Modality-Agnostic Learning for Radar-Lidar Fusion in Vehicle Detection

Yu-Jhe Li Jinhyung Park Matthew O'Toole Kris Kitani
Carnegie Mellon University
{yujheli, jinhyun1, motoole2, kmkitani}@andrew.cmu.edu

Abstract

Fusion of multiple sensor modalities such as camera, Lidar, and Radar, which are commonly found on autonomous vehicles, not only allows for accurate detection but also robustifies perception against adverse weather conditions and individual sensor failures. Due to inherent sensor characteristics, Radar performs well under extreme weather conditions (snow, rain, fog) that significantly degrade camera and Lidar. Recently, a few works have developed vehicle detection methods fusing Lidar and Radar signals, i.e., MVDNet. However, these models are typically developed under the assumption that the models always have access to two error-free sensor streams. If one of the sensors is unavailable or missing, the model may fail catastrophically. To mitigate this problem, we propose the Self-Training Multimodal Vehicle Detection Network (ST-MVDNet) which leverages a Teacher-Student mutual learning framework and a simulated sensor noise model used in strong data augmentation for Lidar and Radar. We show that by (1) enforcing output consistency between a Teacher network and a Student network and by (2) introducing missing modalities (strong augmentations) during training, our learned model breaks away from the error-free sensor assumption. This consistency enforcement enables the Student model to handle missing data properly and improve the Teacher model by updating it with the Student model's exponential moving average. Our experiments demonstrate that our proposed learning framework for multi-modal detection is able to better handle missing sensor data during inference. Furthermore, our method achieves new state-of-the-art performance (5% gain) on the Oxford Radar Robotcar dataset under various evaluation settings.

1. Introduction

In autonomous driving, many vehicles are equipped with multiple sensors, such as camera, Lidar, and Radar, as demonstrated in many datasets [2, 4, 9, 31]. Leveraging different types of sensors can be used to tackle any occasional failures for each sensor and can potentially improve the performance of object detections than using each individual sensor. Existing works [6, 13, 23, 37] mainly focus on fusing Lidar and camera, taking advantage of the camera's higher resolution and rich texture information. However, these visual sensors are sensitive to adverse weather conditions and suffer from degraded performance in harsh weather such as fog [3], snow, and rain.

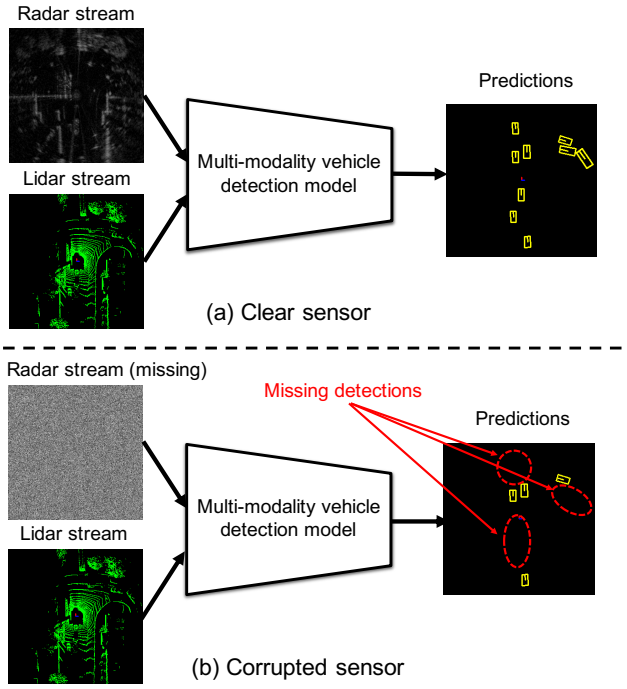

Figure 1. **Illustration of the problem caused by noisy or missing sensor stream**. Models trained on two modalities may suffer from errors when inferenced on missing sensors (only one modality is available).

