The Devil is in the Pose: Ambiguity-free 3D Rotation-invariant Learning via Pose-aware Convolution

Ronghan Chen[1,2,3]   Yang Cong[1,2*]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences [†]
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences

chenronghan@sia.cn, congyang81@gmail.com

Abstract

*Recent progress in introducing rotation invariance (RI) to 3D deep learning methods is mainly made by designing RI features to replace 3D coordinates as input. The key to this strategy lies in how to restore the global information that is lost by the input RI features. Most state-of-the-arts achieve this by incurring additional blocks or complex global representations, which is time-consuming and ineffective. In this paper, we real that the global information loss stems from an unexplored **pose information loss problem**, i.e., common convolution layers cannot capture the relative poses between RI features, thus hindering the global information to be hierarchically aggregated in the deep networks. To address this problem, we develop a P̲ose-a̲ware R̲otation I̲nvariant Conv̲olution (i.e., **PaRI-Conv**), which dynamically adapts its kernels based on the relative poses. Specifically, in each PaRI-Conv layer, a lightweight Augmented Point Pair Feature (APPF) is designed to fully encode the RI relative pose information. Then, we propose to synthesize a factorized dynamic kernel, which reduces the computational cost and memory burden by decomposing it into a shared basis matrix and a pose-aware diagonal matrix that can be learned from the APPF. Extensive experiments on shape classification and part segmentation tasks show that our PaRI-Conv surpasses the state-of-the-art RI methods while being more compact and efficient.*

1. Introduction

With the development of 3D scanning technology, deep learning on 3D point clouds has achieved remarkable progress in various tasks [9]. However, most methods
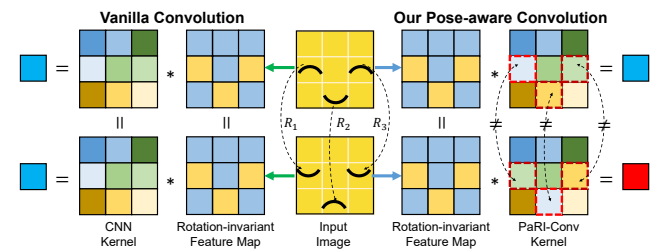
Figure 1. Illustration of the geometric ambiguity caused by the *pose information loss* problem in rotation-invariant (RI) learning. RI features naturally neglect their own pose information, and remain the same under rotations $\{R_i\}_{i=1}^3$. Thus, the **vanilla convolution** (whether on images or 3D point clouds) cannot capture the relative poses between the eyes and the mouth, leading to an ambiguous representation that fails to distinguish a smile face from an angry face. Our **Pose-aware convolution** dynamically adapts the weight based on relative pose, thus eliminating the ambiguity.

[14, 15, 35] assume a strong prior, *i.e.*, the data are pre-aligned to the same canonical pose, and the performance will degrade drastically on unaligned data, even when extensive rotation augmentation is applied. This hinders the application of current methods to real scenarios, where objects typically appear with arbitrary rotations.

To this end, most methods design *rotation-invariant* (RI) representations as input [1, 11, 13, 43], and have achieved consistent invariance to rotation. However, such RI features lose global positional information, comparing to 3D coordinates. Thus, state-of-the-art methods either design auxiliary blocks [42, 45] that directly process coordinates, or handcraft complex representations that record pair-wise relationship in a much larger neighbourhood [13, 38], which incur large computational cost while still being less competitive than rotation-sensitive methods [14, 35] on aligned data. Moreover, they fail to explain why 3D convolutional neural networks (CNNs) [15, 24, 31, 38] cannot acquire intact global information from RI features via hierarchical abstraction, as 2D CNNs can from pixels, since both pixels and RI features are lack of positional information.

In this paper, we reveal that the above issues stem from an inherent *pose information loss problem*, which has not been well exploited before. Intrinsically, being invariant to rotation also means the loss of pose information. Thus, as illustrated in Figure 1, when aggregating RI features via vanilla convolutions, the relative poses between these features are inevitably lost, leading to an ambiguous representation that fails to distinguish the smile face from the angry face. Moreover, this also explains the ineffectiveness of current methods [1, 11, 38, 44], *i.e.*, they only restore 3 degree-of-free (DoF) position rather than 6 DoF pose information.

To address the above challenges, we present Pose-aware Rotation Invariant Convolutions (*i.e.*, PaRI-Conv) to restore the pose information that is lost by the RI features. As shown in Figure 1, in a convolution, our key idea is to generate a dynamic kernel weight for each neighbour based on its relative pose to the center. In this way, the relative poses between local neighbours are fully preserved in the derived feature. Specifically, we first propose an *Augmented Point Pair Feature* (APPF) to represent the relative pose information, which extends the point pair feature (PPF) [5], and is rigorously rotation-invariant. Then the APPFs are applied to dynamically adjust convolution kernels. To reduce the expensive computational cost and memory burden introduced by storing heavy weight banks [31, 39, 40] or regressing large kernel matrix [27, 36], we propose a *factorized dynamic kernel*, which decomposes the kernel weight into a basis matrix that is shared by all neighbours and a low DoF diagonal matrix that is learned from the APPF. Since PaRI-Conv fully preserves the geometric relationship among adjacent patches by incorporating pose information, global context can be automatically acquired by simply stacking PaRI-Conv layers, thus avoiding redundant blocks [45] or complex representations [13, 38] for global information compensation, resulting in a much more compact and efficient network. Extensive experiments on shape classification and part segmentation tasks show that our method surpasses state-of-the-art RI methods, while being consistently rotation-invariant. More importantly, our method approaches or even surpasses the state-of-the-art rotation-sensitive methods on aligned data, verifying PaRI-Conv's ability in preventing information loss.

In conclusion, the main contributions of our work are:

- We reveal the pose information loss problem and address it by proposing a Pose-aware Rotation Invariant Convolution operator (*i.e.*, PaRI-Conv), leading to a more powerful and efficient solution for RI learning.

- A lightweight RI feature, named Augmented Point Pair Feature (APPF), is proposed to fully encode the pose of each neighbour relative to the center.

