ABSTRACT

In recent years, significant progress has been achieved for 3D object detection on point clouds thanks to the advances in 3D data collection and deep learning techniques. Nevertheless, 3D scenes exhibit a lot of variations and are prone to sensor inaccuracies as well as information loss during pre-processing. Thus, it is crucial to design techniques that are robust against these variations. This requires a detailed analysis and understanding of the effect of such variations. This work aims to analyze and benchmark popular point-based 3D object detectors against several data corruptions. To the best of our knowledge, we are the first to investigate the robustness of point-based 3D object detectors. To this end, we design and evaluate corruptions that involve data addition, reduction, and alteration. We further study the robustness of different modules against local and global variations. Our experimental results reveal several intriguing findings. For instance, we show that methods that integrate Transformers at a patch or object level lead to increased robustness, compared to using Transformers at the point level. The code is available at https://github.com/sultanabughazal/robustness3d.

KEYWORDS
3D Detectors, Point Clouds, Robustness, 3D Object Detection

1 INTRODUCTION

3D object detection aims at localizing objects of interest in a given 3D scan. It has applications in numerous areas, such as autonomous driving, robotics, and augmented reality. 3D object detection involves placing 3D bounding boxes around objects, which allows the understanding of real physical dimensions and distances to objects, compared to 2D image detection in which understanding is limited to the image plane.

Most 3D object detection methods rely on the availability of depth information, which is usually acquired using range sensors, such as LiDAR for outdoors and Kinect for indoors. The accuracy of the acquired depth is greatly affected by sensor type and distance to the sensor. Moreover, camera viewpoint and scene structure lead to different representations of the same object, mainly due to occlusions. Also, scene acquisition is sometimes collected from different viewpoints, which requires aggregation of information using various reconstruction techniques, leading to more variations in how a given scene can be represented.

3D scenes are usually represented by point clouds, which are sets of unordered points. These point clouds are typically processed with deep learning methods that are either projection-based, voxel-based, or point-based. Projection-based methods project 3D data into a 2D plane and benefit from the advancement of image feature learning techniques. Voxel-based methods first transform the data.
into a regular grid to then be processed by 3D convolution neural networks (CNNs). Point-based methods directly process raw points, using MLP architectures \[14, 15\], Graph convolutions \[22\], and, recently, Transformers \[20\].

Point-based object detectors have shown promising performance on various datasets \[9, 12, 13, 16\]. Given the sparse nature of point clouds, point-based methods are generally less voluminous when compared to voxel-based and projection-based methods that include empty-space representation and suffer from quantization artifacts. Nonetheless, the robustness of point-based 3D object detection methods against point cloud variations is yet to be investigated. Therefore, we design corruptions to model variations in the point cloud capturing and preparation process. We also analyze and benchmark the performance of popular point-based 3D detection methods against the proposed corruptions.

Corruption in 3D points clouds can occur in various forms. We group corruptions into three main categories: (1) corruptions that lead to information loss, (2) corruptions from added information, and (3) corruptions due to data alteration. Examples of these corruptions are shown in Fig. 1. In our analysis, we simulate those sources of data corruption by generating multiple possible patterns in which each corruption can occur. We design seven point removal corruptions that represent information loss; namely, drop global, drop local, drop object, drop background, drop object parts, and drop floor. For the point addition corruptions, we introduce three different patterns: add global, add local, and scene expansion. Additionally, we simulate point alteration following four patterns: point jitter, background noise, local noise, and floor plane inclination. With our proposed corruptions, we strive to cover a broad variety of variations that are encountered in point cloud generation.

Contributions: We design point cloud corruptions to simulate variations due to errors in data collection and preparation. We analyze and benchmark the robustness of point-based 3D object detection architectures against these corruptions in point clouds. To the best of our knowledge, we are the first to investigate the robustness of point-based 3D object detectors. We also analyze the robustness of methods that use Transformers, which is the new paradigm in numerous computer vision applications including 3D object detection. Our experiments reveal several intriguing findings. For instance, we observe that the recently introduced transformer-based 3D detector shows increased robustness at the patch or object level, and less robustness when extracting point-level features.

