BENCHMARKING 3D PERCEPTION ROBUSTNESS TO COMMON CORRUPTIONS AND SENSOR FAILURE

Lingdong Kong[1,2,*], **Youquan Liu**[1,3,*], **Xin Li**[1,4,*], **Runnan Chen**[1,5], **Wenwei Zhang**[1,6], **Jiawei Ren**[6], **Liang Pan**[6], **Kai Chen**[1], **Ziwei Liu**[6,✉]

[1]Shanghai AI Laboratory [2]National University of Singapore [3]Hochschule Bremerhaven
[4]East China Normal University [5]The University of Hong Kong [6]S-Lab, Nanyang Technological University
[*]The first three authors contributed equally to this work.

{konglingdong,liuyouquan,lixin,zhangwenwei,chenkai}@pjlab.org.cn
{jiawei011,liang.pan,ziwei.liu}@ntu.edu.sg

ABSTRACT

The robustness of the 3D perception system under common corruptions and sensor failure is pivotal for safety-critical applications. Existing large-scale 3D perception datasets often contain data that are meticulously cleaned. Such configurations, however, cannot reflect the reliability of perception models during the deployment stage. In this work, we contribute Robo3D, the first test suite heading toward probing the robustness of 3D detectors and segmentors under out-of-distribution scenarios against natural corruptions that occur in the real-world environment. Specifically, we consider eight corruption types (each with three severity levels) that are likely to happen under 1) *adverse weather conditions*, such as fog, rain, and snow; 2) *external disturbances* that are caused by motions or result in the missing of LiDAR beams; and 3) *internal sensor failure*, including crosstalk, possible incomplete echo, and cross-sensor scenarios. We reveal that, although promising results have been progressively achieved on standard benchmarks, the state-of-the-art 3D perception models are at risk of being vulnerable to data corruptions. Based on our observations, we further draw suggestions on aspects including LiDAR representation, training strategies, and augmentation. We hope this work could inspire follow-up research in designing more robust and reliable 3D perception models. Our robustness evaluation toolkit is publicly available at `https://github.com/ldkong1205/Robo3D`.

1 INTRODUCTION

3D perception aims to detect and segment accurate position, orientation, semantics, and temporary relation of the objects and backgrounds around the ego-vehicle in the three-dimensional world Arnold et al. (2019); Guo et al. (2020). With the emergence of large-scale driving datasets, various approaches in the fields of LiDAR semantic segmentation and 3D object detection advent each year, with record-breaking performances in mainstream benchmarksGeiger et al. (2012); Behley et al. (2019); Caesar et al. (2020); Fong et al. (2022); Kong et al. (2023c); Sun et al. (2020).

Despite the great success achieved on the "clean" evaluation sets, the model's robustness against out-of-distribution (OoD) scenarios remain obscure. Recent attempts mainly focus on probing the OoD robustness from two aspects. The first line focuses on the transfer of 3D perception models to unseen domains, *e.g.*, sim2real Xiao et al. (2022), weather2weather Jaritz et al. (2020), and city2city Kong et al. (2023b) adaptations, to probe the model's generalizability. The second line aims to design adversarial examples which can cause the model to make incorrect predictions while keeping the attacked input close to its original format Rossolini et al. (2022); Cao et al. (2021); Tu et al. (2020).

In this work, different from the above two directions, we aim at understanding the cause of model deterioration under real-world corruptions and sensor failure. Modern 3D perception models often exploit point cloud data collected from LiDAR sensors or RGB-D cameras for feature learning. The corruptions of such data formats are inevitable due to issues with data collection, processing, weather conditions, and scene complexity Ren et al. (2022). While most recent works target creating

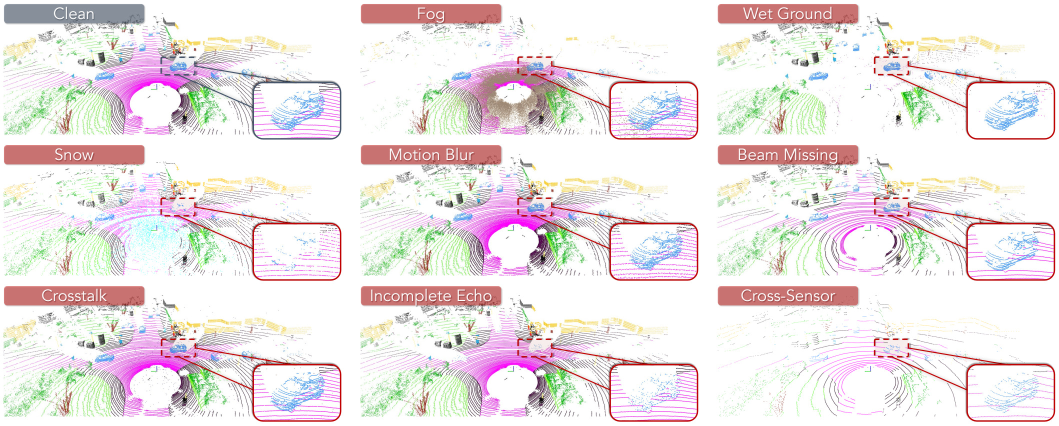


Figure 1: Taxonomy of the **Robo3D** benchmark. Eight corruption types from the following three categories are considered: 1) *Adverse weather conditions*, such as fog, rain, and snow; 2) *External disturbances* that are caused by motion blur or result in the missing of LiDAR beams; and 3) *Internal sensor failure*, including crosstalk, possible incomplete echo, and cross-sensor scenarios. Each corruption is further split into three levels (light, moderate, and heavy) based on its severity.

corrupted point clouds from indoor scenes Kar et al. (2022) or object-centric CAD models Sun et al. (2022); Zhu et al. (2022); Alliegro et al. (2022), we make one of the first attempts to simulate corruptions on large-scale LiDAR point clouds from the complex outdoor driving scenes Geiger et al. (2012); Behley et al. (2019); Caesar et al. (2020); Sun et al. (2020). As shown in Fig. 1, We consider three distinct corruption sources that are with a high likelihood to occur in deployment. 1) ***Adverse weather conditions***, including *fog*, *wet ground*, and *snow*, are causing back-scattering, attenuation, and reflection of the laser pulses Hahner et al. (2021; 2022); Shin et al. (2019). 2) ***External disturbances***, *e.g.*, bumpy surfaces, dust, insects, *etc.*, often lead to nonnegligible *motion blur* and LiDAR *beam missing* issues Phillips et al. (2017). 3) ***Internal sensor failure***, such as the *incomplete echo* or miss detection of instances with a dark color (*e.g.*, black car) and *crosstalk* among multiple sensors, will likely to deteriorate the perception accuracy Yu et al. (2022); Brinon-Arranz et al. (2021). Besides the environmental factors, it is also important to understand the *cross-sensor* discrepancy to avoid sudden failure caused by the sensor configuration change.

In this work, we apply physically-principled corruptions to the validation sets of KITTI Geiger et al. (2012), SemanticKITTI Behley et al. (2019), nuScenes Caesar et al. (2020), and Waymo Oepn Sun et al. (2020), as our corruption suite dubbed **Robo3D**. Analogous to the popular 2D corruption benchmarks Hendrycks & Dietterich (2019); Yi et al. (2021); Michaelis et al. (2019), we create three severity levels for each corruption type in our benchmark and adopt the mean corruption error (mCE) as the main indicator for robustness comparisons. Finally, we conduct in-depth experiments to understand the pros and cons of the design from existing 3D perception models.

The key contributions of this work are summarized as follows.

- We introduce Robo3D, the first systematically-designed robustness evaluation suite for LiDAR-based 3D perception under corruptions and sensor failure.
- We benchmark 22 segmentors and 12 detectors for LiDAR-based semantic segmentation and 3D object detection tasks, respectively, on their robustness against corruptions.
- Based on our observations, we draw in-depth discussions on the design receipt of building more robust 3D perception models.

2 Related Work

LiDAR-based Semantic Segmentation. The design choice of 3D segmentors often correlates with the LiDAR representations, which can be categorized into raw point Thomas et al. (2019), range view Wu et al. (2018); Milioto et al. (2019), bird's eye view Zhang et al. (2020), voxel Choy et al.

(2019), and multi-view fusion Liong et al. (2020); Xu et al. (2021) methods. The projection-based approach rasterizes irregular point clouds into 2D grids, which avoids the need for 3D operators and is thus more hardware-friendly for deployment Cortinhal et al. (2020); Zhao et al. (2021); Cheng et al. (2022); Kong et al. (2023a). The voxel-based methods which retain the 3D structure are achieving better performance than other single modalities Zhu et al. (2021); Ye et al. (2022). Efficient operators like the sparse convolution are widely adopted to ease the memory footprint Tang et al. (2020; 2022). Most recently, some works start to explore possible complementary between two views Liong et al. (2020); Yan et al. (2022); Qiu et al. (2022) or even more views Xu et al. (2021). Although promising results have been achieved, the robustness of 3D segmentors against corruptions remains obscure. As will show in the next sections, these methods are at the risk of being less robust due to the lack of a comprehensive robustness evaluation benchmark.

LiDAR-based 3D Object Detection. The 3D detectors aim to identify and determine the spatial positions and categories of objects through the sensor data. Similar to LiDAR segmentation, modern 3D object detection models also adopt various data representations. Point-based methods Shi et al. (2019); Shi & Rajkumar (2020); Yang et al. (2019; 2020) implicitly capture local structures and fine-grained patterns without any quantization to retain the original geometry of raw point clouds. Voxel-based methods Yan et al. (2018); Zhou & Tuzel (2018); Deng et al. (2021); Yin et al. (2021); Shi et al. (2020b); Mao et al. (2021); Li et al. (2021b) transform irregular point clouds to compact grids while only those non-empty voxels are stored and utilized for feature extraction through the sparse convolution Yan et al. (2018). Recently, some works Mao et al. (2021); Sheng et al. (2021); Zhou et al. (2022) start to explore long-range contextual dependencies among voxels with self-attention Vaswani et al. (2017). Pillar-based methods Lang et al. (2019); Shi et al. (2022a) better balance the accuracy and speed by controlling the resolution in the vertical axis. While point-voxel fusion method Shi et al. (2022c; 2020a) can integrate the merits of both representations to learn more discriminative features. The above methods, however, mainly focus on obtaining better performance on clean point clouds, while paying much less attention to the model robustness. Moreover, they are also prone to suffer from performance degradation under data corruptions and sensor failure.

