Benchmarking the Robustness of LiDAR-Camera Fusion for 3D Object Detection

Kaicheng Yu[2*] Tang Tao[1*†] Hongwei Xie[2*] Zhiwei Lin[3] Tingting Liang[3] Bing Wang[2]
Peng Chen[2] Dayang Hao[2] Yongtao Wang[3] Xiaodan Liang[1§]
[1] Shenzhen Campus, Sun Yat-sen University, China
[2] Autonomous Driving Lab, Alibaba Group, China
[3] Wangxuan Institute of Computer Technology, Peking University, China

{kaicheng.yu.yt, trent.tangtao, hongwei.xie.90, xdliang328, blucewang6}@gmail.com

Abstract

To achieve autonomous driving, developing 3D detection fusion methods, which aim to fuse the camera and LiDAR information, has draw great research interest in recent years. As a common practice, people rely on large-scale datasets to fairly compare the performance of different methods. While these datasets have been carefully cleaned to ideally minimize any potential noise, we observe that they cannot truly reflect the data seen on a real autonomous vehicle, whose data tends to be noisy due to various reasons. This hinders the ability to simply estimate the robust performance under realistic noisy settings. To this end, we collect a series of real-world cases with noisy data distribution, and systematically formulate a robustness benchmark toolkit. It can simulate these cases on any clean dataset, which has the camera and LiDAR input modality. We showcase the effectiveness of our toolkit by establishing two novel robustness benchmarks on widely-adopted datasets, nuScenes and Waymo, then holistically evaluate the state-of-the-art fusion methods. We discover that: i) most fusion methods, when solely developed on these data, tend to fail inevitably when there is a disruption to the LiDAR input; ii) the improvement of the camera input is significantly inferior to the LiDAR one. We publish the robust fusion dataset, benchmark, detailed documents and instructions on https://anonymous-benchmark.github.io/robust-benchmark-website.

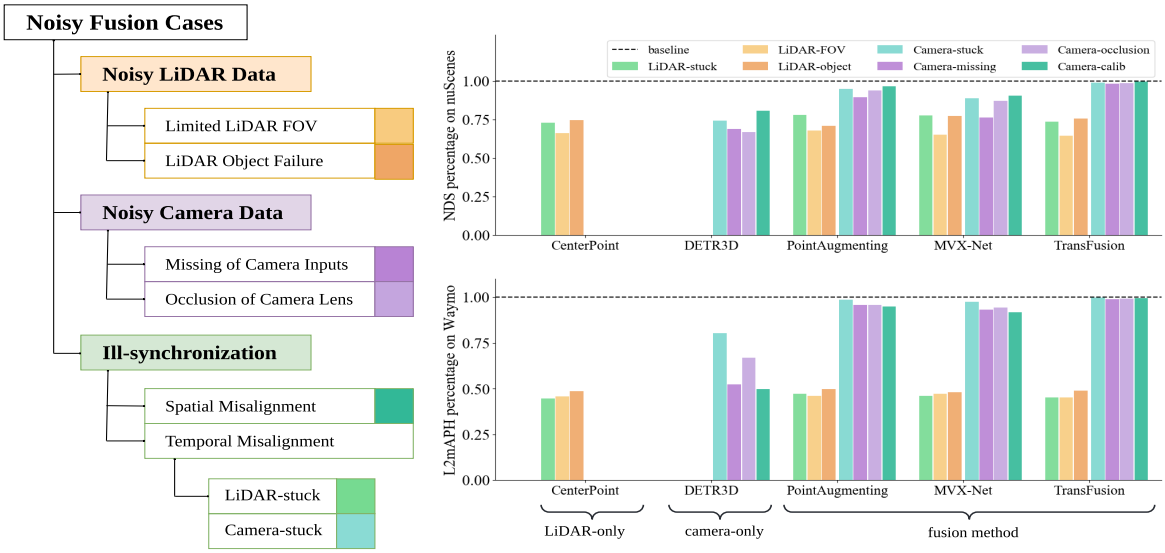
1. Introduction

3D detection has received extensive attention as one of the fundamental tasks in autonomous driving scenarios [11, 16, 20, 33, 38, 40, 41, 55, 57, 58]. Recently, fusing the two common modalities, input from the camera and LiDAR sensors, has become a de-facto standard in the 3D detection domain as each modality has complementary information of the other [5, 12, 42, 49, 50, 56, 60]. Similar to other literature in the computer vision community, a common approach to showcase the effectiveness of a proposed fusion method is to validate it on the existing benchmark datasets [4, 46], which are usually collected from explicitly designed, expensive data collection vehicles to minimize any potential error from the hardware setup.

However, we discover that the data distribution of these popular datasets can be drastically different from the realistic driving scenarios due to various reasons: i) there can be uncontrollable external reasons, such as splatted dirt or BIOS malfunctions of the on-device computer, that temporarily disable the input of certain sensors; ii) the inputs can be difficult to synchronize due to external and internal reasons, such as the spatial misalignment due to the severe vibration when driving on the bumpy road or temporal misalignment due to clock synchronization module malfunctions. Therefore, the methods that are only evaluated on the clean datasets might not be trustworthy in realistic scenarios, and hinders actual deployment on the real autonomous driving vehicles.

To this end, we close this research gap by proposing a novel toolkit that transforms any clean benchmark dataset, which has the camera and LiDAR input modality, into a robustness benchmark to simulate realistic scenarios. We first conduct a systematic overview of potential sensor noisy cases, both for the camera and LiDAR, based on realistic driving data. As in Fig. 1 (a), we identify seven unique cases under three categories, two for noisy LiDAR cases, two for noisy camera cases, and three for ill-synchronization cases. We then carefully study each case and construct a code

*Equal Contribution.
†Work done during an internship at DAMO Academy, Alibaba Group.
§Corresponding Author.