2 RELATED WORK

3D Object Detection From Point Clouds. Many works have been proposed for the 3D object detection task. Most of these works build upon architectures that process 2D projections, voxels, or points. Projection-based methods reduce the computation cost by projecting 3D information into the 2D space \[1, 21, 25\]. Voxel-based methods transform the data into a regular grid, which can be processed with 3D convolution operations \[18, 24, 26\]. Point-based methods directly process raw points to extract feature information. Many of these methods rely on PointNet \[14\] or PointNet++ \[15\], which are MLP-based architectures that can directly operate on 3D points. VoteNet \[13\] uses PointNet++ as a backbone to learn point features and Hough voting to identify object centers and generate object proposals. MLCVNet \[16\] is an extension of VoteNet, in which three context modules are added to exploit context information at the patch, object, and global level. Liu et al. \[9\] introduce a group-free point-based detection method that uses PointNet++ to extract features, and employs self-attention and cross-attention to extract and refine object features. 3DETR \[12\] is an end-to-end transformer detection model that uses an encoder to encode and refine input features and a decoder to predict bounding boxes.

Robustness Benchmarks. Numerous robustness benchmarks have been presented for image classification. Hendrycks and Dietterich \[7\] introduce ImageNet-C which consists of 15 corruptions applied to the validation set of ImageNet \[4\]. ImageNet-R \[6\] enables the study of classifiers’ robustness to abstract visual renditions, attributes shift, and blurliness. Robustness benchmarks have been also proposed for 2D object detection methods. Michaelis et al. \[10\] develop three corrupted versions of the popular object detection datasets, Pascal VOC \[5\], MS COCO \[8\], and Cityscapes \[2\], to assess object detectors’ robustness under different corruptions. Mirza et al. \[11\] study the robustness of object detectors against degrading weather conditions. Robustness against corruptions was also studied in the 3D domain, mainly for the classification task. ModelNet40-C \[19\] includes corruptions of ModelNet40 \[23\] validation set with 15 common corruptions including Gaussian noise and occlusion. It provides comprehensive analysis of six model architectures performance with the proposed benchmark. Ren et al. \[17\] design and apply seven corruptions on ModelNet40 validation set. The benchmark includes 14 classification models and provides a systematic investigation of performance under corruptions.

3 POINT CLOUD CORRUPTIONS

Robustness Benchmark. We design point cloud corruptions to assess the robustness of point-based 3D object detectors. We analyze and benchmark 3D object detection architectures on the corrupted point clouds. Our choice of architectures are: VoteNet, MLCVNet, Group-Free, and 3DETR. The first three models employ PointNet++ as a backbone to learn point features. In VoteNet, Hough voting is applied to cluster object points which get aggregated to generate box proposals. MLCVNet uses the VoteNet framework with the addition of three context modules: patch-patch, object-object and global-scene modules. In Group-Free, a transformer decoder is used after obtaining the point features to leverage all points in a point cloud and compute object features. On the other hand, 3DETR is an end-to-end transformer with an encoder-decoder structure. The point features are learned via multiple layers of self-attention in the encoder, and the decoder learns to predict bounding boxes.

Corruption Types. We introduce corruptions to simulate variations that emanate from different sources, such as sensor’s accuracy, resolution, and location, as well as scene variations and data preprocessing, as shown in Table 1. In this analysis, we focus on indoor scene corruptions. The applied corruptions lead to data reduction, data addition, or simply alter the data. We apply data corruptions on various levels: local, global, object-level, and background.