In addition to Lidar and camera, Radar has also been widely adopted in autonomous system of vehicles [2, 4] and is more robust in certain weather conditions (*e.g.,* fog, snow, rain). To be particular, Radar uses wavelength in the scale of millimeter as ADC chirp signals which is much larger than the size of rain, fog, or even snow [10], making them essentially invisible to Radar. Since the collected

Radar data in existing autonomous driving datasets [4] feature sparse and low resolution data (compared to camera and Lidar), the recent Oxford Radar Robotcar [1] (ORR) dataset, whose Radar sensor has high directionality and much finer spatial resolution, has emerged as a new benchmark for Radar and Lidar fusion. Recently, MVDNet [26] was proposed to fuse Lidar and Radar sensors and achieves state-of-the-art results on the ORR dataset. MVDNet is shown to be successful in adverse conditions such as foggy weather, largely due to the advantageous features of Radar. However, existing Lidar-Radar fusion models [2, 26] are all developed under the assumption that the models will always have access to two reliable sensor streams. If one of the sensors is unavailable or corrupted, performance may suffer (Figure 1). In other words, current fusion models may not be applicable to real-world applications where such failures can occur.

To address this issue, one solution can be to train separate models for processing a number of different sensor streams as input. However, this may be prohibitively expensive. To avoid this, another potential solution is to directly train the fusion model with both clear and missing streams and optimize the model with ground-truth labels. However, such strong data augmentations causes the model to rely on one clear steam and ignore the stream that is missing, which is reflected in our experiments. In other words, a model naively trained with randomly missing sensor streams fails to effectively fuse the two features of the two sensors.

In order to properly leverage data augmentation and mitigate the effect of sensor noise, we propose a framework named Self-Training Multi-modal Vehicle Detection Network (ST-MVDNet) which leverages the backbone of MVDNet [26] and builds upon the self-training pipeline of Mean Teacher (MT) [32] framework. MT was originally proposed for semi-supervised learning, learning two models in parallel, where the Teacher model is used to stabilize the performance of the Student model. To leverage MT to regularize training in our fusion model with strong augmentations (missing streams), our proposed ST-MVDNet also employs two models (Teacher and Student pair), each being architecturally equivalent to MVDNet. The Teacher generates the predictions to train the Student using a consistency constraint while the Student passes the parameters it has learned back to the Teacher via exponential moving average (EMA). The Teacher model only takes clear modalities as input while the Student model additionally takes either missing Lidar or Radar streams as input. We show that by enforcing consistency between the Teacher and the Student, our model is able to prevent a bias towards (over-reliance on) the clear sensor during the training with missing modalities. This pipeline not only allows the model to be more robust to missing sensors but also improves multi-modality feature extraction by forcing the model to better interpret the similarities and relationship between the two modalities. The contributions can be summarized as follows:

- We demonstrate the limitations of a multi-modal detection network when one of its sensors is missing during inference.
- We propose a framework building on Mean Teacher and leverage strong augmentations to address the issue of missing sensor.
- Our developed pipeline is not only able to deal with noisy/missing sensors, which is supported by our designed experiments, but is also able to outperform existing state-of-the-art by a large margin (5%) on ORR dataset in several experimental settings.

