Extended Abstract Track

SeLCA: Self-Supervised Learning of Canonical Axis

Editors: List of editors' names

Abstract

Robustness to rotation is critical for point cloud understanding tasks, as point cloud features can be affected dramatically with respect to prevalent rotation changes. In this work, we propose SeLCA, a novel self-supervised learning framework to learn to the canonical axis of point clouds in a probabilistic manner. In essence, we propose to *learn* rotational-equivariance by predicting the canonical axis of point clouds, and achieve rotational-invariance by aligning the point clouds using their predicted canonical axis. When integrated into a rotation-sensitive pipeline, SeLCA achieves competitive performances on the ModelNet40 classification task under unseen rotations. Most interestingly, our proposed method also shows high robustness to various real-world point cloud corruptions presented by the ModelNet40-C dataset, compared to the state-of-the-art rotation-invariant method.

Keywords: SO(3)-equivariance, Point cloud canonical axis, Point cloud understanding, Rotation Equivariance, Rotation Invariance

1. Introduction

With the recent advances in deep learning, there have been successful attempts to reason about 3D geometric data *e.g.*point clouds (Qi et al., 2017a,b; Wang et al., 2019; Wu et al., 2019). However, presenting consistent inferences with respect to the varying rotation transformations on point clouds remains a challenging problem. Existing approaches which attempt to tackle this issue can be largely divided into a) methods which rely on rotation-equivariant networks or invariant networks (Qi et al., 2017c; Xiao and Wachs, 2021; Rao et al., 2019), and b) methods which aim to align point clouds to their canonical pose using principal component analysis (F.R.S., 1901) to achieve rotation invariance (Li et al., 2021a).

In this paper, we focus on a major drawback of existing methods that has received little attention - their brittleness to real-world corruptions including occlusion, cutout and background noise. Existing methods either have a strong reliance on the plane symmetries of point clouds (Li et al., 2021a; Kim et al., 2020) or the intrinsic geometries of point clouds (Xiao and Wachs, 2021; Rao et al., 2019) which are largely affected by point cloud corruption. This limits their applications on real-world settings or even realistic large-scale point clouds, which contain multiple objects with varying point cloud corruptions.

Our contribution can be summarized as follows:

- We propose SeLCA, a novel self-supervised pipeline to predict the canonical axis of 3D point clouds,
- As core components of SeLCA, we propose a novel alignment scheme on geodesic spherical tessellations to effectively represent the canonical orientation distribution of a point cloud and model its rotational equivariance,
- We evidence the relatively superior robustness of SeLCA against realistic point cloud corruptions of the ModelNet40-C dataset,while being competitive on the ModelNet40 dataset for point cloud classification.

© 2022 .

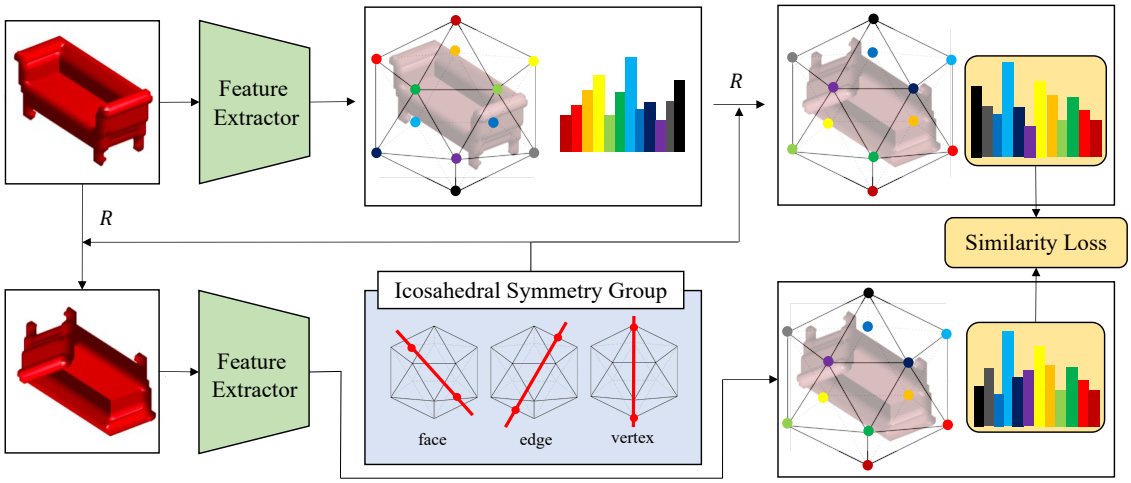

Figure 1: **Overview of the proposed SeLCA pipeline.** Our self-supervised pipeline trains the network by matching the histograms from a pair of rotated point clouds.

