomly rotate the point cloud K times, and the rotated point clouds are denoted as $\{\hat{\mathcal{P}}_k\}_{k=1}^{K}$. Suppose the predicted rotations are $\{\hat{R}_k\}_{k=1}^{K}$. The average distance computes the mean of the pairwise distances:

$$AD = \frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\|p_i - \hat{R}_k \cdot \hat{p}_{ki}\|_2, \tag{5}$$

where $p_i \in \mathcal{P}$ and $\hat{p}_{ki} \in \hat{\mathcal{P}}_k$. The pose is considered to be correct if the average distance is smaller than 10% of the 3D object diameter, following (Xiang et al. 2018).

Method	z*/z*	z*/SO(3)	SO(3)/SO(3)
PointNet (Qi et al. 2017a)	79.4	16.7	54.7
PointNet++ (Qi et al. 2017b)	87.8	15.0	47.4
PointCNN (Li et al. 2018)	**89.9**	14.6	63.7
DGCNN (Wang et al. 2019b)	87.3	17.7	71.8
RIConv (Zhang et al. 2019)	-	78.4	78.1
LGR-Net (Zhao et al. 2022)	-	81.2	81.4
Li et al. (w/o TTA) (Li et al. 2021)	84.3	84.3	84.3
Li et al. (w/ TTA) (Li et al. 2021)	86.7	**86.7**	**86.7**
CRIN (ours)	84.7	84.7	84.7

Table 3: Classification results on ScanObjectNN.

Method	mAD ↓	Accuracy ↑
ICP (Rusinkiewicz and Levoy 2001)	0.6965	0.2
FPFH (Rusu, Blodow, and Beetz 2009)	0.0968	84.1
SHOT (Tombari, Salti, and Di Stefano 2010)	0.0699	93.6
CGF (Khoury, Zhou, and Koltun 2017)	0.0700	92.0
SpinNet (Ao et al. 2021)	**0.0672**	95.2
DenseFusion (Wang et al. 2019a)	0.0689	96.4
EPN (Chen et al. 2021)	0.0684	98.1
CRIN(ours)	0.0678	**98.9**

Table 4: Rotation estimation results on ModelNet40.

We compare CRIN with three types of methods. The first one is the Iterative Closest Point (ICP) (Rusinkiewicz and Levoy 2001) algorithm, which minimizes the similarity between point clouds iteratively. We use the implementation in Open3D (Zhou, Park, and Koltun 2018) with 2000 maximum iterations. The second type is estimating the rotation by calculating the descriptors of points and using RANSAC (Fischler and Bolles 1981) to find the correspondences between two point clouds. The third type is learning to estimate the rotation. We train the point cloud head of DenseFusion (Wang et al. 2019a) and EPN (Chen et al. 2021) with rotation supervision on ModelNet40.

The test set of ModelNet40 is used for evaluation. The rotation number K is set to 16. To validate the robustness of rotation estimation, we add the Gaussian noise with the standard deviation 0.01. The mean AD and accuracy (%) across 40 classes are listed in Tab. 4. ICP aligns objects rotated with large angles in a totally wrong orientation, as the objective of ICP is to align the object to a pose with the largest overlap with the target. Finding correspondences by descriptors and RANSAC can improve the results, while these methods consume excessive time to match two point clouds. Compared with learning methods, CRIN outperforms them even without rotation supervision. The visualization of alignment results is shown in Fig. 7. Objects in blue are in target poses and objects in yellow are randomly rotated. The alignment results are shown in the third column. Objects aligned by our CRIN almost coincide with the target objects. Therefore, we have validated that the anchor point predicted by CRIN can estimate the pose of the object.

Ablation Study

We conduct ablation studies to validate the design of CRIN. The results are shown in Tab. 5.