- To synthesize pose-aware kernel, a factorized dynamic kernel is designed by decomposing the kernel weight into a shared basis matrix and a pose-aware diagonal matrix, which is more compact and efficient without sacrificing the flexibility.

2. Related Work

Canonical Pose Prediction Methods. These methods learn to transform the shape into its canonical pose to avoid the rotation perturbation [6, 12, 23, 29]. PointNet [23] and following works [14, 35] use the Spatial Transformer Networks (STN) [10]. RotPredictor [6] introduces self-consistency and enables approximately equivariant pose estimation. Its stability is improved in [28] by using Spherical CNNs [2] as backbone. Sun *et al*. [29] further learn category-level canonical pose by decomposing each shape into several capsules. Li *et al*. [12] leverage the prior of principal component analysis (PCA), and tackle its ambiguity by blending 24 PCA-based poses. Generally, above methods rely on exhaustive augmentation while only achieve approximate invariance that may not generalize to novel or partial objects.

Rotation Invariant Learning on 3D Point Clouds. Most methods achieve rotation invariance by designing hand-crafted RI features based on distances and angles [1, 13, 38, 42–45]. However, they only encode local geometry and lose global positional information [44]. To alleviate this, ClusterNet [1] has to build a strongly connected $k$nn-graph with $k = 80$. Recent works design auxiliary blocks [42, 45] to encode canonicalized points for global information compensation. Other methods [13, 38] introduce RI representations that encode more global context, such as gram matrix, which suffers from high complexity for computing relative distances or angles between arbitrary point pairs. LRF-based methods [11] transform local point coordinates w.r.t the LRF to stay RI, which also loses the pose information. Equivariant methods allow the learned features to undergo predictable linear transformations following the rotations of the inputs [3, 8, 22, 32], thus can also achieve invariance without losing pose information. Yet they involve strict constraints on the convolution kernels, sacrificing their flexibility. On the contrary, our PaRI-Conv can perfectly solve the pose information loss problem, with informative convolution kernels, resulting in better performance.

Dynamic Kernels for Convolution on 3D Point Clouds. A line of works [15, 34, 36] directly regress weight kernels based on the positions of neighbour points, leading to prohibitively large memory. Others maintain a weight bank and generate new weight matrix via linear combination [18, 31, 39, 40]. They typically relate the weights with anchor points and assemble new weight in a handcrafted manner [18, 31], which limits their flexibility. Recently, PAConv [39] learns to dynamically assemble the kernel weights via score prediction. Generally, the weight bank requires high memory and is hard to be optimized jointly. On the contrary, our factorized dynamic kernel is more compact and efficient without sacrificing the performance.

3. Problem Definition and Background

We first provide necessary background and mathematically explain the pose information loss problem in existing RI networks, which we aim at solving in this paper.

Rotation Invariant Functions. Given a point cloud $P = [p_1, p_2, ..., p_N] \in \mathbb{R}^{N \times 3}$ with N points, we can apply arbitrary rotation to it by $P = PR$, where $R \in \mathrm{SO}(3)$ is a 3×3 rotation matrix. A *rotation-invariant* (RI) function Φ should satisfy:

$$\Phi(P) = \Phi(PR), \quad \forall R \in \mathrm{SO}(3). \tag{1}$$

Convolutions on 3D Point Clouds. Generally, the input of a convolution layer on 3D point clouds is a point cloud $P \in \mathbb{R}^{N \times 3}$ with features $X = [x_1, ..., x_N] \in \mathbb{R}^{N \times c_{in}}$, and output features $X' = [x'_1, ..., x'_N] \in \mathbb{R}^{N \times c_{out}}$. The convolution operation f at the reference point p_r can be formulated as:

$$f(p_r) = \bigwedge_{j \in \mathcal{N}(p_r)} W_j \cdot h(p_j), \tag{2}$$

where $\mathcal{N}(p_r)$ denotes a local patch around p_r, W_j is the kernel weight, and $h : \mathbb{R}^3 \rightarrow \mathbb{R}^c$ denotes a non-linear function that maps the point coordinate p_j to its feature x_j. $\bigwedge$ denotes an aggregation function, such as MAX, AVG or SUM. Early *multi-layer perceptron* (MLP) based methods [23, 24] design isotropic kernels, meaning that $W_i = W_j, \forall i, j \in \mathcal{N}(p_r)$, which limits the ability of expressing location relationship of neighbours. More recently, some works [18, 36, 47] design position-adaptive kernels to solve this problem with $W_j = W(p_j, p_r)$.

Pose Information Loss Problem in Current RI Methods. Though above convolutions have achieved impressive performance on 3D raw coordinates, we discover that applying above convolutions on RI features inevitably loses *relative pose information* between them, leading to local geometric ambiguity. Here, we extend Equation 2 as:

$$f(p_r | \mathbf{\Gamma}(p_r)) = \bigwedge_{j \in \mathcal{N}(p_r)} W_j \cdot h(p_j | \mathbf{\Omega}(p_j)), \tag{3}$$

where $\mathbf{\Omega}(p_j)$ is the receptive field of h at point p_j, meaning that $h(p_j)$ only depends on $\mathbf{\Omega}(p_j)$. Then, the receptive field of f in Equation 3 is defined as $\mathbf{\Gamma}(p_r) = \bigcup_{j \in \mathcal{N}(p_r)} \mathbf{\Omega}(p_j)$.