2. Self-supervised Learning of Canonical Axis

The main motivation behind proposing a self-supervised pipeline is the absence of standard rules to pre-define the canonical axis of a point cloud, and it is therefore impossible to provide strong supervision. To this end, we train our SeLCA pipeline to output a canonical axis distribution that satisfies the following property: when a rotation R is applied to the point cloud then the predicted canonical axis distribution should reflect the corresponding rotation by R, *i.e.* the canonical axis distribution should be equivariant to the rotation of point clouds. Figure 1 illustrates the overall training pipeline.

2.1. Geodesic Spherical Tessellation: Regular Icosahedron

We take inspiration from the axis-angle representation (Rowenhorst et al., 2015) of 3D rotation to represent our canonical axis. As we want the output of our network to represent the canonical axis distribution, we need a discrete representation (1) which can represent as many points on a unit sphere, and (2) where the bins of the representation are equispaced on the unit sphere i.e., a geodesic spherical tessellation. We propose to use a regular icosahedron to represent our canonical axis distribution, noting that the icosahedral group is the largest finite subgroup of SO(3).

2.2. Self-supervised Training Pipeline

First, we randomly select a rotation matrix from the icosahedral group to transform a point cloud. We feed the original point cloud and the rotated point cloud into our DGCNN (Wang et al., 2019) feature extraction network to extract a 12-dimensional feature for each point cloud. We apply softmax on this feature to yield a 12-bin histogram, where each bin represents a vertex on a regular icosahedron, and each bin value represents the probability

Extended Abstract Track

Table 1: **Classification accuracy (%) on the ModelNet40 dataset.** The Acc. drop column represents the drop in accuracy from SO(3)/SO(3) to z/SO(3).

	Method	Inputs	z/z	z/SO(3)	SO(3)/SO(3)	Acc. drop
Rotation-sensitive	PointNet (Qi et al., 2017a)	xyz	88.5	16.4	70.5	54.1
	DGCNN (Wang et al., 2019)	xyz	92.2	20.6	81.1	60.5
	PointNet++ (Qi et al., 2017b)	xyz	89.3	28.6	85.0	56.4
	PointConv (Wu et al., 2019)	xyz	91.6	-	85.6	-
Rotation-equivariant	a^3SCNN (Liu et al., 2019)	voxel	89.6	87.9	88.7	0.8
	SFCNN (Rao et al., 2019)	xyz	91.4	84.8	90.1	5.3
	SFCNN (Rao et al., 2019)	xyz + normal	92.3	85.3	91.0	5.7
	RotPredictor (Fang et al., 2020)	xyz	92.1	-	90.8	-
Rotation-invariant	Li et al. (2021b)	feature	89.4	89.4	89.3	0.1
	RI-GCN (Kim et al., 2020)	xyz	89.5	89.5	89.5	0.0
	RI-GCN (Kim et al., 2020)	xyz + normal	91.0	91.0	91.0	0.0
	LGR-Net (Zhao et al., 2019)	xyz + normal + feature	90.9	90.9	91.1	0.2
	Li et al. (2021a)	xyz	90.2	90.2	90.2	0.0
	Ours ($k = 1$)	xyz	88.4	86.1	87.0	0.9
	Ours ($k = 3$)	xyz	89.4	86.9	87.9	1.0

of a vector from the origin to the vertex to be the canonical axis of the point cloud. Note that we can map the selected rotation matrix (in the first step) to a permutation order of the histogram due to the closure of the icosahedral group. Then, we permute the histogram from the original point cloud using this permutation order, then supervise the two histograms - the permuted histogram from the original point cloud, and the histogram from the rotated histogram - to be close to one another, such that the canonical axis distribution of a point cloud is equivariant with respect to rotation. Formally, our loss is defined as:

$$\mathcal{L}(p_o, p_R) = -\log(\sum_{i=1}^{12} p_{o,i} p_{R,i}), \tag{1}$$

where p_o and p_R denote the output histogram bin of the original and rotated point clouds.