Setup. We introduce corruptions to ScanNetv2 \[3\], a richly annotated dataset of 3D meshes reconstructions of indoor scenes. It contains 1513 scenes out of which 312 form the validation set. We
Table 1: The proposed data corruptions and their causes. We design corruptions to simulate variations emerging from sensor accuracy (sens. acc.), sensor resolution (sens. res.), sensor location (sens. loc.), scene variations (scene var.), and pre-processing. The corruptions are applied on various levels: local, global, object-level, and background (BG), and involve data addition, reduction, or alteration.

| Corruption   | Data Reduction | Data Addition | Data Alteration |
|--------------|----------------|---------------|-----------------|
|              | Global | Local | Object | BG | Part | Global | Local | Expand | Jitter | Local | BG | Incline |
| sens. acc.   | ✓     | ✓     | ✓     | ✓   | ✓    | ✓     | ✓     | ✓     | ✓     | ✓     | ✓   | ✓     |
| sens. res.   | ✓     | ✓     | ✓     | ✓   | ✓    | ✓     | ✓     | ✓     | ✓     | ✓     | ✓   | ✓     |
| sens. loc.   | ✓     | ✓     | ✓     | ✓   | ✓    | ✓     | ✓     | ✓     | ✓     | ✓     | ✓   | ✓     |
| scene var.   | ✓     | ✓     | ✓     | ✓   | ✓    | ✓     | ✓     | ✓     | ✓     | ✓     | ✓   | ✓     |
| preprocess   | ✓     | ✓     | ✓     | ✓   | ✓    | ✓     | ✓     | ✓     | ✓     | ✓     | ✓   | ✓     |

benchmark four different detectors on a corrupted validation set. For methods with multiple variants, we choose the top performing variant in our experiments.

We note that the augmentations used in the training of the selected models and our corruptions don’t overlap. All four models use the same three augmentations during training: (1) Flipping along the YZ plane, (2) Flipping along the XZ plane, (3) Rotation along Z-axis in the range \([-5, 5]\).

**Evaluation Metrics.** We use Corruption Error (CE) and mean Corruption Error (mCE) [7] for evaluation. We substitute top-1 error \(E_{1,s,c}^f\) in the original formula for mean Average Precision (mAP) to measure object detection performance. We report mAP at \(0.25\) IoU threshold. Similar to [7], we adjust for the varying difficulties imposed by different corruptions by dividing with a baseline’s mAP. Any choice of a baseline will work, we choose VoteNet. The Corruption Error is formulated as follows:

\[
CE_i = \frac{\sum_{l=1}^{n} (1 - mAP_{l,i})}{\sum_{l=1}^{n} (1 - mAP_{VoteNet, l,i})},
\]

where \(mAP_{l,i}\) is the mAP on a corrupted test set \(i\) at corruption level \(l\), \(mAP_{VoteNet, l,i}\) is the baseline’s mAP. mCE is the average of CE over \(N\) corruptions:

\[
mCE = \frac{1}{N} \sum_{i=1}^{N} CE_i.
\]

3.1 Robustness Against Point Removal

While designing our suite of point removal corruptions, we use 6 different techniques in dropping points from a given scan: Drop Local, Drop Global, Drop Object, Drop Background, Drop Object Parts, and Drop Floor.

3.1.1 Drop Global. This corruption represents a uniformly distributed loss of points across a 3D scene, as shown in Fig. 2b. We implement drop global at 5 severity levels, ranging from dropping 25% of the points in the least severe level (level 1) with increment of 12.5% in every level up to 75% of the points in the most severe level (level 5). Dropped points are replaced by duplicates of remaining points to keep a fixed sized input. All detectors show robustness against this corruption with mAP decrease of less than 2.5% at level 3 (50% of the points dropped) and decent performance at higher levels. We plot the decreasing mAP over the 5 severity levels in Fig. 1a in the supplementary material.