The baseline performs well in the z/z setting. However,

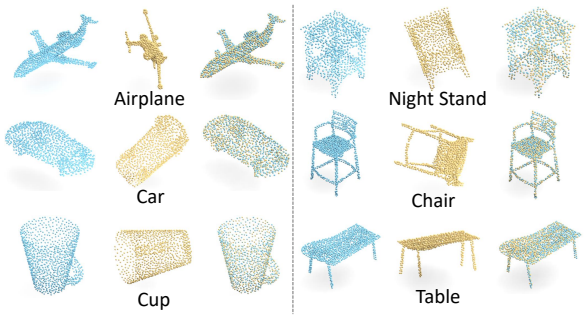

Figure 7: Visualizations of rotation estimation results.

CRF	Continuous Distribution	Relation Module	Attn.-based Sampling	z/z	z/SO(3)
-	-	-	-	90.7	16.8
polar	-	-	-	88.8	77.0
√	-	-	-	89.1	89.1
√	-	-	√	89.3	89.3
√	-	√	-	90.0	90.0
√	√	-	-	90.9	90.9
√	√	√	√	**91.8**	**91.8**

Table 5: Ablation study.

it is sensitive to rotations. The PCRF narrows the gap between results in z/z and z/SO(3), which is not true rotation invariance. It just constrains the 3D rotation to one degree of freedom. The augmentation of basic rotations about the z-axis during training compensates for the information about the remaining degree of freedom while is not good enough. Hence, a complete CRF ensures rigorous rotation invariance. Despite improving minor, the attention-based sampling ensures rotation robust sampling. The continuous distribution makes up for the deficiency of discrete rotations, improving the performance by a remarkable margin. The relation module increases the dependencies between sampled points. The improvement by the relation module is less than other modules because the receptive fields of anchor points are partially overlapped. Part of the relations has been included during the down-sampling process.

Robustness Analysis

Robustness to Point Density Our two-step representation, CRF, is robust to local structural changes. To validate it, we compare CRF with other rotation-invariant LRFs: 1) RI-GCN (Kim, Park, and Han 2020), LRF based on PCA of local points; 2) RIConv (Zhang et al. 2019), low-level features estimated by the barycenter, geometric center of local points and the origin; 3) LRF based on the covariance matrix of local points (Gojcic et al. 2019); 4) AECNN (Zhang et al. 2020), LRF estimated by centrifugal vector and the barycenter of local points. Tab. 6 lists different method results tested with the point dropout ratio from 0 to 0.9. It shows that CRIN is more independent of local points. Actually, these LRFs rely heavily on local structure. The local structure perturbation easily influences their performances. CRF utilizes two PCRFs to avoid over-reliance on local points. Besides, the processing times for a batch of point clouds are 1.74ms

Method	0	0.1	0.3	0.5	0.7	0.9	avg. ↑	var. ↓
RI-GCN	89.2	86.1	62.7	6.9	4.6	3.2	42.1	1456.2
RIConv	86.5	85.0	80.3	70.1	57.0	45.7	70.8	226.6
Cov.	89.6	88.9	88.1	88.2	86.7	71.1	85.4	41.9
AECNN	90.9	90.4	90.0	89.7	88.3	78.1	87.9	19.9
CRIN	91.8	91.4	91.4	91.0	90.6	86.3	**90.4**	**3.5**

Table 6: Robustness analysis of different LRFs.

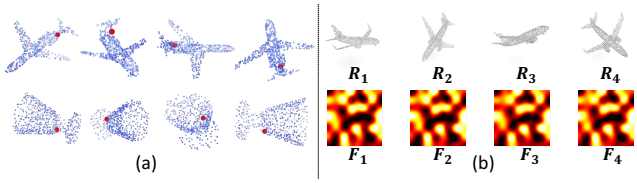

Figure 8: Visualizations of anchor points and point features.

(CRF), 8.30ms (RIConv), 7.85ms (AECNN), and 8.27ms (Cov.). CRF is more efficient, as CRF is free of grouping points or calculating centers, distances, etc.

Anchor Point for Rotation Estimation To ensure the robustness of anchor point selection, we visualize the anchor point used for rotation estimation under different rotations. In Fig. 8(a), the anchor point (red) used for rotation estimation is relatively fixed on the objects with different poses.