(a) Overview of noisy cases (b) Performance degradation of existing 3D detectors

Figure 1. **Benchmarking the robustness of state-of-the-art 3D detection methods. (a)** We provide an overview of all noisy fusion cases. **(b)** We report the performance ratio (robust/clean) of current methods on two robustness datasets, Waymo-R and nuScenes-R, which are generated by our toolkit.

toolkit to transform the clean data into associated realistic data distribution of each case.

To verify the effectiveness of our approach, we apply our toolkit to two large-scale popular benchmark datasets for autonomous driving, nuScenes and Waymo. Note that though these noisy cases rarely appear in realistic scenarios, we convert all data of the given dataset to fully evaluate the robustness of a given method in an extreme manner. And we only investigate one failure case at a time, and do not create a robust benchmark that has multiple malfunctions at the same time. We then collect two single modalities and three fusion state-of-the-art methods and benchmark them on the generated benchmarks. In Fig. 1 (b), we observe several surprising findings: i) state-of-the-art fusion methods tend to fail inevitably when the LiDAR sensor encounters failures due to their fusion mechanism heavily relies on the LiDAR input; ii) fusing the camera input only brings a marginal improvement, suggesting either the current methods fail to sufficiently leverage the information from the camera or the camera information did not carry the complementary information as intuited.

In summary, our main contributions are as follows:

- We systematically study the noisy sensor data in the realistic driving scenarios and propose a novel toolkit that can transform any autonomous driving benchmark datasets, that contain camera and LiDAR input, into a robustness benchmark;

- To the best of our knowledge, we are the first to benchmark existing methods under the noisy settings and find that current fusion methods have a fundamental flaw and can fail inevitably when there is a LiDAR malfunction.

We hope our work can shed light on developing robust fusion method that can be truly deployed to the autonomous vehicles.

2. Related Work

Here, we provide a literature review of current fusion methods in the 3D detection and the robustness evaluation.

Fusion methods in 3D detection. LiDAR and camera are two types of complementary sensors for 3D object detection in autonomous driving. In essence, the LiDAR sensor provides an accurate depth and shape information of the surrounding world in form of sparse point clouds [19, 35–37, 40, 41, 53, 58, 59, 67], while the camera sensor provides an RGB-based image that contains rich semantic and texture information [11, 15, 28, 29, 34, 38, 38, 39, 51, 52, 55, 64, 66]. Recently, fusing these modalities to leverage the complementary information becomes a de-facto standard in the 3D detection domain. Based on the fusion mechanism location, these methods can be divided into three categories, early, deep, and late fusion schemes. Early fusion methods mainly concatenate the image features to the original LiDAR point to enhance the representation power. Specifically, these methods rely on the LiDAR-to-world and camera-to-world calibration matrix to project a LiDAR point on the image plane, where it serves as a query of image features [12, 42, 49, 50, 56, 60]. Deep fusion methods extract deep features

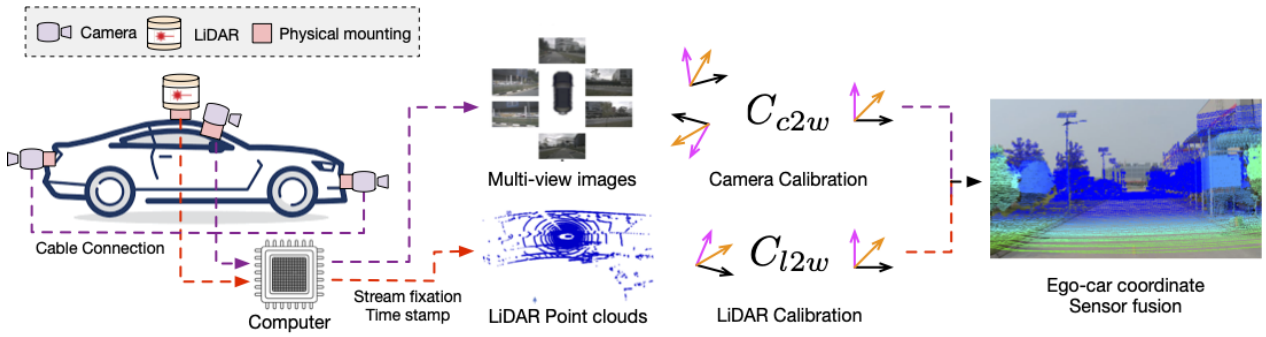


Figure 2. **Autonomous driving perception system with camera and LiDAR sensors.**

from some pre-trained neural networks for both modalities under a unified space [1,5,6,14,17,18,21,22,24,27,31,61], where a popular choice of such space is the bird's eye view (BEV) [1, 24, 27, 61]. While both early and deep fusion mechanisms usually occur within a neural network pipeline, the late fusion scheme usually contains two independent perception models to generate 3D bounding box predictions for both modalities, then fuse these predictions using post-processing techniques [5, 32]. One benefit of these works is their robustness against single modality input failure. However, it is difficult to jointly optimize this line of methods due to the post-processing technique being usually non-differentiable. In addition, this pipeline has a potential higher deployment cost as it has three independent modules to be maintained.

