# MODELNET40-C: A ROBUSTNESS BENCHMARK FOR 3D POINT CLOUD RECOGNITION UNDER CORRUPTION

**Jiachen Sun**[†], **Qingzhao Zhang**[†], **Bhavya Kailkhura**[*], **Zhiding Yu**[‡], **Z. Morley Mao**[†]
[†] University of Michigan, [*] Lawrence Livermore National Laboratory, [‡] NVIDIA

## ABSTRACT

Deep neural networks on 3D point cloud data have been widely used in the real world, especially in safety-critical applications. However, their robustness against corruptions is less studied. In this paper, we present ModelNet40-C, the first comprehensive benchmark on 3D point cloud *corruption robustness*, consisting of 15 common and realistic corruptions. Our evaluation shows a significant gap between the performances on ModelNet40 and ModelNet40-C for state-of-the-art (SOTA) models. We also demonstrate the effectiveness of different data augmentation strategies in enhancing robustness for different corruption types. We hope our in-depth analysis will motivate the development of robust training strategies or architecture designs in the 3D point cloud domain. Our codebase and dataset are included in https://github.com/jiachens/ModelNet40-C.

## 1 INTRODUCTION

Point clouds are one of the most acknowledged data format in 3D computer vision tasks, as they are inherently flexible representations and can be retrieved from a variety of sensors and computer-aided design (CAD) models. Because of these strengths, point clouds have been increasingly utilized in real-world applications.

As opposed to stellar progress on model architectures in 2D computer vision, deep 3D point cloud recognition is emerging where various architectures and operations are being proposed. Classic approaches discretize the point cloud into 3D cells, which causes cubic complexity. PointNet (Qi et al., 2017a) innovates to achieve end-to-end learning on point clouds. A few studies optimize the convolutional operation to be preferable for 3D point cloud learning (Wang et al., 2019; Liu et al., 2019b). Transformer (Vaswani et al., 2017) blocks are also applied as backbones in point cloud recognition (Guo et al., 2021). The most extensively utilized benchmark for comparing methods of point cloud recognition is ModelNet40 (Wu et al., 2015). Although the accuracy on ModelNet40 over the past several years has been steadily improved, it merely shows a single perspective of model performance on the clean data. Given the importance of 3D point cloud in the safety-critical application, a comprehensive *robustness* benchmark for point cloud recognition models is necessary.

In the literature, the vast majority of research on robustness in 3D point cloud recognition has concentrated on the critical difficulties of robustness against adversarial examples. Adversarial training has been adapted to defend against various threats to point cloud learning (Sun et al., 2020b; 2021a). However, we find that the inevitable sensor inaccuracy and physical constraints will result in a number of *common corruption* on point cloud data. For example, occlusion is a typical corruption for scanning devices, rendering partially visible point clouds. Deformation is also ubiquitous in AR/VR games. Such corruptions pose a even bigger threat in most real-world application scenarios. Thus, it is imperative to study the corruption robustness of 3D point cloud recognition.

**Summary of Our Contributions**:

In this paper, we create, to our knowledge, the *first* systematic corruption robustness benchmark, ModelNet40-C, for 3D point cloud recognition and present an in-depth analysis. To construct the dataset, we meticulously design and formulate 75 corruptions (15 types with 5 severity levels) that cover the majority of real-world point cloud distortion cases. We further provide a taxonomy of these corruptions into three categories (*i.e*, density, noise and transformation) and discuss their application scenarios. We anticipate that ModelNet40-C will serve as a first step towards 3D point cloud corruption-resistant models.

We conduct extensive evaluation on our ModelNet40-C. Specifically, we compare 9 representative models including PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), DGCNN (Wang et al.,

