

# Supplementary Material

## Local-consistent Transformation Learning for Rotation-invariant Point Cloud Analysis

Yiyang Chen<sup>1</sup> Lunhao Duan<sup>2</sup> Shanshan Zhao<sup>3†</sup> Changxing Ding<sup>1,4†</sup> Dacheng Tao<sup>5</sup>  
<sup>1</sup>South China University of Technology <sup>2</sup>Wuhan University <sup>3</sup>The University of Sydney  
<sup>4</sup>Pazhou Lab <sup>5</sup>Nanyang Technological University

eeyychen@mail.scut.edu.cn, lhduan@whu.edu.cn  
 {sshan.zhao00, dacheng.tao}@gmail.com, chxding@scut.edu.cn

In this Supplementary Material, we first provide the proof of the properties of LCRF in Sec. A. Additional experiments are presented in Sec. B for further analysis. Next, we summarize the limitation of our work in Sec. C. Finally, we show more visualization results on LCRF and LRF built by the Gram-Schmidt process for comparison in Sec. D.

### A. Proofs of LCRF Properties

In this section, we analyze the orthogonality of  $u_{r,1}$  and  $u_{r,2}$  in LCRF and the rotation invariance in Eq. 13 of the main text.

#### A.1. Orthogonality of $u_{r,1}$ and $u_{r,2}$ in LCRF

**Theorem 1.** Given two normalized vectors  $\tilde{v}_{r,1}, \tilde{v}_{r,2} \in \mathbb{R}^{3 \times 1}$ , with  $\theta \in (0, \frac{\pi}{2})$  being half the angle between them,  $u_{r,1}$  and  $u_{r,2}$  in LCRF, defined as follows, are orthogonal:

$$\sin \theta = \sqrt{\frac{1 - \tilde{v}_{r,1}^T \tilde{v}_{r,2}}{2}}, \cos \theta = \sqrt{\frac{1 + \tilde{v}_{r,1}^T \tilde{v}_{r,2}}{2}}, \quad (1)$$

$$\bar{v}_r = \frac{(\tilde{v}_{r,1} + \tilde{v}_{r,2})}{\|\tilde{v}_{r,1} + \tilde{v}_{r,2}\|} (\sin \theta + \cos \theta), \quad (2)$$

$$u_{r,1} = \frac{\bar{v}_r - \tilde{v}_{r,1}}{\|\bar{v}_r - \tilde{v}_{r,1}\|}, u_{r,2} = \frac{\bar{v}_r - \tilde{v}_{r,2}}{\|\bar{v}_r - \tilde{v}_{r,2}\|}. \quad (3)$$

*Proof.* Fig. 1 illustrates our LCRF. The orthogonality of non-zero vectors  $u_{r,1}$  and  $u_{r,2}$  depends on whether  $u_{r,1}^T u_{r,2} = 0$ . To demonstrate this, we first compute  $u_{r,1}^T u_{r,2}$ :

$$u_{r,1}^T u_{r,2} = \frac{(\bar{v}_r - \tilde{v}_{r,1})^T (\bar{v}_r - \tilde{v}_{r,2})}{\|\bar{v}_r - \tilde{v}_{r,1}\| \|\bar{v}_r - \tilde{v}_{r,2}\|}. \quad (4)$$

†Corresponding authors

For brevity, we only focus on the numerator in Eq. 4, which can be written as:

$$(\bar{v}_r - \tilde{v}_{r,1})^T (\bar{v}_r - \tilde{v}_{r,2}) = \bar{v}_r^T \bar{v}_r - \bar{v}_r^T \tilde{v}_{r,2} - \tilde{v}_{r,1}^T \bar{v}_r + \tilde{v}_{r,1}^T \tilde{v}_{r,2}. \quad (5)$$