Robustness of LiDAR-Camera fusion. Though there are some works [2,9,30,44,45,47,48,63] explore the robustness of 3D dectors from different perspectives, e.g., the challenging weather. In the domain of autonomous driving, there lacks such a benchmark dataset for robustness analysis of the fusion models to the best of our knowledge. There only a few preliminary attempts to investigate this robustness issue [1, 18, 26]. TransFusion [1] evaluates the robustness of different fusion strategies under three scenarios: splitting validation set into daytime and nighttime, randomly dropping images for each frame, misaligning LiDAR and camera calibration by randomly adding a translation offset to the transformation matrix from camera to LiDAR sensor. Instead, we also add a rotation offset to the transformation matrix. Overall, TransFusion mainly explores the robustness against camera inputs, and ignores the noisy LiDAR and temporal misalignment cases. DeepFusion [18] examines the model robustness by adding noise to LiDAR reflections and camera pixels. Though the noise settings of DeepFusion are straightforward and brief, the noisy cases almost never appear in real scenes. Therefore, previous methods don't provide a more thorough study useful for fusion methods. By contrast, we systematically review the autonomous driving perception system and identify three categories, in a total of seven cases of robustness scenarios, and propose a toolkit that can transform an existing dataset into a robustness benchmark. We hope our work can help future research to benchmark the robustness of their methods fairly, and give researchers more insights about designing a more robust fusion framework. An ideal fusion framework should work better than a single modality, and will not be worse than the single modality model while the other modality fails. We hope the deep fusion method is better than late fusion methods that use complex post-processing techniques.

3. Robust Fusion Benchmark

In this section, we first provide a systematic overview of current autonomous driving vehicle systems with LiDAR and camera sensors to show why the data distribution of each case in clean datasets can differ from real-world scenarios. These noisy data cases can be categorized into three broad classes: noisy LiDAR, noisy camera, and ill-synchronization cases. Then, we present a toolkit that can transform current clean datasets into realistic scenarios.

3.1. An overview of modern autonomous driving vehicle system

In Fig. 2, we visualize a common design of the autonomous driving perception system, whose main components include the camera and LiDAR sensors, and an on-device computer. Specifically, the camera and LiDAR sensors are physically mounted on certain fixed locations of the vehicle and are connected to the computer via certain cables with communication protocols. In essence, the computer can access the data stream from the sensors and capture the data into a point cloud or image with a certain timestamp. As the raw data are in the sensor coordinate system, sensor calibration plays a major role in performing efficient coordinate transformation such that the perception system can recognize objects with respect to the ego-car coordinate system. Based on our experience, each step of the aforementioned system can encounter certain failures or disruptions, and yield noisy data that are drastically different from the normal clean data. We identify three categories of cases

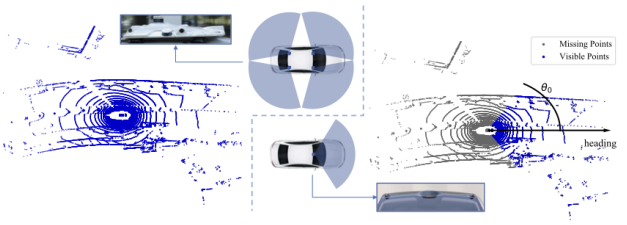


Figure 3. **Limited LiDAR field-of-view**. (Left) We visualize the complete LiDAR point clouds that come from the data collection vehicle which has a complete sensor rack. (Right) In realistic scenarios, the LiDAR is installed in a front-facing manner, yielding a limited FOV. Better view in color.

and briefly discuss the potential reasons and consequences in Tab. 5 of the appendix, and provide a detailed case analysis later.

3.2. Case analysis

In this section, we analyze the collected real-world noisy data cases of autonomous driving in detail.

3.2.1 Noisy LiDAR Data

We identify two common cases that can cause noisy LiDAR data in practice.

Limited LiDAR field-of-view (FOV). While most companies collect the LiDAR data whose field-of-view is 360 degrees, certain LiDAR data might not always be available for various reasons. For example, a certain type of vehicle only installs a front-facing semi-solid LiDAR sensor on the roof of the car instead of using a full rack, as shown in the right part of Fig. 3. Without loss of generality, we first convert the coordinate of LiDAR points from Euclidean (in x, y, z) to polar coordinate system (r, θ, z). We then can simulate such limited FOV by keeping the points that satisfy $\theta \in (-\theta_0, \theta_0)$. In practice, we set θ_0 to 0, 60, and 90 degrees to simulate commonly seen scenarios, which have realistic meanings. We clarify that, the two settings of limited FOV have different causes: i.complete failure (no lidar data) is due to the temporary hardware malfunctions; ii.reduced FOV such as [-60, 60] is due to the difference between data collection vehicle and final production ones.

LiDAR object failure. One common scenario that people tend to overlook is that the LiDAR can be blind to objects under certain constraints. We show one example from the realistic data captured on a commercialized autonomous driving system in Fig. 4. We observe that the LiDAR point clouds are drastically different from two side-by-side cars, where the black car has nearly zero points while the white car has a normal point distribution. We dub this phenomenon LiDAR object failure, which is usually caused by low reflection rate of objects due to object texture, inappropriate reflection angle or water film. Without loss of generality, we simulate such scenarios by randomly dropping the points within a bounding box with a probability of 0.5. Note that we do not alter the camera input because the purpose is to benchmark the single modality input data.

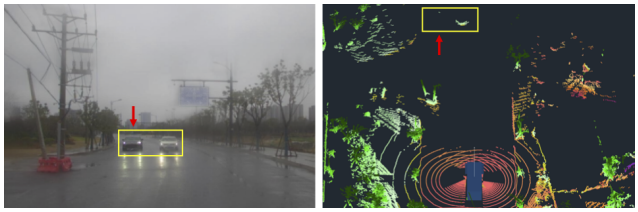


Figure 4. **LiDAR object failure**. On rainy days, the reflection rate of some common objects (e.g., the black car) is below the threshold of LiDAR hence causing the issue of object failure.

3.2.2 Noisy Camera Data

Different from the LiDAR module, the camera module is usually installed on much lower locations of the autonomous driving vehicles to cover the blind region of the LiDAR sensor. Such blind region is due to the fact that the LiDAR is usually installed on the roof of the car to maximize the visualization distance, while it cannot see the near-car region due to blockage. As such, the camera can be easily affected by the surrounding environment such as temporary generic object coverage or lens occlusion of dirt. We discuss these two scenarios in detail.