Common Corruption. Hendrycks & Dietterich (2019) contributes the pioneering work in this line of research which benchmarks image classification models to common corruptions and perturbations on ImageNet-C. Follow-up studies extend on aspect to other perception tasks, *e.g.*, object detection Michaelis et al. (2019), image segmentation Kamann & Rother (2020), navigation Chattopadhyay et al. (2021), video classification Yi et al. (2021), and pose estimation Wang et al. (2021). The importance of evaluating model robustness has been constantly proven. Since we are targeting a different sensor, *i.e.*, LiDAR, most of the well-studied corruption types become realistic or suitable for such a data format. This motivates us to explore new taxonomy for defining more proper corruption types for the 3D perception tasks in autonomous driving scenarios.

3D Perception Robustness. Several recent studies proposed to investigate the vulnerability of point cloud classifiers and detectors in indoor scenes Kar et al. (2022); Ren et al. (2022); Sun et al. (2022); Zhu et al. (2022); Alliegro et al. (2022). Recently, there are works started to explore the robustness of 3D object detectors under adversarial attacks Park et al. (2019); Tu et al. (2020); Xie et al. (2023). In the context of corruption robustness, we notice three concurrent works Yu et al. (2022); Li et al. (2021a); Albreiki et al. (2022). These works, however, all consider 3D object detection alone and might be constrained by either a limited number of corruption types or datasets. Our benchmark properly defines a more diverse range of corruptions for the general 3D perception task and includes significantly more models from both LiDAR-based semantic segmentation and 3D object detection.

3 ROBO3D BENCHMARK

3.1 TAXONOMY

Given a point $p \in \mathbb{R}^4$ in a LiDAR point cloud with coordinates (p^x, p^y, p^z) and intensity p^i, our goal is to simulate a corrupted point via a mapping $\hat{p} = \mathcal{F}(p)$, with rules constrained by physical principles or engineering experience. Tailored for the LiDAR-based 3D perception task, we summarize eight corruption types commonly occurring in real-world deployment in our benchmark (see Fig. 1).

Specifically, we consider common corruptions from three main categories: 1) *Adverse weather conditions*, such as fog, rain, and snow; 2) *External disturbances* that are caused by motions or result in the missing of LiDAR beams; and 3) *Internal sensor failure*, including LiDAR crosstalk, incomplete echo, and cross-sensor scenarios.

Fog. The LiDAR sensor emits laser pulses for accurate range measurement. Back-scattering and attenuation of LiDAR points tend to happen in foggy weather since the water particles in the air will cause inevitable pulse reflection Bijelic et al. (2020). In our benchmark, we adopt the physically valid fog simulation method Hahner et al. (2021) to create fog-corrupted data. For each p, we calculate its attenuated response $p^{i_{\text{hard}}}$ and the maximum fog response $p^{i_{\text{soft}}}$ as follows:

$$\hat{p} = \mathcal{F}_{\text{fog}}(p) = \begin{cases} (\hat{p}^x, \hat{p}^y, \hat{p}^z, p^{i_{\text{soft}}}), & \text{if} \quad p^{i_{\text{soft}}} > p^{i_{\text{hard}}}, \\ (p^x, p^y, p^z, p^{i_{\text{hard}}}), & \text{else}. \end{cases} \tag{1}$$

Wet Ground. The laser pulses will likely lose certain amounts of energy when hitting wet surfaces, which causes significantly attenuated laser echoes depending on the water height d_w and mirror refraction rate Shin et al. (2019). We follow Hahner et al. (2022) to model the attenuation caused by ground wetness. A pre-processing step is taken to estimate the ground plane with existing semantic labels or RANSAC Fischler & Bolles (1981). Next, a ground plane point of its measured intensity $\hat{p}^i$ is obtained based on the modified reflectivity, and the point is only kept if its intensity is greater than the noise floor i_n:

$$\hat{p} = \mathcal{F}_{\text{wet}}(p) = \begin{cases} (p^x, p^y, p^z, \hat{p}^i), & \text{if} \quad \hat{p}^i > i_n \text{ and } p \in \text{ground plane}, \\ \text{None}, & \text{elif} \quad \hat{p}^i < i_n \text{ and } p \in \text{ground plane}, \\ (p^x, p^y, p^z, p^i), & \text{elif} \quad p \notin \text{ground plane}. \end{cases} \tag{2}$$

Snow. For each LiDAR beam, the set of particles that intersect with it and derive the angle of the beam cross-section that is reflected by each particle, taking potential occlusions into account. We follow Hahner et al. (2022) Seppänen et al. (2022) to simulate snow-corrupted data which is similar to the fog simulation. This physically principled method samples snow particles in 2D space for each LiDAR line and modifies the measurement for each LiDAR beam in accordance with the induced geometry. Note that the number of sampling snow particles is set based on the snowfall rate r_s.

Motion Blur. Since the LiDAR sensor is often mounted on the rooftop or side of the vehicle, it inevitably suffers from the blur caused by vehicle movement, especially on bumpy surfaces. To simulate blur-corrupted data, we add a jittering noise to each p with a translation value sampled from the Gaussian distribution with standard deviation σ_t.

Beam Missing. The dust and insect tend to form agglomerates in front of the LiDAR surface and will not likely disappear without human intervention, such as drying and cleaning Phillips et al. (2017). This type of occlusion causes zero readings on masked areas and results in the loss of certain light impulses. To mimic such a behavior, we randomly sample a total number of m beams and drop points on these beams from the original point cloud.

Crosstalk. Considering that the road is often shared by multiple vehicles, the time-of-flight of light impulses from one sensor might interfere with impulses from other sensors within a similar frequency range Brinon-Arranz et al. (2021). Such a crosstalk phenomenon often creates noisy points within the mid-range areas in between two (or multiple) sensors. To simulate this corruption, we randomly sample a subset of k_t percent points from the original point cloud and add large jittering noise with a translation value sampled from the Gaussian distribution with standard deviation σ_c.

Incomplete Echo. The near-infrared spectrum of the laser pulse emitted from the LiDAR sensor is vulnerable to vehicles or other instances with dark colors Yu et al. (2022). The LiDAR readings are thus incomplete in such scan echoes, resulting in significant point miss detection. We simulate this corruption by randomly querying k_e percent points for *vehicle*, *bicycle*, and *motorcycle* classes, via either semantic masks or 3D bounding boxes. Next, we drop the queried points from the original point cloud, along with their point-level semantic labels. Note that we do not alter the ground-truth bounding boxes since they should remain at their original positions in the real world.

Cross-Sensor. Due to the large variety of LiDAR sensor configurations (*e.g.*, beam number), it is important to design robust 3D perception models that are capable of maintaining satisfactory performance under cross-device cases Yang et al. (2019). While previous works directly form such settings with two different datasets, the domain idiosyncrasy in between (*e.g.*, different label mappings and data collection protocols) further hinders the direct robustness comparison. In our benchmark, we follow Wei et al. (2022) and generate cross-sensor point clouds by first dropping points of certain beams from the point clouds and then sub-sample k_c percent points from each beam.

3.2 Corruption Set

Following the above taxonomy, we create new robustness evaluation sets upon the validation sets of existing large-scale 3D perception datasets Geiger et al. (2012); Behley et al. (2019); Caesar et al. (2020); Fong et al. (2022); Sun et al. (2020) to fulfill SemanticKITTI-C, KITTI-C, nuScenes-C, and WOC-C. They are constructed with eight corruption types under three severity, resulting in a total number of 97704, 90456, 144456, and 143424 annotated LiDAR point clouds, respectively.

KITTI-C adopts 3769 samples from the KITTI dataset Geiger et al. (2012) under the *val* split defined in Chen et al. (2015). The mean average precision (mAP) is used to measure the detection accuracy, which is calculated among three classes: *car*, *pedestrian*, and *cyclist*.

SemanticKITTI-C is generated using samples from the validation set of SemanticKITTI Behley et al. (2019), which consists of 4071 scans collected by a 64-beam LiDAR sensor. 19 semantic classes are chosen for evaluation after merging classes with distinct moving statuses and discarding classes with very few points. The mean intersection-over-union (mIoU) is used to measure accuracy.

nuScenes-C uses samples from the official *val* split in the nuScenes dataset Caesar et al. (2020) and collected by a 32-beam LiDAR sensor. There are 16 semantic classes, where the 10 *thing* classes are used for detection. We adopt mIoU/NDS to measure the segmentation/detection accuracy.

WOD-C is built with the 5976 scans from the 3D semantic segmentation validation set in the Waymo Open dataset Sun et al. (2020). We adopt to same scans for 3D segmentation and detection to keep consistency. The 3D segmentation task aims to classify 23 semantic classes, including one ignored and 22 valid semantic categories, while the 3D detection task targets to detect the *vehicle*, *pedestrian*, and *cyclist* classes. We adopt mIoU/mAP to measure the segmentation/detection accuracy.

3.3 Evaluation Metric

Corruption Error (CE). We follow Hendrycks & Dietterich (2019) and use the mean CE (mCE) as the primary metric in comparing models' robustness. To normalize the severity effects, we choose SECOND Yan et al. (2018) and MinkUNet Tang et al. (2020) as the baseline models for detectors and segmentors, respectively. The CE and mCE scores are calculated as follows.

$$\text{CE}_i = \frac{\sum_{l=1}^{3}(1-\text{Acc}_{i,l})}{\sum_{l=1}^{3}(1-\text{Acc}_{i,l}^{\text{baseline}})}~, \quad \text{mCE} = \frac{1}{N}\sum_{i=1}^{N}\text{CE}_i~, \tag{3}$$

where $\text{Acc}_{i,l}$ denotes the task-specific accuracy score, *i.e.*, mIoU, AP, NDS, or APH, on a corruption type i at severity level l. $N = 8$ denotes the total number of corruption types in our benchmark.

Resilience Rate (RR). We define mean RR (mRR) as the relative robustness indicator for measuring how much accuracy can a model retain when evaluated on our corruption sets. The RR and mRR scores are calculated as follows.

$$\text{RR}_i = \frac{\sum_{l=1}^{3}\text{Acc}_{i,l}}{3\times\text{Acc}_{\text{clean}}}~, \quad \text{mRR} = \frac{1}{N}\sum_{i=1}^{N}\text{RR}_i~, \tag{4}$$

where $\text{Acc}_{\text{clean}}$ denotes the task-specific accuracy score on the "clean" evaluation set.