3.1.2 Drop Local. In contrast to drop global, drop local represents a clustered loss of points, similar to the occlusion effect (Fig. 2c). In each scan, we use a uniform random distribution function to determine the position, size, and the number of clusters. We implement drop local in 5 increasing severity levels. In the least severe corruption (level 1), 25% of all points are uniformly selected in clusters and replaced by duplicates of the remaining points. With increasing the level of severity, the dropout ratio is increased by 12.5% up to level 5 where 75% of the points are dropped. At the first level, VoteNet, MLCVNet, and 3DETR’s drop rate does not exceed 7% while Group-Free suffers greatly, with 3x reduction compared to other methods. We reason that this significant drop is due to Group-Free’s reliance on local patch and object information, which is heavily affected by the local drop. The accuracy of all four models drop below 20% by level 5 on the severity scale.

3.1.3 Drop Floor. Drop Floor represents the loss of all the points that make up the floor in the 3D scene (Fig. 2d). This aims at analyzing the ability of different methods to detect objects from shape...
We note that the performance of all models steadily declines as the severity level increases. It is worthy to mention that even with 62.5% of the object’s points being dropped, all detectors were able to perform exceptionally well with only a maximum drop of around 3% mAP. This suggests that detectors rely on contextual cues to infer the object from surrounding points. Pushing the severity level up to 75% drop rate leads to performance degradation due to the high loss in local object shape information. The detailed results are plotted in Fig. 2a in the supplementary.

### Table 2: mAP for various object detection methods before and after applying Drop Floor corruption.

| Method  | mAP Before | mAP After |
|---------|------------|-----------|
| VoteNet | 58.44      | 43.81     |
| MLCVNet | 65.01      | 52.80     |
| Group-Free | 69.05   | 63.86     |
| 3DETR   | 61.04      | 55.21     |

The floor plane in indoor scans provides a good contextual cue, and the estimated floor height can be used as an additional input feature. The floor height is usually estimated using a percentile along the vertical dimension. Since natural scenes are very diverse, this estimation is not always correct. We show our results in Table 2. We observe that Group-Free and 3DETR are more robust than VoteNet and MLCVNet. This is due to the latter methods using estimated height as an additional input feature.

#### 3.1.4 Drop Object.

Similar to drop global, in this corruption, we drop random points uniformly. However, the selected points are taken only from objects of interest excluding the background, as shown in Fig. 3b. We implement this with 5 gradually increasing levels of severity. In level 1, 25% of the object points are dropped and replaced by duplicates of the remaining points from the same object. The drop percentage is increased by 12.5% in every level. At the highest severity level, 75% of the object points are dropped. We note that the performance of all models steadily declines as the severity level increases.

#### 3.1.5 Drop Background.

We also consider scenarios where the missing points are exclusively background points, which belong to the floor, wall, or the ceiling. An example of this corruption is shown in Fig. 3c. Similar to Drop Object, we implement this corruption with 5 levels of severity, with 25% to 75% drop in background points. Do detectors rely on local shape information to infer the object class and position or rely more on context? In this corruption, we aim at analyzing the behavior of detectors once context information is gradually dropped. All methods show robustness against removing up to 75% of the background points, with less than 1% decrease. This shows that minor scene structure is helpful to encode contextual information. Detailed results are plotted in Fig. 2b in the supplementary.

#### 3.1.6 Drop Object Parts.

Amodal object detection estimates the physical size and structure of an object even if partially visible, it has benefits in applications such as robotics navigation, grasp estimation, and Augmented Reality. We investigate the ability of 3D detectors to estimate amodal bounding boxes given partial objects. For all object instances, we slice a portion of the object on a random axis and a random direction, where the portion size is 10, 20, 30, 40, or 50% of the object, representing 5 levels of corruption. An example of this corruption is shown in Fig. 3d. With minor corruption, Group-Free was able to maintain good performance compared to the other methods. We believe this is facilitated by the global sharing of shape, size, and location information in the self-attention module, allowing detection with a broader understanding. We also perform evaluation with updated bounding boxes that enclose the remaining object part. In this case the four models exhibit a more significant drop in accuracy over the five severity levels. This shows that methods tend to regress to the smaller bounding box at low corruption levels, but favor the amodal bounding box at high corruption levels. Both trends are plotted in Fig. 3 in the supplementary material.