3. Experiments

3.1. Object classification

We evaluate our model on the ModelNet40 dataset (Wu et al., 2015) on the task of classification. The results are presented in Table 1. We largely divide existing methods to rotation-sensitive, rotation-equivariant and rotation-invariant methods. We can see that our methods achieves competitive performances compared to most rotation-equivariant methods and rotation-invariant methods despite not using equivariance-guaranteed architectures or representations. Also, we can see that the accuracy drop of our method from SO(3)/SO(3) to z/SO(3) is smaller than the majority of rotation-equivariant methods, showing that our learned rotational equivariance to model rotational invariance is empirically effective. We can also select up to top-k canonical axis candidates to perform test-time augmentation *e.g.*, $k = 3$ denotes using a soft ensemble of output results from using top-k argmax positions of the output canonical axis distribution.

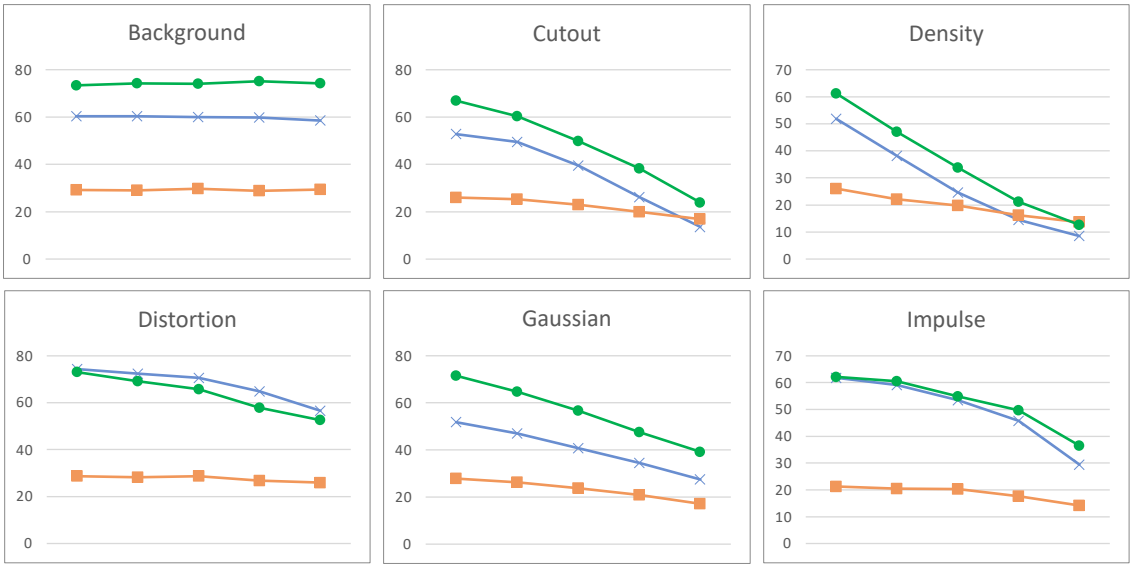

Figure 2: **Overall accuracy on the ModelNet40-C dataset.** We report the classification results of ours (green circle), Li et al. (2021a) (blue cross), and DGCNN (Wang et al., 2019) (orange square) with 6 different corruptions and five severity levels increasing from left to right.

3.2. Classification on noisy real-world datasets

We evaluate our method on the ModelNet40-C dataset (Sun et al., 2022) on the task of classification against DGCNN (Wang et al., 2019) and Li et al. (2021a), the current state-of-the-art PCA-based rotation-invariant method. The ModelNet40-C dataset is based on the ModelNet40 validation set, but augmented with 15 corruption types at 5 severity levels. We report the results for 6 types of corruptions in Figure 2. We refer the readers to the appendix for the full results. It can be seen that our method shows the highest performance overall under various types of point cloud corruptions, where our robustness is especially pronounced under background, cutout and density corruptions.

4. Discussion and Future Work

Albeit the rotational equivariance of our canonical axis distribution is not theoretically guaranteed as we do not rely on equivariant or invariant networks, the probabilistic nature of our approach enables our approach to be significantly more robust to various types of point cloud corruptions compared to existing rotation-equivariant and -invariant representations, while yielding competent results on synthetic datasets. A promising future direction would be to build on our approach to improve the empirical rotational equivariance of the predicted canonical axis distribution. Another limitation of our approach is that we only predict the canonical axis of a point cloud, but not the canonical angle around the predicted canonical axis. Correctly predicting the canonical angle together with the canonical angle is expected to achieve a significant performance improvement, which we leave for future endeavors.