3.4 Protocol

Most of the models in our benchmark follow similar data augmentation, pre-training, and validation configurations. We thus directly use public checkpoints for model evaluation whenever applicable, or re-train the model following default settings. We notice that some models used extra tricks on the original validation sets, *e.g.*, test-time augmentation and model ensemble. For such cases, we re-train their models with conventional settings and report the reproduced results. To ensure a fair comparison, we encourage future works to follow our standard protocols.

Table 1: Results for the **Corruption Error (CE)** of each method on *SemanticKITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row. Symbol † denotes the baseline model adopted in calculating the CE scores.

Method	mCE↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
MinkUNet$_{18}$ Choy et al. (2019)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SqueezeSeg Wu et al. (2018)	164.9	183.9	158.0	165.5	122.4	171.7	188.1	158.7	170.8
SqueezeSegV2 Wu et al. (2019)	152.5	168.5	141.2	154.6	115.2	155.2	176.0	145.3	163.5
RGNet$_{21}$ Milioto et al. (2019)	136.3	156.3	128.5	133.9	102.6	141.6	148.9	128.3	150.6
RGNet$_{53}$ Milioto et al. (2019)	130.7	144.3	123.7	128.4	104.2	135.5	129.4	125.8	153.9
SalsaNext Cortinhal et al. (2020)	116.1	147.5	112.1	116.6	77.6	115.3	143.5	114.0	102.5
FIDNet Zhao et al. (2021)	113.8	127.7	105.1	107.7	88.9	116.0	121.3	113.7	130.0
CENet Cheng et al. (2022)	103.4	129.8	92.7	99.2	70.5	101.2	131.1	102.3	100.4
PolarNet Zhang et al. (2020)	118.6	138.8	107.1	108.3	86.8	105.1	178.1	112.0	112.3
KPConv Thomas et al. (2019)	99.5	103.2	91.9	98.1	110.7	97.6	111.9	97.3	85.4
PIDS$_{1.2\times}$ Zhang et al. (2023)	104.1	118.1	98.9	109.5	114.8	103.2	103.9	97.0	87.6
PIDS$_{2.0\times}$ Zhang et al. (2023)	101.2	110.6	95.7	104.6	115.6	98.6	102.2	97.5	84.8
WaffleIron Puy et al. (2023)	109.5	123.5	90.1	108.5	99.9	93.2	186.1	**91.0**	84.1
MinkUNet$_{34}$ Choy et al. (2019)	100.6	105.3	99.4	106.7	98.7	97.6	99.9	99.0	98.3
Cy3D$_{\text{SPC}}$ Zhu et al. (2021)	103.3	142.5	92.5	113.6	70.9	97.0	105.7	104.2	99.7
Cy3D$_{\text{TSC}}$ Zhu et al. (2021)	103.1	142.5	101.3	116.9	61.7	98.9	111.4	99.0	93.4
SPVCNN$_{18}$ Tang et al. (2020)	100.3	101.3	100.0	104.0	97.6	99.2	100.6	99.6	100.2
SPVCNN$_{34}$ Tang et al. (2020)	**99.2**	**98.5**	100.7	102.0	97.8	99.0	**98.4**	98.8	98.1
RPVNet Xu et al. (2021)	111.7	118.7	101.0	104.6	78.6	106.4	185.7	99.2	99.8
CPGNet Li et al. (2022)	107.3	141.0	92.6	104.3	**61.1**	**90.9**	195.6	95.0	**78.2**
2DPASS Yan et al. (2022)	106.1	134.9	**85.5**	110.2	62.9	94.4	171.7	96.9	92.7
GFNet Qiu et al. (2022)	108.7	131.3	94.4	**92.7**	61.7	98.6	198.9	98.2	93.6

4 Experimental Analysis

4.1 Benchmarked Method

We benchmark a total number of 34 LiDAR-based 3D perception models, including model variants. **Detectors:** SECOND Yan et al. (2018), PointPillars Lang et al. (2019) PointRCNN Shi et al. (2019), Part-A2 Shi et al. (2020b), PV-RCNN Shi et al. (2020a), CenterPoint Yin et al. (2021), and PV-RCNN++ Shi et al. (2022b). **Segmentors:** SqueezeSeg Wu et al. (2018), SqueezeSegV2 Wu et al. (2019), RangeNet++ Milioto et al. (2019), SalsaNext Cortinhal et al. (2020), FIDNet Zhao et al. (2021), CENet Cheng et al. (2022), PolarNet Zhang et al. (2020), KPConv Thomas et al. (2019), PIDS Zhang et al. (2023), WaffleIron Puy et al. (2023), MinkUNet Choy et al. (2019), Cylinder3D Zhu et al. (2021), SPVCNN Tang et al. (2020), RPVNet Xu et al. (2021), CPGNet Li et al. (2022), 2DPASS Yan et al. (2022), and GFNet Qiu et al. (2022).

4.2 LiDAR Semantic Segmentation

Corruption Robustness. We show the robustness evaluation results in Table 1, Table 2, Table 5, Table 6, Table 7 and Table 8. For SemanticKITTI-C, we observe that corruptions of adverse weather conditions and internal sensor failure affect the performance of models the most. All corruptions cause significant performance drops for the projection-based methods. For example, *fog* leads to a 30.37% mCE on average higher than the baseline for all models, which brings the great detrimental effect of adverse weather conditions on 3D segmentors. Besides, *crosstalk* is also a challenging one for all models and has a great impact on projection-based methods and fusion models, with 152.06% mCE and 159.02% mCE on average, respectively. On the other hand, most models exhibit robustness under *motion blur*, mainly due to model training with similar data augmentations.

For nuScenes-C, we find that the *motion blur* corruption significantly worsens all models, with on average 154.60% mCE. Similar to SemanticKITTI-C, the corruptions caused by *beam missing* and *cross-sensor* remarkably destroy the robustness of projection-based models. The notable difference with SemanticKITTI-C is that most models are not resistant to weather-related corruptions. We conjecture that the adverse weather samples (*e.g.*, *fog* and *wet ground*) contained in the nuScenes dataset enable the 3D segmentors to predict robustly under our simulated weather conditions.

Table 2: Results for the **Corruption Error (CE)** of each method on *nuScenes-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
MinkUNet$_{18}$ Choy et al. (2019)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
FIDNet Zhao et al. (2021)	122.4	75.9	122.6	68.8	192.0	164.8	58.0	141.7	155.6
CENet Cheng et al. (2022)	112.8	71.2	115.5	64.3	156.7	159.0	53.3	129.1	153.4
PolarNet Zhang et al. (2020)	115.1	90.1	115.3	59.0	208.2	121.1	80.7	128.2	118.2
WaffleIron Puy et al. (2023)	106.7	94.7	99.9	84.5	152.4	110.7	91.1	106.4	114.2
MinkUNet$_{34}$ Choy et al. (2019)	96.4	93.0	96.1	104.8	**93.1**	**95.0**	96.3	**96.9**	**95.9**
Cy3D$_{\text{SPC}}$ Zhu et al. (2021)	111.8	86.6	104.7	70.3	217.5	113.0	75.7	109.2	117.8
Cy3D$_{\text{TSC}}$ Zhu et al. (2021)	105.6	83.2	111.1	69.7	165.3	114.0	74.4	110.7	116.2
SPVCNN$_{18}$ Tang et al. (2020)	106.7	88.4	105.6	98.8	156.5	110.1	86.0	104.3	103.6
SPVCNN$_{34}$ Tang et al. (2020)	97.5	95.2	99.5	97.3	95.3	98.7	97.9	**96.9**	98.7
2DPASS Yan et al. (2022)	98.6	76.6	**89.1**	76.4	142.7	102.2	89.4	101.8	110.4
GFNet Qiu et al. (2022)	**92.6**	**65.6**	93.8	**47.2**	152.5	112.9	**45.3**	105.5	117.6

Table 3: Results for the **Corruption Error (CE)** of each method on *KITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row. Symbol † denotes the baseline model adopted in calculating the CE scores.

Method	mCE ↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
CenterPoint Yin et al. (2021)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SECOND Yan et al. (2018)	95.9	99.7	100.6	87.6	97.6	91.5	96.5	99.2	94.8
PointPillars Lang et al. (2019)	110.7	115.8	106.4	124.9	101.6	95.3	117.6	109.9	113.9
PointRCNN Shi et al. (2019)	91.9	93.2	90.1	96.8	93.1	86.1	100.9	92.4	**82.5**
PartA^2-F Shi et al. (2020b)	**82.2**	**89.4**	**75.8**	**81.3**	**86.2**	**80.9**	**71.8**	**83.6**	88.9
PartA^2-A Shi et al. (2020b)	88.6	92.6	83.2	94.6	86.4	87.0	83.2	89.3	92.7
PVRCNN Shi et al. (2020a)	90.0	95.2	86.6	93.1	87.5	86.0	87.1	90.0	94.7

Comparisons among Segmentors. For SemanticKITTI-C: Among 22 segmentors, we observe that KPConv Thomas et al. (2019) and SPVCNN Tang et al. (2020) have better relative robustness than other models, with mCE scores lower than the baseline. The worst-performing model is SqueezeSeg Wu et al. (2018), with a 164.87% mCE and showing weak robustness to all corruptions. For nuScenes-C (Seg3D): We observe that among the 12 segmentors, GFNet Qiu et al. (2022) and MinkUNet Choy et al. (2019) possess greater ability against common corruptions, with 7.45% mCE and 3.63% mCE lower than the baseline, respectively. FIDNet Zhao et al. (2021) is the worst model with the highest mCE compared to others. In contrast to other models and SemanticKITTI-C, the projection-based models Milioto et al. (2019); Zhao et al. (2021); Cheng et al. (2022) demonstrate greater robustness to weather-related corruptions and less sensitivity to crosstalk. Unsurprisingly, GFNet Qiu et al. (2022) possesses a greater ability against weather-related corruptions and the *crosstalk* corruption, due to the way it is constructed by both range view and polar view.

Key Observations.