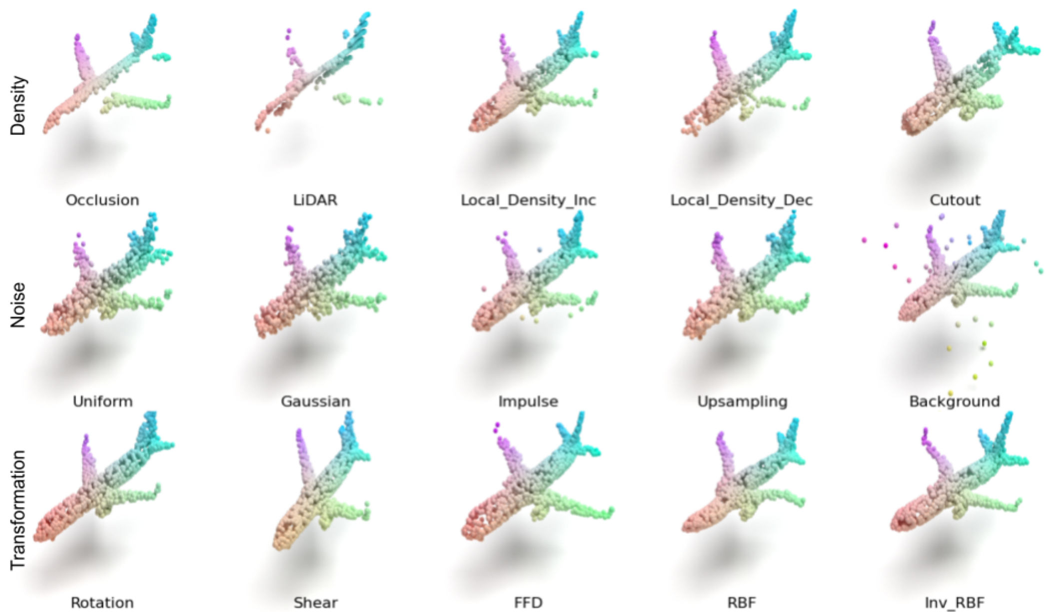


Figure 1: Visualizations of Our Constructed ModelNet40-C. Our ModelNet40-C dataset consists of 15 corruption types that represent different out-of-distribution shifts in real-world applications of point clouds. Similar to ImageNet-C (Hendrycks & Dietterich, 2019), each corruption type has 5 severity levels. We carefully examine the generated point clouds and ensure they preserve their original semantics. More visualization samples are shown in Appendix A.

2019), RSCNN (Liu et al., 2019b), PCT (Guo et al., 2021), SimpleView (Goyal et al., 2021), CurveNet (Xiang et al., 2021), GDANet (Xu et al., 2021), and PointMLP (Ma et al., 2022). We find that current models are vulnerable to our created corruptions and there are nearly 3$\times$ error rate gaps between model performances on ModelNet40 and ModelNet40-C. Our results reveal that ***there is still considerable room for point cloud recognition models to improve on robustness against common corruptions***. We also leverage data augmentation (or regularization) strategies including PointCutMix-R, PointCutMix-K (Zhang et al., 2021), PointMixup (Chen et al., 2020), RSMix (Lee et al., 2021), and adversarial training (Sun et al., 2021a) to show their potential in improving corruption robustness on our ModelNet40-C.

## 2 3D Point Cloud Corruption Robustness

In this section, we introduce the design principles of our 3D corruption benchmark. Extensive studies have been carried out to improve both architectures and training strategies for point cloud recognition on in-distribution data (Qi et al., 2017a; Wang et al., 2019; Chen et al., 2020; Lee et al., 2021). However, there has not been any systematic study on the model robustness against common corruption. To bridge this gap, we design 15 common corruptions for benchmarking *corruption robustness* of point cloud recognition models. It is worth noting that such designs are *non-trivial* since the manipulation space of 3D point clouds is completely different from 2D images where the corruptions come from the RGB modification (Hendrycks & Dietterich, 2019). In particular, we have three principles to design our benchmarks: i) Since we directly manipulate the position of points, we need to take extra care to preserve the *original semantics* of point clouds (Fig. 1). ii) we should ensure the constructed corruptions are *realistic* in various applications. iii) We should take *diversity* as an important factor to emulate a wide range of natural corruptions for 3D point clouds.

Our 15 corruption types can be naturally grouped into three categories (*i.e.*, density, noise, and transformation) , and we will introduce them in the following subsections.

### 2.1 Density Corruption Patterns

Test-time point clouds may have different density patterns from the training samples due to sensor capability and physical constraints. For example, VR scanning (in indoor scenes) and LiDAR sensors may suffer from occlusion, so that only a portion of the point cloud is visible (Geiger et al., 2012; Dai et al., 2017). Besides, the direct reflection of lasers on metal materials will cause local missing points in LiDAR point clouds (Liu et al., 2018). The local density of 3D scanned point clouds rely on how frequently the device passes that area (Nguyen & Le, 2013). We hence formulate five corruption

Table 1: Error Rates of Different Model Architectures on ModelNet40-C with Standard Training.

