# *Supplementary Material for* The Devil is in the Pose: Ambiguity-free 3D Rotation-invariant Learning via Pose-aware Convolution

In this supplementary material, we first provide theoretical analysis on the rotation invariance of our proposed PaRI-Conv in Section A. We further show more visualization on the learned feature map in Section B. Section C investigates the robustness of PaRI-Conv. Finally, in Section D, we provide detailed comparison results for the discussion section in the main paper.

## A. Theoretical Analysis on Rotation Invariance

Here, we provide theoretical analysis on the rotation invariance of our proposed PaRI-Conv.

### A.1. Augmented Point Pair Feature

We first introduce some lemmas and prove that our proposed Augmented Point Pair Feature (APPF) is rotation-invariant, which is the building block of the rotation invariance of our PaRI-Conv.

**Lemma 1.** *Given two vector $v_1, v_2 \in \mathbb{R}^{1\times 3}$, the angle between them is invariant to arbitrary rotations $R \in \mathrm{SO}(3)$:*

$$\angle(v_1, v_2) = \angle(v_1 R, v_2 R). \tag{1}$$

*Proof.* Given that $\angle(v_1, v_2) \in [0, \pi]$, Equation 1 is equivalent to $\cos(\angle(v_1, v_2)) = \cos(\angle(v_1 R, v_2 R))$, which is given by:

$$\begin{aligned}\cos(\angle(v_1 R, v_2 R)) &= \frac{\langle v_1 R, v_2 R\rangle}{\|v_1 R\|\|v_2 R\|} = \frac{v_1 R R^\top v_2^\top}{\|v_1 R\|\|v_2 R\|} \\ &= \frac{v_1 v_2^\top}{\|v_1\|\|v_2\|} = \frac{\langle v_1, v_2\rangle}{\|v_1\|\|v_2\|} = \cos(\angle(v_1, v_2)),\end{aligned} \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes inner product. $\square$

**Lemma 2.** *Given two points $p_1, p_2 \in \mathbb{R}^{1\times 3}$, the distance between them is invariant to arbitrary rotations $R \in \mathrm{SO}(3)$:*

$$\mathrm{dist}(p_1, p_2) = \mathrm{dist}(p_1 R, p_2 R). \tag{3}$$

*Proof.*

$$\begin{aligned}\mathrm{dist}(p_1 R, p_2 R) &= \sqrt{(p_1 - p_2) R R^\top (p_1 - p_2)^\top} \\ &= \sqrt{(p_1 - p_2)(p_1 - p_2)^\top} \\ &= \mathrm{dist}(p_1, p_2)\end{aligned} \tag{4}$$

$\square$

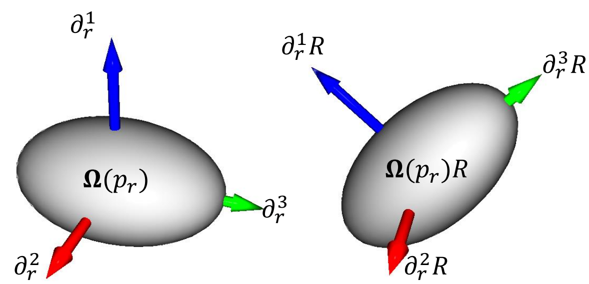


Figure 1. Illustration of the rotation equivariance of LRF.

The rotation invariance of our APPF relies on the stability of LRF, which is expected to satisfy

$$\partial_r^*(\boldsymbol{\Omega}(p_r)R) = \partial_r^*(\boldsymbol{\Omega}(p_r))R, \tag{5}$$

where $\boldsymbol{\Omega}(p_r)$ denotes a local patch around point $p_r$. This implies an axis of the LRF $\partial_r^*$ built at $\boldsymbol{\Omega}(p_r)$ is equivariant to arbitrary rotations $\forall R \in \mathrm{SO}(3)$. As shown in Figure 1, intuitively, this means a stable LRF should rotate together with the rotation of local patch $\boldsymbol{\Omega}(p_r)$.