2. Related Works

Vehicle Detection using Lidar. Vehicle detection methods on Lidar point clouds are broadly categorized by how they represent the point cloud. One stream of work leverages pioneering works PointNet [24] and PointNet++ [25] for feature extraction directly on unordered point sets. PointRCNN [28] generates object proposals from foreground predicted points, and STD [42] improves PointRCNN with circular proposals and sparse-to-dense refinement. Other works [22, 41] use point-wise voting predictions to move foreground points closer to object centers. Another line of work instead proposes to discretize the 3D space into regular 2D or 3D grids and leverage mature CNN architectures. The pioneering work MV3D [7], processes both range view and bird's eye view 2D projections of the point cloud with 2D CNNs. VoxelNet [44] proposes to work with 3D voxels, leveraging a small PointNet for initial intra-voxel processing and uses 3D convolutions for inter-voxel feature extraction. Observing that most 3D voxels are empty, some subsequent works [8, 11, 29, 38] propose to improve efficiency by only performing convolutions on regions with points. Finally, other works propose to primarily extract features via 2D convolutions in BEV since objects in outdoor scenes rarely overlap when viewed from above. PIXOR [40] generates a BEV occupancy map for different height ranges, and PointPillars [16] uses a PointNet for initial per-grid feature extraction. In our work, we follow MVDNet [26] in adapting PIXOR for multi-sensor fusion because the voxel feature representation is easy to be combined with BEV Radar data.

Vehicle Detection using Multiple Sensor. To enhance 3D perception, many works have proposed to fuse different imaging modalities, including cameras, Lidars, and Radars—the three most common sensors used in autonomous driving. For camera-Lidar fusion, some works constrain the 3D search space with 2D object detections

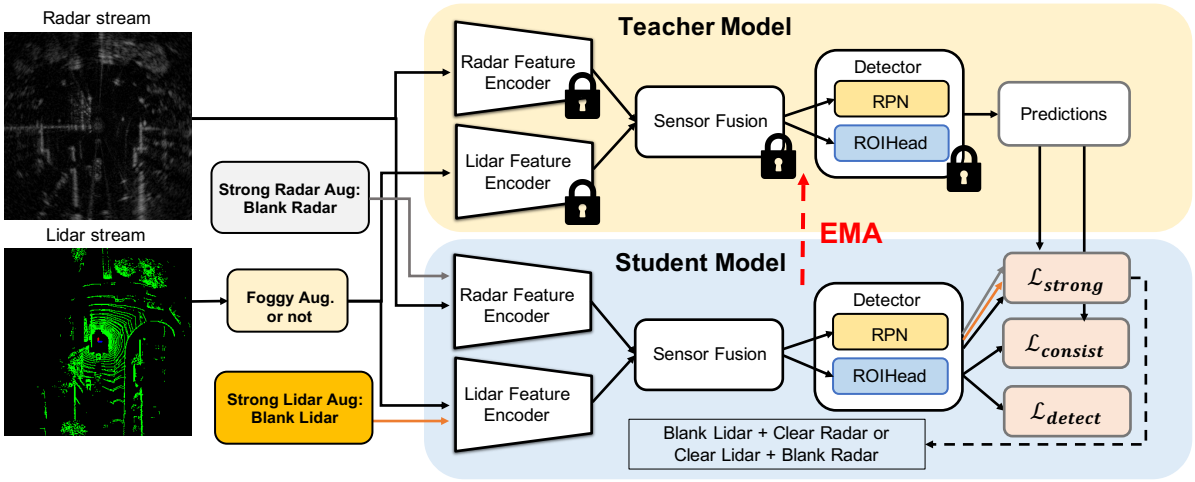

Figure 2. **Overview of our proposed Self-Training Multimodal Vehicle Detection Network (ST-MVDNet).** Our model consists of two modules: 1) Teacher model taking clean data streams for both Radar and Lidar (foggy or not) and 2) Student model taking additional missing streams from both sensors. We train our model using two learning processes: 1) supervised detection learning ($\mathcal{L}_{detect}$) in the Student model, and 2) the mutual learning with strong ($\mathcal{L}_{strong}$) and normal ($\mathcal{L}_{consist}$) consistency losses. The Teacher generates predictions to train the Student with while the Student updates the Teacher via exponential moving average (EMA).