From Eq. 1,  $\tilde{v}_{r,1}^T \tilde{v}_{r,2}$  can be derived by:

$$\tilde{v}_{r,1}^T \tilde{v}_{r,2} = 2\cos^2 \theta - 1. \quad (6)$$

From Eq. 2 and Eq. 6, we can derive  $\bar{v}_r^T \bar{v}_r$ ,  $\bar{v}_r^T \tilde{v}_{r,2}$  and  $\tilde{v}_{r,1}^T \bar{v}_r$  separately:

$$\begin{aligned} \bar{v}_r^T \bar{v}_r &= \frac{(\tilde{v}_{r,1} + \tilde{v}_{r,2})^T (\tilde{v}_{r,1} + \tilde{v}_{r,2})}{\|\tilde{v}_{r,1} + \tilde{v}_{r,2}\|^2} (\sin \theta + \cos \theta)^2 \\ &= (\sin \theta + \cos \theta)^2, \\ \bar{v}_r^T \tilde{v}_{r,2} &= \frac{(\tilde{v}_{r,1} + \tilde{v}_{r,2})^T \tilde{v}_{r,2}}{\|\tilde{v}_{r,1} + \tilde{v}_{r,2}\|} (\sin \theta + \cos \theta) \\ &= \frac{(\tilde{v}_{r,1}^T \tilde{v}_{r,2} + 1)}{\sqrt{2 + 2\tilde{v}_{r,1}^T \tilde{v}_{r,2}}} (\sin \theta + \cos \theta) \\ &= \frac{(2\cos^2 \theta - 1 + 1)}{\sqrt{2 + 2(2\cos^2 \theta - 1)}} (\sin \theta + \cos \theta) \\ &= \cos \theta (\sin \theta + \cos \theta), \\ \tilde{v}_{r,1}^T \bar{v}_r &= \frac{\tilde{v}_{r,1}^T (\tilde{v}_{r,1} + \tilde{v}_{r,2})}{\|\tilde{v}_{r,1} + \tilde{v}_{r,2}\|} (\sin \theta + \cos \theta) \\ &= \frac{(1 + \tilde{v}_{r,1}^T \tilde{v}_{r,2})}{\sqrt{2 + 2\tilde{v}_{r,1}^T \tilde{v}_{r,2}}} (\sin \theta + \cos \theta) \\ &= \frac{(1 + 2\cos^2 \theta - 1)}{\sqrt{2 + 2(2\cos^2 \theta - 1)}} (\sin \theta + \cos \theta) \\ &= \cos \theta (\sin \theta + \cos \theta). \end{aligned} \quad (7)$$

Thus, we can proceed to derive:

$$\begin{aligned}
(\bar{v}_r - \tilde{v}_{r,1})^T (\bar{v}_r - \tilde{v}_{r,2}) &= \bar{v}_r^T \bar{v}_r - \bar{v}_r^T \tilde{v}_{r,2} - \tilde{v}_{r,1}^T \bar{v}_r + \tilde{v}_{r,1}^T \tilde{v}_{r,2} \\
&= (\sin \theta + \cos \theta)^2 - \cos \theta (\sin \theta + \cos \theta) \\
&\quad - \cos \theta (\sin \theta + \cos \theta) + 2 \cos^2 \theta - 1 \\
&= 1 + 2 \sin \theta \cos \theta - 2 \cos^2 \theta - 2 \sin \theta \cos \theta \\
&\quad + 2 \cos^2 \theta - 1 \\
&= 0,
\end{aligned} \tag{8}$$

which can be used to derive:

$$u_{r,1}^T u_{r,2} = \frac{(\bar{v}_r - \tilde{v}_{r,1})^T (\bar{v}_r - \tilde{v}_{r,2})}{\|\bar{v}_r - \tilde{v}_{r,1}\| \|\bar{v}_r - \tilde{v}_{r,2}\|} = 0. \tag{9}$$

Therefore,  $u_{r,1}$  and  $u_{r,2}$  are orthogonal.

