Supplementary Material for A Closer Look at Rotation-invariant Deep Point Cloud Analysis (ICCV 2021)

Feiran Li[1] Kent Fujiwara[2] Fumio Okura[1] Yasuyuki Matsushita[1]

[1]Osaka University [2] LINE Corporation

1. Visualization of ambiguities

As claimed, there are 4 (sign ambiguities) $\times$ 6 (order ambiguities) $=$ 24 ambiguities when PCA is used for retrieving the canonical pose of a given point cloud. In this supplementary material, we present a detailed visualization of the 24 ambiguities of the PCA-based canonical poses as well as the 4 and 6 counterparts obtained with order-disambiguation and sign-disambiguation, respectively. The results are presented in Fig. 1.

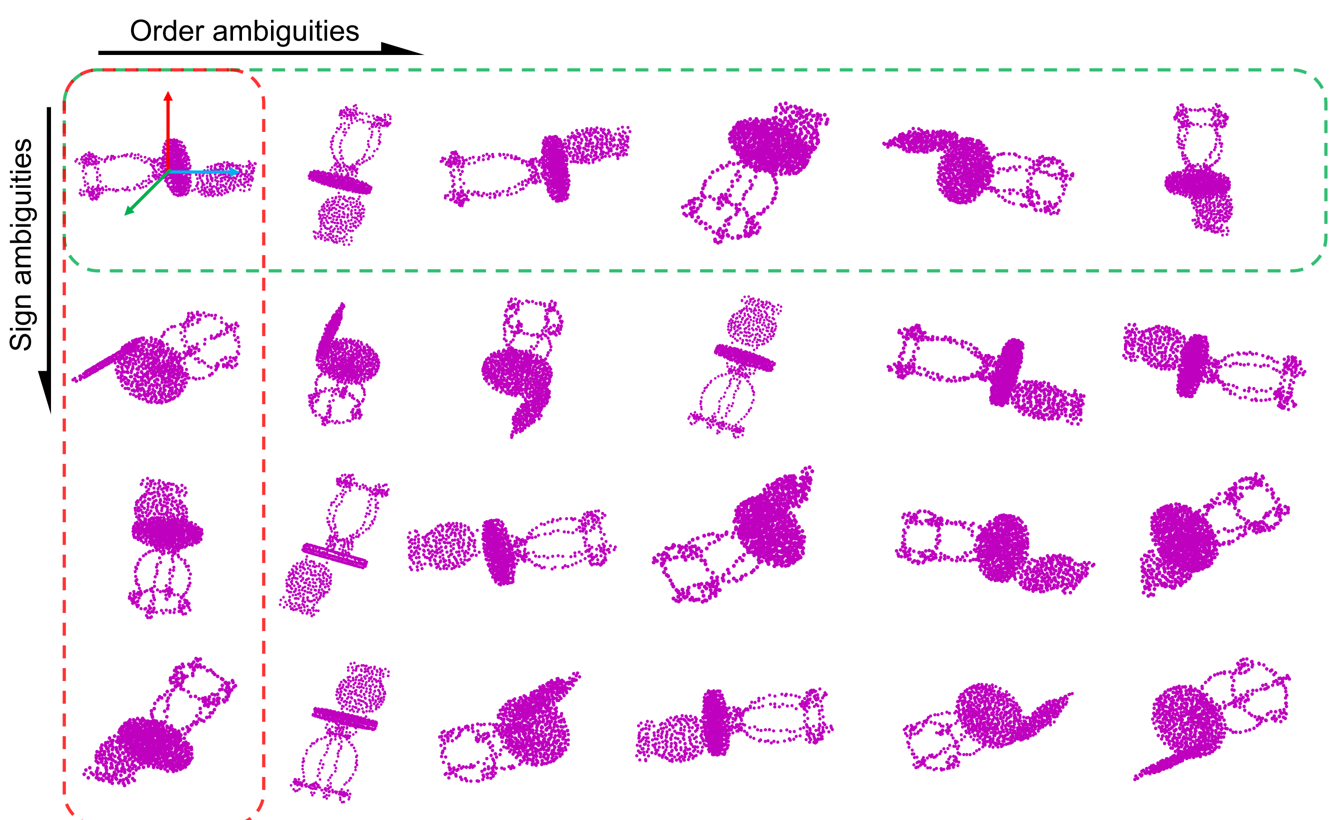

Figure 1: Visualization of the 24 ambiguities of the PCA-based canonical poses of a chair. The top row and left column respectively consist of the 6 order ambiguities and 4 sign ambiguities as mentioned in Section 6.2 of the main paper.