We show Corruption Error (CE) and mean Corruption Error (mCE) for all point removal corruptions in Table 3. We observe that the Transformer-based Group-free method is the most robust against point removal corruptions. The least robust is also a Transformer-based method (3DETR). This suggests that employing Transformers at the patch and object level realizes more robustness than extracting point-level features used in 3DETR.

### 3.2 Robustness Against Point Addition

During the acquisition process of point cloud scenes, outlier points and noise may be captured as well. We introduce 3 corruptions: Add Global, Add Local and Scene Expansion. Each corruption aims at understanding how models deal with such variations. We apply the corruption with different levels of severity to model different scenarios in the real-world setting.

![Figure 3: An example scene with its corrupted versions (Drop Object, Drop Background, and Drop Object Parts). Removed points are replaced with duplicates of the remaining points. Each corruption is implemented in 5 levels of severity where the ratio of the dropped points increases with every level.](image-url)
3.2.2 Add Local. We follow the setting of ModelNet40-C [17]. First, we normalize the scene and then randomly select $C$ centroids, where $C \in \{1, 312\}$. Around each centroid, we generate a random number of points due to the self-attention mechanism, which will include the added noise at the local level which affects local interactions. 3DETR drop to below 10% mAP as early as level 2, while other methods maintain an accuracy well above the 25% mark. We plot the results in Fig. 4 in the supplementary material.

### 3.2.3 Scene Expansion.
This corruption aims to replicate a scene with multiple floor planes like a stairwell for instance. We introduce new points in the form of a plane that is below the original floor, but outside the scene (Fig. 4d). We note that point-based architectures tend to be relatively robust to such corruptions. On the other hand, the performance of the transformer-based architecture drops significantly at the lowest level of severity. We show those trends in Fig. 4c in the supplementary material.

As illustrated in Table 4, we report CE and mCE over Points Addition corruptions. 3DETR performs the worst in all three corruptions. The Detector scores the largest CE of 1.430 in Add Local corruption. We conclude that while the transformers’ wide receptive field is desirable, the self-attention mechanism that is needed to achieve that is extremely sensitive to noise.

### 3.3 Robustness Against Alterations

Sensors noise leads to spatial inaccuracies, where points’ position, scale, and angle are wrongly measured. We aim to capture the same kind of effect in the subsequent corruptions. We introduce Jitter, Local Noise, Background Noise, and Floor Plane Inclination.

#### 3.3.1 Point Jitter.
To apply this corruption, we normalize the scene and then add Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ to (X,Y,Z) coordinates for all scene points. We set $\sigma = 0.004 \times \text{level}$, where level $\in \{1, 2, 3, 4, 5\}$.

We inspect the performance of all four detectors over 5 levels of jitter. VoteNet shows good robustness against jitter, most probably due to its local grouping nature. By the time the applied corruption is raised to severity level 3, all four methods fall below 20% in mAP. This trend is plotted in Fig. 5a in the supplementary material.

#### 3.3.2 Floor Plane Inclination.
In this corruption, we want to investigate the robustness of detectors toward orientation changes. This scenario might arise with unlevel measurement devices or in inclined indoor scenes. We apply rotation about the Y-axis with different angles ranging from 5 to 25 degrees in steps of 5. We observe that the accuracy worsens with increasing degree of rotation across detectors. Initially, Group-Free shows more robustness toward inclinations of small angles. All four methods perform similarly and gradually drop below 40% in mAP at severity level 4.