| | | Density Corruptions | | | | | Noise Corruptions | | | | | Transformation Corruptions | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model (%) ↓ | $\text{ER}_{\text{cor}}$ | Occlusion | LiDAR | Density Inc. | Density Dec. | Cutout | Uniform | Gaussian | Impulse | Upsampling | Background | Rotation | Shear | FFD | RBF | Inv. RBF |
| PointNet | 28.3 | 52.3 | **54.9** | **10.5** | 11.6 | 12.0 | 12.4 | 14.4 | 29.1 | 14.0 | 93.6 | 36.8 | 25.4 | 21.3 | 18.6 | 17.8 |
| PointNet++ | 23.6 | 54.7 | 66.5 | 16.0 | **10.0** | **10.7** | 20.4 | 16.4 | 35.1 | 17.2 | 18.6 | 27.6 | 13.4 | 15.2 | 16.4 | 15.4 |
| DGCNN | 25.9 | 59.2 | 81.0 | 14.1 | 17.3 | 15.4 | 14.6 | 16.6 | 24.9 | 19.1 | 53.1 | 19.1 | 12.1 | 13.1 | 14.5 | 14.0 |
| RSCNN | 26.2 | **51.8** | 68.4 | 16.8 | 13.2 | 13.8 | 24.6 | 18.3 | 46.2 | 20.1 | **18.3** | 29.2 | 17.0 | 18.1 | 19.2 | 18.6 |
| PCT | 25.5 | 56.6 | 76.7 | 11.8 | 14.3 | 14.5 | 12.1 | 13.9 | 39.1 | 17.4 | 57.9 | 18.1 | 11.5 | 12.4 | 13.0 | 12.6 |
| SimpleView | 27.2 | 55.5 | 82.2 | 13.7 | 17.2 | 20.1 | 14.5 | 14.2 | 24.6 | 17.7 | 46.8 | 30.7 | 18.5 | 17.0 | 17.9 | 17.2 |
| CurveNet | **22.7** | 55.1 | 66.0 | 10.5 | 15.3 | 13.9 | **11.7** | **13.2** | **23.7** | **11.8** | 61.0 | **15.8** | **9.8** | **10.7** | **11.4** | **10.6** |
| GDANet | 25.6 | 60.5 | 72.1 | 11.0 | 14.5 | 13.8 | 13.5 | 34.1 | 28.9 | 16.0 | 52.6 | 17.4 | 11.5 | 12.0 | 13.1 | 12.7 |
| PointMLP | 31.9 | 64.3 | 95.2 | 12.1 | 14.6 | 14.4 | 25.7 | 35.9 | 49.3 | 42.5 | 56.9 | 19.7 | 11.5 | 11.1 | 12.8 | 11.9 |
| PointMLP-Elite | 33.4 | 64.8 | 93.3 | 14.0 | 18.2 | 18.7 | 21.7 | 31.3 | 46.8 | 36.2 | 81.1 | 19.9 | 13.2 | 12.9 | 14.4 | 13.8 |
| Average | 27.0 | 57.5 | 75.6 | 13.0 | 14.6 | 14.7 | 17.1 | 20.8 | 34.8 | 21.2 | 54.0 | 23.4 | 14.4 | 14.4 | 15.1 | 14.5 |

types to cover the density corruption patterns: {`Occlusion`, `LiDAR`, `Local_Density_Inc`, `Local_Density_Dec`, `Cutout`}. Specifically, `Occlusion` and `LiDAR` both simulate occlusion patterns using ray tracing on the original meshes (Zhou et al., 2018), and `LiDAR` additionally incorporates the vertically line-styled pattern of LiDAR point clouds (Liu et al., 2018). `Local_Density_Inc` and `Local_Density_Dec` will randomly select several local clusters of points using $k$-nearest neighbors ($k$NN) to increase and decrease their density, respectively. Similarly, `Cutout` discards several randomly chosen local clusters of points using $k$NN.

### 2.2 Noise Corruption Patterns

Noise evidently exists in all real-world point cloud applications. For example, the inevitable digital noise of scanning sensors (*e.g.*, medical imaging) (Wolff et al., 2016) and the random reflections and inaccuracy of LiDAR lasers (Geiger et al., 2012) will contribute to a substantial variation of points. Compression and decompression will potentially result in noisy point clouds as well (Cao et al., 2019). Besides, real-time rendering in VR games is another source of noise (Bonatto et al., 2016). We thus formulate five noise perturbations: {`Uniform`, `Gaussian`, `Impulse`, `Upsampling`, `Background`}. As their names indicate, `Uniform` and `Gaussian` apply different distributional noise to each point in a point cloud. `Impulse` applies deterministic perturbations to a subset of points. `Upsampling` assigns new perturbation points around the existing points. `Background` randomly adds new points in the bounding box space of the pristine point cloud.

### 2.3 Transformation Corruptions Patterns

We use both linear and non-linear 3D transformations to formulate the corruptions. For the linear ones, we leverage 3D `Rotation` and `Shear` as our corruption types and exclude translation and scale transformations since they can be easily restored by normalization (*i.e.*, the inverse transformation matrix). `Rotation` of point clouds is common in the real world and the robustness against adversarial rotations has been investigated by a few studies (Zhao et al., 2020; Shen et al., 2021a). We here do not use aggressive rotations that might affect human perception as well, but instead enable a milder rotation ($\leq 15°$) along $xyz$ axes. We consider `Shear` on the $xy$ plane to represent the motion distortion in 3D point clouds (Yang et al., 2021). We utilize free-form deformation (FFD) (Sederberg & Parry, 1986) and radial basis function (RBF)-based deformation (Forti & Rozza, 2014) for non-linear transformations. Such deformations are also common in VR/AR games and point clouds from generative models (GAN) (Li et al., 2018a; Zhou et al., 2021). Specifically, we use multi quadratic ($\varphi(\boldsymbol{x}) = \sqrt{\boldsymbol{x}^2 + r^2}$) and inverse multi quadratic splines ($\varphi(\boldsymbol{x}) = (\boldsymbol{x}^2 + r^2)^{-\frac{1}{2}}$) as the representative RBFs to cover a wide range of deformation types. As a result, we in total formulate {`Rotation`, `Shear`, `FFD`, `RBF`, `Inv_RBF`} as our transformation-based corruptions.

## 3 ModelNet40-C Robustness Benchmark

**Setup**. ModelNet40 is the most popular dataset for benchmarking point cloud recognition performance, containing 12,308 point clouds from 40 classes (Wu et al., 2015). Point clouds from ModelNet40 are extracted from CAD models, rendering a perfectly clean dataset. We create ModelNet40-C with five severity levels for each corruption type, the same as ImageNet-C. Fig. 1 illustrates samples from ModelNet40-C with severity level four, and they clearly still preserve the semantics of the "airplane" class. Since it is hard to qualify and quantify the corruption severity for `LiDAR` and `Occlusion`, we instead leverage five different view angles to create their corrupted point clouds. These designed corruptions are applied to the *validation* set of ModelNet40, resulting in ModelNet40-C a 75× larger dataset to test the corruption robustness of pre-existing models.