1. The projection-based methods (range view and polar view) tend to be the most vulnerable ones to common corruption according to the mCE metric. Adverse weather conditions like *fog* and *snow* cause messy point clouds and further hamper the spherical projection. The external disturbances like *beam missing* and the internal sensor failure of the *cross-sensor* scenario drop point clouds with severity, and result in information losses for the range view methods.

2. The point-based methods show superior performance on the "clean" distribution. They are the most robust against weather-related corruptions but are sensitive to *crosstalk*. WaffleIron is greatly robust against the scenario of *incomplete echo*, which drops point clouds at the instance level.

3. The pure voxel-based methods show the most superior robustness across all corruptions, except for the *fog* corruption. We conjecture that the voxelization process is conducive to mitigating the local noise and thus such quantization operation can help models against the missing of points in local regions. The cylindrical partition type of voxelization greatly improves the robustness of *motion blur* but is more vulnerable to other corruptions, *i.e.*, adverse weathers.

Table 4: Results for the **Corruption Error (CE)** of each method on *nuScenes-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row. Symbol † denotes the baseline model adopted in calculating the CE scores.

Method	mCE ↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
CenterPoint Yin et al. (2021)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SECOND Yan et al. (2018)	97.5	95.4	96.0	96.1	100.8	99.3	92.2	97.6	102.6
PointPillars Lang et al. (2019)	102.9	102.9	104.6	102.5	106.4	102.4	100.9	102.4	**101.1**
CenterPoint-LR Yin et al. (2021)	98.7	97.9	96.5	97.7	102.2	101.1	95.5	**95.6**	103.5
CenterPoint-HR Yin et al. (2021)	**95.8**	**93.0**	**92.0**	**94.9**	**97.6**	**98.4**	**91.1**	96.2	103.2

4.3 3D Object Detection

Corruption Robustness. Table 13, Table 3, Table 12, Table 4, Table 14, and Table 15 show the 3D detection robustness on the KITTI-C and nuScenes-C (Det3D) datasets. Among all corruption types in these two datasets, sensor-level ones degrade the model performance the most, and the normal working state of the sensor plays a key role in the safety of autonomous driving. The two datasets are equipped with different LiDAR sensors, KITTI-C uses 64 beams while nuScenes-C only has 32 beams. It can be seen that the overall mRR and mCE of KITTI-C are better than that of nuScenes-C, which also proves that high-precision LiDAR has a better effect on robustness for different corruptions. From Table 13 and Table 15 we can observe that the two worst corruptions are *cross-sensor* and *snow*, which almost degrades the model performance over 5% AP or 20% NDS in KITTI-C and nuScenes-C, respectively, for all detectors. Among these corruptions, sensor-level ones are significantly more detrimental to both the two-stage and one-stage detectors than the weather-related corruptions. In particular, we note that the *wet ground* corruption brings a tiny effect to different 3D object detection models.

Comparisons among Detectors. As shown in Table 13 and Table 15, two-stage detectors Liu et al. (2019); Shi et al. (2020b) are more robust against the corruptions compared to one-stage ones Yan et al. (2018); Lang et al. (2019). The voxel-based methods Yan et al. (2018); Yin et al. (2021); Liu et al. (2019) also outperform the pillar-based methods Lang et al. (2019) and point-based methods Shi et al. (2019). A smaller grid size can bring lower mCE and higher mRR, improving the robustness of detectors under different corruptions. One explanation is that the spatial quantization of the point cloud by voxelization mitigates the local randomness and the absence of points caused by corruptions. We also notice that PointPillars Lang et al. (2019) perform the worst among these corruptions, which was developed for improving the efficiency of 3D object detection.

Key Observations.

1. The pillar-based methods Lang et al. (2019) are the most vulnerable ones to common corruptions according to the mCE metric. There exists a clear margin of robustness among the voxel-based Yan et al. (2018), point-based Shi et al. (2019), and point-voxel fusion Shi et al. (2020a) detectors.

2. The voxel-based two-stage detectors show superior performance against adverse weather corruptions but are less robust to sensor-level corruptions. They often rasterize the point clouds into voxel grids to mitigate the local noise and enhance the models against the corruption of point clouds.

3. Part-A2 shows a great ability against corruptions on KITTI-C, even better than the point-voxel fusion method Shi et al. (2020a). It also indicates that the point-based feature representation of point clouds is sensitive to corruptions and tends to cause performance degradation.

5 Conclusion

In this work, we establish a comprehensive evaluation suite dubbed *Robo3D* for probing the robustness of LiDAR-based 3D perception models. We define eight distinct corruption types with three severity on large-scale datasets. We systematically benchmarked and analyzed representative 3D detectors and segmentors to understand their resilience under real-world corruptions and sensor failure. We hope this work could lay a solid foundation for future research on 3D robustness.

A Appendix

In this appendix, we supplement the following materials to support the findings and observations in the main body of this paper.

- Sec. A.1 lists implementation details for corruption generation.
- Sec. A.2 includes visual comparisons for each corruption type across three severity levels.
- Sec. A.3 provides additional benchmarking results for a more comprehensive comparison.
- Sec. A.4 acknowledges the public resources used during the course of this work.

A.1 Implementation detail

We include necessary details for generating the eight common corruptions under three severity in the four corruption sets, *i.e.*, KITTI-C, SemanticKITTI-C, nuScenes-C, and WOD-C.

Fog Simulation. Following Hahner et al. (2021), we uniformly sample the attenuation coefficient α from $[0, 0.005, 0.01, 0.02, 0.03, 0.06]$. For the *SemanticKITTI-C*, *KITTI-C*, *nuScenes-C*, and *WOD-C* datasets, we set the back-scattering coefficient β to $\{0.008, 0.05, 0.2\}$ to split severity levels into light, moderate, and heavy levels. The semantic classes of*fog* are 21, 41, and 23 for *SemanticKITTI-C*, *nuScenes-C*, and *WOD-C*, respectively. And $\mathbf{p}$ belongs to fog class will be mapped to class 0 or 255 (*i.e.*, the *ignored* label).

Wet Ground Simulation. We follow Hahner et al. (2022) and set the parameter of water height d_w to $\{0.2\ mm, 1.0\ mm, 1.2\ mm\}$ for different severity levels of *wet ground.* Note that the ground plane estimation method is different across four benchmarks. We estimate the ground plane via RANDSAC Fischler & Bolles (1981) for the *KITTI-C* since it only provides detection labels. For *SemanticKITTI-C*, we use semantic classes of *road*, *parking*, *sidewalk*, and *other ground* to build the ground plane. The *driveable surface*, *other flat*, and *sidewalk* classes are used to construct the ground plane in *nuScenes-C*. For *WOD-C*, the ground plane is estimated by *curb*, *road*, *other ground*, *walkable*, and *sidewalk* classes.

Snow Simulation. We use the method proposed in Hahner et al. (2022) to construct *snow* corruptions. The value of snowfall rate parameter r_s is set to $\{0.5, 1.0, 2.5\}$ to simulate light, moderate, and heavy snowfall for the *SemanticKITTI-C*, *KITTI-C*, *nuScenes-C*, and *WOD-C* datasets, and the ground plane estimation is the same as the *wet ground* simulation. The semantic class of snow is 22, 42, and 24 for the *SemanticKITTI-C*, *nuScenes-C*, and *WOD-C* datasets, respectively. And $\mathbf{p}$ belongs to snow class will also be mapped to class 0 or 255 (*i.e.*, the *ignored* label).

Motion Blur Simulation. We add jittering noise from Gaussian distribution with standard deviation σ_t to simulate motion blur. The σ_t is set to $\{0.20, 0.25, 0.30\}$, $\{0.04, 0.08, 0.10\}$, $\{0.20, 0.30, 0.40\}$ and $\{0.06, 0.10, 0.13\}$ for the *SemanticKITTI-C*, *KITTI-C*, *nuScenes-C*, and *WOD-C* datasets, respectively.

Beam Missing Simulation. The value of parameter m (number of beams to be dropped) is set to $\{48, 32, 16\}$ for the benchmark of *SemanticKITTI-C*, *KITTI-C* and *WOD-C*, respectively, while set as $\{24, 16, 8\}$ for the *nuScenes-C* dataset.

Crosstalk Simulation. We set the parameter of k_t to $\{0.006, 0.008, 0.01\}$ for the *SemanticKITTI-C*, *KITTI-C*, and *WOD-C* datasets, respectively, and $\{0.03, 0.07, 0.12\}$ for *nuScenes-C* dataset. The semantic class of crosstalk is assigned to 23, 43, and 25 for *SemanticKITTI-C*, *nuScenes-C*, and *WOD-C* datasets, respectively. Meanwhile, the $\mathbf{p}$ belongs to crosstalk class will also be mapped to class 0 or 255 (*i.e.*, the *ignored* label).

Incomplete Echo Simulation. For *SemanticKITTI-C*, the point labels of classes *car*, *bicycle*, *motorcycle*, *truck*, *other-vehicle* are used as the semantic mask. For *nuScenes-C*, we include *bicycle*, *bus*, *car*, *construction vehicle*, *motorcycle*, *truck* and *trailer* class label to build semantic mask. For *WOD-C*, we adopt the point labels of classes *car*, *truck*, *bus*, *other-vehicle*, *bicycle*, *motorcycle* as the semantic mask. For *KITTI-C*, we use 3D bounding box labels to create the semantic mask. The value of parameter k_e is set to $\{0.75, 0.85, 0.95\}$ for the four corruption sets during the *incomplete echo* simulation.

Cross-Sensor Simualtion. The value of parameter m is set to $\{48, 32, 16\}$ for the *SemanticKITTI-C*, *KITTI-C*, and *WOD-C* datasets, respectively, and $\{24, 16, 12\}$ for the *nuScenes-C* dataset. Based on Wei et al. (2022), we then sub-sample 50% points from the remaining point clouds with an equal interval.

A.2 Additional Visual Example

Additional examples for each corruption under three severity levels are shown in Fig. 2 and Fig. 3.

A.3 Additional Experimental Result

Additional statistical analyses are shown in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9.