	Pre-aligned pose	Data augmentation with random rotations	Disambiguation with Strategy III	All possible canonical poses
PointNet [5]	88.5	70.5	82.8	86.7
PointConv [9]	92.4	85.6	86.7	89.1
DGCNN [8]	92.9	81.1	87.2	91.6

Table 1: Classification accuracy (%) w.r.t. distinct networks. Strategy III stands for the disambiguation method described in Section 6.2 of the main paper, with which all the ambiguities are eliminated.

Method	Overall mIoU	aero	bag	cap	car	cha.	earph.	guit.	knif.	lam.	lap.	mot.	mug	pist.	rock.	ska.	tab.
PointNet [5]	74.4	81.6	68.7	74.0	70.3	87.6	68.5	88.9	80.0	74.9	83.6	56.5	77.6	75.2	53.9	69.4	79.9
PointNet++ [6]	76.7	79.5	71.6	**87.7**	70.7	88.8	64.9	88.8	78.1	79.2	**94.9**	54.3	92.0	76.4	50.3	68.4	81.0
PointCNN [4]	71.4	78.0	80.1	78.2	68.2	81.2	70.2	82.0	70.6	68.9	80.8	48.6	77.3	63.2	50.6	63.2	**82.0**
DGCNN [8]	73.3	77.7	71.8	77.7	55.2	87.3	68.7	88.7	85.5	81.8	81.3	36.2	86.0	**77.3**	51.6	65.3	80.2
RIConv [11]	75.5	80.6	80.2	70.7	68.8	86.8	70.4	87.2	84.3	78.0	80.1	57.3	91.2	71.3	52.1	66.6	78.5
Li *et al*. [3]	82.5	81.4	**84.5**	85.1	75.0	88.2	72.4	**90.7**	84.4	80.3	84.0	**68.8**	**92.6**	76.1	52.1	74.1	80.0
LGR-Net [12]	82.8	81.7	78.1	82.5	**75.1**	87.6	**74.5**	89.4	86.1	**83.0**	86.4	65.3	**92.6**	75.2	**64.1**	**79.8**	80.5
Ours (w/o TTA)	81.7	81.9	58.2	77.0	71.8	89.6	64.2	89.1	85.9	80.7	84.7	46.8	89.1	73.2	45.6	66.5	81.0
Ours (w/ TTA)	**83.1**	**83.7**	62.9	79.1	73.4	**90.1**	64.2	90.3	**86.4**	82.5	87.3	46.5	89.1	75.4	46.1	66.6	81.3

Table 2: Per-class mIoU (%) of the $\mathrm{SO}(3)/\mathrm{SO}(3)$ setup. RI-GCN [2], Triangle-Net [10] and SRI-Net [7] are omitted since the per-class mIoUs are not reported in their respective paper.

2. Additional experiments

In this section, we conduct more experiments to test how different network designs could affect the performance when the PCA-based canonical poses are used as the inputs. The elapsed time of the proposed pose selector module and the per-category mIoU on the ShapeNet part segmentation task are also reported.

2.1. Effect of different backbone networks

We carry out experiments to study how different network structures can affect performance. For experimental setup, we select PointNet [5], PointConv [9], and DGCNN [8] as the backbones and conduct classification on ModelNet40. The results are presented in Table 1. As shown in the table, the performance of all the networks can be boosted by providing PCA-based canonical poses with correctly handled ambiguities.

2.2. Effect of the spatial transform networks

Spatial transform networks (STN) [1] are self-attention modules designed to take the point cloud as input and learn to apply transformations on it. In general, such a module can effectively handle small rotational perturbations but is weak for large ones. Therefore, although STN is commonly mentioned as a component for various existing network structures [5, 6, 8], it is often omitted in realistic implementation for being either redundant (on pre-aligned data) or ineffective (on randomly rotated data). However, we find that STN does contribute when PCA-based canonical poses are used as the inputs. Specifically, the classification accuracy declines from 91.6% to 89.7% if STN is omitted. This result agrees with our observation that PCA can achieve rough alignments with small rotational disorders. Therefore, we consider STN as a necessary module for complementing PCA-based canonical poses.

2.3. Time consumption of the pose selector module

As mentioned in Section 4 of the main paper, our proposed pose selector is a lightweight module that barely increases the computation time for inference. For demonstration, we record the time required for classifying a point cloud of shape (1024×3) with both the vanilla DGCNN network and the one that consists of the pose selector. In our test, the former version takes 6.03 milliseconds to infer a certain input pose and the latter requires 6.51 milliseconds on a Tesla V100 GPU. The main time consumption lies in the data transfer between CPU and GPU.