**Metrics**. We use the error rate (ER) and class-wise mean error rate (mER) as the main metrics for ModelNet40-C benchmarking. We denote $\text{ER}^f_{\text{clean}}$ as the error rate for a classifier $f$ on the clean dataset (*i.e.*, ModelNet40) and $\text{ER}^f_{s,c}$ as the error rate for $f$ on corruption $c$ with severity $s$. Similarly,

Table 2: Error Rates of Architectures on ModelNet40-C with Different Data Augmentation Strategies.

| | Standard | PointCutMix-R | | | | PointCutMix-K | | | | PointMixup | | | | RSMix | | | | PGD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model (%) $\downarrow$ | $\mathrm{ER}_{\mathrm{cor}}$ | $\mathrm{ER}_{\mathrm{cor}}$ | Density | Noise | Trans. | $\mathrm{ER}_{\mathrm{cor}}$ | Density | Noise | Trans. | $\mathrm{ER}_{\mathrm{cor}}$ | Density | Noise | Trans. | $\mathrm{ER}_{\mathrm{cor}}$ | Density | Noise | Trans. | $\mathrm{ER}_{\mathrm{cor}}$ | Density | Noise | Trans. |
| PointNet | 28.3 | 21.8 | 30.5 | 18.0 | 16.9 | 21.3 | 26.8 | 21.8 | 15.4 | 25.4 | **28.3** | 28.9 | 19.0 | 22.5 | **24.8** | 27.3 | 15.5 | 25.9 | **28.8** | 28.4 | 20.5 |
| PointNet++ | **23.6** | 19.1 | 28.1 | 12.2 | 17.0 | 20.2 | 26.3 | 16.9 | 17.3 | **19.3** | 30.8 | **14.3** | 12.9 | 23.3 | 27.0 | 19.3 | 23.7 | - | - | - | - |
| DGCNN | 25.9 | 17.3 | 28.9 | 11.4 | 11.5 | 17.3 | 29.1 | **11.9** | **10.9** | 20.4 | 32.1 | 16.8 | 12.3 | 18.1 | 28.8 | 13.0 | **12.6** | 20.7 | 36.8 | **13.8** | 11.5 |
| RSCNN | 26.2 | 17.9 | **25.0** | 13.0 | 15.8 | 21.6 | 28.3 | 19.0 | 17.6 | 19.8 | 29.7 | 15.5 | 14.1 | 21.2 | 26.8 | 17.4 | 19.3 | - | - | - | - |
| PCT | 25.5 | **16.3** | 27.1 | **10.5** | **11.2** | **16.5** | **25.8** | 12.6 | 11.1 | 19.5 | 30.3 | 16.7 | **11.5** | **17.3** | 25.0 | **12.0** | 15.0 | **18.4** | 29.3 | 14.7 | **11.1** |
| SimpleView | 27.2 | 19.7 | 31.2 | 11.3 | 16.5 | 20.6 | 29.1 | 15.6 | 17.0 | 21.5 | 32.7 | 17.1 | 14.8 | 20.4 | 28.4 | 14.6 | 18.3 | - | - | - | - |
| Average | 26.1 | 18.7 | 28.5 | 12.7 | 14.8 | 19.6 | 27.6 | 16.3 | 14.9 | 21.0 | 30.6 | 18.2 | 14.1 | 20.5 | 26.8 | 17.3 | 17.4 | - | - | - | - |

$\mathrm{ER}_c^f = \sum_{s=1}^{5} \mathrm{ER}_{s,c}^f$ and $\mathrm{ER}_{\mathrm{cor}}^f = \sum_{c=1}^{15} \mathrm{ER}_{\mathrm{c}}^f$. We will release our leaderboard publicly to facilitate future studies on robustness of point cloud learning.

## 4 Experiments and Results

In this section, we elaborate our comprehensive evaluation and rigorous analysis in detail.

**Setup**. As mentioned in § 1, we leverage 9 representative models. These models stand for distinct architecture designs, and have achieved good accuracy on the clean dataset. They are also well-recognized by the 3D vision community, and have been extensively applied to complex tasks like semantic segmentation (Nguyen & Le, 2013) and object detection (Shi et al., 2019; 2020). We adopt the original hyper-parameter settings from the official data augmentation implementations in our study. We only enable adversarial training for PointNet, DGCNN, and PCT since the other methods will hinder the gradients from backward propagating to the original point cloud, making adversarial training inapplicable.

