Towards Reliable Point-Cloud Segmentation: A Robustness Evaluation of Benchmark Architectures

Su Mon Ko, Ida Bagus Krishna Yoga Utama, Yeong Min Jang
Department of Electronics Engineering
Kookmin University
Seoul, South Korea
sumonko001@gmail.com; ibkyutama@ieee.org; yjang@kookmin.ac.kr

***Abstract*—Reliable point-cloud segmentation models are critical for robotics and autonomous applications. However, existing models often lack sufficient robustness in real-time environments, leading to significant performance degradation due to the fact that the data quality varies based on distance, sensor noise, or partial visibility. This work presents a comprehensive study of widely used segmentation architectures—Point CNN, Dynamic Graph CNN (DGCNN), and Point Transformer—on the ShapeNet Part dataset by applying three data distortions (density drop, Gaussian jitter noise, and spherical occlusion). Experimental results show that while all models achieve competitive accuracy on clean data, their resilience varies substantially under corruption, with Point Transformer consistently demonstrating the highest robustness, PointCNN showing moderate degradation, and DGCNN being most sensitive to noise and occlusion. These findings highlight the importance of robustness as a core evaluation dimension for 3D perception models and provide practical guidance for designing and selecting architectures suitable for deployment in noisy or dynamically changing environments.**

***Index Terms*—Point-cloud segmentation, model robustness, 3D perception, real-time applications**

I. Introduction

3D point-cloud segmentation has become a foundational technology for many real-world applications such as robotics, autonomous navigation, AR/VR, and object part recognition because point clouds offer a flexible, sensor-agnostic representation of 3D geometry. As sensor technologies spread and laser scanning or depth-sensing becomes mainstream, point clouds are increasingly common in scenarios where data quality is not guaranteed: objects may be far away (leading to sparsity), sensors may introduce jitter or noise, and occlusion or partial visibility may hide important geometry. Under such variable conditions, robustness of segmentation models is critical: a model that works well on pristine registered data may fail when applied to real-world, noisy, sparse, or partially observed point clouds.

Despite recent progress in deep learning methods for point cloud segmentation, most architectures have focused on achieving high performance under "clean" conditions, leaving their behaviour under real-world distortions unexplored. For instance, traditional clean-data performance often degrades significantly when point clouds contain noise or outliers [1]. Moreover, existing models primarily focus on clean, high-quality datasets, such as ShapeNet Part and S3DIS, which contain uniformly sampled, complete point clouds with minimal noise. As a result, their dependability in real-time applications characterised by sparse, noisy, or partially occluded point clouds remains predominantly unverified [2] [3]. While some studies have explored robustness in classification tasks or under limited corruptions, these evaluations are typically inconsistent across architectures, making fair comparison difficult [1] [4]. Recent efforts have begun addressing robustness in point-cloud processing. PointASNL [2] demonstrated that adaptive sampling combined with local–nonlocal operations can significantly mitigate the impact of noise and outliers, improving performance on both classification and segmentation tasks under corrupted real-world point clouds. Overall, there is a clear gap between high benchmark performance and real-world reliability, which points to a controlled, systematic evaluation framework. Such a framework would allow multiple segmentation models to be assessed under uniform corruptions, including density reduction, Gaussian jitter, and occlusion, to quantify robustness, reveal architectural failure modes, and provide actionable guidance for real-time deployment.

In this paper, we address this gap by conducting a robustness evaluation of three representative segmentation models (PointCNN, DGCNN, and Point Transformer) on a standard part-segmentation dataset (ShapeNet Part). We apply three realistic types of corruption (density drop, Gaussian jitter, and occlusion) at multiple severity levels to simulate real-world distortions such as sparsity due to distance, sensor jitter, and partial visibility. By systematically measuring performance across these conditions, we aim to reveal how different architectures degrade under data corruption, identify which models are more robust to which perturbations, and provide insights and recommendations for deploying point-cloud segmentation in real-world or real-time applications.

II. Related Work

Early deep models for point sets introduced architectures that operate directly on unordered point clouds. PointNet [5] proposed per-point MLPs with symmetric pooling to handle permutation invariance, enabling both classification and part segmentation. Subsequent works introduced local and geometric operators to capture neighbourhood structures. PointCNN [6] learnt X-transformations to enable convolution-like operations on irregular points, while DGCNN [7] used dynamic graph constructions and EdgeConv to explicitly model local edges. More recently, transformer-style models (Point Transformers) exploited self-attention and learnt positional encodings to capture long-range context and deliver strong segmentation performance on large scenes.