[15,23,34], while others fuse 2D and 3D features at the task-level [20, 21, 33, 36] or feature-level [7, 12, 14, 17, 30, 43]. However, since Lidar's wavelengths are shorter than those of Radar, making Lidar more susceptible to weather interference, recent works [5, 19, 39] also leverage Radar for autonomous perception. DEF [2] proposes an baseline fusion detector with all of the sensors including Lidar, Radar, and camera. Yet, their doppler-based Radar has only low resolution spectral maps which demonstrates inferior performance. Specifically, compared to Oxford Radar Robotcar [1] (ORR) dataset which uses Navtech, the front-view Radar in DEF have only limited field of view (FOV) which depends on the density of receivers and may not be easily adapted to 360° detection with Lidar. Recently, MVDNet [26] proposes fusion model with attention mechanism for vehicle detection in foggy weather conditions on the ORR dataset. However, existing Lidar-Radar fusion models are developed under the assumption that the trained model can only take a fixed number of sensor streams into account. If one of the sensors is unavailable, the model may suffer from noise due to missing input data. In this paper, we leverage the design of MVDNet and demonstrate how our proposed self-training framework is able to mitigate this problem.

3. The Proposed Method

3.1. Problem Formulation and Overview

Given Radar intensity maps and Lidar point clouds, we aim to detect the vehicle in bird-eye-view (BEV) projection maps. Specifically, we are given N Radar intensity maps denoted by $X_r = \{x_r^i\}_{i=1}^N$ where $x_r \in \mathcal{R}^{H\times W\times 1}$ with only one channel, and N Lidar occupancy and intensity maps $X_l = \{x_l^i\}_{i=1}^N$ where $x_l \in \mathcal{R}^{H\times W\times (C+1)}$ with C occupancy channels and 1 intensity channel. The occupancy channels and intensity channel use PIXOR's [40] BEV projection of point clouds. We further denote the annotations as $Y = \{y^i\}_{i=1}^N$.

The overview of our framework is presented in Figure 2. Our Self-Training Multimodal Vehicle Detection Network (ST-MVDNet) consists of two architecturally identical fusion models: Teacher model and Student model. Each of the models is composed of one feature encoder for each of the two sensors, the fusion module, and the detector. Both the teacher model and the student model take as input two sensor streams: Radar and Lidar. We train our model using Teacher-Student mutual learning with strong augmentations on Radar and Lidar. The Lidar stream will be randomly foggified using the fog model in DEF [2] with a probability of 0.5 following [26]. To begin with, we train the student model object detector using the available annotations $Y = \{y^i\}_{i=1}^N$ in the first stage with the standard detection loss as in Equation 1. Then, to start the stage of mutual learning (Sec. 3.3), we copy the entire network parameters from Student to Teacher (duplicate the Student model to Teacher model). In this second stage, the Teacher generates predictions to train the Student with consistency loss while the Student updates the knowledge it learned back to the Teacher via exponential moving average (EMA) of its weights. We introduce two strong augmentations: strong Radar augmentation and strong Lidar augmentation to force the student model to learn to derive features which are invariant to missing sensors. Specifically, this is achieved by

using the consistency loss, which we define as "strong consistency loss", to ensure the predictions obtained from the student model with one missing sensor derive the consistent outputs as the Teacher model. During the inference stage, we only keep the the Teacher model since the Teacher is the temporal ensemble of the Student models in different time steps and is more robust.

3.2. Sensor Fusion of Radar and Lidar

The backbone of each model in Teacher and Student contains one feature encoder for each of the sensors, a sensor fusion model, and a detector, which is also shown in Figure 2. The model architecturally employs MVDNet [26] with the same design. Similar to classical anchor-based detectors, it consists of two stages: RPN and ROI head for sensor fusion. The region proposal network (RPN) derives feature maps from Lidar and Radar, and then generates candidate proposals. The region of interest (ROI) head pools the region-wise features from both of the two sensors and fuse them to obtain oriented bounding boxes of detected objects (vehicles). Since the focus and contribution of the paper is not the architecture design of the fusion model, we only briefly review the design as below.