As presented in Table 1, there is no overarching model that dominates our ModelNet40-C dataset, unlike robustness benchmarking in 2D vision (Hendrycks & Dietterich, 2019). Point cloud recognition models have various designs and no consensus has been reached as deep learning in the 3D space is a relatively nascent field. The model performances on ModelNet40-C are found to be in good alignment with their design attributes. PointNet does not encode local feature. Such a design has been regarded as a main drawback of PointNet. However, we find it robust against the variations in density. PCT achieves a much balanced result under all corruption types by adopting self-attention modules as its backbone. CurveNet innovates advanced grouping in the graph frequency domain to the strongest robustness under standard training (ER $= 22.7\%$). To our surprise, the latest PointMLP performs the worst on ModelNet40-C, showing its overfittng to the clean data and poor generalization capability. Similarly, SimpleView cannot achieve better robustness under common corruptions than other architectures, despite it high performance on clean data, suggesting point cloud-specific designs are indeed desired.

Due to time and resource constraints, we select 6 models for data augmentation experiments. We find that no single data augmentation can rule them all. Different augmentation methods have expertise on distinct corruption patterns.

As Table 2 presents, PointCutMix-R performs the best on noise corruptions (ER $= 12.7\%$), PointMixup specializes the transformation corruptions (ER $= 14.1\%$), and RSMix is especially robust against density corruptions (ER $= 26.8\%$). Such results also relate to the design of augmentation strategies. In details, given two point cloud samples $\boldsymbol{x}_a$,$\boldsymbol{x}_b$ from class $a$ and $b$, PointCutMix-R simply merges ($\oplus$) two randomly selected ($\odot$) subsets together based on hyper-parameter $\lambda$ ($\boldsymbol{x}_{aug} = \lambda \odot \boldsymbol{x}_a \oplus (1-\lambda) \odot \boldsymbol{x}_b$). The two subsets will overlap in the resulting point cloud $\boldsymbol{x}_{aug}$. Each point cloud subset can be regarded as a special noise by the other. Thus, it naturally includes noise corruptions with mixing into data augmentations. PointMixup leverages interpolation-based mixing that the transition between two point clouds ($\boldsymbol{x}_{aug} = \lambda \boldsymbol{x}_a + (1-\lambda)\zeta(\boldsymbol{x}_a, \boldsymbol{x}_b)$, where $\zeta(\boldsymbol{x}_a, \boldsymbol{x}_b)$ finds the shortest path for every pair in $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$). The augmented point cloud is thus locally smooth, which aligns with the transformation corruptions. In contrast, RSMix acts similarly with PointCutMix-K but guarantee a *rigid* mixing of two partial point clouds. There will be no overlaps and each point cloud subset is clustered and isolated in the 3D space. Such patterns correspond to density corruptions in point cloud data.

## 5 Conclusion

To conclude, we have presented ModelNet40-C, the first comprehensive benchmark for corruption robustness of point cloud recognition models. We have unveiled the massive performance degradation on our ModelNet40-C for representative models. We also provided critical insights on how different architecture and data augmentation designs affect model robustness on different corruptions. We hope that our ModelNet40-C benchmark will benefit future research in developing robust 3D point cloud models and training strategies!

## REFERENCES


Daniele Bonatto, Ségolène Rogge, Arnaud Schenkel, Rudy Ercek, and Gauthier Lafruit. Explorations for real-time point cloud rendering of natural scenes in virtual reality. In *2016 International Conference on 3D Imaging (IC3D)*, pp. 1–7. IEEE, 2016.

Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021.

Chao Cao, Marius Preda, and Titus Zaharia. 3d point cloud compression: A survey. In *The 24th International Conference on 3D Web Technology*, pp. 1–9, 2019.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 330–345. Springer, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11513–11521. IEEE, 2020.

Davide Forti and Gianluigi Rozza. Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems. *International Journal of Computational Fluid Dynamics*, 28(3-4):158–169, 2014.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. *arXiv preprint arXiv:2106.05304*, 2021.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. *arXiv preprint arXiv:2203.01441*, 2022.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4558–4567, 2018.

Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15900–15909, 2021.

Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018a.

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, 2018b.

Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2279–2283. IEEE, 2019a.

Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6186–6195, 2021.

Jingyun Liu, Qiao Sun, Zhe Fan, and Yudong Jia. Tof lidar development in autonomous vehicle. In *2018 IEEE 3rd Optoelectronics Global Conference (OGC)*, pp. 185–190. IEEE, 2018.

Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904, 2019b.

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. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=3Pbra-_u76D`.

Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928. IEEE, 2015.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. *arXiv preprint arXiv:2102.11273*, 2021.

Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pp. 225–230. IEEE, 2013.

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*, pp. 652–660, 2017a.

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*, 2017b.

Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151–160, 1986.

Wen Shen, Qihan Ren, Dongrui Liu, and Quanshi Zhang. Interpreting representation quality of dnns for 3d point cloud processing. *Advances in Neural Information Processing Systems*, 34, 2021a.

Wen Shen, Zhihua Wei, Shikun Huang, Binbin Zhang, Panyue Chen, Ping Zhao, and Quanshi Zhang. Verifiability and predictability: Interpreting utilities of network architectures for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10703–10712, June 2021b.

Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4548–4557, 2018.

Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 770–779, 2019.

Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.

Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 877–894, 2020a.

Jiachen Sun, Karl Koenig, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of 3d point cloud classification under adaptive attacks. *arXiv preprint arXiv:2011.11922*, 2020b.