2.4. Per-category mIoU on ShapeNet

We report the per-category mIoU of the ShapeNet part segmentation task in Table 2. We note that, different from the other methods, the results originally reported in Li *et al*. [3] and LGR-Net [12] in their papers are the mean values of the mIoU over all categories. Therefore, we recalculate their mIoU over all the instances using their reported per-category mIoU and summarize these converted results for a straightforward comparison.

We also show some examples of segmentation results from each class in Figs. 2 and 3. We compare our results with the ground truth and the results obtained by using only one of the poses that is randomly selected. As shown in the figures, some of the shapes can be correctly segmented just by using one of the poses. However, for some shapes, chairs for example (right side of the first row in Fig. 2), wrong classes are assigned. For other shapes such as laptops and tables (left side of the second row, right side of the last row in Fig. 3, respectively), the labels are reversed, showing the negative effects of random orientation. On the other hand, these shapes are successfully segmented into correct parts using our pose selector, despite their varying poses. Our method especially excels at shapes whose self-symmetry is prominent.

References

[1] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015. 2

[2] SeoHyun Kim, JaeYoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2020. 2

[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. *arXiv preprint arXiv:2003.07238*, 2020. 2

[4] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on χ-transformed points. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2018. 2

[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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[6] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[7] Xiao Sun, Zhouhui Lian, and Jianguo Xiao. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In *Proceedings of International Conference on Multimedia (ACMMM)*, 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. *Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 2

[9] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[10] Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[11] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *Proceedings of International Conference on 3D Vision (3DV)*, 2019. 2