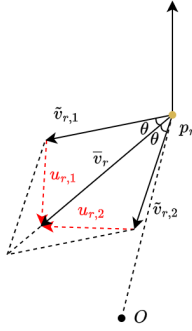


Figure 1. Illustration of LCRF.

## A.2. Rotation Invariance in Eq. 13 of the main text

To prove the rotation invariance in Eq. 13 of the main text, we begin with analyzing how LCRF introduces rotation invariance since the local rotation-invariant representation  $U_r^T p$  is the input of the equation. The detailed proof is shown as follows.

**Theorem 2.** For any reference point  $p_r$  and its  $K$  neighbors  $\{p_j\}_{j=1}^K$ , the equation defined as follows can achieve rotation invariance under arbitrary rotation  $R \in \text{SO}(3)$ :

$$x_r = \max_{j \in \mathcal{N}(p_r)} \psi(U_r^T p_r, U_r^T (p_j - p_r)), \tag{10}$$

where  $U_r$  is LCRF,  $\mathcal{N}(\cdot)$  denotes the KNN operation invariant to rotation, and  $\psi$  represents a MLP.

*Proof.* For a point  $p \in P$ , when applying a random rotation matrix  $R$  to it, for an equivariant feature  $v$ , we have:

$$v^* = h(Rp) = Rh(p) = Rv, \tag{11}$$

where  $h(\cdot)$  is the equivariant network and the superscript  $*$  indicates that the result corresponds to the rotated input.  $\tilde{v}_r = [\tilde{v}_{r,1}, \tilde{v}_{r,2}]$  is the equivariant feature used to construct LCRF, so it also satisfies:

$$\tilde{v}_r^* = R\tilde{v}_r = [R\tilde{v}_{r,1}, R\tilde{v}_{r,2}]. \tag{12}$$

Therefore, for Eq. 1, Eq. 2 and Eq. 3 we have:

$$\begin{aligned}
\sin^* \theta &= \sqrt{\frac{1 - (R\tilde{v}_{r,1})^T R\tilde{v}_{r,2}}{2}} = \sqrt{\frac{1 - \tilde{v}_{r,1}^T \tilde{v}_{r,2}}{2}} = \sin \theta, \\
\cos^* \theta &= \sqrt{\frac{1 + (R\tilde{v}_{r,1})^T R\tilde{v}_{r,2}}{2}} = \sqrt{\frac{1 + \tilde{v}_{r,1}^T \tilde{v}_{r,2}}{2}} = \cos \theta,
\end{aligned} \tag{13}$$

$$\begin{aligned}
\bar{v}_r^* &= \frac{(R\tilde{v}_{r,1} + R\tilde{v}_{r,2})}{\|R\tilde{v}_{r,1} + R\tilde{v}_{r,2}\|} (\sin^* \theta + \cos^* \theta) \\
&= \frac{R(\tilde{v}_{r,1} + \tilde{v}_{r,2})}{\|\tilde{v}_{r,1} + \tilde{v}_{r,2}\|} (\sin \theta + \cos \theta) \\
&= R\bar{v}_r,
\end{aligned} \tag{14}$$

$$\begin{aligned}
u_{r,1}^* &= \frac{\bar{v}_r^* - R\tilde{v}_{r,1}}{\|\bar{v}_r^* - R\tilde{v}_{r,1}\|} = \frac{R(\bar{v}_r - \tilde{v}_{r,1})}{\|\bar{v}_r - \tilde{v}_{r,1}\|} = Ru_{r,1}, \\
u_{r,2}^* &= \frac{\bar{v}_r^* - R\tilde{v}_{r,2}}{\|\bar{v}_r^* - R\tilde{v}_{r,2}\|} = \frac{R(\bar{v}_r - \tilde{v}_{r,2})}{\|\bar{v}_r - \tilde{v}_{r,2}\|} = Ru_{r,2}.
\end{aligned} \tag{15}$$