To explain the problem, we change the relative pose between local patches via a transformation $\mathcal{T}$:

$$\mathcal{T}(\mathbf{\Gamma}(p_r)) = \bigcup_{j \in \mathcal{N}(p_r)} \mathbf{\Omega}(p_j) R_j, \tag{4}$$

which means we separately rotate each patch $\mathbf{\Omega}(p_j)$ around p_j via R_j (*e.g.*, rotating the eyes and mouth in Figure 1, changing the smile face to the angry face). Then we perform convolution on the transformed data $\mathcal{T}(\mathbf{\Gamma}(p_r))$ by

$$f(p_r | \mathcal{T}(\mathbf{\Gamma}(p_r))) = \bigwedge_{j \in \mathcal{N}(p_r)} W_j \cdot h(p_j | \mathbf{\Omega}(p_j) R_j). \tag{5}$$

Given that h is an RI function, we have $h(p_j | \mathbf{\Omega}(p_j)) = h(p_j | \mathbf{\Omega}(p_j) R_j)$, which is the definition of RI functions given in Equation 1. Thus, the right-hand side of Equation 3 and Equation 5 is equal, leading to

$$f(p_r | \mathbf{\Gamma}(p_r)) = f(p_r | \mathcal{T}(\mathbf{\Gamma}(p_r))), \tag{6}$$

which means current CNNs cannot identify the changes of the relative pose between local patches achieved by $\mathcal{T}$, and proves the derived feature will *inevitably lose the relative pose information* between local patches. This can cause severe geometric ambiguity in the derived feature, *i.e.*, as shown in Figure 1, the feature derived from vanilla CNNs cannot distinguish the happy face from the angry face.

4. Method

To counter the pose information loss problem, we propose a *Pose-aware Rotation Invariant Convolution* (*i.e.*, PaRI-Conv), which replaces the fixed kernels in vanilla convolutions with pose-aware dynamic kernels. Specifically, the general formulation of PaRI-Conv at a reference point p_r can be defined as:

$$x'_r = \bigwedge_{j \in \mathcal{N}(p_r)} \mathcal{W}(\mathcal{P}_r^j) \cdot x_j, \tag{7}$$

where $\mathcal{W}$ is a dynamic kernel function that maps the relative pose $\mathcal{P}_r^j$ between the center point p_r and its neighbour p_j into respective kernel weight. With the pose-aware kernel $\mathcal{W}(\mathcal{P}_r^j)$, W_j in Equation 3 and 5 will no longer be equal, because rotating p_j with R_j changes its relative pose $\mathcal{P}_r^j$ to the center point. Thus, the ambiguity defined in Equation 6 is readily eliminated. Meanwhile, the rotation invariance property is maintained because a global rotation will not change the relative poses $\mathcal{P}_r^j$ between arbitrary points (see supplementary material for more details). The pipeline of our PaRI-Conv is illustrated in Figure 2. An RI feature, named *Augmented Point Pair Feature* (APPF) is introduced to fully embed the pose information $\mathcal{P}_r^j$. Afterwards, we propose a *factorized dynamic kernel* to formulate the kernel function $\mathcal{W}$, and the final output is aggregated from the derived features via the EdgeConv [35].

4.1. Augmented Point Pair Feature

LRF Construction. To build RI representations, the LRF $\mathcal{L}_r$ is first constructed at each point $p_r \in P$, which consists of three orthonormal basis: $\mathcal{L}_r = [\partial_r^1, \partial_r^2, \partial_r^3] \in \mathbb{R}^{3 \times 3}$. Typically, $\mathcal{L}_r$ is constructed by first defining two axes e_r^1, e_r^2, and then performing orthonormalization via

$$\partial_r^1 = e_r^1, \ \partial_r^3 = \frac{\partial_r^1 \times e_r^2}{\|\partial_r^1 \times e_r^2\|}, \ \partial_r^2 = \partial_r^3 \times \partial_r^1. \tag{8}$$

There are a few choices for axes e_r^1, e_r^2. RI-GCN [11] computes LRF via PCA, which can be sensitive to perturbations.
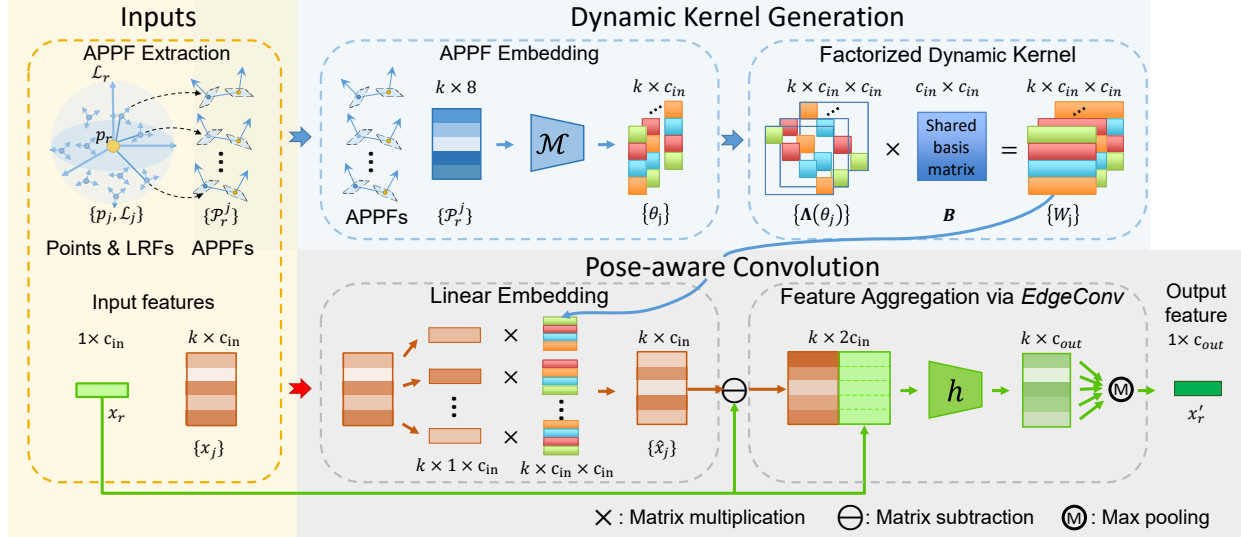
Figure 2. Overview of the proposed PaRI-Conv. The inputs of a PaRI-Conv layer are the LRF $\mathcal{L}_r$ and feature x_r for a center point p_r, as well as the LRFs $\{\mathcal{L}_j\}_{j=1}^k$ and features $\{x_j\}_{j=1}^k$ for its k neighbours $\{p_j\}_{j=1}^k$. To address the pose information loss problem, the key idea is to synthesize dynamic kernels $\{W_j\}_{j=1}^k$ based on the relative poses. We encode the relative pose between each neighbour p_j and the center p_r via the proposed APPF $\mathcal{P}_r^j$, which is extracted from their respective LRFs $\mathcal{L}_j$ and $\mathcal{L}_r$. Then $\mathcal{P}_r^j$ is encoded into θ_j via an MLP $\mathcal{M}$, and converted to a diagonal matrix $\Lambda(\theta_j)$, which is multiplied with the shared basis matrix B to synthesize the final kernel weight W_j. We name this strategy *factorized dynamic kernel*. After the linear embedding, we aggregate the derived features $\{\hat{x}_j\}_{j=1}^k$ via EdgeConv.
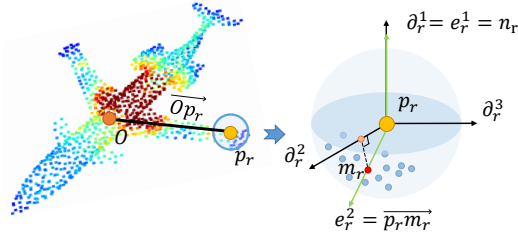