Feature Encoders and RPN. Our ST-MVDNet has one feature encoder for each of the sensors, ending up with two encoders. The feature encoder utilizes several coarse-grained convolutions and merges the multi-scale feature maps via a residual connection, similar to Unet [27]. Each of the two sensor specific feature extractor will derive feature representations for each of the input sensor streams. The region proposal network (RPN) takes the feature maps as input and generates proposals for the detector later.

Sensor Fusion and ROI head. Following the fusion techniques in [26], our sensor fusion model employs self-attention and cross-attention blocks to merge the feature maps from synchronized pair of Lidar and Radar frames and output the locations of each bounding boxes using RIO head. More details can be obtained in [26].

3.3. Mutual Learning between Teacher and Student

Following the teacher-student framework or Mean Teacher (MT) [32], initially proposed for semi-supervised object detection, our model also consists of two models with identical architecture: a Student model and a Teacher model. The Student model is trained using standard gradient back-propagation algorithm, and the Teacher model is updated with the exponential moving average (EMA) weights of the student model. Since the Teacher model can be seen as an ensemble of the Student model's current and earlier versions, the Teacher model is able to guide the Student model with more accurate predictions.

Supervised Learning of the Detectors. We first optimize the object detector in the student model using the ground-truth labeled data $\mathcal{D} = \{(X_r, X_l, Y\}$ to optimize our model with the loss $\mathcal{L}_{detect}$. Since it is important to have a reliable initialized weights for the Teacher model, we firstly copy the weights to the teacher model from the student model. Therefore the loss for training the model with the annotations can be written as:

$$\begin{aligned}\mathcal{L}_{detect}(X_r, X_l, Y) =& \mathcal{L}^{rpn}_{cls}(X_r, X_l, Y) \\ +\mathcal{L}^{rpn}_{reg}(X_r, X_l, Y) &+ \mathcal{L}^{roi}_{cls}(X_r, X_l, Y) + \mathcal{L}^{roi}_{reg}(X_r, X_l, Y),\end{aligned} \tag{1}$$

where RPN loss $\mathcal{L}^{rpn}$ is for the Region Proposal Network (RPN) module which is used for proposal generation, and ROI loss $\mathcal{L}^{roi}$ is for the region of interest module (ROI). Both of the modules perform bounding box regression (reg) and classification (cls) on the proposals. We use binary cross-entropy loss for $\mathcal{L}^{rpn}_{cls}$ and $\mathcal{L}^{roi}_{cls}$, and l_1 loss for $\mathcal{L}^{rpn}_{reg}$ and $\mathcal{L}^{roi}_{reg}$

Optimize Student with Predictions from Teacher. To regularize the Student model using the Teacher model, we generate predictions from the Teacher for training the Student with consistency loss. To prevent the propagated errors from noisy pseudo-labels, we filter the false positives with a confidence threshold δ. In addition, we remove duplicated boxes using non-maximum suppression (NMS). After obtaining the predictions from the Teacher model on the input sensor streams, we can construct a consistency loss on the Student model as:

$$\begin{aligned}\mathcal{L}_{consist}(X_r, X_l, \hat{C}_t) = \mathcal{L}^{rpn}_{cls}(X_r, X_l, \hat{C}_t) \\ + \mathcal{L}^{roi}_{cls}(X_r, X_l, \hat{C}_t),\end{aligned} \tag{2}$$

where $\hat{C}_t$ denotes the predictions generated by the Teacher model. Note that, we do not apply losses for the bounding box regression since the confidence score of predicted bounding boxes on the unlabeled data can only represent the confidence of the categories for each object instead of the locations for the produced bounding boxes.

Update Teacher via Exponential Moving Average. To obtain more stable predictions following MT, we apply Exponential Moving Average (EMA) in each step to gradually update the Teacher model. The update can be written as:

$$\theta_t \leftarrow \alpha\theta_t + (1-\alpha)\theta_s, \tag{3}$$

where θ_t and θ_s denote the network parameters of Teacher and Student, respectively.