For LCRF  $U_r = [u_{r,1}, u_{r,2}, u_{r,1} \times u_{r,2}]$ , it satisfies:

$$\begin{aligned}
U_r^* &= [u_{r,1}^*, u_{r,2}^*, u_{r,1}^* \times u_{r,2}^*] \\
&= [Ru_{r,1}, Ru_{r,2}, Ru_{r,1} \times Ru_{r,2}] \\
&= RU_r,
\end{aligned} \tag{16}$$

which can be used to achieve rotation invariance through:

$$U_r^{*T} R p = (RU_r)^T R p = U_r^T R^T R p = U_r^T p. \tag{17}$$

Hence, we can derive:

$$\begin{aligned}
x_r^* &= \max_{j \in \mathcal{N}(Rp_r)} \psi(U_r^{*T} R p_r, U_r^{*T} (R p_j - R p_r)) \\
&= \max_{j \in \mathcal{N}(p_r)} \psi(U_r^T p_r, U_r^T (p_j - p_r)) \\
&= x_r,
\end{aligned} \tag{18}$$

which proves Eq. 10 can achieve rotation invariance.

## B. Additional Experiments

### B.1. Semantic Segmentation

**Dataset.** The S3DIS [1] dataset consists of 6 large-scale indoor areas with 271 rooms in 13 categories. Following previous work [7], we select Area-5 for testing, while the other five areas are used for training.

**Results.** Given that few works perform experiments on S3DIS, we compare our method with our backbone. As shown in Tab. 1, DGCNN [8] struggles to process complex scene-level data under arbitrary rotations, and our method outperforms it by a large margin, especially in the z/SO(3) setting. The results indicate that our LocoTrans can also work on large-scale point cloud data.

Method	Input	z/SO(3)		SO(3)/SO(3)	
		mIOU	Accuracy	mIOU	Accuracy
DGCNN [8]	pc	3.0	18.1	42.8	80.7
Ours	pc	54.2	84.8	56.0	84.7

Table 1. Semantic segmentation results on S3DIS dataset, which are reported by mIOU (%) and accuracy (%) separately.

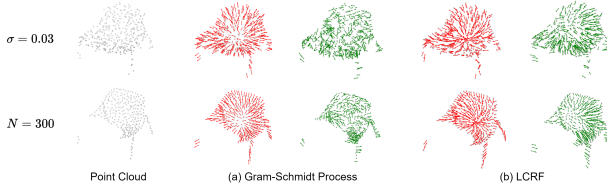


Figure 2. Visualization of  $u_r^1$  (Red) and  $u_r^2$  (Green) in LRF under perturbation.

$\sigma$	0	0.01	0.02	0.03	$N$	0	100	200	300
	G-S	91.3	85.9	74.0		46.0	G-S	91.3	87.1
LCRF	91.6	86.8	77.1	48.5	LCRF	91.6	87.2	80.3	72.4

Table 2. Classification accuracy (%) under different scales of noises  $\sigma$  (Left) and number of dropout points  $N$  (Right) on ModelNet40 z/SO(3) setting. G-S represents LRF built from Gram-Schmidt process.

## B.2. Performance under Various Perturbations

In Tab. 2, we evaluate the performance of our LCRF and LRF built from Gram-Schmidt process under perturbations by introducing noise and random dropout. Despite both methods showing decreased performance under perturbations, our LCRF consistently outperforms Gram-Schmidt process. Moreover, the visualization in Fig. 2 shows that LCRF still maintains a degree of local consistency in both  $u_r^1$  (Red) and  $u_r^2$  (Green). We further try to introduce noises during training to improve the robustness on noised data, getting 88.7% with our LCRF and 88.3% with LRF from Gram-Schmidt process for  $\sigma = 0.03$ . It shows training with noises benefits the performance on noised data and our LCRF still performs better.