Figure 3. Illustration of the construction of LRF at point p_r.

Figure 4. Illustration of the traditional PPF (a) and our proposed APPF (c). In (b), the PPF allows rotations around the axis ∂_r^1, leading to ambiguity. In (c), we fix the rotation via the angle $\beta_{r,j}$.

Many other methods turn to apply the vector between the global center O and the local center point p_r [13, 42, 43]. Though being more consistent under rotations, the axis $\overrightarrow{Op_r}$ is related to the global location of the point p_r rather than the local geometry, which leads to two serious problems: 1) the network struggles to learn consistent representations for identical structures that locate differently in the 3D shape, and 2) when partiality and background exist, the global center will be severely shifted, limiting the robustness to these perturbations. Thus, we propose to build the LRF upon local geometry only. Specifically, as shown in Figure 3, we define the first two axes as $e_r^1 = n_r$ and $e_r^2 = \overrightarrow{p_r m_r} = m_r - p_r$, where n_r is the normal and $m_r = \frac{1}{k}\sum_{j=1}^k p_r^j$ is the barycenter of k nearest neighbours around point p_r.

Feature Extraction. To capture the relative pose between the LRFs of the reference point p_r and a neighbour point p_j, we design our RI feature based on point pair feature (PPF) [5]. As shown in Figure 4(a), given their primal axes
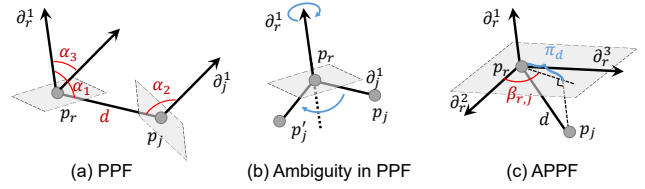
$\partial_r^1, \partial_j^1$, the PPF is a 4D vector:

$$\text{PPF}(p_r, p_j) = (\|d\|_2, \alpha_1, \alpha_2, \alpha_3),$$
$$\alpha_1 = \angle(\partial_r^1, d), \ \alpha_2 = \angle(\partial_j^1, d), \ \alpha_3 = \angle(\partial_r^1, \partial_j^1)), \quad (9)$$

where $d = p_j - p_r$, and $\angle(\cdot, \cdot)$ denotes the angle between two vectors. However, as shown in Figure 4(b), PPF cannot fully define the relative pose because it allows p_j to freely rotate around the axis ∂_r^1 of p_r [4], and vice versa. Thus, we propose to fix the rotation by augmenting PPF with azimuth angles $\beta_{r,j}$ and $\beta_{j,r}$. As shown in Figure 4(c), we project d on the $\partial_r^2 \times \partial_r^3$ plane into π_d, and we record the azimuth angle as $\beta_{r,j} = \angle(\partial_r^2, \pi_d)$. Similarly, $\beta_{j,r}$ can be derived by switching the roles of p_r and p_j. To this end, our APPF $\mathcal{P}_r^j \in \mathbb{R}^8$ can be given by:

$$\mathcal{P}_r^j = (\|d\|_2, \cos(\alpha_1), \cos(\alpha_2), \cos(\alpha_3),$$
$$\cos(\beta_{r,j}), \sin(\beta_{r,j}), \cos(\beta_{j,r}), \sin(\beta_{j,r})), \quad (10)$$

where we further embed the angles with $\sin(\cdot)$ and $\cos(\cdot)$, following recent positional embedding techniques [19, 30]. For analysis on the rotation invariance of APPF, please refer to the supplementary material. To restore the pose information lost by general convolution layers, the APPFs are then utilized to synthesize pose-aware dynamic kernel, which is introduced as follows.

4.2. Factorized Dynamic Kernel

Current methods synthesize dynamic kernels by assembling memory intensive weight banks [18, 39, 40] or regressing large kernel matrix [27, 36], which incur heavy memory usage and large computational cost. To address this problem, we propose a new *factorized dynamic kernel* (FDK). The key idea is to decompose the dynamic kernel $W_j = \mathcal{W}(\mathcal{P}_r^j) \in \mathbb{R}^{c_{in} \times c_{in}}$ into a lightweight pose-aware diagonal matrix $\mathbf{\Lambda}(\theta_j) \in \mathbb{R}^{c_{in} \times c_{in}}$ and a pose-agnostic basis matrix $B \in \mathbb{R}^{c_{in} \times c_{in}}$ that is shared by all neighbours:

$$W_j = \mathbf{\Lambda}(\theta_j)B. \tag{11}$$

As shown in Figure 2, given APPF $\mathcal{P}_r^j$ for each neighbour point, we first synthesize the diagonal matrix $\mathbf{\Lambda}(\theta_j)$ by:

$$\theta_j = \mathcal{M}(\mathcal{P}_r^j), \tag{12}$$

where $\theta_j \in \mathbb{R}^{c_{in}}$ is the diagonal entries of $\mathbf{\Lambda}(\theta_j)$, and $\mathcal{M}$ is a Multi-layer Perceptron (MLP). Then the pose-aware kernel $\mathbf{\Lambda}(\theta_j)$ is multiplied with the shared basis matrix B to generate the final kernel weight W_j, following Equation 11.