A.4 Public Resources Used

In this section, we acknowledge the use of the following public resources, during the course of this work:

• SemanticKITTI[1]	CC BY-NC-SA 4.0
• SemanticKITTI-API[2]	MIT License
• nuScenes[3]	CC BY-NC-SA 4.0
• nuScenes-devkit[4]	Apache License 2.0
• Waymo Open Dataset[5]	Waymo Dataset License
• RangeNet++[6]	MIT License
• SalsaNext[7]	MIT License
• FIDNet[8]	Unknown
• CENet[9]	MIT License
• KPConv-PyTorch[10]	MIT License
• PIDS[11]	MIT License
• WaffleIron[12]	Apache License 2.0
• PolarSeg[13]	BSD 3-Clause License
• MinkowskiEngine[14]	MIT License
• Cylinder3D[15]	Apache License 2.0
• PyTorch-Scatter[16]	MIT License
• SpConv[17]	Apache License 2.0
• TorchSparse[18]	MIT License

[1] http://semantic-kitti.org.

[2] https://github.com/PRBonn/semantic-kitti-api.

[3] https://www.nuscenes.org/nuscenes.

[4] https://github.com/nutonomy/nuscenes-devkit.

[5] https://waymo.com/open.

[6] https://github.com/PRBonn/lidar-bonnetal.

[7] https://github.com/TiagoCortinhal/SalsaNext.

[8] https://github.com/placeforyiming/IROS21-FIDNet-SemanticKITTI.

[9] https://github.com/huixiancheng/CENet.

[10] https://github.com/HuguesTHOMAS/KPConv-PyTorch.

[11] https://github.com/lordzth666/WACV23_PIDS-Joint.

[12] https://github.com/valeoai/WaffleIron.

[13] https://github.com/edwardzhou130/PolarSeg.

[14] https://github.com/NVIDIA/MinkowskiEngine.

[15] https://github.com/xinge008/Cylinder3D.

[16] https://github.com/rusty1s/pytorch_scatter.

[17] https://github.com/traveller59/spconv.

[18] https://github.com/mit-han-lab/torchsparse.

SPVCNN[19]	MIT License
CPGNet[20]	Unknown
2DPASS[21]	MIT License
GFNet[22]	Unknown
PointPillars[23]	Unknown
second.pytorch[24]	MIT License
OpenPCDet[25]	Apache License 2.0
PointRCNN[26]	MIT License
PartA2-Net[27]	Apache License 2.0
PV-RCNN[28]	Unknown
CenterPoint[29]	MIT License
lidar-camera-robust-benchmark[30]	Unknown
LiDAR-fog-sim[31]	NonCommercial 4.0
LiDAR-snow-sim[32]	NonCommercial 4.0
mmdetection3d[33]	Apache License 2.0

[19] https://github.com/mit-han-lab/spvnas.

[20] https://github.com/huixiancheng/No-CPGNet.

[21] https://github.com/yanx27/2DPASS.

[22] https://github.com/haibo-qiu/GFNet.

[23] https://github.com/zhulf0804/PointPillars.

[24] https://github.com/traveller59/second.pytorch.

[25] https://github.com/open-mmlab/OpenPCDet.

[26] https://github.com/sshaoshuai/PointRCNN.

[27] https://github.com/sshaoshuai/PartA2-Net.

[28] https://github.com/sshaoshuai/PV-RCNN.

[29] https://github.com/tianweiy/CenterPoint.

[30] https://github.com/kcyu2014/lidar-camera-robust-benchmark.

[31] https://github.com/MartinHahner/LiDAR_fog_sim.

[32] https://github.com/SysCV/LiDAR_snow_sim.

[33] https://github.com/open-mmlab/mmdetection3d.

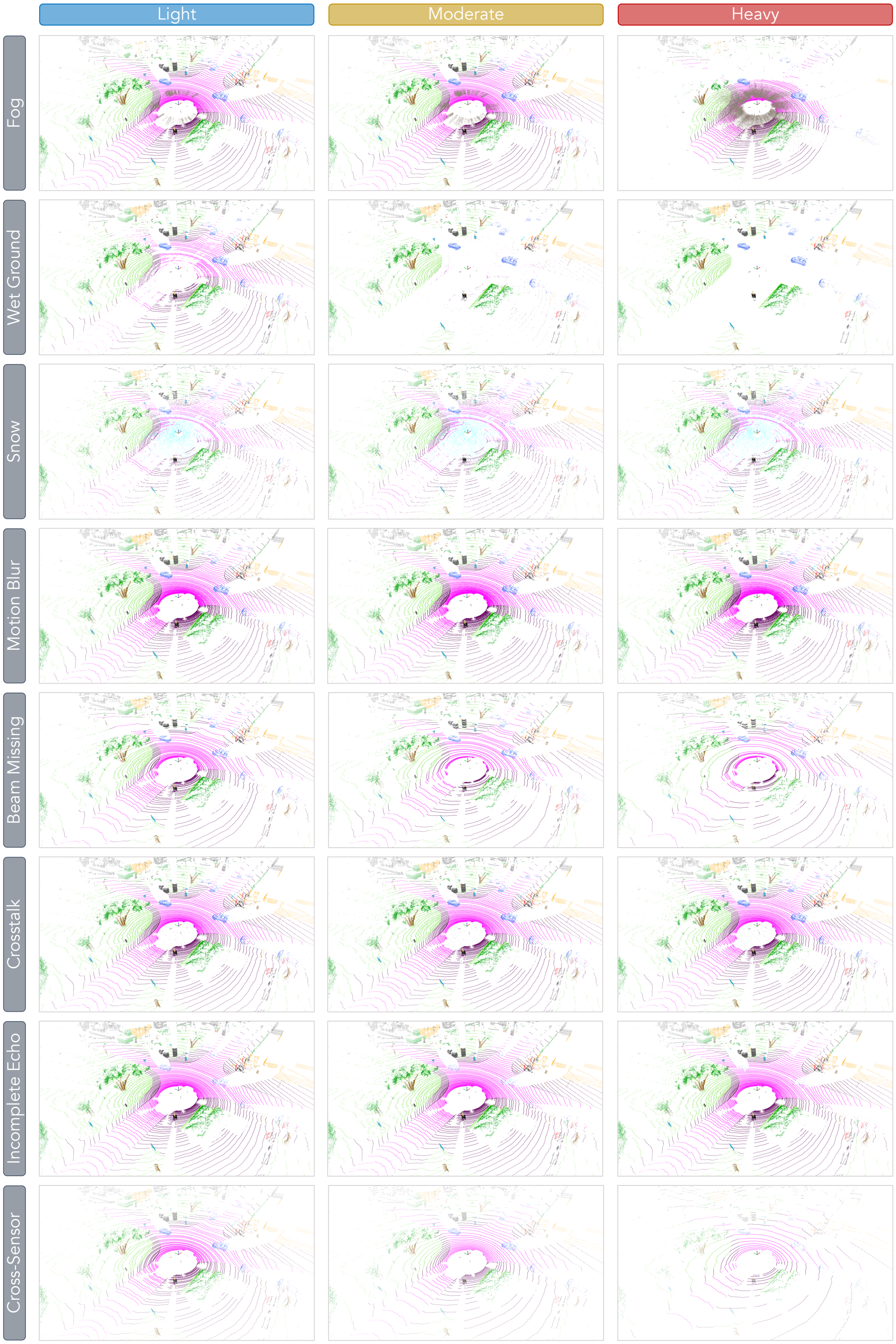


Figure 2: Visual examples of each corruption under three severity levels in our *Robo3D* benchmark.

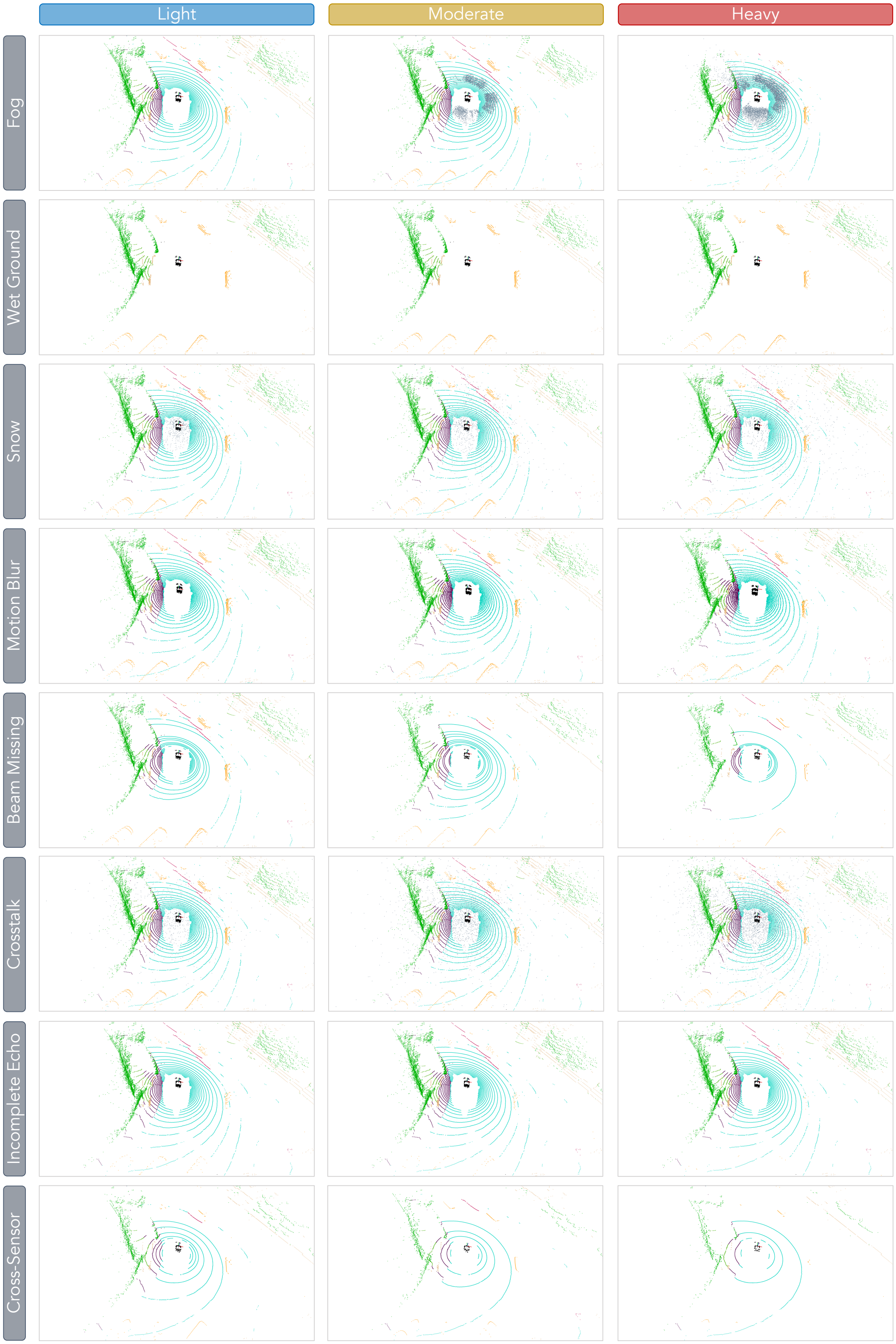
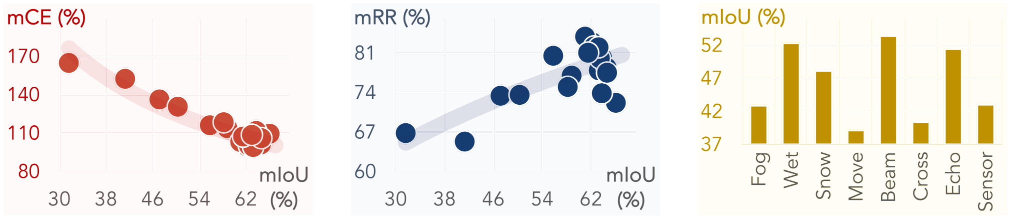
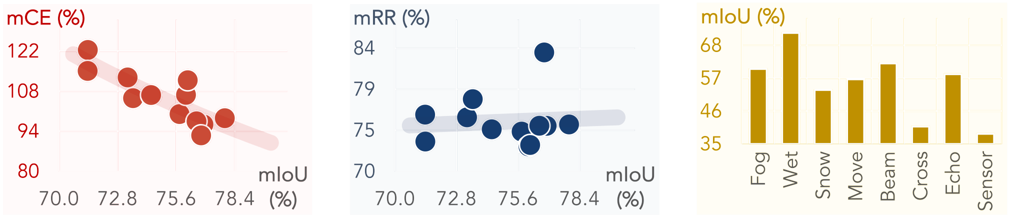