## B.3. Robustness to the Number of Neighbours

The number of neighbors in KNN operation determines the scope of local feature extraction. We conduct experiments to investigate the robustness of LocoTrans to the number of neighbors. As shown in Tab. 3, our method achieves the best performance with  $K = 20$ .

Number of neighbors	$K = 10$	$K = 20$	$K = 40$
Ours	91.3	91.6	90.8

Table 3. Classification accuracy (%) under different neighbor size  $K$  on ModelNet40 z/SO(3) setting.

## B.4. More Backbones

We further conduct experiments to investigate the robustness of other mainstream point cloud analysis models to rotations and explore the effects of LocoTrans on these backbones. In Tab. 4, we introduce three networks: 1) PointNet++ [6], a classic framework using PointNet [5] as local extractors, 2) the attention-based method PCT [3], which captures relationships between points well, and 3) PointMLP [4], which gives a pure residual MLP to replace sophisticated local extractors. Although these backbones can achieve state-of-the-art on the well-aligned data, they fail to accurately classify the rotated data in z/SO(3) setting. In contrast, their performance is significantly improved when using our LocoTrans. The results show that mainstream point cloud models lack rotation robustness, and LocoTrans can effectively address this issue.

Row	Backbone	z/SO(3)	
		w/o LocoTrans	w LocoTrans
#1	PointNet++ [6]	28.6	90.7
#2	PCT [3]	25.6	90.7
#3	PointMLP [4]	31.1	91.1

Table 4. Classification accuracy (%) on ModelNet40 z/SO(3) setting. ‘w/o LocoTrans’ denotes not using LocoTrans while ‘w LocoTrans’ represents applying LocoTrans on these backbones.

## B.5. Model Output

Here we analyze three types of outputs from the invariant branch, the equivariant branch, and fusion in our network. From Tab. 5, we can see fusion can significantly improve performance under different datasets and different settings. In addition, for z/SO(3) and SO(3)/SO(3) settings, both invariant branch and equivariant branch achieve similar performance in ModelNet40 while they suffer from performance fluctuation in ScanObjectNN, especially the equivariant branch. We guess the reason is that we use the vector-based equivariant network [2] as our equivariant branch, which combines input vectors linearly to achieve equivariance and thus has limited learning ability. Hence, facing changes in randomness brought by different rotations in two settings, equivariant branch cannot achieve relatively stable and consistent performance in ScanObjectNN containing background noise.

## C. Limitation

Although efficient, our approach may encounter a potential challenge when handling real-world data containing background noise. LocoTrans relies on equivariant features and as a result, its performance might be influenced by the efficacy of the equivariant branch. As mentioned in Sec. B.5,

Source	ModelNet40		ScanObjectNN	
	z/SO(3)	SO(3)/SO(3)	z/SO(3)	SO(3)/SO(3)
invariant branch	90.6	90.4	82.3	83.1
equivariant branch	90.2	90.3	79.2	76.9
fusion	91.6	91.5	85.0	84.5

Table 5. Classification accuracy (%) on ModelNet40 dataset and ScanObjectNN dataset.

due to limited learning ability and changes in randomness, our equivariant branch suffers from performance fluctuation under z/SO(3) and SO(3)/SO(3) settings in ScanObjectNN, hindering our network from yielding better performance on both settings. In future work, we aim to improve the equivariant branch to address this issue.

## D. Visualization

Here, we provide additional visualization results of LRF built by the Gram-Schmidt process and our LCRF in Fig. 3(a) and Fig. 3(b) separately. The results demonstrate that our method achieves local consistency along different axes (Red and Green), while the previous LRF only works effectively along one axis (Red).