In this way, the complexity and memory usage can be significantly reduced since FDK only regresses a c_{in}-dim vector θ_j rather than a full $c_{in} \times c_{in}$ matrix as in [27, 36]. Moreover, FDK only requires one basis matrix B shared by all neighbours. Thus, it requires much less parameters than the weight bank assembling methods [40]. We also compare their complexity in the ablation study (see Table 6).

Pose-aware Convolution. Before performing convolution with the learned pose-aware kernels W_j, there remains one problem. Generally, the convolution also considers the feature of the reference point x_r. However, in this case, the APPF $\mathcal{P}_r^r$ is a zero vector, leading to an invalid kernel matrix W_r. Thus, naive convolution will lose critical information from the feature of the reference point x_r. We address this problem by excluding x_r when linear embedding:

$$\hat{x}_j = W_j x_j, \quad j \neq r, \tag{13}$$

and then aggregating x_r with neighbour features $\hat{x}_j$ via the EdgeConv [35]:

$$x_r' = \underset{j \in \mathcal{N}(p_r)}{\text{MAX}} \, g((\hat{x}_j - x_r) \oplus x_r), \tag{14}$$

where g is a one-layer MLP and $\oplus$ denotes feature concatenation operation.

4.3. Network Architecture

Apart from the convolution, specialized architectures can also affect the performance, and obscure the true effectiveness of the convolution operators [16]. To fairly evaluate the intrinsic effectiveness of PaRI-Conv and its compatibility with existing backbones, we directly integrate our PaRI-Conv into two known architectures.

Shape Classification. We employ the classical DGCNN [35] by simply replacing EdgeConv in it with our PaRI-Conv. Following DGCNN, we also perform k nearest neighbour ($k$nn) search ($k = 20$) in Euclidean space for the first layer and in the feature space for the rest of the layers.

Shape Part Segmentation. For part segmentation, larger context is very important. Thus, we apply pooling layers to enlarge the receptive field following AdaptConv [47]. Generally, it is similar to the classification network but deeper with 5 convolution layers and 3 pooling layers. We replace all the Conv-layers with our PaRI-Conv, except for the last graph convolution layer. Here, the $k$nn search is performed in the Euclidean space with $k = 40$ in all layers.

5. Experiments

We evaluate the performance of the proposed PaRI-Conv on three challenging datasets, *i.e.*, ModelNet40 [37] for 3D shape classification, ShapeNetPart [41] for part segmentation and ScanObjectNN [33] for real-world shape classification. To evaluate the invariance under various rotations, we conduct experiments under 3 train/test settings, *i.e.*, z/z, $z/\text{SO}(3)$ and $\text{SO}(3)/\text{SO}(3)$, where z denotes that the inputs are processed with rotations around the vertical axis, and $\text{SO}(3)$ denotes arbitrary rotations.

5.1. Implementation Details

Weight Kernel Generation. We implement the function $\mathcal{M}$ in Equation 12 as a 2-layer MLP that ends with a linear layer. The shared basis matrix B is randomly initialized and optimized together with other network parameters.

LRF Setting. The default LRF introduced in Section 4.1 utilizes the normal vector. We denote the results obtained under this setting as (pc+normal) in the experiments. For fair comparison with methods that only require point positions, we replace the normal with the vector between the global center and the point (*i.e.*,$\overrightarrow{Op_r}$), and leave the second axis unchanged. We also evaluate the performance of PaRI-Conv under more LRFs settings in the ablation study.

Network Inputs. The rotation invariance of input is an essential condition for the rotation invariance of the network. We assign each point P_i with an initial RI attribute following Spherical CNN [2], which are $\|p_i\|_2$, $\sin(\angle(\partial_i^1, p_i))$ and $\cos(\angle(\partial_i^1, p_i))$, where ∂_i^1 is the primal axis of the LRF.

Training Strategy. We implement the above networks in PyTorch [20] and PyTorch Geometric [7]. We use SGD with

initial learning rate 0.1, which is gradually reduced to 0.001 using cosine annealing [17]. The batch size is 32 and all networks are trained for 300 epochs. We apply 80% dropout rate in the fully connected layers.

5.2. 3D Shape Classification

Dataset. ModelNet40 [37] consists of 12,311 CAD mesh models from 40 categories, with 9,843 for training and 2,468 for testing. For fair comparison, we use the sampled point clouds provided by PointNet [23] and uniformly sample 1,024 points as input. When training, we augment the input with random scaling and translation.

Results. As shown in Table 1, we compare our proposed PaRI-Conv with rotation sensitive, equivariant and invariant methods. PaRI-Conv stays consistently rotation-invariant regardless of rotation augmentation during training and achieves outstanding accuracy of **92.3%** with normal and **91.4%** without normal, which outperform all comparison methods on unaligned data. Comparing to rotation-sensitive methods, their performance degrades drastically when testing on unseen rotations under $z/\mathrm{SO}(3)$ setting. With augmentation ($\mathrm{SO}(3)/\mathrm{SO}(3)$), there is still a noticeable performance gap ($\sim 10\%$). Comparing to RI methods [11, 38, 43–45], our method significantly improves former best performance from LGR-Net [45] by 1.2%. Specially, ours with point only can still outperform other RI methods that apply normals, which justifies the significance in countering the pose information loss suffered by these RI methods. Moreover, comparing to equivariant methods [3, 22, 32] that are free of the pose information loss, PaRI-Conv surpasses them even more, which can be attributed to the high flexibility of our proposed factorized dynamic kernel over their constrained kernels. Finally, our method is also more effective than methods that learn to transform the input point cloud globally [6,12], while avoiding exhaustive rotation or permutation augmentation.