Missing camera inputs. As the camera module is usually much smaller (within one centimeter), the most common covering scenario is covering the whole camera sensor. Thus, we drop the entire camera input to simulate such covering scenarios and the situation when camera sensor is damaged. In practice, we design two finer cases to perform a robust benchmark, dropping one camera at a time as it's common that one camera is covered or damaged, and dropping all other cameras except the front one as some patrol robots or logistics robots only have one camera on the front.

Figure 5. **Visualization of camera occlusion**. We display the original images from different scenarios in the nuScenes dataset (Left), the randomly sampled dirt occlusion masks (Middle), and the final composed images that simulating the occlusion (Right).

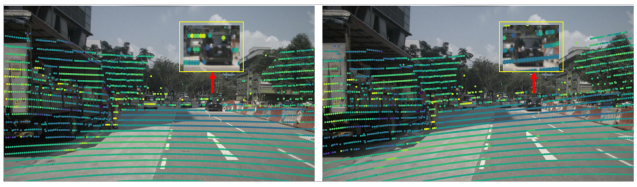


Figure 6. **Visualization of spatial misalignment**. We provide one visual example to showcase the spatial misalignment caused by noisy extrinsic parameters.

Camera lens occlusion. Another commonly seen camera covering problem is lens occlusion caused by non-transparent liquid or dirt. Some works also which introduces a soiling dataset [48], and [47] which uses these data to train a GAN model to generate realistic lens occlusions together with corresponding annotations. Instead, to simulate the occlusion of camera lenses in real scenes, we spray mud dots on a transparent film and cover the dirty film on the camera lens to take photos on a white background. Then, we adopt an image matting algorithm to cut out the background part in the images and separate the masks of mud spots. Finally, the separated masks are pasted on the images of clean datasets to simulate the occlusion of their camera lenses, as illustrated in the Fig. 5. In addition, we spray mud dots of different sizes and randomly move and rotate the film to create masks with different occlusion areas and occlusion ranges to enhance the diversity of the mask.

3.3. Ill-synchronization

As illustrated in Fig. 2, the data stream is firstly fixed into a data frame with a given timestamp when passed into the on-device computer, then one needs to perform the coordinate transform via the camera-to-world and LiDAR-to-world matrix that is obtained by the calibration process. However, this leads to two potential ill-synchronization issues, spatial misalignment due to the external reasons of calibration matrix and temporal misalignment for both LiDAR and camera data due to internal system reasons.

Spatial misalignment. As the physical size of a camera module is drastically smaller than the size of a vehicle, the relative position of car center to the camera center will inevitably change due to various reasons, like the vibration during driving on the bumpy road, and since such noise happens all the time, it cannot be avoided using online calibration. In addition, such errors can accumulate while the mileage of a vehicle is increasing. To simulate such a situation, we add random rotation and translation noise to the calibration of each camera independently. The range of noise rotation angle is from 1° to 5° andthe translation range is from 0.5 cm to 1.0 cm to accord with the noise range in the real scene. Sensor calibration misalignment will cause spatial misalignment between point cloud and image, as shown in Fig. 6.

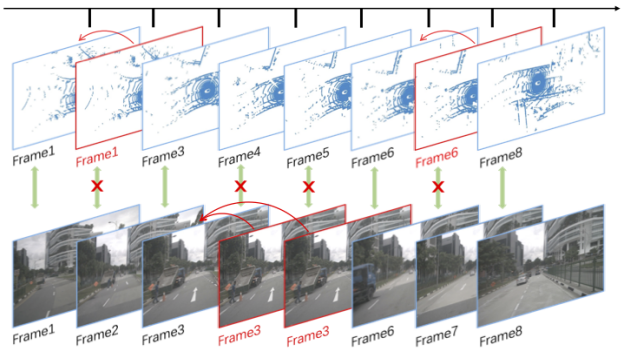


Figure 7. **Visualization of temporal misalignment**. The timestamp of two modalities might not always be aligned on realistic vehicles. We show one concrete example of temporal misalignment.

Temporal misalignment. In a realistic autonomous driving system, failure of the system components is quite common throughout the time. As the streaming data is first fixed with a certain time stamp then passed into the corresponding code module of deep learning model via system sockets, the timestamp of both modality sensors might not be always synchronized. In some rare cases, e.g., sensor connection failure or temporary insufficient cable bandwidth, the data frame of one modality can be stuck by over one minute depending on different system implementations. To simulate such effect, we let the frame remains the same as previous frame when the data is ill-synchronized and we dub the phenomenon data-stuck. Initially, we apply nine levels of severity according to the percentage of stuck frames in all frames. Besides, we consider two ways to select the stuck frames randomly, discrete selection and consecutive selection. In discrete selection, the discrete stuck frames are selected randomly. While in consecutive selection, the continuous multiple frames are selected. And one illustration is shown in Fig. 7, LiDAR-stuck by discrete selection on the top of this figure and camera-stuck by consecutive selection on the bottom.

3.4. A toolkit to transform generic autonomous driving dataset into robustness benchmark

To remove the randomness of benchmark comparison, we compose a toolkit that can transform an autonomous driving dataset into a robustness benchmark [1]. In essence, we only simulate noisy data cases by altering the image and LiDAR data, the ground-truth annotation will remain the same as the 3D position of the object in the surrounding worlds will not change when the sensors malfunction. To facilitate future research, we leverage two popular large-scale autonomous driving datasets, nuScenes and Waymo,

[1] See our website for more details. https://anonymous-benchmark.github.io/robust-benchmark-website.