### Table 3: Corruption Error(CE) and mean Corruption Error(mCE) for Points Removal corruptions using various architectures.

| Architectures | mAP | mCE | Global | Local | Object | Background | Object-parts | Floor |
|---------------|-----|-----|--------|-------|--------|------------|--------------|-------|
| VoteNet[13]   | 58.44 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| MLCVNet[16]   | 65.01 | 0.897 | 0.851 | 0.887 | 0.840 | 0.850 | 0.855 | 0.868 |
| Group-Free[9] | 69.05 | 0.842 | 0.774 | 1.018 | 0.779 | 0.770 | 0.642 | 0.685 |
| 3DETR[12]     | 61.04 | 1.067 | 0.922 | 0.957 | 0.924 | 0.946 | 0.995 | 0.832 |

On the Robustness of 3D Object Detectors
Table 4: Corruption Error (CE) and mean Corruption Error (mCE) for Point Addition and Point Alteration corruptions in different architectures.

| Architectures | mAP ↑ | mCE ↓ | Global | Local | Scene-Expansion | Jitter | Floor-P-Incl. | Local-N | Background-N |
|---------------|-------|-------|--------|-------|----------------|--------|---------------|---------|--------------|
| VoteNet[13]   | 58.44 | 1.000 | 1.000  | 1.000 | 1.000          | 1.000  | 1.000         | 1.000   | 1.000        |
| MLCVNet[16]   | 65.01 | 0.897 | 0.955  | 0.903 | 0.794          | 1.016  | 0.8854        | 0.951   | 1.010        |
| Group-Free[9] | 69.05 | 0.842 | 0.861  | 0.869 | 0.741          | 1.031  | 0.7933        | 0.951   | 1.028        |
| 3DETR[12]     | 61.04 | 1.067 | 1.329  | 1.430 | 1.372          | 1.092  | 0.9911        | 1.024   | 1.049        |

Table 5: Detectors mAP on clean validation-set of ScanNet, after adding Local Noise and Background Noise.

| Method         | Clean | Local-Noise | Background-Noise |
|----------------|-------|-------------|-----------------|
| VoteNet        | 58.44 | 55.67       | 4.52            |
| MLCVNet        | 65.01 | 63.10       | 3.94            |
| Group-Free     | 69.05 | 65.75       | 1.31            |
| 3DETR          | 61.04 | 56.37       | 0.44            |

3.3.3 Local Noise. How large is the effect of context on object detection? In Local Noise, our goal is to answer this question. We distort the shape information of the object by replacing all instances of an object class in all scenes with random noise. We choose chair class as it is the most occurring instance with 1368 chair instances out of 4364 total instances. Here, we apply the previously defined Jitter corruption with a severity level of 5 to the chair points. Table 5 shows of the detectors accuracy on the clean validation-set and after Local and Background Noise. The AP of the class chair drops as expected. In particular, Group-Free chair’s AP drops from 93.17 to 53.04. We suspect the sharper drop is due to the noise affecting the self-attention modules. In the other hand, MLCVNet performs the best. The added context modules helped the model recognize chair instances from other objects. Furthermore, for all detectors, the AP of the table category decreases significantly, with a reduction of 5.43,5.24,8.89 and 13.4 in VoteNet, MLCVNet, Group-Free and 3DETR respectively. We reason that behind this drop is noisy points interfering with the table points. We report per category AP in the supplementary.

3.3.4 Background Noise. Introducing background noise aims to investigate how the detector performance would be affected if there is no useful context information but rather noise around the object. Here, we keep the local object information intact. We apply Jitter (level 5) on all points except class chair points. All categories AP deteriorate. Although the local shape information of the chair is preserved, all detectors fail to detect it with certainty. We can see that VoteNet performs the best with AP of 16%, while 3DETR performs the worst with AP of less than 1 %. This aligns with VoteNet’s superior robustness to noise and 3DETR extreme sensitivity to noise. Further details are reported in the supplementary.