5.3. Shape Part Segmentation

Dataset. For shape part segmentation, we evaluate our method on ShapeNetPart Dataset [41]. It contains 16,881 3D shapes from 16 categories. In each category, shapes are labeled with $2 \sim 5$ parts, resulting in 50 parts in total. We follow the commonly applied train-test split in [24] and randomly sample 2,048 points from each shape as input.

Results. We utilize mean Intersection-over-Union (mIoU) over all instances as evaluation metric. As shown in Table 2, similar conclusion can be drawn. Our method outperforms all comparison methods and shows consistent performance under unseen rotations. Moreover, ours without normal also surpasses methods with normal, *e.g.*, LGR-Net [45] and RI-GCN [11], showing the significance of preserving pose information. We also visualize segmentation results under $z/\mathrm{SO}(3)$ setting in Figure 5, where most parts are well seg-

	Method	Inputs	z/z	$z/\mathrm{SO}(3)$	$\mathrm{SO}(3)/$ $\mathrm{SO}(3)$
Rotation-sensitive	PointNet [23]	pc	89.2	16.4	75.5
	PointNet++ [24]	pc+n	91.8	18.4	77.4
	PointCNN [14]	pc	92.2	41.2	84.5
	DGCNN [35]	pc	92.2	20.6	81.1
	RotPredictor [6]	pc	92.1	-	90.7
Rotation-equivariant	TFN [32]	pc	88.5	85.3	87.6
	TFN-NL [22]	pc	89.7	89.7	89.7
	VN-DGCNN [3]	pc	89.5	89.5	90.2
Rotation-invariant	SF-CNN [25]	pc+n	92.3	85.3	91.0
	RI-CNN [44]	pc	86.5	86.4	86.4
	Li *et al.* [13]	pc	89.4	89.3	89.4
	SGMNet [38]	pc	90.0	90.0	90.0
	AECNN [43]	pc	91.0	91.0	91.0
	LGR-Net [45]	pc+n	90.9	90.9	91.1
	RI-GCN [11]	pc+n	91.0	91.0	91.0
	Li *et al.* [12]	pc	90.2	90.2	90.2
	Ours	pc	**91.4**	**91.4**	**91.4**
	Ours	pc+n	**92.4**	**92.4**	**92.3**

Table 1. Shape classification accuracy (%) on ModelNet40 dataset under three train/test settings. 'pc' and 'n' stands for 3d coordinates and normals of the input point cloud, respectively.

mented under unseen rotations.

Method	$z/\mathrm{SO}(3)$	$\mathrm{SO}(3)/\mathrm{SO}(3)$
Input pc only		
PointNet [23]	37.8	74.4
PointNet++ [24]	48.2	76.7
DGCNN [35]	37.4	73.3
RSCNN [15]	50.7	73.3
RI-CNN [44]	75.3	75.3
VN-DGCNN [3]	81.8	81.8
RI-Framework [13]	82.2	82.5
Li *et al.* [12]	81.7	81.7
Ours (pc)	**83.8**	**83.8**
Input pc+normal		
RI-GCN [11]	77.2	77.3
LGR-Net [45]	-	82.8
Ours (pc+normal)	**84.6**	**84.6**

Table 2. Part segmentation results on ShapeNetPart dataset [41]. We report mIoU (%) over all instances under two train/test settings. 'pc' stands for 3d coordinates of the input point cloud.

5.4. Real-world Classification

Dataset. To test the robustness of our RI method for real-world applications, we evaluate the proposed PaRI-Conv on scanned indoor objects from ScanObjectNN [33]. We choose the OBJ_BG subset, which contains 2,902 objects from 15 categories. In addition to common nuisances, such as noise, incompleteness and deformations, data in this subset are also cluttered with background, which poses great challenges to the stability of rotation invariance. Note that normal is not applied since it is not available on their processed dataset.

Results. As shown in Table 3, our method can also achieve consistent invariance to rotations even when various real-world nuisances exist. Since the axis $\overrightarrow{Op_r}$ applied by **ours**
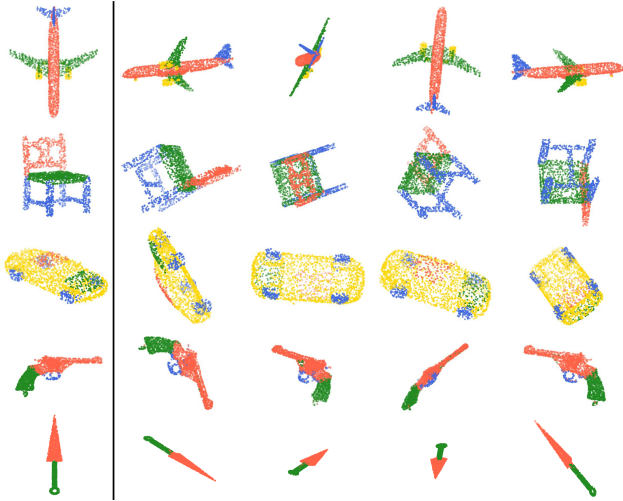
Figure 5. Visualization of part segmentation results on ShapeNet-Part Dataset [41] under $z/\text{SO}(3)$ setting. The ground truth is in the left-most column. The rest columns are our testing results under shown rotations.

(pc) can be very unstable due to the shift of global center O when cluttered background exists. We replace $\overrightarrow{Op_r}$ with a PCA-based primal axis, denoted as **ours (PCA)**. Though ours (pc) is less competitive, ours (PCA) achieves the best performance, and outperforms the strongest competitor LGR-Net by 2.1%. More importantly, when comparing the results on aligned data (*i.e.*, z/z), ours (PCA) even surpasses most standard 3D deep learning methods (*e.g.*, DGCNN [35]), which demonstrates that explicitly modeling the relative poses via our PaRI-Conv can not only eliminate the pose information loss but also counter various perturbations existing in real-world data.