[12] 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. 2

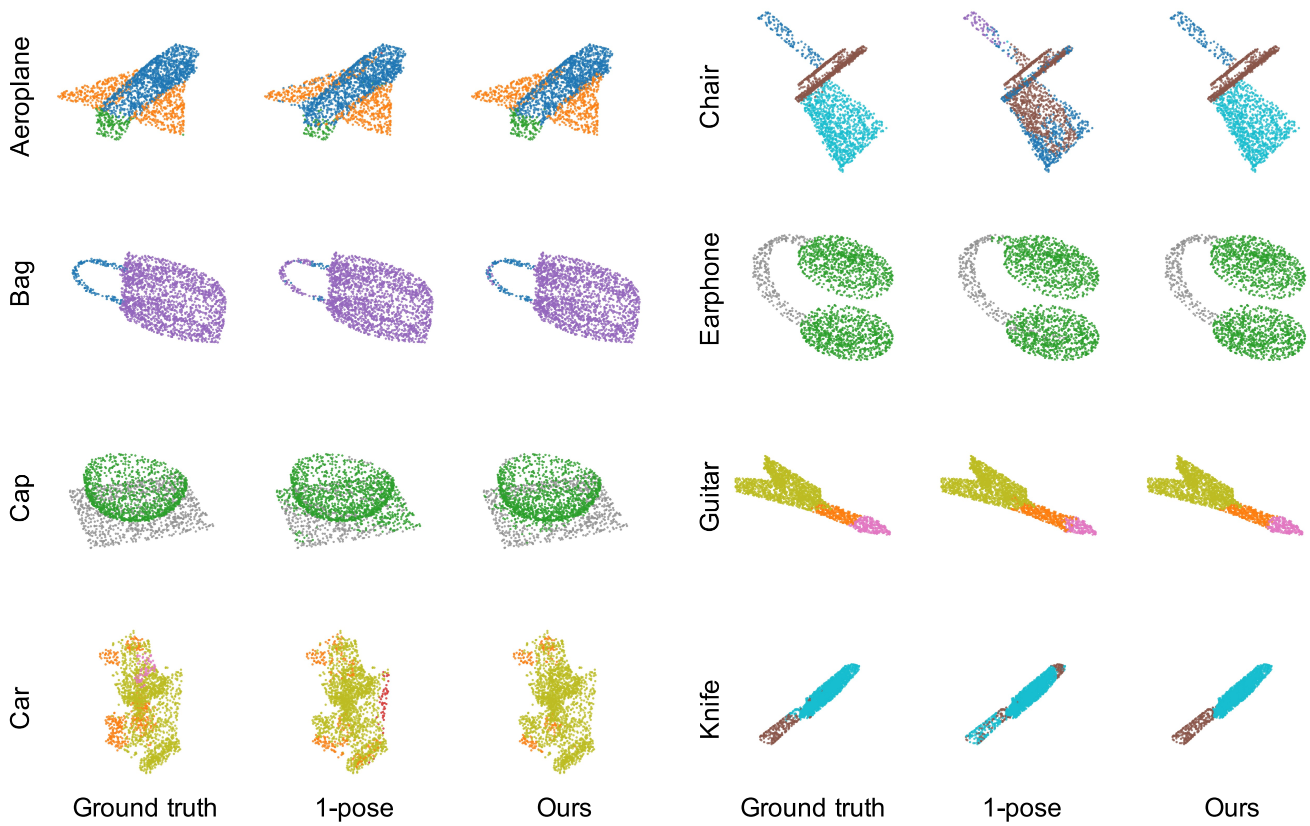

Figure 2: Visualization of the ground truth segmentation, results using 1 of the 24 possible poses, and our results from the former 8 classes in ShapeNet part segmentation dataset. The poses are drawn from the 24 possible PCA-based canonical poses.

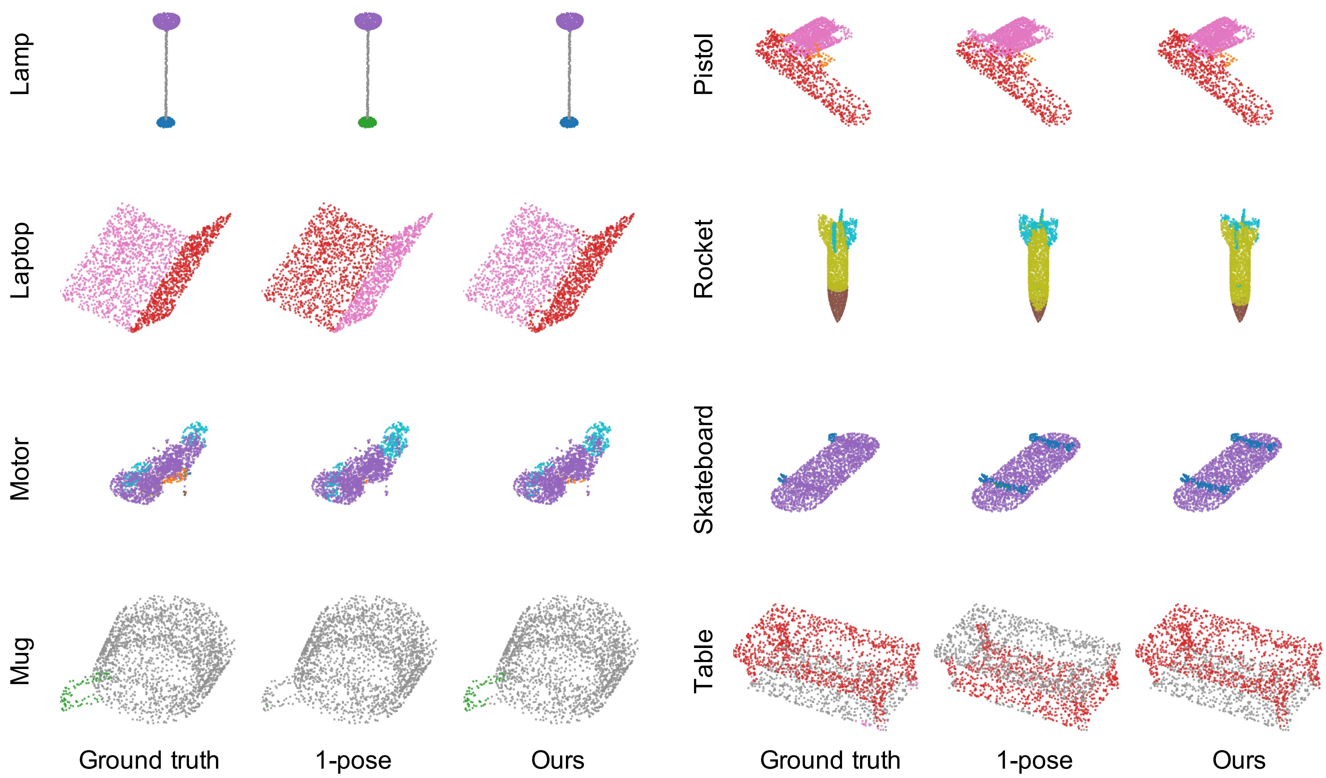

Figure 3: Visualization of the ground truth segmentation, results using 1 of the 24 possible poses, and our results from the latter 8 classes in ShapeNet part segmentation dataset. The poses are drawn from the 24 possible PCA-based canonical poses.