Jiachen Sun, Yulong Cao, Christopher B Choy, Zhiding Yu, Anima Anandkumar, Zhuoqing Morley Mao, and Chaowei Xiao. Adversarially robust 3d point cloud recognition using self-supervisions. *Advances in Neural Information Processing Systems*, 34, 2021a.

Jiachen Sun, Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, Dan Hendrycks, Jihun Hamm, and Z Morley Mao. Certified adversarial defenses meet out-of-distribution corruptions: Benchmarking robustness and simple baselines. *arXiv preprint arXiv:2112.00659*, 2021b.

Saeid Asgari Taghanaki, Jieliang Luo, Ran Zhang, Ye Wang, Pradeep Kumar Jayaraman, and Krishna Murthy Jatavallabhula. Robustpointset: A dataset for benchmarking robustness of point cloud classifiers. *arXiv preprint arXiv:2011.11572*, 2020.

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*, pp. 6411–6420, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, volume 1, pp. 10–15607. Rome, Italy, 2015.

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.

Katja Wolff, Changil Kim, Henning Zimmer, Christopher Schroers, Mario Botsch, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Point cloud noise and outlier removal for image-based 3d reconstruction. In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 118–127. IEEE, 2016.

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*, pp. 9621–9630, 2019.

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*, pp. 1912–1920, 2015.

Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, 2019.

Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 915–924, October 2021.

Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. *arXiv preprint arXiv:2012.10921*, 2, 2021.

Wen Yang, Zheng Gong, Baifu Huang, and Xiaoping Hong. Lidar with velocity: Motion distortion correction of point clouds from oscillating scanning lidars. *arXiv preprint arXiv:2111.09497*, 2021.

Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujing Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *arXiv preprint arXiv:2101.01461*, 2021.

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*, pp. 16259–16268, 2021.

Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1201–1210, 2020.

Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1961–1970, 2019.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. *arXiv preprint arXiv:2104.03670*, 2021.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

# A ModelNet40-C

We elaborate the creation of ModelNet40-C in this section. The detailed implementation can be found in our codebase, which is included in the supplementary materials.

`Occlusion` and `LiDAR` share similar general corruption features. We leverage five viewing angles to construct these two corruptions on ModelNet40, as shown in Fig. A. Specifically, we utilize ray tracing algorithms on the original meshed from ModelNet40 to generate the point cloud. Let the facing direction of the object as $0°$ pivoting the $z$ axis, we use $0°$, $72°$, $144°$, $216°$, and $288°$ as our viewing angles, the viewing angles between the $xy$ plane are randomly sampled from is $30° - 60°$. For `LiDAR`, we additionally render the generated point cloud into the vertically multi-line style to simulate the pattern of the LiDAR sensor.

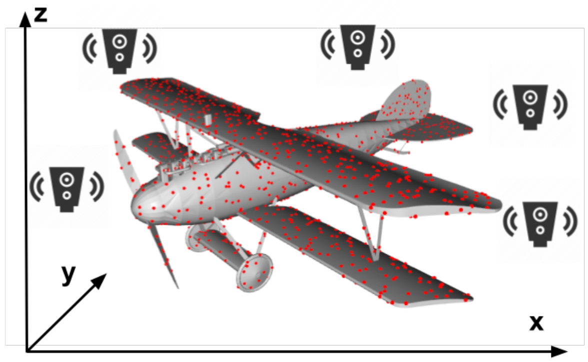


Figure 2: Illustration of `Occlusion` and `LiDAR` Corruption Generation.

For `Local_Density_Inc` and `Local_Density_Dec`, we first sample a number of anchor points based the severity level. We further find the $k$NN of the anchor points and up-sample or down-sample them to increase and decrease their local density, respectively. Similarly, `Cutout` discards the full $k$NN ($k = 50$) subsets of the anchor points to simulate the sensor limitations of LiDAR and other scanning devices.

`Gaussian` and `Uniform` noises are sampled from Gaussian and uniform distributions with different $\sigma$ and $\epsilon$ based on the severity level. For the `Background` noise, we randomly sample different numbers of points in the edge-length-2 cube that bounds the point cloud based on the severity level. For `Impulse` noise, we first sample different numbers of points based on the severity level and assign the maximum magnitude of perturbation $\ell_\infty = 0.05$ to them. For the `Upsampling` noise, we first choose different numbers of points based on the severity level and generate new points around the selected anchors, bounded by $\ell_\infty = 0.05$.

For `Rotation` and `Shear`, we have introduced their construction in § 2. As mentioned, we allow relatively small transformations since we find larger ones will affect the human perception of the object class as well.

For deformation-based corruptions `FFD`, `RBF`, and `Inv_RBF`, we assign 5 control points along each $xyz$ axis, resulting in 125 control points in total. We choose the deformation distance based on the severity level and randomly assign their directions in the 3D space. The deformations then are formulated based on the interpolation functions that we choose in § 2.

We visualize three additional groups of sample point clouds from ModelNet40-C in Fig. 3, Fig. 4 and 5.