Rotation-Invariant Features To validate CRIN predicts per-point rotation-invariant features, we visualize point features in Fig. 8(b). We select 64 points uniformly from the point cloud and rearrange their features into an 8×8 matrix. We can see that these features almost keep same during rotation transformations.

Conclusion and Limitation

We present the CRIN for rotation invariance analysis of 3D point cloud. We introduce the centrifugal reference frame as a rotation-invariant representation of the point cloud, which reserves the input data structure without losing geometry information. To avoid localizing one specific CRF, we turn the global rotation invariance to a local one. Each point with CRF can be treated as a discrete rotation, and a continuous distribution of rotation space is further built based on points. Furthermore, CRIN utilizes a down-sampling strategy robust to rotations and a relation module to reinforce the relationship between sampled points in feature space. The relation module at the last layer predicts an anchor point for unsupervised rotation estimation. Experiments show that CRIN is qualified for rotation-invariant point cloud analysis.

In real-world applications, our CRIN is inevitably influenced by the shift of the global center due to the occlusion or the background points. A straightforward solution to this issue is applying the whole CRIN on each local patch with the local geometric center as the origin. Therefore, we can get rotation-invariant features of the local patch, regardless of the global center. The global rotation-invariant features can be fused with all local features. In addition, some center voting strategies (You et al. 2022; Lin et al. 2022) also can improve the robustness of CRIN. We will focus on rotation invariance in real-world applications in our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (72192821), National Key R&D Program of China (2021ZD0110700), SmartMore Corporation, Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, SHEITC (2018-RGZN-02046), and Shanghai Science and Technology Commission (21511101200).

References

Alperin, R. C. 1989. The Matrix of a Rotation. *The College Mathematics Journal*, 20(3): 230–230.

Ao, S.; Hu, Q.; Yang, B.; Markham, A.; and Guo, Y. 2021. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11753–11762.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4994–5002.

Chen, H.; Liu, S.; Chen, W.; Li, H.; and Hill, R. 2021. Equivariant Point Network for 3D Point Cloud Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14514–14523.

Cohen, T. 2013. *Learning Transformation Groups and their Invariants*. Ph.D. thesis, PhD thesis, University of Amsterdam.

Cohen, T.; and Welling, M. 2014. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*, 1755–1763. PMLR.

Cohen, T.; and Welling, M. 2016. Group equivariant convolutional networks. In *International conference on machine learning*, 2990–2999. PMLR.

Cohen, T. S.; Geiger, M.; Köhler, J.; and Welling, M. 2018. Spherical cnns. *arXiv preprint arXiv:1801.10130*.

Deng, H.; Birdal, T.; and Ilic, S. 2018a. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 602–618.

Deng, H.; Birdal, T.; and Ilic, S. 2018b. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 195–205.

Drost, B.; Ulrich, M.; Navab, N.; and Ilic, S. 2010. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 998–1005. Ieee.

Esteves, C.; Allen-Blanchette, C.; Makadia, A.; and Daniilidis, K. 2018. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–68.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 605–613.

Fang, J.; Zhou, D.; Song, X.; Jin, S.; Yang, R.; and Zhang, L. 2020. Rotpredictor: Unsupervised canonical viewpoint learning for point cloud classification. In *2020 International Conference on 3D Vision (3DV)*, 987–996. IEEE.

Fischler, M. A.; and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395.

Fix, E.; and Hodges, J. L. 1989. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3): 238–247.

Gojcic, Z.; Zhou, C.; Wegner, J. D.; and Wieser, A. 2019. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5545–5554.

Khoury, M.; Zhou, Q.-Y.; and Koltun, V. 2017. Learning compact geometric features. In *Proceedings of the IEEE international conference on computer vision*, 153–161.

Kim, S.; Park, J.; and Han, B. 2020. Rotation-Invariant Local-to-Global Representation Learning for 3D Point Cloud. *Advances in Neural Information Processing Systems*, 33.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, F.; Fujiwara, K.; Okura, F.; and Matsushita, Y. 2021. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16218–16227.