3.4. Learning Robustness Against Biased Fusion

We have now briefly introduced the method of the sensor fusion and the algorithm of mutual learning used to regularize the model to be more stable and robust. However,

since the fusion model is trained with two modalities, the model may be disproportionately negatively impacted when a modality is missing. One potential solution is to simply train the model with both clear and missing modalities, targeting the ground-truth labels. However, we found that such a training scheme weakens the fusion of modalities - the model learns to work with each modality more separately so that it is able to generate predictions even when one sensor is missing. When both sensors are present, the model performs worse when compared to when it is trained without this strong augmentation training scheme.

To properly apply **strong augmentations** to Lidar and Radar sensors, we utilize the consistency loss $\mathcal{L}_{consist}$ to regularize the model instead of naively using ground-truth detection loss $\mathcal{L}_{detect}$. As shown in Figure 2, the Teacher model always takes as input a complete version of both modalities while the Student model additionally takes as input a missing Lidar or Radar stream (with the other stream being clear). By forcing the Student's predictions, which are the result of a single clear modality, to be similar to the Teacher's predictions, which took advantage of two clear modalities, the model is trained to recover features from the missing modality to generate better detections. This is fundamentally different from the simple addition of augmentation, which decreases the importance of fusion between modalities. Instead, our self-training pipeline strengthens the focus on multi-modal fusion. Our method can be considered a form of cross-modal distillation - the Teacher model distills its multi-modal fusion features to the Student in the form of detections, forcing the student to recover multi-modality features from a single modality. We can write the strong augmentation consistency loss, which includes both scenarios of missing Lidar plus clear Radar and clear Lidar plus missing Radar, as follows:

$$\begin{aligned} &\mathcal{L}_{strong}(X_r, X_l, X_{rn}, X_{ln}, , \hat{C}_t) = \\ &\mathcal{L}_{cls}^{rpn}(X_r, X_{ln}, \hat{C}_t) + \mathcal{L}_{cls}^{roi}(X_r, X_{ln}, \hat{C}_t) \\ &+\mathcal{L}_{cls}^{rpn}(X_{rn}, X_l, \hat{C}_t) + \mathcal{L}_{cls}^{roi}(X_{rn}, X_l, \hat{C}_t), \end{aligned} \tag{4}$$

where $\hat{C}_t$ denotes the predictions generated by the Teacher model. X_{rn} and X_{ln} denote the augmented missing streams for Radar and Lidar, respectively.

3.5. Full Objective and Inference

The total loss $\mathcal{L}$ for training our proposed ST-MVDNet is summarized as follows:

$$\mathcal{L} = \mathcal{L}_{\text{detect}} + \lambda_{\text{consist}} \cdot \mathcal{L}_{\text{consist}} + \lambda_{\text{strong}} \cdot \mathcal{L}_{\text{strong}}, \tag{5}$$

where λ_{consist} and λ_{strong} are the hyper-parameters used to control the weighting of the corresponding losses. We note that $\mathcal{L}_{\text{detect}}$, $\mathcal{L}_{\text{consist}}$, and $\mathcal{L}_{\text{strong}}$ are developed to learn the feature encoder and detector in the Student model. The Teacher model is only updated through EMA discussed in the Sec 3.3.

With the interaction between the Teacher and the Student, both models can evolve jointly and continuously to improve detection accuracy under strong augmentation. From another perspective, we can also regard the Teacher as the temporal ensemble of the Student models in different time steps, which aligns with the observation that the accuracy of the Teacher is consistently higher than the Student. As a result, during the inference stage we only keep the Teacher model for evaluating on the testing dataset.