References

J. Fang, D. Zhou, X. Song, S. Jin, R. Yang, and L. Zhang. Rotpredictor: Unsupervised canonical viewpoint learning for point cloud classification. In *2020 International Conference on 3D Vision (3DV)*, pages 987–996, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. doi: 10.1109/3DV50981.2020.00109. URL https://doi.ieeecomputersociety.org/10.1109/3DV50981.2020.00109.

Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2 (11):559–572, 1901. doi: 10.1080/14786440109462720.

Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. *Advances in Neural Information Processing Systems*, 33:8174–8185, 2020.

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, 2021a.

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*, pages 1–1, 2021b. doi: 10.1109/tvcg.2021.3092570. URL https://doi.org/10.1109%2Ftvcg.2021.3092570.

Min Liu, Fupin Yao, Chiho Choi, Sinha Ayan, and Karthik Ramani. Deep learning 3d shapes using alt-az anisotropic 2-sphere convolution. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkeSiiA5Fm.

Charles Qi, Hao Su, Kaichun Mo, and Leonidas Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *cvpr*, pages 77–85, 07 2017a. doi: 10.1109/CVPR.2017.16.

Charles Qi, Li Yi, Hao Su, and Leonidas Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *neurips*, 06 2017b.

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)*, July 2017c.

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.

David Rowenhorst, AD Rollett, GS Rohrer, Mike Groeber, Michael Jackson, Peter Joachim Konijnenberg, and Marc De Graef. Consistent representations of and conversions between 3d rotations. *Modelling and Simulation in Materials Science and Engineering*, 23(8): 083501, 2015.

Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*, 2022.

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.

Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *cvpr*, pages 9613–9622, 06 2019. doi: 10.1109/CVPR.2019.00985.

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.

Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 826–835, 2021.

Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. Rotation invariant point cloud classification: Where local geometry meets global topology, 2019. URL https://arxiv.org/abs/1911.00195.

Appendix A. Geometry-aware Icosahedral Permutation

We go over the details of the geometry-aware icosahedral permutation, which is applied on the output of our ATP method in correspondence to the rotation applied on the point cloud, in order to provide self-supervision. The main idea here is that each histogram bin *i.e.*, entry in our normalized output, corresponds to a single vertex on the icosahedron. Therefore, applying a rotation sampled from the icosahedral rotational symmetry group to the icosahedron results in a permutation of the vertices *i.e.*, the histogram bins. This guarantees the rotated coordinates $\hat{v}$ are a permutation of v. For further details, refer to Algo 1.

Algorithm 1 Geometry-aware Icosahedral Permutation

Original Histogram Bin: $\{h_i\}_{i=1}^{12}$
Coordinates of Icosahedron: $\{v_i\}_{i=1}^{12}$
Rotation Matrix R
Output: Permuted Histogram $\hat{h}$
for $i = \{1, 2, \cdots, 12\}$ **do**
  $\hat{v}_i = Rv_i$
  $j^* = \text{argmax}_{j=1}^{12} ||\hat{v}_i - v_j||^2$
  $\hat{h}_i = h_{j^*}$
end for

Extended Abstract Track

Appendix B. Full Results on the ModelNet40-C Dataset

In this section, we present the full set of results on the ModelNet40-C dataset, which accounts for a total of 3 settings (z/z, z/SO(3) and SO(3)/SO(3)), 15 corruption types and 5 severity levels.

The results are presented in Figure 3, Figure 4 and Figure 5. On the z/z setup (Figure 3), DGCNN Wang et al. (2019) shows the best performance overall, as the point clouds are already assumed to be aligned. While ours falls behind DGCNN, it shows a consistently high performance compared to Li *et al.*, the current state of the art, on the z/z setting. On the z/SO(3) setup, which was partially reported in the main paper, ours shows to perform the best overall compared to DGCNN and Li *et al.*. Finally, on the SO(3)/SO(3) setup, ours and DGCNN show equally competitive performances overall. On the other hand, Li *et al.* shows to fall behind, which we conjecture is due to the brittleness of principal component analysis to point cloud corruptions.