## References

- [1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [2] 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, 2021. 3
- [3] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 3
- [4] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 3
- [5] 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 (CVPR)*, 2017. 3
- [6] 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*. Curran Associates, Inc., 2017. 3
- [7] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2
- [8] 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)*, 2019. 2, 3

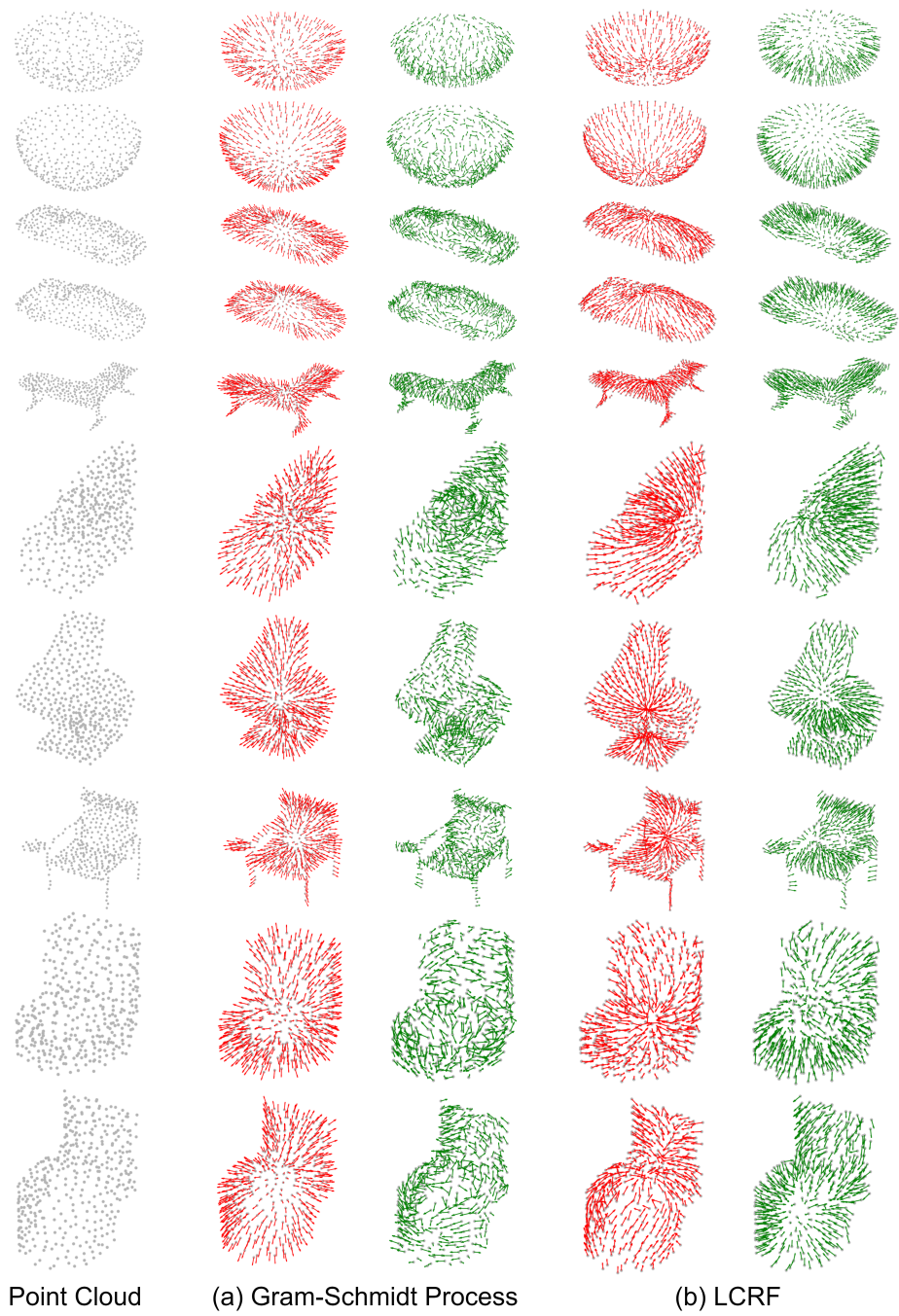


Figure 3. Visualization of the learned orientations for the LRF built by the Gram-Schmidt process and our LCRF.