**Theorem 1.** *The APPF defined as follows is rotation-invariant.*

$$\begin{aligned}\mathcal{P}_r^j =& (\|d\|_2, \cos(\alpha_1), \cos(\alpha_2), \cos(\alpha_3), \\ & \cos(\beta_{r,j}), \sin(\beta_{r,j}), \cos(\beta_{j,r}), \sin(\beta_{j,r})),\end{aligned} \tag{6}$$

*Proof.* The first four elements $\|d\|_2, \alpha_1, \alpha_2, \alpha_3$ are from the original Point Pair Feature (PPF), whose rotation invariance has been proved in [1]. Thus, here we only need to prove that $\beta_{r,j}, \beta_{j,r}$ are rotation invariant. Without loss of generality, we only prove the rotation invariance of $\beta_{r,j}$. And the rotation invariance of $\beta_{j,r}$ can be derived similarly.

As shown in Figure 4(c) of the main paper, the definition of $\beta_{r,j}$ is given by $\beta_{r,j} = \angle(\partial_r^2, \pi_d)$. We define an induced transformation $L_R$, that acts on the functions $f$ of the point cloud $P \in \mathbb{R}^{N\times 3}$ as

$$[L_R \circ f](P) = f(PR), \tag{7}$$

where $f$ can be vectors or angles in the point cloud $P$ such as $\pi_d, \beta_{r,j}$. Intuitively, $L_R$ maps a feature $f$ to itself after being rotated by $R$. Considering the property given in Equation 5, $L_R \circ \partial_r^2$ can be derived by:

$$L_R \circ \partial_r^2 = \partial_r^2(PR) = \partial_r^2(\mathbf{\Omega}(p_r)R) = \partial_r^2(\mathbf{\Omega}(p_r))R = \partial_r^2 R. \tag{8}$$

Thus, we can proceed to derive:

$$\begin{aligned} L_R \circ \beta_{r,j} =& \angle(L_R \circ \partial_r^2, L_R \circ \pi_d) \\ =& \angle(L_R \circ \partial_r^2, L_R \circ (d - \langle d, \partial_r^1 \rangle \partial_r^1)) \\ =& \angle(\partial_r^2 R, dR - \langle dR, \partial_r^1 R\rangle \partial_r^1 R) \\ =& \angle(\partial_r^2 R, (d - \langle d, \partial_r^1 \rangle \partial_r^1)R) \\ =& \angle(\partial_r^2 R, \pi_d R) \\ =& \angle(\partial_r^2, \pi_d) \\ =& \beta_{r,j}, \end{aligned} \tag{9}$$

which proves $\beta_{r,j}$ is rotation invariant. Detailed explanation of each procedure is given as follows. Line 2 is the extension of the projection $\pi_d$. Line 3 is given by Equation 8 and the definition of $L_R$ given in Equation 7. Line 4 is simply derived by algebraic operations. Line 5 is derived by the definition of $\pi_d$. Line 6 utilizes Lemma 1.

□

### A.2. PaRI-Conv

**Lemma 3.** *The network inputs defined as* $\|p_i\|_2$, $\sin(\angle(\partial_i^1, p_i))$, $\cos(\angle(\partial_i^1, p_i))$ *is rotation invariant.*

*Sketch of the proof.* The above representation is composed of distances and angles, which are proved to be rotation invariant in Lemma 1 and Lemma 2. Consequently, the representation is also rotation invariant. □

**Theorem 2.** *The $l$-th PaRI-Conv layer defined as follows is rotation invariant.*

$$x_r^{l+1} = \bigwedge_{j\in\mathcal{N}(p_r)} \mathcal{W}(\mathcal{P}_r^j) \cdot x_j^l, \tag{10}$$

*Proof.* We first assume the statement is true for layer $k-1$, which implies that the output features $x^k$ is rotation invariant. Then, given that $\mathcal{P}_r^j$ is also rotation invariant, we can proceed to derive:

$$\begin{aligned} L_R \circ x_r^{k+1} =& L_R \circ \left( \bigwedge_{j\in\mathcal{N}(p_r)} \mathcal{W}(\mathcal{P}_r^j) \cdot x_j^k \right) \\ =& \bigwedge_{j\in\mathcal{N}(p_r)} \mathcal{W}(L_R \circ \mathcal{P}_r^j) \cdot \left(L_R \circ x_j^k\right) \\ =& \bigwedge_{j\in\mathcal{N}(p_r)} \mathcal{W}(\mathcal{P}_r^j) \cdot x_j^k \\ =& x_r^{k+1}, \end{aligned} \tag{11}$$

which proves the statement is also true for layer $k$.