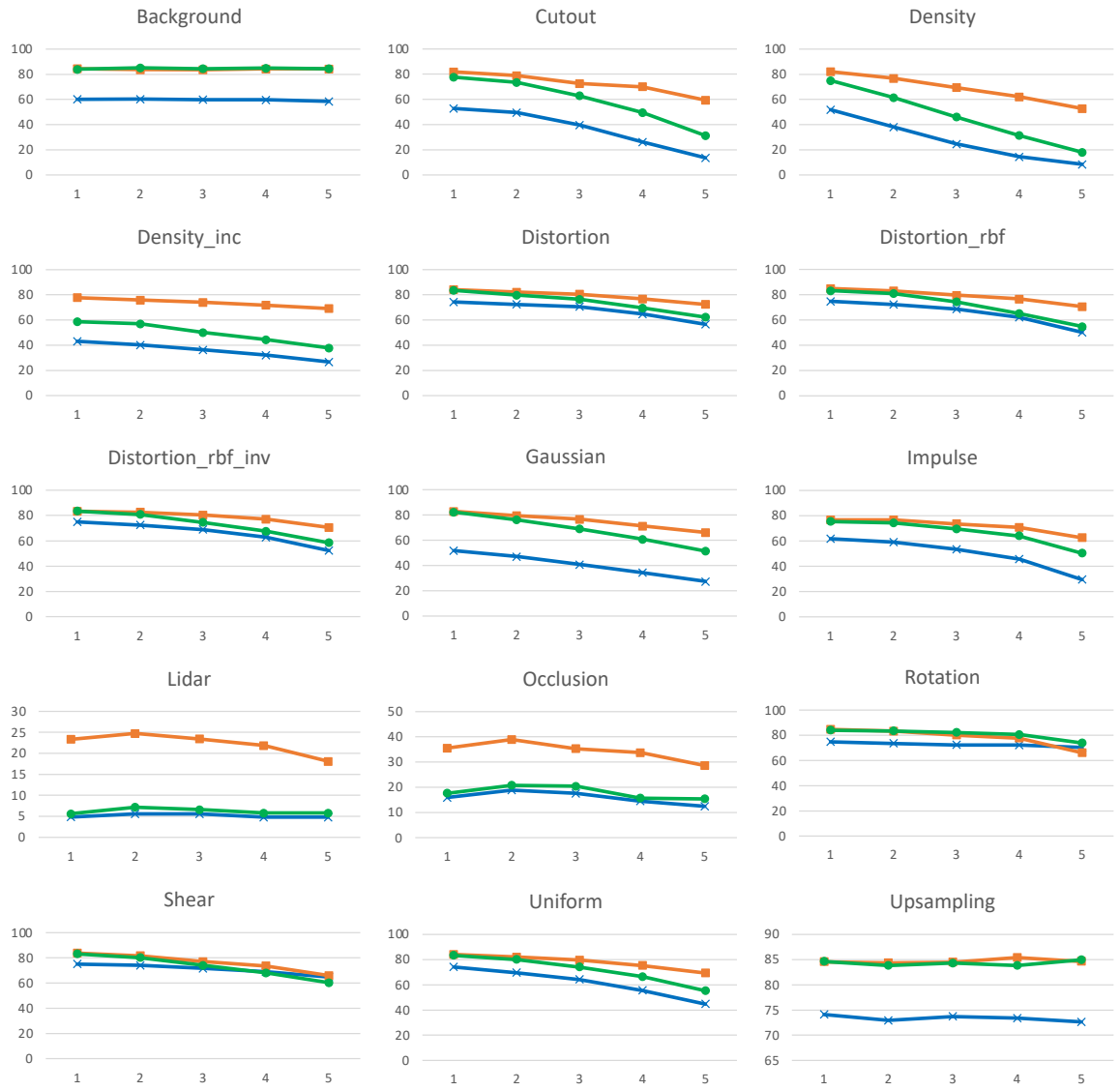

Figure 3: **Overall accuracy on the ModelNet40-C dataset using the (z/z) setting.** We report the classification results of ours (green circle), Li et al. (2021a) (blue cross), and DGCNN (Wang et al., 2019) (orange square) with 15 different corruptions and five severity levels increasing from left to right. Under most corruptions, Li et al. (2021a) shows lower performances than other methods.