Figure 3: Visual examples of each corruption under three severity levels in our *Robo3D* benchmark.



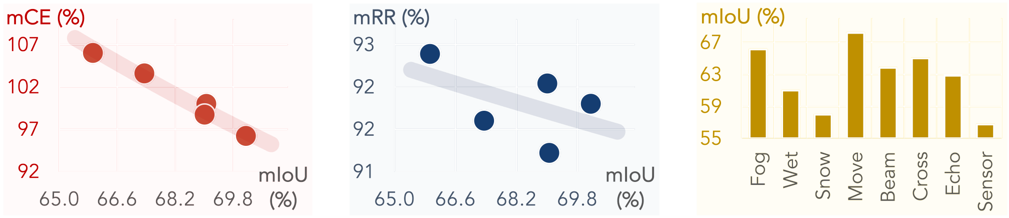
(a) mCE *vs.* mIoU (b) mRR *vs.* mIoU (c) Per-Corruption mIoUs

Figure 4: Statistical analysis among 22 LiDAR semantic segmentation models on the proposed *SemanticKITTI-C* dataset in our *Robo3D* benchmark.



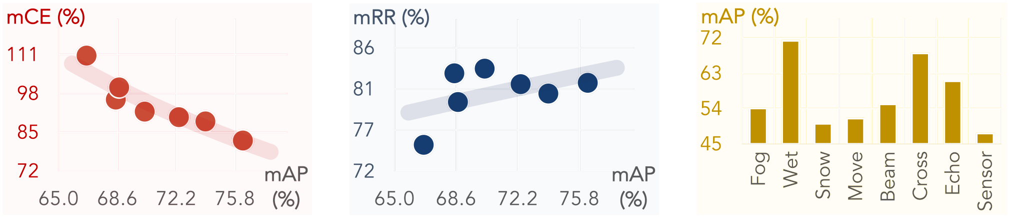
(a) mCE *vs.* mIoU (b) mRR *vs.* mIoU (c) Per-Corruption mIoUs

Figure 5: Statistical analysis among 12 LiDAR semantic segmentation models on the proposed *nuScenes-C (Seg)* dataset in our *Robo3D* benchmark.



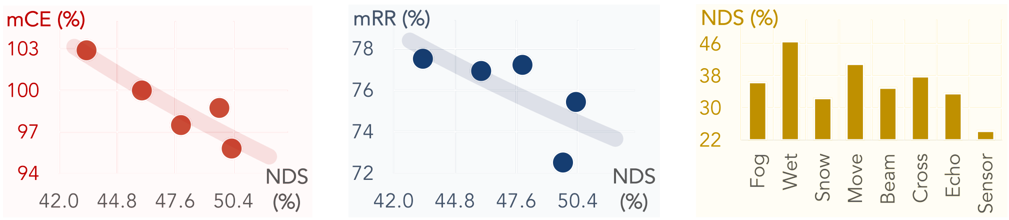
(a) mCE *vs.* mIoU (b) mRR *vs.* mIoU (c) Per-Corruption mIoUs

Figure 6: Statistical analysis among 5 LiDAR semantic segmentation models on the proposed *WOD-C (Seg)* dataset in our *Robo3D* benchmark.



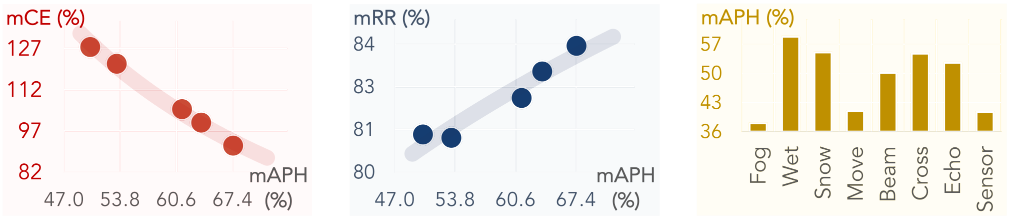
(a) mCE *vs.* mAP (b) mRR *vs.* mAP (c) Per-Corruption mAPs

Figure 7: Statistical analysis among 7 3D object detection models on the proposed *KITTI-C* dataset in our *Robo3D* benchmark.



(a) mCE *vs.* NDS (b) mRR *vs.* NDS (c) Per-Corruption NDSs

Figure 8: Statistical analysis among 5 3D object detection models on the proposed *nuScenes-C (Det)* dataset in our *Robo3D* benchmark.



(a) mCE *vs.* NDS (b) mRR *vs.* NDS (c) Per-Corruption mAPHs

Figure 9: Statistical analysis among 5 3D object detection models on the proposed *WOD-C (Det)* dataset in our *Robo3D* benchmark.

Table 5: Results for the **Resilience Rate (RR)** of each method on *SemanticKITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mIoU ↑
SqueezeSeg	66.81	59.63	86.37	71.81	56.72	79.12	68.49	87.50	24.83	31.61
SqueezeSegV2	65.29	62.11	84.84	67.22	55.11	77.98	64.63	81.88	28.54	41.28
RangeNet$_{21}$	73.42	65.83	86.70	79.38	66.09	80.93	80.55	88.10	39.79	47.15
RangeNet$_{53}$	73.59	72.24	85.64	79.58	59.85	81.13	91.63	84.85	33.76	50.29
SalsaNext	80.51	62.53	86.81	81.63	85.90	88.94	72.06	86.08	80.14	55.80
FIDNet	76.99	74.25	87.81	84.49	68.67	83.88	84.12	81.92	50.77	58.80
CENet	81.29	68.27	91.67	85.76	84.27	89.18	72.53	85.37	73.29	62.55
PolarNet	74.98	66.60	87.21	84.96	71.81	93.00	44.34	84.17	67.80	58.17
KPConv	82.90	87.60	92.81	87.10	41.34	92.25	85.86	89.50	86.71	62.17
PIDS$_{1.2\times}$	77.94	75.73	86.13	77.25	36.32	86.85	89.64	88.24	83.35	63.25
PIDS$_{2.0\times}$	78.42	79.30	86.71	79.18	34.84	88.23	88.94	86.06	84.07	64.55
WaffleIron	72.18	68.93	88.66	74.65	50.00	89.76	34.04	88.66	82.71	**66.04**
MinkUNet$_{18}$	81.90	89.02	86.03	84.89	52.45	89.74	92.96	86.73	73.37	62.76
MinkUNet$_{34}$	80.22	83.94	85.09	78.66	52.99	89.92	91.53	86.05	73.61	63.78
Cylinder3D$_{SPC}$	80.08	58.50	90.59	74.01	82.70	90.89	88.27	82.80	72.88	63.42
Cylinder3D$_{TSC}$	**83.90**	60.84	87.54	74.41	**96.13**	93.13	87.85	89.97	81.34	61.00
SPVCNN$_{18}$	82.15	88.55	86.41	82.31	55.27	90.72	93.00	87.40	73.56	62.47
SPVCNN$_{34}$	82.01	**89.42**	84.91	82.81	54.40	89.78	**93.32**	86.95	74.45	63.22
RPVNet	73.86	74.73	83.98	80.20	74.18	83.94	35.51	85.95	72.42	63.75
CPGNet	81.05	61.45	93.32	83.35	96.02	**98.03**	30.08	**92.23**	**93.97**	61.50
2DPASS	77.50	62.62	**93.92**	75.11	89.46	90.98	44.05	86.43	77.40	64.61
GFNet	77.92	66.73	89.79	**90.02**	93.00	90.40	27.21	87.67	78.54	63.00