## B Related Work

**Adversarial & Corruption Robustness of 2D Images**. Deep neural networks are known to be vulnerable to adversarial examples and common corruptions (Bulusu et al., 2020). Hendrycks & Dietterich (2019); Hendrycks et al. (2021) developed corruption robustness benchmarking datasets CIFAR-10/100-C, ImageNet-C, and ImageNet-R to facilitate robustness evaluations of CIFAR and ImageNet classification models. Michaelis et al. (2019) extended this benchmark to object detection models. Mintun et al. (2021) further proposed ImageNet-$\bar{\text{C}}$ dataset that is comprised of a set of corruptions that are perceptually dissimilar to ImageNet-C. Recently, Sun et al. (2021b) proposed a comprehensive benchmarking suite CIFAR-10/100-F that contains corruptions from different regions in the spectral domain. (Koh et al., 2021) presented WILDS, a curated benchmark of 10 datasets reflecting a diverse range of distribution shifts that naturally arise in real-world applications. Hendrycks et al. (2019); Cubuk et al. (2018); Calian et al. (2021); Kar et al. (2022) proposed augmentation methods to improve the corruption robustness in 2D vision tasks. On the adversarial robustness benchmarking front, Carlini et al. (2019) discussed the methodological foundations, reviewed commonly accepted best practices, and suggested new methods for evaluating defenses to

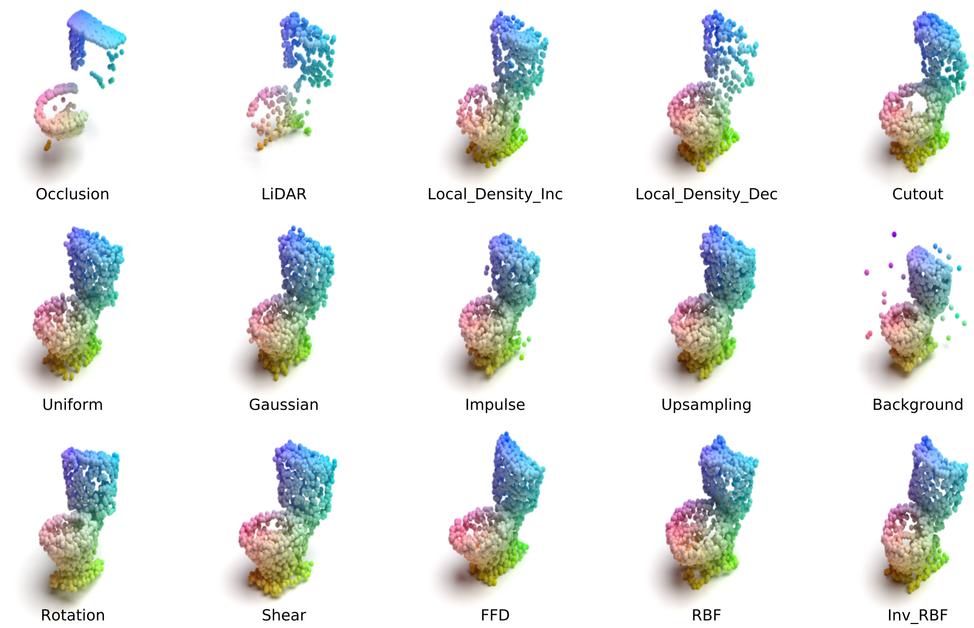


Figure 3: Visualization of Samples from ModelNet40-C - "Toliet" Class.

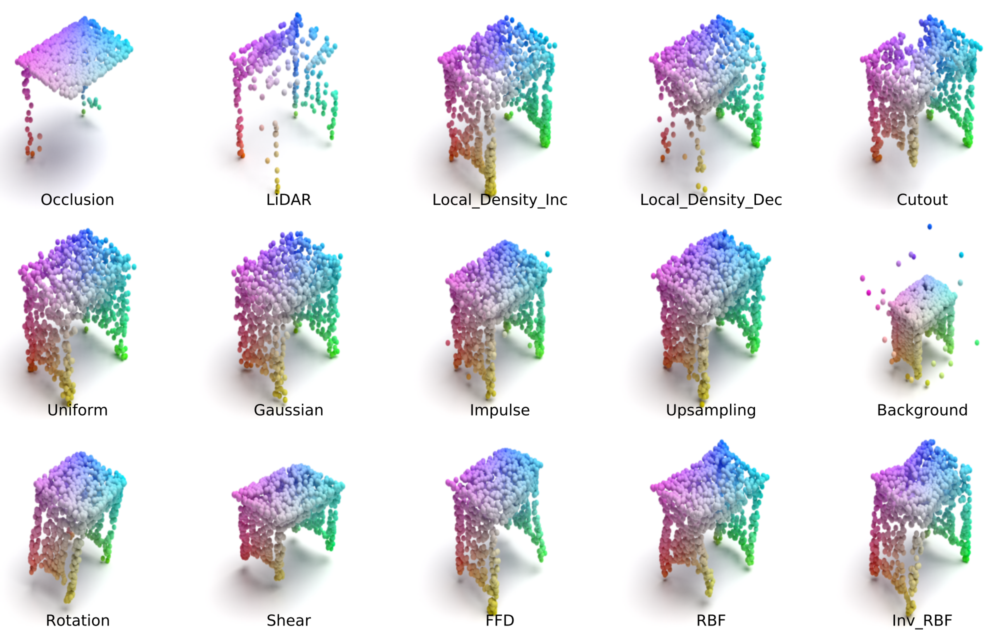


Figure 4: Visualization of Samples from ModelNet40-C - "Desk" Class.

adversarial examples. Croce et al. (2020) proposed a standardized leaderboard called RobustBench, which evaluates the adversarial robustness with AutoAttack (Croce & Hein, 2020), a comprehensive ensemble of white- and black-box attacks.