CE and mCE of Points Alteration are shown in Table 4. All models perform unfavorably in Jitter and Background Noise. MLCVNet struggles with Jitter and Background Noise but performs relatively better in Local Noise and Floor Plane Inclination. Group-Free follows a similar trend. Furthermore, 3DETR scores the highest CE in Local Noise which is consistent with our previous findings.

Figure 5: An example scene with its corrupted versions (Point Jitter, Local Jitter, and Background Jitter).

4 CONCLUSION

We investigate the robustness of four of the most influential 3D detection architectures; VoteNet [13], MLCVNet [16], 3DETR [12], and Group-Free [9]. Each of the four architectures is trained on a carefully corrupted version of the ScanNet dataset using one or more of 13 point cloud corruptions that cover point addition, point removal, and 3D scene alteration. We identify common natural errors and imperfections in point clouds and then design corruptions that simulate those inaccuracies. We present our experiments and make comparisons across corruptions and different architectures. Each of the four architectures is trained on a version of the ScanNet dataset that is partially modified using different corruptions with different probabilities and severity, and then evaluated on the clean ScanNet test set. We find that a transformer-based 3D detector is more robust at the patch or object level, whereas its robustness diminishes when used to extract point-level features. We hope the analysis presented in this work will contribute toward designing more robust 3D object detection frameworks in the future.
REFERENCES

[1] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. 2017. Multiview-3D object detection network for autonomous driving. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 1907–1915.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition. 3213–3223.

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. Ieee, 248–255.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. International journal of computer vision 88, 2 (2010), 303–338.

[6] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. ICCV (2021).

[7] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. Proceedings of the Inter-National Conference on Learning Representations (2019)

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In European conference on computer vision. Springer, 740–755.

[9] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. 2021. Group-Free 3D Object Detection via Transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021), 2929–2938.

[10] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgienia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. 2019. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming. arXiv preprint arXiv:1907.07484 (2019).

[11] Muhammad Jahanzeb Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Buschlof. 2021. Robustness of Object Detectors in Degrading Weather Conditions. 2021 IEEE International Transportation Systems Conference (ITSC) (2021), 2719–2724.

[12] Ishan Misra, Rohit Girdhar, and Armand Joulin. 2021. An End-to-End Transformer Model for 3D Object Detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021), 2886–2897.

[13] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In Proceedings of the IEEE International Conference on Computer Vision.

[14] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space (NIPS’17). Curran Associates Inc., Red Hook, NY, USA, 5105–5114.

[16] Xie Qian, Lai Yu-kun, Wu Jing, Wang Zhoutao, Zhang Yiming, Xu Kai, and Wang Jun. 2020. MLCVNet: Multi-Level Context VoteNet for 3D Object Detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[17] Jiawei Ren, Liang Pan, and Ziwei Liu. 2022. Benchmarking and Analyzing Point Cloud Classification under Corruptions. CoRR abs/2202.03377 (2022). arXiv:2202.03377 https://arxiv.org/abs/2202.03377

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

[19] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. 2022. Benchmarking Robustness of 3D Point Cloud Recognition Against Common Corruptions. CoRR abs/2201.12296 (2022). arXiv:2201.12296 https://arxiv.org/abs/2201.12296

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (NIPS’17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[21] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. 2020. Pillar-based object detection for autonomous driving. In European Conference on Computer Vision. Springer, 18–34.

[22] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay P. Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cpm for learning on point clouds. ACM Transactions On Graphics (tog) 38, 5 (2019), 1–12.

[23] Zhixiong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lingxuan Zhang, Xiaosou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1912–1920.

[24] Yan Yan, Xuejiao Mao, and Bo Li. 2018. Second: Sparsely embedded convolutional detection. Sensors 18, 10 (2018), 3337.

[25] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. Pixor: Real-time 3d object detection from point clouds. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 7652–7660.

[26] Yin Zhou and Oncel Tuzel. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4480–4499.