Table 6: Results for the **Intersection-over-Union (IoU)** of each method on *SemanticKITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	mIoU ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
SqueezeSeg	164.87	66.81	31.61	18.85	27.30	22.70	17.93	25.01	21.65	27.66	7.85
SqueezeSegV2	152.45	65.29	41.28	25.64	35.02	27.75	22.75	32.19	26.68	33.80	11.78
RangeNet$_{21}$	136.33	73.42	47.15	31.04	40.88	37.43	31.16	38.16	37.98	41.54	18.76
RangeNet$_{53}$	130.66	73.59	50.29	36.33	43.07	40.02	30.10	40.80	46.08	42.67	16.98
SalsaNext	116.14	80.51	55.80	34.89	48.44	45.55	47.93	49.63	40.21	48.03	44.72
FIDNet	113.81	76.99	58.80	43.66	51.63	49.68	40.38	49.32	49.46	48.17	29.85
CENet	103.41	81.29	62.55	42.70	57.34	53.64	52.71	55.78	45.37	53.40	45.84
PolarNet	118.56	74.98	58.17	38.74	50.73	49.42	41.77	54.10	25.79	48.96	39.44
KPConv	99.54	82.90	62.17	54.46	57.70	54.15	25.70	57.35	53.38	55.64	53.91
PIDS$_{1.2\times}$	104.13	77.94	63.25	47.90	54.48	48.86	22.97	54.93	56.70	55.81	52.72
PIDS$_{2.0\times}$	101.20	78.42	64.55	51.19	55.97	51.11	22.49	56.95	57.41	55.55	54.27
WaffleIron	109.54	72.18	**66.04**	45.52	58.55	49.30	33.02	59.28	22.48	**58.55**	54.62
MinkUNet$_{18}$	100.00	81.90	62.76	55.87	53.99	53.28	32.92	56.32	58.34	54.43	46.05
MinkUNet$_{34}$	100.61	80.22	63.78	53.54	54.27	50.17	33.80	57.35	58.38	54.88	46.95
Cylinder3D$_{\text{SPC}}$	103.25	80.08	63.42	37.10	57.45	46.94	52.45	57.64	55.98	52.51	46.22
Cylinder3D$_{\text{TSC}}$	103.13	**83.90**	61.00	37.11	53.40	45.39	58.64	56.81	53.59	54.88	49.62
SPVCNN$_{18}$	100.30	82.15	62.47	55.32	53.98	51.42	34.53	56.67	58.10	54.60	45.95
SPVCNN$_{34}$	**99.16**	82.01	63.22	**56.53**	53.68	52.35	34.39	56.76	**59.00**	54.97	47.07
RPVNet	111.74	73.86	63.75	47.64	53.54	51.13	47.29	53.51	22.64	54.79	46.17
CPGNet	107.34	81.05	61.50	37.79	57.39	51.26	**59.05**	**60.29**	18.50	56.72	**57.79**
2DPASS	106.14	77.50	64.61	40.46	**60.68**	48.53	57.80	58.78	28.46	55.84	50.01
GFNet	108.68	77.92	63.00	42.04	56.57	**56.71**	58.59	56.95	17.14	55.23	49.48

Table 7: Results for the **Resilience Rate (RR)** of each method on *nuScenes-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mIoU ↑
FIDNet	73.33	90.78	95.29	82.61	68.51	67.44	80.48	68.31	33.20	71.38
CENet	76.04	**91.44**	95.35	84.12	79.57	68.19	83.09	72.75	33.82	73.28
PolarNet	76.34	81.59	97.95	90.82	62.49	86.75	57.12	75.16	58.86	71.37
WaffleIron	72.78	73.71	97.19	65.19	78.16	85.70	43.54	80.86	57.85	76.07
MinkUNet$_{18}$	74.44	70.80	97.56	53.26	96.87	90.47	35.08	84.25	67.25	75.76
MinkUNet$_{34}$	75.08	74.01	97.44	48.76	**97.84**	**91.16**	38.13	84.47	**68.87**	76.90
Cylinder3D$_{\text{SPC}}$	72.94	78.59	95.46	76.26	55.33	84.64	58.36	79.45	55.46	76.15
Cylinder3D$_{\text{TSC}}$	78.08	83.52	96.57	79.41	76.18	87.23	61.68	81.55	58.51	73.54
SPVCNN$_{18}$	74.70	79.31	97.39	55.22	78.44	87.85	49.50	83.72	66.14	74.40
SPVCNN$_{34}$	75.10	72.95	96.70	54.79	97.47	90.04	36.71	**84.84**	67.35	76.57
2DPASS	75.24	82.78	**98.51**	69.89	79.62	87.06	44.11	81.10	58.82	**77.92**
GFNet	**83.31**	90.62	98.35	**93.54**	77.39	83.96	**86.96**	80.56	55.09	76.79

Table 8: Results for the **Intersection-over-Union (IoU)** of each method on *nuScenes-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	mIoU ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
FIDNet	122.42	73.33	71.38	64.80	68.02	58.97	48.90	48.14	57.45	48.76	23.70
CENet	112.79	76.04	73.28	67.01	69.87	61.64	58.31	49.97	60.89	53.31	24.78
PolarNet	115.09	76.34	71.37	58.23	69.91	64.82	44.60	61.91	40.77	53.64	42.01
WaffleIron	106.73	72.78	76.07	56.07	73.93	49.59	59.46	65.19	33.12	61.51	44.01
MinkUNet$_{18}$	100.00	74.44	75.76	53.64	73.91	40.35	73.39	68.54	26.58	63.83	50.95
MinkUNet$_{34}$	96.37	75.08	76.90	56.91	74.93	37.50	**75.24**	**70.10**	29.32	**64.96**	**52.96**
Cylinder3D$_{\text{SPC}}$	111.84	72.94	76.15	59.85	72.69	58.07	42.13	64.45	44.44	60.50	42.23
Cylinder3D$_{\text{TSC}}$	105.56	78.08	73.54	61.42	71.02	58.40	56.02	64.15	45.36	59.97	43.03
SPVCNN$_{18}$	106.65	74.70	74.40	59.01	72.46	41.08	58.36	65.36	36.83	62.29	49.21
SPVCNN$_{34}$	97.45	75.10	76.57	55.86	74.04	41.95	74.63	68.94	28.11	**64.96**	51.57
2DPASS	98.56	75.24	**77.92**	64.50	**76.76**	54.46	62.04	67.84	34.37	63.19	45.83
GFNet	**92.55**	**83.31**	76.79	**69.59**	75.52	**71.83**	59.43	64.47	**66.78**	61.86	42.30

Table 9: Results of the **Corruption Error (CE)** of each method on *WOD-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mIoU ↑
MinkUNet$_{18}^{\dagger}$	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	69.06
MinkUNet$_{34}$	**96.21**	**96.00**	94.90	99.53	**96.20**	**95.43**	**96.79**	**96.75**	**94.08**	**70.15**
Cylinder3D$_{\text{TSC}}$	106.02	111.81	104.08	**98.39**	110.30	105.77	106.87	108.24	102.69	65.93
SPVCNN$_{18}$	103.60	105.63	104.79	99.17	105.41	104.85	99.74	104.28	104.91	67.35
SPVCNN$_{34}$	98.72	99.67	**96.36**	100.43	100.00	98.55	101.93	97.87	94.97	69.01

Table 10: Results of the **Resilience Rate (RR)** of each method on *WOC-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mIoU ↑
MinkUNet$_{18}$	91.22	97.00	88.31	83.62	99.80	92.89	94.66	91.75	81.73	69.06
MinkUNet$_{34}$	91.80	**97.38**	89.78	82.61	**99.93**	93.78	94.77	92.02	84.13	**70.15**
Cylinder3D$_{\text{TSC}}$	**92.39**	95.69	90.10	**88.62**	99.68	**94.16**	95.54	91.52	83.83	65.93
SPVCNN$_{18}$	91.60	96.70	87.78	86.27	99.84	92.67	**97.19**	91.74	80.62	67.35
SPVCNN$_{34}$	92.04	97.23	**90.44**	83.42	99.87	93.71	93.75	**92.94**	**84.96**	69.01

Table 11: Results of the **Intersection-over-Union (IoU)** of each method on *WOD-C (Seg3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	mIoU ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
MinkUNet$_{18}$	100.00	91.22	69.06	66.99	60.99	57.75	68.92	64.15	65.37	63.36	56.44
MinkUNet$_{34}$	**96.21**	91.80	**70.15**	**68.31**	62.98	57.95	**70.10**	**65.79**	**66.48**	**64.55**	**59.02**
Cylinder3D$_{\text{TSC}}$	106.02	**92.39**	65.93	63.09	59.40	58.43	65.72	62.08	62.99	60.34	55.27
SPVCNN$_{18}$	103.60	91.60	67.35	65.13	59.12	58.10	67.24	62.41	65.46	61.79	54.30
SPVCNN$_{34}$	98.72	92.04	69.01	67.10	**62.41**	**57.57**	68.92	64.67	64.70	64.14	58.63

Table 12: Results for the **Resilience Rate (RR)** of each method on *KITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mAP ↑
PointPillars	74.94	68.52	100.01	53.63	70.60	78.32	89.97	82.22	56.22	66.70
SECOND	82.94	77.73	100.03	**80.19**	71.82	79.05	98.10	86.51	70.08	68.49
PointRCNN	**83.46**	**80.15**	**102.22**	71.45	73.33	**80.90**	93.51	**88.27**	**77.90**	70.26
Part-A2$_{\text{Free}}$	81.87	76.11	100.01	76.26	72.30	77.95	**99.10**	86.08	67.15	**76.28**
Part-A2$_{\text{Anchor}}$	80.67	76.49	99.99	69.37	74.40	76.21	96.95	85.55	66.44	73.98
PV-RCNN	81.73	76.51	100.73	72.03	**75.23**	78.61	97.28	87.06	66.35	72.36
CenterPoint	79.73	77.29	100.01	70.68	69.78	72.61	96.07	85.74	65.68	68.70

Table 13: Results for the **Average Precision (AP)** of each method on *KITTI-C*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	mAP ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
PointPillars	110.67	74.94	66.70	45.70	66.71	35.77	47.09	52.24	60.01	54.84	37.50
SECOND	95.93	82.94	68.49	53.24	68.51	54.92	49.19	54.14	67.19	59.25	48.00
PointRCNN	91.88	**83.46**	70.26	56.31	71.82	50.20	51.52	56.84	65.70	62.02	**54.73**
Part-A2$_{\text{Free}}$	**82.22**	81.87	**76.28**	**58.06**	**76.29**	**58.17**	**55.15**	**59.46**	**75.59**	**65.66**	51.22
Part-A2$_{\text{Anchor}}$	88.62	80.67	73.98	56.59	73.97	51.32	55.04	56.38	71.72	63.29	49.15
PV-RCNN	90.04	81.73	72.36	55.36	72.89	52.12	54.44	56.88	70.39	63.00	48.01
CenterPoint	100.00	79.73	68.70	53.10	68.71	48.56	47.94	49.88	66.00	58.90	45.12