Li, J.; Chen, B. M.; and Lee, G. H. 2018. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9397–9406.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. PointCNN: Convolution on χ-transformed points. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 828–838.

Lin, H.; Liu, Z.; Cheang, C.; Fu, Y.; Guo, G.; and Xue, X. 2022. SAR-Net: Shape Alignment and Recovery Network for Category-Level 6D Object Pose and Size Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6707–6717.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8895–8904.

Lou, Y.; You, Y.; Li, C.; Cheng, Z.; Li, L.; Ma, L.; Wang, W.; and Lu, C. 2020. Human correspondence consensus for 3d object semantic understanding. In *European Conference on Computer Vision*, 496–512. Springer.

Mo, K.; Zhu, S.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; and Su, H. 2019. Partnet: A large-scale benchmark

for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 909–918.

Palais, B.; and Palais, R. 2007. Euler's fixed point theorem: The axis of a rotation. *Journal of fixed point theory and applications*, 2(2): 215–220.

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

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++ deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5105–5114.

Rusinkiewicz, S.; and Levoy, M. 2001. Efficient variants of the ICP algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, 145–152. IEEE.

Rusu, R. B.; Blodow, N.; and Beetz, M. 2009. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, 3212–3217. IEEE.

Spezialetti, R.; Stella, F.; Marcon, M.; Silva, L.; Salti, S.; and Di Stefano, L. 2020. Learning to orient surfaces by self-supervised spherical cnns. *Advances in Neural Information Processing Systems*, 33: 5381–5392.

Taylor, K. 2014. *Euler's Rotation Theorem: Rotating objects in 3-space*. Texas Woman's University.

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

Tombari, F.; Salti, S.; and Di Stefano, L. 2010. Unique signatures of histograms for local surface description. In *European conference on computer vision*, 356–369. Springer.

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

Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; and Savarese, S. 2019a. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3343–3352.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019b. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (TOG)*.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9621–9630.

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

Xiang, Y.; Schmidt, T.; Narayanan, V.; and Fox, D. 2018. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. *Robotics: Science and Systems (RSS)*.

Xu, J.; Tang, X.; Zhu, Y.; Sun, J.; and Pu, S. 2021. SGMNet: Learning Rotation-Invariant Point Cloud Representations via Sorted Gram Matrix. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10468–10477.

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

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6): 1–12.

You, Y.; Lou, Y.; Li, C.; Cheng, Z.; Li, L.; Ma, L.; Lu, C.; and Wang, W. 2020a. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13647–13656.

You, Y.; Lou, Y.; Liu, Q.; Tai, Y.-W.; Ma, L.; Lu, C.; and Wang, W. 2020b. Pointwise rotation-invariant network with adaptive sampling and 3d spherical voxel convolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12717–12724.

You, Y.; Lou, Y.; Shi, R.; Liu, Q.; Tai, Y.-W.; Ma, L.; Wang, W.; and Lu, C. 2021. Prin/sprin: On extracting point-wise rotation invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

You, Y.; Shi, R.; Wang, W.; and Lu, C. 2022. CPPF: Towards Robust Category-Level 9D Pose Estimation in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6866–6875.

Yu, R.; Wei, X.; Tombari, F.; and Sun, J. 2020. Deep Positional and Relational Feature Learning for Rotation-Invariant Point Cloud Analysis. In *European Conference on Computer Vision*, 217–233. Springer.

Zhang, J.; Yu, M.-Y.; Vasudevan, R.; and Johnson-Roberson, M. 2020. Learning rotation-invariant representations of point clouds using aligned edge convolutional neural networks. In *2020 International Conference on 3D Vision (3DV)*, 200–209. IEEE.

Zhang, Z.; Hua, B.-S.; Rosen, D. W.; and Yeung, S.-K. 2019. Rotation invariant convolutions for 3d point clouds deep learning. In *2019 International Conference on 3D Vision (3DV)*, 204–213. IEEE.

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. *Pattern Recognition*, 127: 108626.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*.