Moreover, given that the input feature $x_j^0$ of layer $l=0$ is rotation invariant (Lemma 3), the statement is true for layer 0. Thus, by mathematical induction, the statement is true for all $l$. □

## B. Visualization

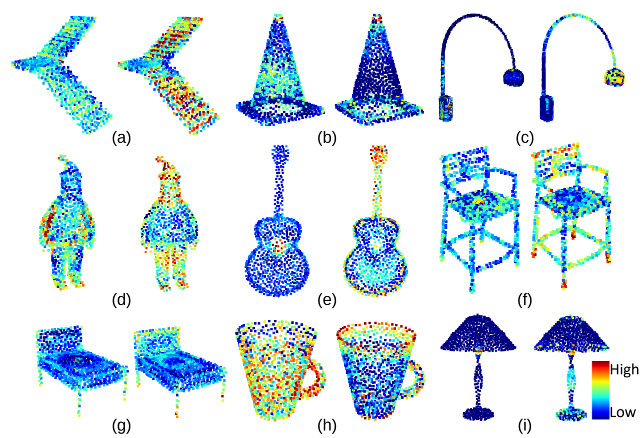


Figure 2. Low-level and high-level features learned by PaRI-Conv. In each image pair, the left shows low-level features learned at layer 2 and the right shows high-level features learned at layer 4.

### B.1. Feature Activation

We visualize the features learned by our PaRI-Conv on ModelNet40. As shown in Figure 2, we colourize the features according to the level of activation at layer 2 (left) and layer 4 (right). We observe that the shallower layer tends to capture low-level features such as planes (a, c, f), conical surface (b, h), circles (e, h) and corners (d, i). In deeper layer, PaRI-Conv learns high-level structures such as stairs (a), cones (b), lamp holders (c), heads (d), guitar necks and head-stocks (e), *etc.*

### B.2. Dense Rotation Invariant Features

**Implementation Detail for Figure 6 in the Main Paper.** We use T-SNE to map the learned features into 3D representations, which are then normalized to $[0, 1]$ to represent the RGB values.

Moreover, we observe that our PaRI-Conv is also invariant to reflections, *i.e.*, PaRI-Conv can map identical structures in the same shape (*e.g.*, left and right wings and engines) into similar representations. However, the representations learned by DGCNN tend to be bound with the absolute location information and fail to stay invariant. We assume that the reason for this invariance of PaRI-Conv is that the reflection (similar to rotation) preserves the relative poses between neighbouring structures. Thus, the $\mathcal{W}(\mathcal{P}_r^j)$ in Equation 10 remains the same, leading to invariance to reflections. We leave further exploration on this property for future work.

**Feature Similarity.** We further investigate the feature similarity between different points on the same shape. As shown in Figure 3, we visualize the similarity between the circled point and other points. Different form rotation-sensitive methods, such as DGCNN [4], our PaRI-Conv can also map identical structures in different poses (*e.g.*, the left, right wings and edge of the bathtub) into the same features. This indicates even on aligned data that are free of rotation perturbation, PaRI-Conv still has the potential in developing more compact networks via kernel weight sharing.

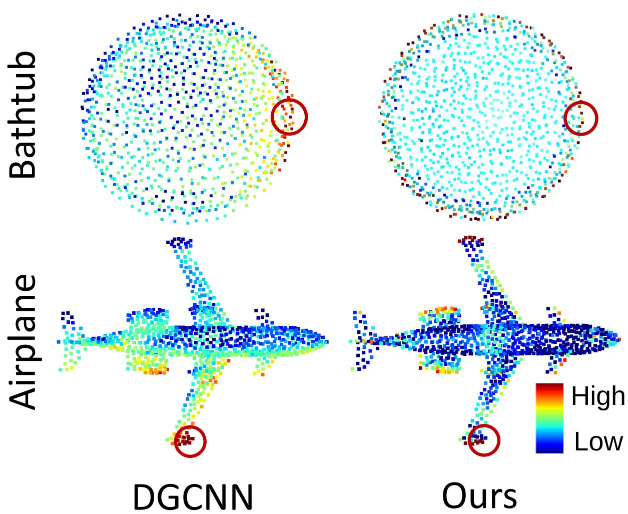


Figure 3. Feature similarity between the circled point and other points learned by DGCNN and our PaRI-Conv on ModelNet40 dataset.