Method	z/z	$z/\text{SO}(3)$	$\text{SO}(3)/\text{SO}(3)$
	Input pc only		
PointNet [23]	73.3	16.7	54.7
PointNet++ [24]	82.3	15.0	47.4
PointCNN [14]	86.1	14.6	63.7
DGCNN [35]	82.8	17.7	71.8
RI-CNN [44]	74.6	75.3	75.5
RI-Framework [13]	-	79.8	-
LGR-Net [45]	81.2	81.2	81.4
Ours (pc)	77.8	77.8	78.1
Ours (PCA)	**83.3**	**83.3**	**83.3**

Table 3. Real-world classification accuracy (%) on ScanObjectNN dataset under three train/test settings. 'PCA' denotes the primal axis ∂_r^1 is constructed via PCA in local neighbours.

5.5. Model Complexity

Efficiency and lightness are also our key advantages. Former methods suffer from large computational cost caused by larger receptive fields [43] and additional network [45] to incorporate the lost global information. In contrast, due to the pose-awareness, our method can obtain intact global context by simply stacking PaRI-Conv layers as in normal 2D CNNs. Moreover, PaRI-Conv is also free of the pose predictor utilized in global methods [12]. Consequently, as shown in table 4, our method achieves the best performance with the smallest model size and much less computational cost ($>$**48%** FLOPs reduction).

	DGCNN	Li *et al*. [12]	AECNN [43]	LGRNet [45]	Ours
FLOPs	2495M	3747M	4841M	5828M	**1938M**
Params	**1.81M**	2.91M	1.99M	5.55M	1.85M
Acc.	75.5	90.2	91.0	91.1	**91.4**

Table 4. Comparison of model complexity on ModelNet40 dataset, where floating point operation/sample (FLOPs), number of parameters (Params) and accuracy (Acc.) are reported.

5.6. Ablation Studies

To demonstrate the effectiveness of each proposed component in PaRI-Conv, we further conduct ablation studies on the relative pose representation (APPF), the factorized dynamic kernel and the LRF construction. All experiments are evaluated by accuracy (%) on ModelNet40 [37] dataset. **Relative Pose Representation.** The pose is the key to preventing information loss. Thus, how to represent it is worth exploring. As shown in Table 5, we compare our proposed APPF with: **1)** the rotation matrix and translation vector (R, t), **2)** the PPF [5] defined as Equation 9, and **3)** APPF without direction information, which is defined as $(\|d\|_2, \cos(\alpha_1), \cos(\beta_{r,j}), \sin(\beta_{r,j}))$, where $\|d\|_2, \alpha_1, \beta_{r,j}$ can be considered as the 3-dim polar coordinates, which only encode the relative position information. First of all, APPF notably outperforms representations that only partially encode the pose information (**2,3**), demonstrating the critical role of pose-awareness in preventing information loss that exists in RI learning. Moreover, APPF also surpasses (R, t) by 1.1%, which shows APPF is a more effective representation for deep networks to embed.

No.	RI Representation	Dim	$\text{SO}(3)/\text{SO}(3)$
1	(R, t)	12	91.2
2	PPF	4	91.5
3	APPF-w/oDirection	5	91.4
4	APPF	8	**92.3**

Table 5. Ablation study on the relative pose representation $\mathcal{P}_r^j$.

Factorized Dynamic Kernel. As shown in Table 6, we compare the proposed factorized dynamic kernel (FDK) with other strategies that allow pose-aware convolution, including: **1)** Concat where we ablate the FDK and directly concatenate the APPFs with the latent features, **2)** Ours-w/oEdge where we ablate the final *EdgeConv*, and **3)** other dynamic kernels [15,36,39]. With the proposed FDK, ours-w/oEdge reduces the number of parameters (up to **26%**) and FLOPs ($>$**12%**), while achieving better performance. This shows that FDK is more compact and flexible. Our

full PaRI-Conv can further boost the performance to **92.3**% with moderate parameters and FLOPs increment.

Dynamic Kernels	Params	FLOPs	SO(3)/SO(3)
Concat.	1.84M	-	91.3
RSCNN [15]	1.83M	1913M	91.2
PointConv [36]	2.45M	2264M	91.2
PAConv [39]	2.44M	1768M	89.6
Ours-w/oEdge	**1.81M**	**1579M**	91.7
Ours	1.85M	1938M	**92.3**

Table 6. Ablation study on the factorized dynamic kernel.

LRF Construction. The selection of LRF is crucial for the robustness of rotation invariance. Recall that we introduce three relatively stable axes in Section 4.1, which are the normal n_r, the direction of the local barycenter $\overrightarrow{p_r m_r} = m_r - p_r$ and the direction relative to the global center $\overrightarrow{Op_r} = p_r - O$. Based on these axes, we construct various LRFs by assigning two of them as (e_r^1, e_r^2) and derive the LRF following Equation 8. As shown in Table 7, the LRFs with normal generally outperform the counterparts. Moreover, replacing the local barycenter in $(n_r, \overrightarrow{p_r m_r})$ with the global center in $(n_r, \overrightarrow{Op_r})$ results in severe performance degradation (0.6%), which proves our assumption that LRF should be built upon local geometry only.

(e_r^1, e_r^2)	$(\overrightarrow{p_r m_r}, \overrightarrow{Op_r})$	$(\overrightarrow{Op_r}, \overrightarrow{p_r m_r})$	$(n_r, \overrightarrow{Op_r})$	$(n_r, \overrightarrow{p_r m_r})$
Acc.	91.2	91.4	91.7	**92.3**

Table 7. Ablation study on LRF construction.

5.7. Visualization of Learned RI Features

We visualize the learned feature map under various rotations in Figure 6. While the features learned by DGCNN [35] change vastly as the shape rotates, our PaRI-Conv can learn consistent representation for corresponding points between point clouds under different rotations.

The above results demonstrate that developing deep networks like PaRI-Conv that respect natural symmetries in 3D space, such as rotations and reflections, allows identical geometric structures to share the same convolution kernels. On the contrary, rotation-sensitive methods [35] are forced to learn redundant filters for the same structures under different poses, which indicates PaRI-Conv has the potential in developing more compact networks.