Extended Abstract Track

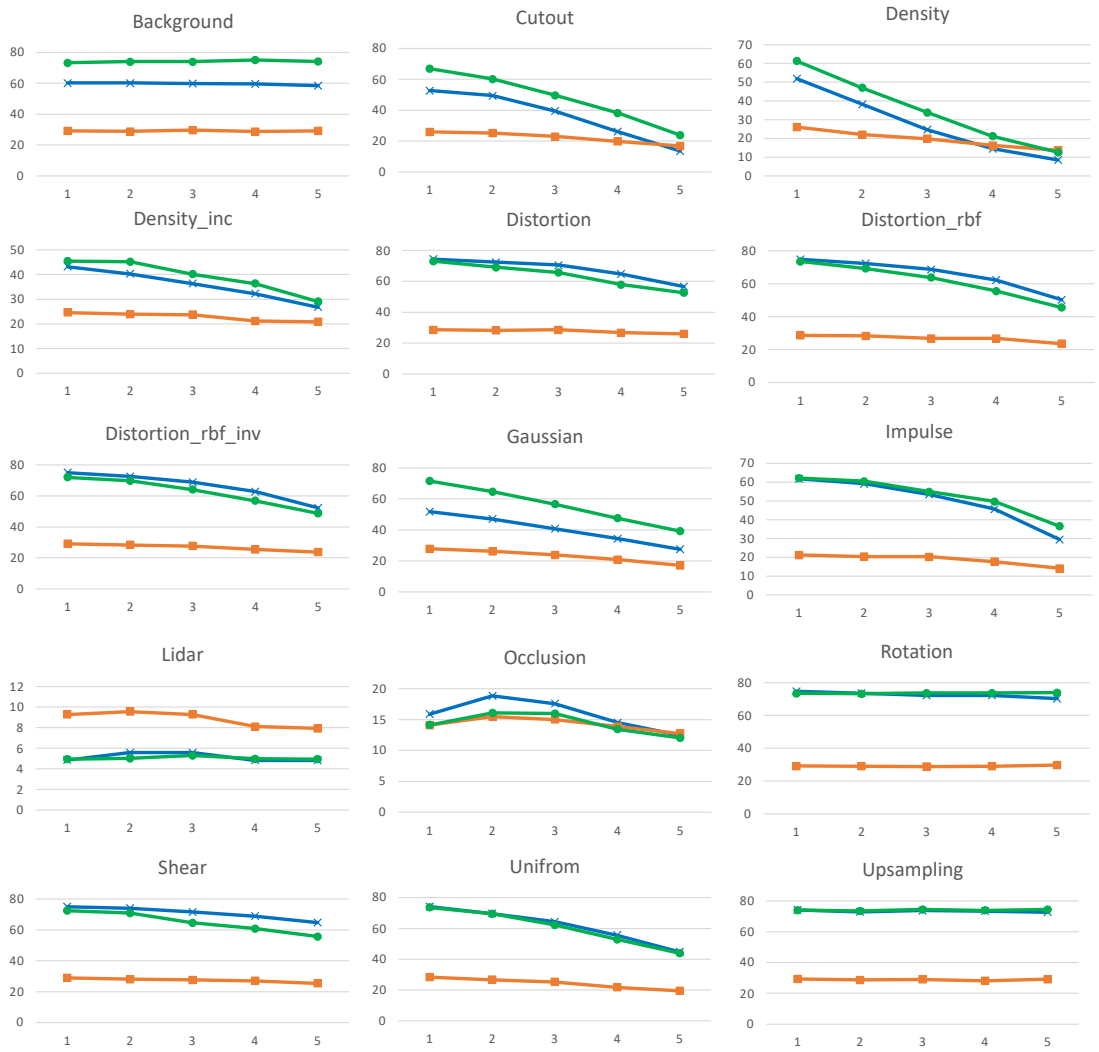

Figure 4: **Overall accuracy on the ModelNet40-C dataset using the (z/SO(3)) setting.** We report the classification results of ours (green circle), Li et al. (2021a) (blue cross), and DGCNN (Wang et al., 2019) (orange square) with 15 different corruptions and five severity levels increasing from left to right. In most corruptions, DGCNN shows much lower classification accuracy than the others. Ours shows the best performance overall.