# C. Robustness analysis

## C.1. Robustness to Sampling Density

Figure 4 a) shows the results under varying density. During training, the input point clouds have 1024 points and are augmented with random point dropout. Our method can achieve reasonable performance with even half of the points and consistently outperforms Li *et al.* [3]. Moreover, it also shows that applying normal as the primal axis can significantly improve the resistance to lower density.

## C.2. Robustness to Noises

Figure 4 b) shows the results under Gaussian noises of different standard variation (Std). Here, the normals are directly extracted from noisy point clouds via PCA. Our method has certain robustness to noises and can outperform RI-GCN [2] when $\sigma < 0.04$. The inferior performance under extreme noise could be attributed to the instability of current LRF, which can be addressed by applying more stable LRFs.

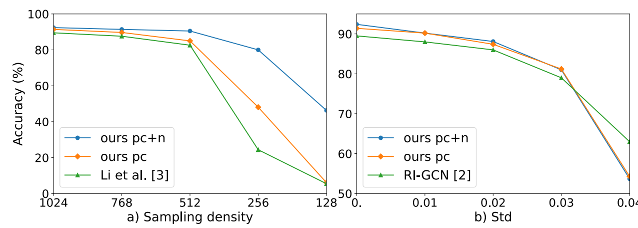


Figure 4. Comparison results with **a)** varying sampling density and **b)** different scales of noises on ModelNet40 dataset under $z/\mathrm{SO}(3)$ setting.

## C.3. Performance Under Different Neighbour Sizes

As shown in Table. 1, PaRI-Conv is not quite sensitive to the neighbour size $k$, and achieves the best result with $k = 20$.

| | $k = 10$ | $k = 20$ | $k = 40$ |
|---|---|---|---|
| PaRI-Conv (pc) | 90.5 | **91.4** | 90.4 |
| PaRI-Conv (pc+normal) | 91.5 | **92.4** | 91.7 |

Table 1. Performance under different neighbour size $k$ on ModelNet40 dataset. Results are evaluated under $z/SO(3)$ setting.

# D. Detailed Results on the Discussion

Here, we provide detailed comparison results with more recent state-of-the-art methods in Table 2. Similar to other methods, we also directly take 3D coordinates as input. When normal is available, different from PointASNL [6] that directly uses normal as additional input attributes, we only utilize them to construct a more stable LRF for the extraction of APPF. Surprisingly, with normal as a stable axis, our PaRI-Conv achieves **93.8%** overall accuracy, which outperforms recent point convolution method PAConv [5] and powerful transformer based method [7]. We believe that above results reveal the significance of pose information, which has been greatly overlooked by current point cloud analysis techniques. Moreover, this also demonstrates that the proposed PaRI-Conv is an effective operator in capturing pose-variant geometric structures.

| Methods | input | mAcc | OA |
|---|---|---|---|
| DGCNN [4] | pc | 90.2 | 92.9 |
| PointASNL [6] | pc | - | 92.9 |
| PointASNL [6] | pc+normal | - | 93.2 |
| PAConv [5] | pc | - | 93.6 |
| AdaptConv [8] | pc | 90.7 | 93.4 |
| PointTransformer [7] | pc | 90.6 | 93.7 |
| Ours | pc | 90.4 | 93.2 |
| Ours (normal in LRF) | pc | **91.3** | **93.8** |

Table 2. Performance comparison with state-of-the-art rotation-sensitive methods on ModelNet40 dataset. Rotation perturbation is not applied. Our proposed PaRI-Conv directly takes 3D coordinates as input.

## References


[1] 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. 1

[2] 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. 3

[3] 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. 3

[4] 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. 3, 4

[5] 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. 3, 4

[6] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. 3, 4

[7] 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. 3, 4

[8] 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. 4