5.8. Discussion

Though our PaRI-Conv solves the information loss problem in RI learning, there remains small gap between PaRI-Conv and more recent rotation-sensitive methods [39, 46] on aligned data. We assume that this may be partially due to some prior that exists in aligned data. For instance, shape of beds and wardrobes are very similar, while the canonical poses (*i.e.*, lying down vs. standing) provide strong prior, making it easier to distinguish between them, which, however, cannot be leveraged by our PaRI-Conv. To preclude
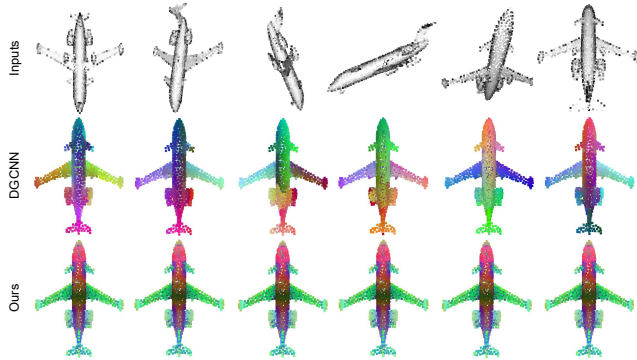

Figure 6. Visualization of dense features learned on shape classification task. While the features learned by DGCNN (second row) change vastly, our PaRI-Conv (third row) learns consistent representation for identical structures under different rotations.

this factor, we further evaluate our PaRI-Conv on aligned data by directly taking coordinates as input. Surprisingly, with normal as a stable axis, PaRI-Conv achieves 93.8% overall accuracy, which outperforms recent point convolution method PAConv [39] (93.6%) and powerful transformer based method [46] (93.7%) (see the supplementary material for details). We believe that above results reveal the significance of pose information, which has not been well exploited before, and our proposed PaRI-Conv is an effective operator in capturing pose-variant geometric structures.

6. Conclusion, Limitation and Future Work

We have presented PaRI-Conv, a Pose-aware convolution operator for RI learning on 3D point clouds. We have revealed that the inferior performance of current RI methods is caused by the inherent pose information loss problem, and demonstrated that PaRI-Conv can perfectly solve the problem by dynamically adapting the kernel based on the relative pose. Specifically, a new Augmented Point Pair Feature (APPF) has been proposed to effectively encode the pose. To synthesize the dynamic kernel, we factorize it into a shared basis and a low DoF dynamic kernel, which has been shown to be more lightweight without sacrificing flexibility. Experimental results show that PaRI-Conv is consistently invariant, and, by eliminating the pose information loss, can noticeably surpass the state-of-the-art RI methods with less parameters and computational cost.

Though achieving certain robustness to real-world nuisance, our method could be sensitive to extreme noises due to the instability of LRFs. Since PaRI-Conv provides a general framework for RI learning, this can be addressed by applying more robust LRFs [21, 26].

In the future, we hope PaRI-Conv can pave the way for introducing rotation invariance to more tasks, *e.g.*, semantic segmentation, object detection and registration *etc*.

References

[1] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. 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*, pages 4994–5002, 2019. 1, 2

[2] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *ICLR*, 2018. 2, 5

[3] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12200–12209, October 2021. 2, 6

[4] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018. 4

[5] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. 2, 4, 7

[6] Jin Fang, Dingfu Zhou, Xibin Song, Shengze Jin, Ruigang Yang, and Liangjun Zhang. Rotpredictor: Unsupervised canonical viewpoint learning for point cloud classification. In *2020 International Conference on 3D Vision (3DV)*, pages 987–996. IEEE, 2020. 2, 6

[7] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. 5

[8] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[9] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 2

[11] Park Jaeyoo Kim, Seohyun and Bohyoung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 6

[12] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16218–16227, 2021. 2, 6, 7

[13] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. A rotation-invariant framework for deep point cloud analysis. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 1, 2, 4, 6, 7

[14] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 1, 2, 6, 7

[15] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 1, 2, 6, 7, 8

[16] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pages 326–342. Springer, 2020. 5

[17] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2017. 6

[18] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1578–1587, 2019. 2, 3, 5

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 5

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[21] Alioscia Petrelli and Luigi Di Stefano. A repeatable and efficient canonical reference for surface matching. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 403–410. IEEE, 2012. 8

[22] Adrien Poulenard and Leonidas J Guibas. A functional approach to rotation equivariant non-linearities for tensor field networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13174–13183, 2021. 2, 6

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2, 3, 6, 7

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*. 1, 3, 6, 7

[25] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 452–460, 2019. 6

[26] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 8

[27] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017. 2, 5

[28] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to orient surfaces by self-supervised spherical cnns. *Advances in Neural Information Processing Systems*, 33:5381–5392, 2020. 2

[29] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[30] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020. 5

[31] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 2

[32] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 2, 6

[33] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1588–1597, 2019. 5, 6

[34] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. 2

[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 3, 5, 6, 7, 8

[36] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 2, 3, 5, 7, 8

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5, 6, 7

[38] Jianyun Xu, Xin Tang, Yushi Zhu, Jie Sun, and Shiliang Pu. Sgmnet: Learning rotation-invariant point cloud representations via sorted gram matrix. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10468–10477, 2021. 1, 2, 6

[39] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021. 2, 5, 7, 8

[40] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 2, 5

[41] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 5, 6, 7

[42] Ruixuan Yu, Xin Wei, Federico Tombari, and Jian Sun. Deep positional and relational feature learning for rotation-invariant point cloud analysis. In *European Conference on Computer Vision*, pages 217–233. Springer, 2020. 1, 2, 4

[43] Junming Zhang, Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson. Learning rotation-invariant representations of point clouds using aligned edge convolutional neural networks. In *2020 International Conference on 3D Vision (3DV)*, pages 200–209. IEEE, 2020. 1, 2, 4, 6, 7

[44] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *2019 International Conference on 3D Vision (3DV)*, pages 204–213. IEEE, 2019. 2, 6, 7

[45] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. Rotation invariant point cloud classification: Where local geometry meets global topology. *arXiv preprint arXiv:1911.00195*, 2019. 1, 2, 6, 7

[46] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 8

[47] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4965–4974, 2021. 3, 5