Robustness of 3D models has attracted growing attention. Sun et al. [8] introduced ModelNet40-C to quantify degradation under a range of corruptions, showing large gaps between clean and corrupted performance. Efforts such as PointCloud-C [9] expand this idea to shape/segmentation testbeds (ShapeNet-C), providing standardized corruptions (density drop, jitter, occlusion, etc.) for fair model comparisons. These benchmarks reveal that strong clean accuracy does not imply robustness to realistic perturbations.

Several works explicitly target robustness in 3D processing. PointASNL [2] introduced adaptive sampling and nonlocal modules to reduce sensitivity to noise and outliers in raw point clouds. For segmentation with noisy annotations, methods such as PNAL [10] proposed point-wise confidence selection and cluster-wise label correction to improve training under label corruption. Overall, robustness-oriented methods either change input sampling, augment local/nonlocal feature aggregation, or incorporate noise-aware training/label correction strategies. However, most of these methods were evaluated under limited corruption types or on classification tasks, leaving a gap for systematic, head-to-head segmentation robustness comparisons across standard architectures.

While prior studies have investigated architectural innovations for point-cloud segmentation and explored limited robustness conditions, a comprehensive comparison of diverse model families under multiple real-world corruption types remains largely missing. In this work, we address this gap by systematically evaluating three representative segmentation models under control distortions that commonly occur in practical 3D sensing environments. This unified robustness evaluation framework highlights how different architectures degrade under data sparsity, sensor noise, and partial visibility, offering novel perspectives on their reliability for real-time and safety-critical applications.

III. Methodology

This section describes the dataset, corruption framework, model selection and evaluation protocol to assess the robustness of three benchmark point-cloud segmentation models.

A. Datasets and Models

To evaluate robustness under realistic 3D sensing conditions, we use the ShapeNet Part dataset, a standard benchmark for part-level segmentation. It contains 16,881 3D shapes from 16 object categories (e.g., chairs, lamps, airplanes), annotated with 50-part labels in total. Each object is represented by 2048 points sampled from the mesh surface. For each shape, the task is to label each point with its corresponding part category. We follow the official train/test split and report results on the held-out test set. We evaluate three established point-based neural network architectures:

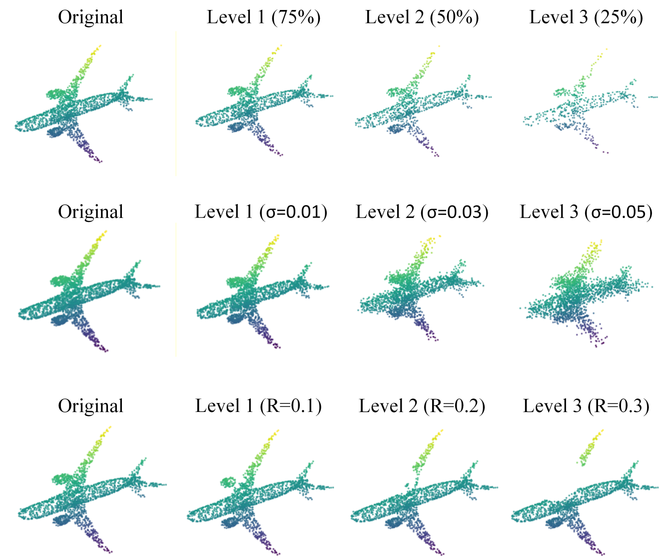

Fig. 1: Point cloud after density reduction, Gaussian noise injection and spherical occlusion

- PointCNN [6]: A convolutional framework that learn an X-transformation to permute and weight neighboring points, enabling a generalization of grid CNNs to irregular point sets. PointCNN reports competitive accuracy on benchmarks.
- DGCNN (Dynamic Graph CNN) [7]: A graph-based model that constructs a k-NN graph on the point cloud at each layer and applies the proposed EdgeConv operation to capture local geometric relations. DGCNN proved effective for both classification and segmentation tasks.
- Point Transformer [11]: A transformer-like network using self-attention on point neighborhoods. Designed for 3D data, it achieves state-of-the-art results (e.g. $\tilde{8}$6.6% mIoU on ShapeNet Part) by capturing both local and global context.

Each model is implemented and trained on the clean ShapeNet Part training set. The reported baseline accuracies (clean test mIoU) are around 83–85% for these models, confirming good performance on the unmodified data.

B. Corruption Protocol

To assess robustness, we generate corrupted versions of the ShapeNet Part test set under three categories of distortions. These mimic real sensor artifacts:

- Density Drop: For each object, we randomly drop points so that only 75%, 50%, or 25% of the original points remain. This emulates distant or low-resolution scans where the point cloud density is reduced.
- Gaussian Noise: We add independent Gaussian noise to each point's (x,y,z) coordinates. Specifically, we sample noise from $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.01, 0.03$, or 0.05. This creates slight random displacements of points, reflecting typical depth-sensor noise.