Approach	M				LiDAR			Camera			
		P_C	mP_R	R	Stuck	FOV	Object	Stuck	Missing	Occlusion	Calib
nuScenes-R (mAP / NDS)											
CenterPoint [59]	L	56.8 / 65.0	23.4 / 46.3	0.41 / 0.71	26.1 / 47.5	15.6 / 43.0	28.4 / 48.5	-	-	-	-
DETR3D [54]	C	34.9 / 43.4	17.6 / 31.5	0.50 / 0.73	-	-	-	17.3 / 32.3	14.5 / 29.9	14.3 / 29.0	24.2 / 35.0
PointAugmenting [50]	LC	46.9 / 55.6	31.9 / 47.0	0.68 / 0.85	25.3 / 43.5	13.3 / 37.7	21.3 / 39.4	42.1 / 52.8	37.0 / 49.8	40.7 / 52.2	43.6 / 53.8
MVX-Net [42]	LC	61.0 / 66.1	37.7 / 53.2	0.62 / 0.81	35.2 / 51.4	17.6 / 43.1	34.0 / 51.1	48.3 / 58.8	32.7 / 50.6	45.5 / 57.6	50.8 / 59.9
TransFusion [1]	LC	66.9 / 70.9	50.2 / 61.8	0.75 / 0.87	33.4 / 52.3	20.3 / 45.8	34.6 / 53.6	65.9 / 70.2	64.9 / 69.7	65.5 / 70.0	66.5 / 70.7
BEVFusion [24]	LC	67.9 / 71.0	51.3 / 61.9	0.76 / 0.87	34.4 / 52.2	21.1 / 45.6	39.2 / 54.6	66.2 / 70.3	65.5 / 70.3	65.3 / 69.6	67.4 / 70.7
Waymo-R(L2 mAP / L2 mAPH)											
CenterPoint [59]	L	66.0 / 63.4	30.6 / 29.4	0.46 / 0.46	29.5 / 28.3	30.3 / 29.1	32.1 / 30.9	-	-	-	-
DETR3D [54]	C	16.2 / 15.7	10.1 / 9.8	0.62 / 0.62	-	-	-	13.0 / 12.6	8.4 / 8.2	10.9 / 10.5	8.0 / 7.8
PointAugmenting [50]	LC	52.5 / 50.7	39.6 / 38.3	0.75 / 0.76	24.7 / 23.9	24.3 / 23.4	26.2 / 25.3	51.7 / 50.0	50.4 / 48.6	50.3 / 48.6	49.8 / 48.1
MVX-Net [42]	LC	59.7 / 54.1	44.3 / 40.1	0.74 / 0.74	27.5 / 24.9	28.8 / 25.6	28.7 / 26.0	58.2 / 52.7	55.9 / 50.5	56.4 / 51.1	54.9 / 49.6
TransFusion [1]	LC	66.7 / 64.1	51.2 / 49.1	0.77 / 0.77	30.2 / 29.0	30.2 / 29.0	32.7 / 31.3	66.5 / 63.9	66.1 / 63.5	66.2 / 63.6	66.3 / 63.7

Stuck: Temporal misalignment for both modalities. FOV: Limited LiDAR FOV. Object: LiDAR object failure.
Missing: Missing camera inputs. Occlusion: Camera Lens Occlusion. Calib: Spatial misalignment of camera-to-world matrix.

Table 1. **Benchmarking the robustness of state-of-the-art methods in all seven scenarios of the nuScenes-R and Waymo-R**. M denotes input modality, camera (C) and LiDAR (L).

Approach	M		LiDAR		Camera			LiDAR		Camera	
		P_C	mP_R	R	mP_R	R	P_C	mP_R	R	mP_R	R
		nuScenes-R (mAP / NDS)					Waymo-R(L2 mAP / L2 mAPH)				
CenterPoint [59]	L	56.8 / 65.0	23.4 / 46.3	0.41 / 0.71	-	-	66.0 / 63.4	30.6 / 29.4	0.46 / 0.46	-	-
DETR3D [54]	C	34.9 / 43.4	-	-	17.6 / 31.5	0.50 / 0.73	16.2 / 15.7	-	-	10.1 / 9.8	0.62 / 0.62
PointAugmenting [50]	LC	46.9 / 55.6	20.0 / 40.2	0.43 / 0.72	40.9 / 52.2	0.87 / 0.94	52.5 / 50.7	25.1 / 24.2	0.48 / 0.48	50.6 / 48.8	0.96 / 0.96
MVX-Net [42]	LC	61.0 / 66.1	28.9 / 48.5	0.47 / 0.73	44.3 / 56.7	0.73 / 0.86	59.7 / 54.1	28.3 / 25.5	0.47 / 0.47	56.4 / 51.0	0.94 / 0.94
TransFusion [1]	LC	66.9 / 70.9	29.4 / 50.6	0.44 / 0.71	65.7 / 70.1	0.98 / 0.99	66.7 / 64.1	31.0 / 29.8	0.46 / 0.46	66.3 / 63.7	0.99 / 0.99
BEVFusion [24]	LC	67.9 / 71.0	31.6 / 50.8	0.46 / 0.72	66.1 / 70.2	0.97 / 0.99	-	-	-	-	-

Table 2. **Robustness against LiDAR and camera modals of state-of-the-art architectures**. In short, the robust metric (R) is computed by averaging the cases by the affecting modality.

and benchmark state-of-the-art methods to evaluate their robustness for the first time to the best of our knowledge. We denote the newly created robustness benchmark nuScenes-R and Waymo-R.