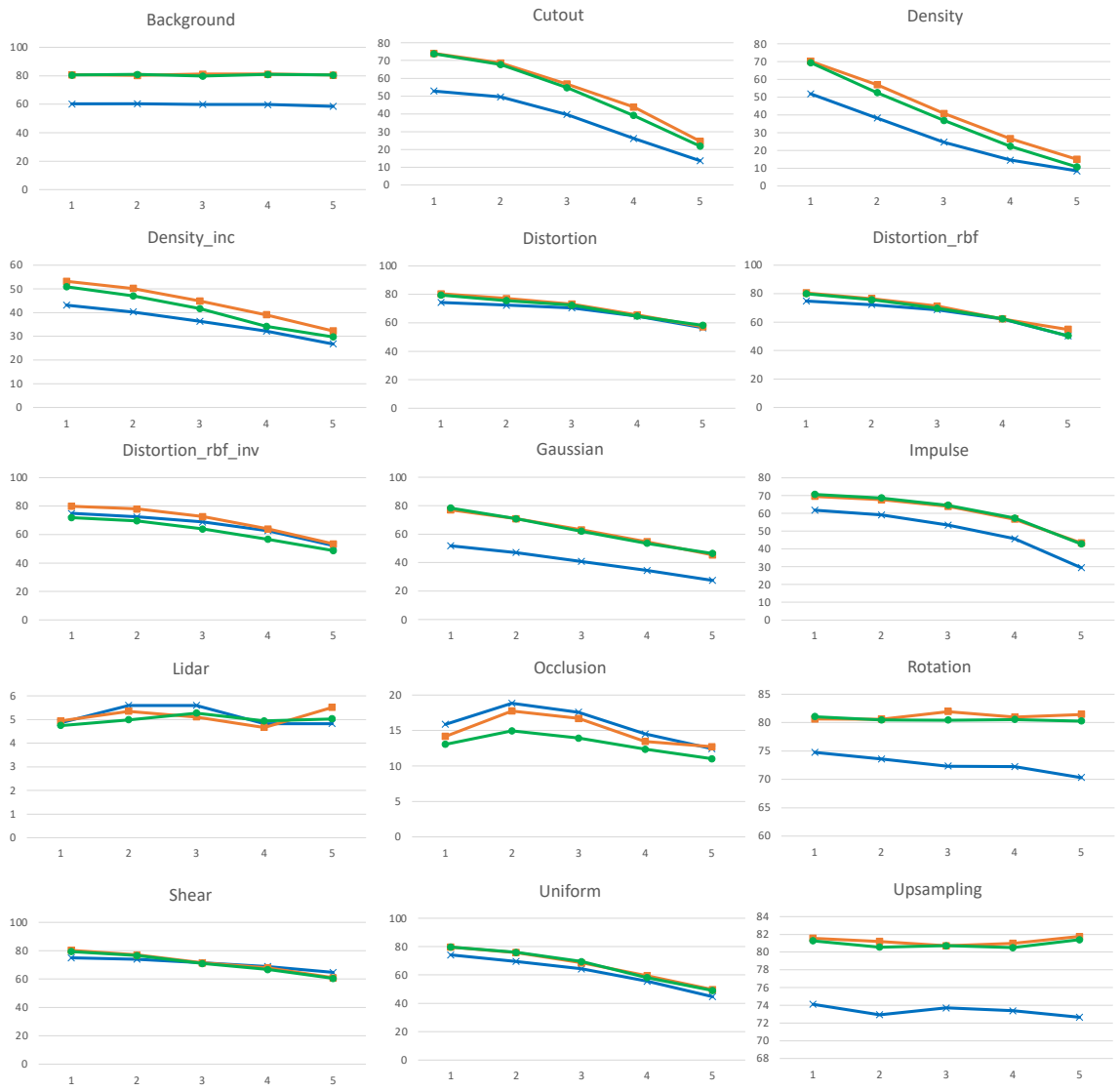

Figure 5: **Overall accuracy on the ModelNet40-C dataset using the (SO(3)/SO(3)) setup.** We report the classification results of ours (green circle), Li et al. (2021a) (blue cross), and DGCNN (Wang et al., 2019) (orange square) with 15 different corruptions and five severity levels increasing from left to right. DGCNN and ours show similar performances, while Li et al. (2021a) shows much lower classification performances under several corruption types.