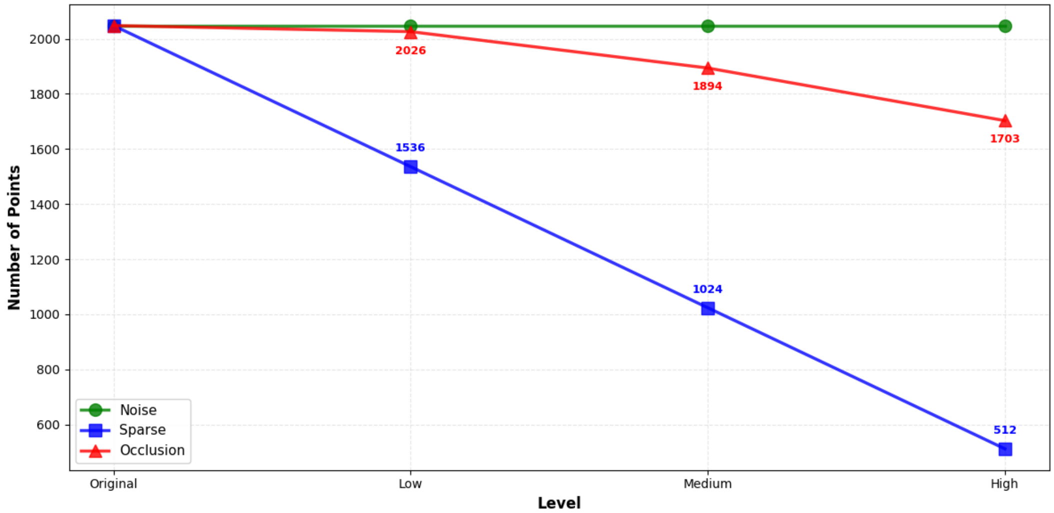

Fig. 2: Number of point cloud after applying sparse, noise and occlution

- Occlusion: We remove all points within a random sphere of a given radius. Concretely, for each test object, we select a random center in the object's bounding volume and delete all points within radius $r = 0.1, 0.2,$ or 0.3. This creates a spherical hole in the point set.

The visualisation diagram of three distinct data corruption techniques applied on a 3D point cloud of an airplane object is illustrated in figure 1.

Figure 2 shows the comparison of how different corruption types affect the number of points in a ShapeNet part point cloud. Gaussian noise (left) does not change the point count, since it only perturbs coordinates while keeping all points. In contrast, density-based sparsification (middle) directly reduces the number of points according to the sampling ratio, dropping from the full 2048 points down to 512 at the highest severity. Occlusion (right) also lowers the point count, but more gradually: removing points within a spherical region reduces the cloud from 2048 to 1703 as the occlusion radius increases.

C. Evaluation Metrics

We measure segmentation performance using the mean Intersection-over-Union (mIoU) over all part classes, a standard metric for part segmentation. For each shape, IoU is computed per part category, then averaged across the parts present in the shape. Finally, we average the shape-level IoUs over the test set. Higher mIoU indicates better segmentation quality.

To quantify robustness, we will report the absolute mIoU values at each corruption level and, optionally, a relative performance drop (e.g., percentage change from clean). In some research, the mean Corruption Error (mCE) metric was used by normalising error against a baseline model, but here we mainly focus on direct comparison of mIoUs across models and conditions. Apart from the intended corruption, we carefully ensure that the evaluation conditions (number of points, normalisation) align with the clean-data setting..

IV. Results and Discussion

This section presents high-level quantitative results. We first establish a baseline performance using clean data to contextualise the degradation. On the clean ShapeNet Part test set, Point Transformer achieves the highest reported mIoU (around 86.6%), PointCNN (around 86.1), and DGCNN achieves around 85.7%. This confirms that all four models are competitive on the standard task, with differences of only a few percent in clean accuracy. All models' accuracy degrades as more points are dropped. Adding jitter to point coordinates generally degrades features for all models. We observe that segmentation accuracy falls steadily as noise increases. The spherical occlusion (removing a local region) has a strong impact, as it removes entire parts of an object. We observe that occlusion with a large radius (0.3) can dramatically reduce accuracy, especially for parts near the occluded region.