Evaluation Metrics. To intuitively show the robustness of LiDAR-camera fusion methods, we simply use the performance and the relative performance degradation on our benchmark datasets as our evaluation metrics. Specifically, the LiDAR-camera fusion model performance on the clean dataset is denoted as P_C and its corresponding robustness performance against disruption type d under severity level l on the benchmark is denoted as $P_R^{d,l}$. Then, we can estimate the model robustness mP_R by averaging over all noise types and severity levels. The formula can be summarized as follows:

$$mP_R = \frac{1}{N_d}\sum_{d=1}^{N_d}\frac{1}{N_l}\sum_{l=1}^{N_l}P_R^{d,l}, \tag{1}$$

where N_d is the number of disruption types and N_l is the number of severity levels. The relative mean robustness performance of the model is defined as $R = mP_R/P_C$. The higher R means the model is more robust to inferior senor fusion conditions. In practice, we adopt the mean Average Precision (mAP) and the weighted consolidated metric NDS as P_C for nuScenes-R and L2-mAP and L2-mAPH as P_C for Waymo-R.

4. Benchmark Existing Methods

We investigate and evaluate existing popular LiDAR-camera fusion methods with opening source code on our benchmark, including PointAugmenting [50], MVX-Net [42], TransFusion [1] and BEVFusion [24]. In addition, we also evaluate a LiDAR-only method, CenterPoint [59], and a camera-only method, DETR3D [54], for better comparison. It is worth noting that the metrics on waymo focus on intersection over union (IoU). However, strictly calculating the IoU of 3D bounding boxes is quite challenging for camera-based methods. Thus we reduce the IoU threshold to 0.3 and report the vehicle class for DETR3D on Waymo.

4.1. Benchmark Results

The fusion robustness results are shown in Tab. 1. Moreover, to analyze the robustness of models against LiDAR and camera disruptions, we present the mP_R and R of Li-

Approach	Modality	nuScenes-R (mAP / NDS)				Waymo-R(L2 mAP / L2 mAPH)			
		P_C	$(-\pi/2, \pi/2)$	$(-\pi/3, \pi/3)$	$(-0, 0)$	P_C	$(-\pi/2, \pi/2)$	$(-\pi/3, \pi/3)$	$(-0, 0)$
CenterPoint [59]	L	56.8 / 65.0	23.5 / 47.7	15.6 / 43.0	0 / 0	66.0 / 63.4	36.6 / 35.2	30.3 / 29.1	0 / 0
PointAugmenting [50]	LC	46.9 / 55.6	19.5 / 41.2	13.3 / 37.7	0 / 0	52.5 / 50.7	29.4 / 28.3	24.3 / 23.4	0 / 0
MVX-Net [42]	LC	61.0 / 66.1	26.0 / 47.8	17.6 / 43.1	0 / 0	59.7 / 54.1	34.5 / 30.8	28.8 / 25.6	0 / 0
TransFusion [1]	LC	66.9 / 70.9	29.3 / 51.4	20.3 / 45.8	0 / 0	66.7 / 64.1	36.8 / 35.3	30.2 / 29.0	0 / 0

Table 3. **Results of the limited LiDAR field-of-view case**. The angle ranges in brackets mean the visible angle range. $(-0, 0)$ means the extreme case when all LiDAR points are missing.

Approach	Modality	nuScenes-R (mAP / NDS)							
		P_C	{F}	{B}	{FL}	{FR}	{BL}	{BR}	Keeping F
DETR3D [54]	C	34.9 / 43.4	25.8 / 39.2	23.9 / 38.0	28.9 / 39.5	29.1 / 39.8	30.0 / 40.7	29.7 / 40.2	3.3 / 20.5
PointAugmenting [50]	LC	46.9 / 55.6	42.4 / 53.0	41.3 / 52.5	43.6 / 53.8	45.8 / 54.6	45.2 / 54.7	44.9 / 54.6	31.6 / 46.5
MVX-Net [42]	LC	61.0 / 66.1	47.8 / 59.4	45.8 / 58.4	53.6 / 61.9	54.1 / 62.5	55.2 / 63.1	54.6 / 62.6	17.5 / 41.7
TransFusion [1]	LC	66.9 / 70.9	65.3 / 70.1	66.0 / 70.4	66.2 / 70.4	66.4 / 70.5	66.3 / 70.5	66.3 / 70.5	64.4 / 69.3
Approach	Modality	Waymo-R(L2 mAP / L2 mAPH)							
		P_C	{F}	{B}	{FL}	{FR}	{BL}	{BR}	Keeping F
DETR3D [54]	C	16.2 / 15.7	9.2 / 8.8	-	13.4 / 13.0	14.2 / 13.8	14.0 / 13.6	14.4 / 14.0	7.7 / 7.5
PointAugmenting [50]	LC	52.5 / 50.7	50.6 / 48.9	-	51.8 / 50.0	52.1 / 50.3	51.8 / 50.0	51.9 / 50.1	50.2 / 48.4
MVX-Net [42]	LC	59.7 / 54.1	57.1 / 51.7	-	57.5 / 52.2	58.1 / 52.7	58.5 / 53.1	58.9 / 53.5	54.3 / 49.2
TransFusion [1]	LC	66.7 / 64.1	66.3 / 63.7	-	66.5 / 64.0	66.4 / 63.8	66.4 / 63.8	66.5 / 63.9	65.8 / 63.2

Camera location abbr. F: front. B: back. FL: front-left. FR: front-right. BL: back-left. BR: back-right.

Table 4. **Results of the missing camera inputs case**. {X} denotes the location of the missing camera, while the last column indicates the case only keeping the input from the front camera. Note that there is no back camera in the Waymo Open Dataset.

DAR and camera modal separately in Tab. 2. In general, existing methods perform poorly on our robust fusion benchmark as shown in Tab. 1, and there is vast room for improvement. Especially, for all LiDAR-camera fusion methods shown in Tab. 2, the robustness of models against noisy LiDAR cases is worse than the one against noisy camera cases. And among the LiDAR-camera fusion methods we investigated, BEVFusion and TransFusion achieve the overall best robustness. It is worth noting that the robustness against camera noise of them is unexpectedly outstanding, while the robustness against LiDAR noise is even worse than other fusion methods.