Table 14: Results for the **Resilience Rate (RR)** of each method on *nuScenes-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	NDS ↑
PointPillars-MH	**77.24**	76.53	99.05	68.06	87.79	**77.57**	79.88	71.31	**57.70**	43.33
SECOND-MH	76.96	**79.38**	**99.42**	**70.86**	86.32	74.45	**84.19**	71.28	49.76	47.87
CenterPoint-PP	76.68	76.13	98.74	67.91	**90.87**	76.45	76.58	70.73	56.06	45.99
CenterPoint-LR	72.49	73.19	95.21	65.99	81.54	69.33	76.65	**71.40**	46.58	49.72
CenterPoint-HR	75.26	78.61	98.93	69.03	85.89	71.97	81.45	69.75	46.47	**50.31**

Table 15: Results for the **nuScenes Detection Score (NDS)** of each method on *nuScenes-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	NDS ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
PointPillars-MH	102.90	**77.24**	43.33	33.16	42.92	29.49	38.04	33.61	34.61	30.90	25.00
SECOND-MH	97.50	76.96	47.87	38.00	47.59	33.92	41.32	35.64	40.30	34.12	23.82
CenterPoint-PP	100.00	76.68	45.99	35.01	45.41	31.23	41.79	35.16	35.22	32.53	**25.78**
CenterPoint-LR	98.74	72.49	49.72	36.39	47.34	32.81	40.54	34.47	38.11	**35.50**	23.16
CenterPoint-HR	**95.80**	75.26	**50.31**	**39.55**	**49.77**	**34.73**	**43.21**	**36.21**	**40.98**	35.09	23.38

Table 16: Results for the **Corruption Error (CE)** of each method on *WOD-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mAPH ↑
CenterPoint[†]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	63.59
SECOND	121.43	117.86	126.51	127.51	113.37	121.25	127.82	123.66	113.48	53.37
PointPillars	127.53	120.76	135.23	129.65	115.23	122.99	151.71	131.64	113.05	50.17
PV-RCNN	104.90	110.08	104.22	95.68	101.33	110.70	101.84	106.00	109.37	61.27
PV-RCNN++	**91.60**	**95.71**	**88.32**	**90.05**	**93.24**	**92.50**	**88.94**	**90.81**	**93.23**	**67.45**

Table 17: Results for the **Resilience Rate (RR)** of each method on *WOD-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mRR ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor	mAPH ↑
PointPillars	81.23	62.27	99.16	92.31	69.62	**87.56**	79.33	86.53	**73.09**	50.17
SECOND	81.12	61.63	99.29	88.44	67.42	83.79	92.34	87.76	68.26	53.37
PV-RCNN	82.43	60.91	**100.00**	**98.55**	69.82	80.84	**97.26**	88.84	63.21	61.27
CenterPoint	83.30	**67.72**	98.82	92.14	68.45	85.56	94.86	89.65	69.16	63.59
PV-RCNN++	**84.14**	67.46	99.60	92.97	**70.20**	85.74	95.94	**90.38**	70.82	**67.45**

Table 18: Results of the **Average Precision (APH)** of each method on *WOD-C (Det3D)*. **Bold**: Best in column. Underline: Second best in column. Dark : Best in row. Red : Worst in row.

Method	mCE ↓	mRR ↑	mAPH ↑	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
PointPillars	127.53	81.23	50.17	31.24	49.75	46.07	34.93	43.93	39.80	43.41	36.67
SECOND	121.43	81.12	53.37	32.89	52.99	47.20	35.98	44.72	49.28	46.84	36.43
PV-RCNN	104.90	82.43	61.27	37.32	61.27	60.38	42.78	49.53	59.59	54.43	38.73
CenterPoint	100.00	83.30	63.59	43.06	62.84	58.59	43.53	54.41	60.32	57.01	43.98
PV-RCNN++	**91.60**	**84.14**	**67.45**	**45.50**	**67.18**	**62.71**	**47.35**	**57.83**	**64.71**	**60.96**	**47.77**

REFERENCES

Fatima Albreiki, Sultan Abughazal, Jean Lahoud, Rao Anwer, Hisham Cholakkal, and Fahad Khan. On the robustness of 3d object detectors. *arXiv preprint arXiv:2207.10205*, 2022.

Antonio Alliegro, Francesco Cappio Borlino, and Tatiana Tommasi. Towards open set 3d learning: A benchmark on object point clouds. *arXiv preprint arXiv:2207.11554*, 2022.

Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pp. 9297–9307, 2019.

Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11682–11692, 2020.

Lara Brinon-Arranz, Tiana Rakotovao, Thierry Creuzet, Cem Karaoguz, and Oussama El-Hamzaoui. A methodology for analyzing the impact of crosstalk on lidar measurements. *IEEE Sensors*, 8:1–4, 2021.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 176–194. IEEE, 2021.

Prithvijit Chattopadhyay, Judy Hoffman, Roozbeh Mottaghi, and Aniruddha Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. In *IEEE/CVF International Conference on Computer Vision*, pp. 15691–15700, 2021.

Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28, 2015.

Huixian Cheng, Xianfeng Han, and Guoqiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *IEEE International Conference on Multimedia and Expo*, 2022.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pp. 207–222, 2020.

Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, pp. 1201–1209, 2021.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7:3795–3802, 2022.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2020.

Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In *IEEE/CVF International Conference on Computer Vision*, pp. 15283–15292, 2021.

Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16364–16374, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12605–12614, 2020.

Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8828–8838, 2020.

Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8963–18974, 2022.

Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. *arXiv preprint arXiv:2303.05367*, 2023a.

Lingdong Kong, Niamul Quader, and Venice Erin Liong. Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. In *IEEE International Conference on Robotics and Automation*, 2023b.

Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023c.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

Shuangzhi Li, Zhijie Wang, Felix Juefei-Xu, Qing Guo, Xingyu Li, and Lei Ma. Common corruption robustness of point cloud detectors: Benchmark and enhancement. *arXiv preprint arXiv:2210.05896*, 2021a.

Xiaoyan Li, Gang Zhang, Hongyu Pan, and Zhenhua Wang. Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. In *IEEE International Conference on Robotics and Automation*, 2022.

Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7546–7555, 2021b.

Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv preprint arXiv:2012.04934*, 2020.

Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, 2019.

Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pp. 3164–3173, 2021.

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.

Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4213–4220, 2019.

Won Park, Nan Liu, Qi Alfred Chen, and Z. Morley Mao. Sensor adversarial traits: Analyzing robustness of 3d object detection sensor fusion models. In *IEEE International Conference on Image Processing*, pp. 484–488, 2019.

Tyson Govan Phillips, Nicky Guenther, and Peter Ross McAree. When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates. *Journal of Field Robotics*, 34: 985–1009, 2017.

Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. *arxiv:2301.10100*, 2023.

Haibo Qiu, Baosheng Yu, and Dacheng Tao. Gfnet: Geometric flow network for 3d point cloud semantic segmentation. *Transactions on Machine Learning Research*, 2022.

Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, 2022.

Giulio Rossolini, Federico Nesti, Gianluca D'Amico, Saasha Nair, Alessandro Biondi, and Giorgio Buttazzo. On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving. *arXiv preprint arXiv:2201.01850*, 2022.

Alvari Seppänen, Risto Ojala, and Kari Tammi. Adverse weather denoising from adjacent point clouds. *IEEE Robotics and Automation Letters*, 8:456–463, 2022.

Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, pp. 2743–2752, 2021.

Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: High-performance pillar-based 3d object detection. *arXiv preprint arXiv:2205.07403*, 2022a.

Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020a.

Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2647–2664, 2020b.

Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 2022b.

Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, pp. 1–21, 2022c.

Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1711–1719, 2020.

Jungil Shin, Hyunsuk Park, and Taejung Kim. Characteristics of laser backscattering intensity to detect frozen and wet surfaces on roads. *Journal of Sensors*, 2019.

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

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, and Benjamin Caine. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pp. 685–702, 2020.

Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. Torchsparse: Efficient point cloud inference engine. In *Conference on Machine Learning and Systems*, 2022.

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 *IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420, 2019.

James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13716–13725, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11855–11864, 2021.

Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: bridging the beam-induced domain gap for 3d object detection. In *European Conference on Computer Vision*, pp. 179–195, 2022.

Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *IEEE International Conference on Robotics and Automation*, pp. 1887–1893, 2018.

Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *IEEE International Conference on Robotics and Automation*, pp. 4376–4382. IEEE, 2019.

Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pp. 2795–2803, 2022.

Shaoyuan Xie, Zichao Li, Zeyu Wang, and Cihang Xie. On the adversarial robustness of camera-based 3d object detection. *arXiv preprint arXiv:2301.10766*, 2023.

Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *IEEE/CVF International Conference on Computer Vision*, pp. 16024–16033, 2021.

Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, 2022.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *IEEE/CVF International Conference on Computer Vision*, pp. 1951–1960, 2019.

Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11040–11048, 2020.

Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Efficient point cloud segmentation with geometry-aware sparse networks. In *European Conference on Computer Vision*, 2022.

Chenyu Yi, Siyuan Yang, Haoliang Li, Yap peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *Advances in Neural Information Processing Systems*, 2021.

Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11784–11793, 2021.

Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Zhongwei Wu, Zhongyu Xia, Tingting Liang, Haiyang Sun, Jiong Deng, Dayang Hao, Yongtao Wang, Xiaodan Liang, and Bing Wang. Benchmarking the robustness of lidar-camera fusion for 3d object detection. *arXiv preprint arXiv:2205.14951*, 2022.

Tunhou Zhang, Mingyuan Ma, Feng Yan, Hai Li, and Yiran Chen. Pids: Joint point interaction-dimension search for 3d point cloud. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1298–1307, 2023.

Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. *arXiv preprint arXiv:2003.14032*, 2020.

Yiming Zhao, Lin Bai, and Xinming Huang. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4453–4458. IEEE, 2021.

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *European Conference on Computer Vision*, 2022.

Lifa Zhu, Changwei Lin, Cheng Zheng, and Ninghua Yang. Point-voxel adaptive feature abstraction for robust point cloud classification. *arXiv preprint arXiv:2210.15514*, 2022.

Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9939–9948, 2021.