Table I shows that all models maintain high accuracy until the most severe drop, at which Point Transformer (85.4) exceeds PointCNN (84.2) and DGCNN (82.1). Table II shows a similar pattern: at $\sigma = 0.05$, the Transformer's mIoU (83.1) is far higher than DGCNN (76.1) and PointCNN (75.3). Table III highlights the largest disparity: even with large occlusions, Point Transformer stays near 86.0 mIoU, whereas PointCNN and DGCNN fall to 83.2 and 79.8. These tables confirm the narrative trends. For example, Table I indicates that under a 25% density drop, DGCNN loses $\approx$ 3.6 mIoU points from baseline while Point Transformer loses only $\approx$ 1.2. Likewise, Table III shows DGCNN suffers a $\approx$ 5.9-point drop under severe occlusion, compared to $\approx$ 0.6 for the Transformer. Overall, the data make clear that Point Transformer consistently tops the performance across corruptions, PointCNN is second, and DGCNN is most affected.

In summary, a key quantitative observation is that Point Transformer consistently has the best robustness profile among the three. The transformer's attention mechanism does help maintain longer-range context, mitigating some local loss.

TABLE I: Evaluation under density drop corruption

Model	Key Features	mIoU				
		Clean	Level 1 (75%)	Level 2 (50%)	Level 3 (25%)	Δ(Clean → Level 3)
PointCNN	Convolution-based	86.1	84.1	84.2	84.2	-1.9
DGCNN	Graph-based	85.7	85.5	85.0	82.1	-3.6
Point Transformer	Transformer-based	86.6	86.4	86.1	85.4	**-1.2**

TABLE II: Evaluation under Noise conditions

Model	Key Features	mIoU				
		Clean	Level 1 (75%)	Level 2 (50%)	Level 3 (25%)	Δ(Clean → Level 3)
PointCNN	Convolution-based	86.1	83.6	80.2	75.3	-10.8
DGCNN	Graph-based	85.7	84.5	80.7	76.1	-9.6
Point Transformer	Transformer-based	86.6	86.5	85.1	83.1	**-3.5**

TABLE III: Evaluation under occlusion conditions

Model	Key Features	mIoU				
		Clean	Level 1 (75%)	Level 2 (50%)	Level 3 (25%)	Δ(Clean → Level 3)
PointCNN	Convolution-based	86.1	84.1	84	83.2	-2.9
DGCNN	Graph-based	85.7	80.2	80.6	79.8	-5.9
Point Transformer	Transformer-based	86.6	86.6	86.3	86	**-0.6**

All models experience accuracy degradation as corruption severity increases, but the magnitude of the drop depends on the architecture. These observations imply that real-world 3D perception systems might prefer local-aggregation networks or incorporate augmentation strategies to handle sensor artefacts.

V. Conclusion

This work presented a robustness evaluation of three widely used point-cloud segmentation models—PointCNN, DGCNN, and Point Transformer—under realistic corruptions that simulate sparsity, sensor noise, and occlusion. Across all conditions, the Point Transformer consistently showed the highest resilience, maintaining strong performance even at severe corruption levels, while PointCNN exhibited moderate degradation and DGCNN suffered the largest drops, especially under heavy noise and occlusion. These results highlight how architectural choices, such as attention-based feature aggregation versus graph-based neighbourhood construction, directly influence robustness. Our results show that robustness should be a key factor in evaluating 3D vision models. Future research should look into corruption-aware training and hybrid architectures to make models even more stable in real-world settings.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No.2022R1A2C1007884).

References

[1] Z. Zhang, L. Lin, and X. Zhi, "R-pointnet: Robust 3d object recognition network for real-world point clouds corruption," *Applied Sciences*, vol. 14, no. 9, p. 3649, 2024.

[2] X. Yan, C. Zheng, Z. Li, S. Wang, and S. 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*, 2020, pp. 5589–5598.

[3] J. Sun, Q. Zhang, B. Kailkhura, Z. Yu, C. Xiao, and Z. M. Mao, "Benchmarking robustness of 3D point cloud recognition against common corruptions," *arXiv preprint arXiv:2201.12296*, 2022.

[4] S. Ye, D. Chen, S. Han, and J. Liao, "Learning with noisy labels for robust point cloud segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6443–6452.

[5] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[6] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[7] A. V. Phan, M. Le Nguyen, Y. L. H. Nguyen, and L. T. Bui, "Dgcnn: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, vol. 108, pp. 533–543, 2018.

[8] J. Sun, Q. Zhang, B. Kailkhura, Z. Yu, C. Xiao, and Z. M. Mao, "Modelnet40-c: A robustness benchmark for 3d point cloud recognition under corruption," in *ICLR 2022 Workshop on Socially Responsible Machine Learning*, vol. 7, 2022.

[9] J. Ren, L. Kong, L. Pan, and Z. Liu, "Pointcloud-c: Benchmarking and analyzing point cloud perception robustness under corruptions," *preprint*, vol. 3, 2022.

[10] D. Cazorla López, "Pnal: Un modelo algebraico para procesos probabilísticos y no deterministas," 2001.

[11] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.