We speculate that this is mainly due to the fact that fusing the camera input only brings a marginal improvement, suggesting either the current methods fail to sufficiently leverage the information from the camera or the camera information did not carry the complementary information as intuited. And the fusion mechanism of most of the current popular fusion-based 3D object detection methods rely heavily on accurate LiDAR input. Some of them [50] decorate LiDAR features with corresponding camera features based on calibration matrices on input level. Others [1, 24, 42] use deep feature-level fusion where the features are combined after feature extraction, such as projecting point clouds onto the BEV plane, and then using them as queries to select corresponding image features or using calibration matrices to lift camera features to the same BEV plane, to obtain fused features. Thus, if the LiDAR sensor input is missing, current fusion methods fail to produce meaningful results.

Moreover, when comparing the performance of LiDAR-camera fusion methods with single modality methods on our benchmark, we find all fusion methods have greater robustness on both LiDAR and camera modality than single modality methods. It indicates that when encountering imperfect modality inputs, the fusion methods somehow have the ability to utilize other modal information to enhance the features and predict the final outputs.

4.2. A complete analysis of each noisy data case

We analyze the robustness of existing popular methods on each noisy case proposed in Sec. 3.2.

4.2.1 Noisy LiDAR Data

Limited LiDAR field-of-view. We investigate the situations when the LiDAR points with limited field-of-view in angle range $(-\pi/3, \pi/3)$, $(-\pi/2, \pi/2)$ and $(-0, 0)$. The angle range of $(-0, 0)$ is an extreme case when the LiDAR sensor is completely damaged. The results are shown in Tab. 3. For both LiDAR-only and fusion methods, their performance decreases largely in three situations. Especially, in the extreme case where all LiDAR points are missing, current fusion methods fail to predict any objects like the LiDAR-only method. Thus, for existing fusion methods, the LiDAR modality is the main modality and the camera

modality is auxiliary. An ideal fusion model should still work as long as there is single modality input.

LiDAR object failure. The results of the LiDAR object failure case are shown in Tab. 1. We can find that, with 50% probability to drop all points of the objects, the performance of both LiDAR-only and LiDAR-camera fusion methods reduce by half approximately. This indicates current fusion methods fail to work when the foreground LiDAR points are missing, even the objects appear in the images. From another perspective, it shows that, for the fusion mechanisms of current LiDAR-camera fusion methods, camera information is not well exploited. The fusion process still largely relies on LiDAR information. And more results of object failure setting can be found in Appendix C.

4.2.2 Noisy Camera Data

Missing of camera inputs. In the case of missing camera inputs, we consider several combinations of cameras installed in different positions and report the comprehensive results in Tab. 4, in which we can find that the missing front camera or back camera (for nuScenes) has a greater impact on the detection results. So we consider the case of the missing front camera and the extreme case where all cameras except the front camera are missing in our benchmark. When all cameras except the front camera are missing, the performance of PointAugmenting and TransFusion decreases no more than 50% on both nuScenes-R and Waymo-R. This demonstrates that the robustness of PointAugmenting and TransFusion against camera noise is much better than the other two methods. Besides, the performance degradation on Waymo-R is much smaller than that on nuScenes-R , which indicates the robustness on the various datasets is different.

Occlusion of camera lens. The results for the case of the dirty camera lens are shown in Tab. 1. We observe that among the fusion models, TransFusion is the most robust one compared to the clean settings, while DETR3D is the most sensitive one. Interestingly, although MVX-Net significantly outperforms PointAugmenting in the clean setting, it suffers from more severe performance degradation against occlusions.

4.2.3 Ill-synchronization

Spatial misalignment. For spatial misalignment, the effect of the noise rotation and translation matrix on fusion models is comparable to that of the noisy camera sensor cases, as shown in Tab. 1. We find that the TransFusion is the most robust one compared to the clean settings, while the DETR3D is the most sensitive to the spatial misalignment.

Temporal misalignment. For temporal misalignment, we explore 9 levels of severity and two ways to select the stuck frames, discrete selection and consecutive selection. The results are shown in Fig. 8. A trend can be observed that the performance degradation of all methods is linear to the percentage of stuck frames among all frames. Thus, to reduce the load of the benchmarks, we only consider the case where the stuck frames are 50% of all frames as the final benchmark setting. Interestingly, although TransFusion performs well against the stuck camera frame case, we can observe that the perfromance of TransFusion decreases faster than other fusion methods when the LiDAR-stuck frame ratio increases.

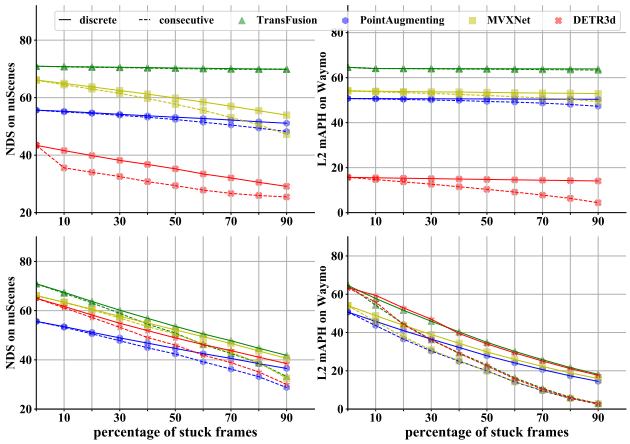


Figure 8. **Temporal misalignment case**. The solid line denotes the discrete selection. The dash line denotes the consecutive selection. Camera-stuck on the top and LiDAR-stuck on the bottom.