**3D Point Cloud Deep Learning**. Deep learning models are increasingly being proposed to process point cloud data. Early works attempted to use 3D voxel grids for perception, which have cubic complexity (Maturana & Scherer, 2015; Wang & Posner, 2015). PointNet (Qi et al., 2017a) pioneered to leverage shared multi-layer perceptrons and a global pooling operation to achieve permutation-invariance and thus enable end-to-end training. Qi et al. (2017b) further proposed PointNet++ to hierarchically stack PointNet for multi-scale local feature encoding. PointCNN and RSCNN refactor the traditional pyramid CNN to improve the local feature learning for point cloud recognition (Li et al., 2018b; Liu et al., 2019b). The graph data structure is also heavily used in point cloud learn-

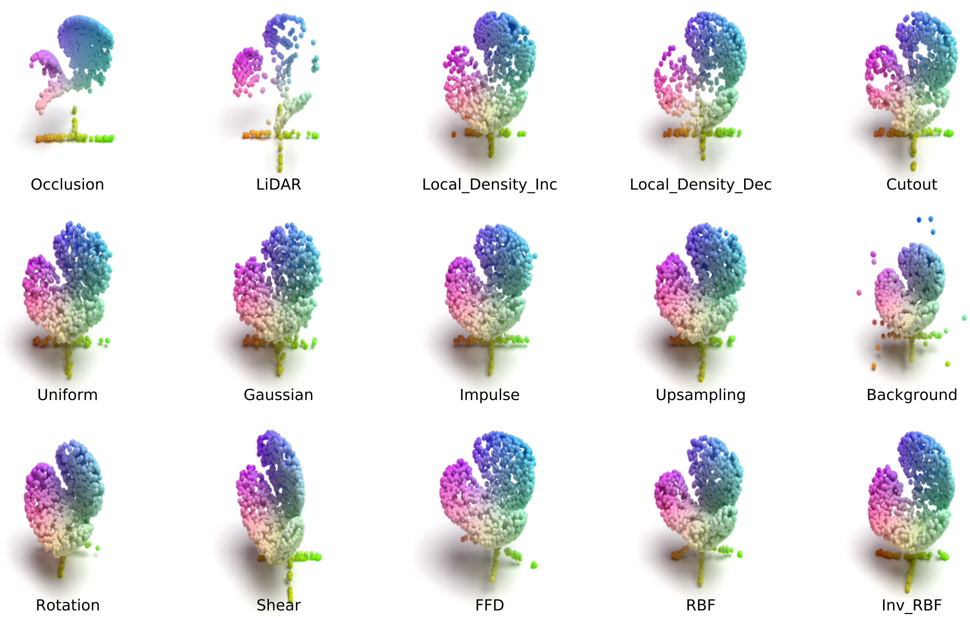


Figure 5: Visualization of Samples from ModelNet40-C - "Chair" Class.

ing (Landrieu & Simonovsky, 2018; Shen et al., 2018). For example, DGCNN built a dynamic graph of point cloud data for representation learning (Wang et al., 2019). PointConv and KPConv improve the convolution operation for point cloud learning (Wu et al., 2019; Thomas et al., 2019). Recent work demonstrated that ResNet (He et al., 2016) on multi-view 2D projections of point clouds could also achieve high accuracy (Goyal et al., 2021). PointTransformer and PCT advance Transformer (Vaswani et al., 2017) blocks into point cloud learning and achieve state-of-the-art performance (Zhao et al., 2021; Guo et al., 2021).

**Robustness Enhancements for 3D Point Cloud**. Several recent efforts tackle improving the robustness of 3D point cloud learning (Sun et al., 2020a). Xiang et al. (2019) and Liu et al. (2019a) first demonstrated that point cloud recognition is vulnerable to adversarial attacks. Zhou et al. (2019) and Dong et al. (2020) proposed to leverage input randomization techniques to mitigate such vulnerabilities. Sun et al. (2020b) conducted adaptive attacks on existing defenses and analyzed the application of adversarial training on point cloud recognition. Zhao et al. (2020) discovered that adversarial rotation greatly degrades the perception performance. Sun et al. (2021a) further showed that pre-training on self-supervised tasks enhances the adversarial robustness of point cloud recognition. Recent studies presented a framework that uses the Shapley value (Roth, 1988) to assess the quality of representations learned by different point cloud recognition models (Shen et al., 2021a;b). Recent efforts also proposed certified adversarial defenses(Liu et al., 2021). Taghanaki et al. (2020) proposed several simple corruption types to benchmark the robustness of point cloud recognition models. However, their formulations cannot represent realistic distortions in the physical world. In this work, we aim to present a more systematic benchmark and rigorously analyze the corruption robustness of representative deep point cloud recognition models.