5. Discussion and Conclusion

In this work, we collect a series of real-world cases with noisy data distribution, and systematically formulate a robustness benchmark toolkit, that simulates these cases on any clean autonomous driving datasets. We showcase the effectiveness of our toolkit by establishing the robustness benchmark nuScenes-R and Waymo-R, then holistically benchmark the state-of-the-art fusion methods. We further provide a simple robust training strategy in Appendix C, which finetunes the models on these robustness scenarios, and show that it moderately improves the robustness. However, there is still a large performance gap when compared to the results of the clean settings.

We also provide some insights into developing robust fusion models. In general, we believe an ideal sensor fusion framework should be able to do the following: i) given both modality data, it can significantly surpass the performance of single modality methods; ii) when there is a disruption of one modality, the performance should not be worse than the single modality method of the other. We hope our robust benchmark can be a tool for the community to fully exploit this research direction to develop truly robust methods that can be deployed on realistic vehicles.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. *arXiv preprint arXiv:2203.11496*, 2022. 3, 6, 7, 14

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 3

[3] 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. *arXiv preprint arXiv:1903.11027*, 2019. 13

[4] 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 *CVPR*, 2020. 1

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 1, 3

[6] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *arXiv preprint arXiv:2203.10642*, 2022. 3, 16

[7] MMDetection3D Contributors. Mmdetection3d: Openmmlab next-generation platform for general 3d object detection. *https://github.com/open-mmlab/mmdetection3d*, 2020. 14

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 13

[9] Carlos A Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, et al. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21383–21392, 2022. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 13

[11] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2

[12] Tengteng Huang, Zhe Liu, Xiwu Chen, and X. Bai. EPNet: Enhancing point features with image semantics for 3d object detection. In *ECCV*, 2020. 1, 2

[13] Jyh-Jing Hwang, Henrik Kretzschmar, Joshua Manela, Sean Rafferty, Nicholas Armstrong-Crews, Tiffany Chen, and Dragomir Anguelov. Cramnet: Camera-radar fusion with ray-constrained cross-attention for robust 3d object detection. *ECCV*, 2022. 15

[14] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018. 3

[15] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *CVPR*, 2021. 2

[16] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1, 13, 14, 15, 16

[17] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 3, 16

[18] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Bo Wu, Yifeng Lu, Denny Zhou, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. *arXiv preprint arXiv:2203.08195*, 2022. 3

[19] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*, 2021. 2

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1

[21] Ming Liang, Binh Yang, Yun Chen, Rui Hu, and R. Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *CVPR*, 2019. 3

[22] Ming Liang, Binh Yang, Shenlong Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 3

[23] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibing Ling. Cbnet: A composite backbone network architecture for object detection. *arXiv preprint arXiv:2107.00420*, 2021. 14, 15

[24] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *NIPS*, 2022. 3, 6, 7, 14, 16

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 13

[26] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 3

[27] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 3, 16

[28] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *ICCV*, 2021. 2

[29] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, 2021. 2

[30] Muhammad Jehanzeb Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Bischof. Robustness of object detectors in degrading weather conditions. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2719–2724. IEEE, 2021. 3

[31] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3

[32] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020. 3

[33] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 1

[34] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 2, 15, 16

[35] C. Qi, W. Liu, Chenxia Wu, Hao Su, and L. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 2

[36] C. Qi, Hao Su, Kaichun Mo, and L. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[38] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 1, 2

[39] Thomas Roddick, Alex Kendall, and R. Cipolla. Orthographic feature transform for monocular 3d object detection. In *BMVC*, 2019. 2

[40] 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 *CVPR*, 2020. 1, 2

[41] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1, 2

[42] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *International Conference on Robotics and Automation (ICRA)*, 2019. 1, 2, 6, 7, 13, 14, 15

[43] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, pages 1–6, 2021. 15, 16

[44] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 877–894, 2020. 3

[45] 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. 3

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1

[47] Michal Uricar, Ganesh Sistu, Hazem Rashed, Antonin Vobecky, Varun Ravi Kumar, Pavel Krizek, Fabian Burger, and Senthil Yogamani. Let's get dirty: Gan based data augmentation for camera lens soiling detection in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 766–775, 2021. 3, 5

[48] Michal Uricár, Jan Ulicny, Ganesh Sistu, Hazem Rashed, Pavel Krizek, David Hurych, Antonin Vobecky, and Senthil Yogamani. Desoiling dataset: Restoring soiled areas on automotive fisheye cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3, 5

[49] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, 2020. 1, 2

[50] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, 2021. 1, 2, 6, 7, 14

[51] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *CVPR*, 2021. 2

[52] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. In *NeurIPS*, 2021. 2

[53] Yue Wang, Alireza Fathi, Abhijit Kundu, David A. Ross, Caroline Pantofaru, Thomas A. Funkhouser, and Justin M. Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020. 2

[54] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning (CoRL)*, 2022. 6, 7, 13

[55] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M. Alvarez. M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 1, 2

[56] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Bin Zhou, and Liangjun Zhang. FusionPainting: Multimodal fusion with adaptive attention for 3d object detection. In *IEEE*

International Conference on Intelligent Transportation Systems (ITSC), 2021. 1, 2

[57] Yan Yan, Yuxing Mao, and B. Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 2018. 1, 12, 14

[58] Zetong Yang, Y. Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-based 3d single stage object detector. In *CVPR*, 2020. 1, 2

[59] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 2, 6, 7, 12, 14

[60] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multi-modal virtual point 3d detection. In *NeurIPS*, 2021. 1, 2

[61] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and J. Choi. 3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *ECCV*, 2020. 3

[62] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018. 13, 14

[63] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015. 3

[64] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 2

[65] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020. 13

[66] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *CVPR*, 2021. 2

[67] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 2, 12, 14

[68] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 12, 14