4. Experiments

4.1. Experimental Settings

Dataset Following MVDNet [26], we use the Oxford Radar Robotcar [1] (ORR) dataset for our experiments. The datset has $8,862$ sample pairs of Lidar and Radar frames, which are split into training and testing sets ($7,071$ and $1,791$ for each respectively) without being overlapped in terms of geography. We use the ground-truth annotations created in [26] for training the model, which created 3D bounding boxes of vehicles in one of sequential 20 frames. The annotations of the rest interval 19 frames are generated by [26] who interpolate the bounding boxes by using the visual odometry data provided in ORR. The ORR Navtech Radar scans the environment with 360° field of view at a step of 0.9° every 0.25 seconds while the Lidar scans at a step of 0.33° every 0.05 seconds. The scanning results of Radar and Lidar are saved into the formats of 2D BEV image map and 3D point cloud, respectively, where both share the same world coordinate origin with odometry parameters. As we know, significant scanning delay in Radar suffers from non-synchronization with the Lidar cause frame-wise misalignment, we also use the synchronized and processed streams from [26] to address this issue. Similar to [26], the RoI for the sensors is set to [-32,32] × [-32,32] meter and BEV projection is conducted with a 0.2 meter quantization. The height range is set to [-2.5,1] meter while all Lidar 3D points are vertically divided into 35 slices with a bin size of 0.1 meter. Plus one dimension intensity map, the size of the input Lidar is 320×320×36. Since Radar intensity image has only one intensity channel, the input size of Radar is 320×320×1.

Evaluation settings We evaluate the model on two settings: 1) simulated foggy weather and 2) missing sensors. Following [26], we train the model on the clear Radar stream and the randomly foggified Lidar stream. To foggify Lidar as needed, we change the Lidar point clouds in the training samples using the fog model in DEF [2] with a probability of 0.5. Specifically, for each point in Lidar,

Table 1. The average precision (AP, in %) on different experimental settings of the **simulated foggy weather** with different methods. The number in bold indicates the best score.

Method	Train	Clear + Foggy (Lidar)						Clear					
	Test	Clear			Foggy Lidar			Clear			Foggy Lidar		
	IoU	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8
PIXOR [40]		72.8	68.3	41.2	62.6	58.9	35.7	71.0	67.2	40.6	61.8	58.3	35.7
PointRCNN [28]		78.2	73.8	45.7	69.7	65.6	41.6	78.2	72.8	43.4	68.7	64.0	37.6
PointPillars [16]		85.8	83.0	58.3	72.8	70.3	48.6	85.8	82.9	60.6	71.3	68.3	47.8
DEF [2]		86.6	78.2	46.2	81.4	72.5	41.1	85.9	78.1	44.2	71.8	63.7	32.4
MVDNet [26]		90.9	88.8	74.6	87.4	84.6	68.9	87.2	86.1	72.6	78.0	75.9	61.6
MVDNet [26] + Strong aug.		88.2	85.1	71.7	83.4	81.2	66.1	84.5	85.5	72.1	77.4	71.8	60.0
ST-MVDNet (w/o strong aug.)		94.5	**93.7**	80.2	90.0	86.7	71.4	**91.7**	89.4	77.8	80.1	79.7	63.4
ST-MVDNet (Ours)		**94.7**	93.5	**80.7**	**91.8**	**88.3**	**73.6**	91.4	**89.9**	**78.4**	**81.2**	**80.8**	**64.9**

the fog model will drop it by setting the distance threshold. Each threshold is corresponding to the fog density and if the points are beyond this threshold, they will be dropped.

To evaluate on **simulated foggy weather**, we test the model on either clear or foggy Lidar stream along with normal Radar stream, following [26]. To evaluate on the **missing sensor**, we test the model on two settings:

- Missing Radar (Clear Lidar)
- Missing Lidar (Clear Radar)

We use entirely blank Radar intensity map to simulate the corrupted Radar while we use entirely blank occupancy and intensity maps for the missing Lidar.

Evaluation protocol Following [26], we evaluate the model using mean average precision (mAP) in COCO evaluation [18] with different IOU: 0.5, 0.65, and 0.8 for fair comparison.

4.2. Implementation Details

Following [26], we implement ST-MVDNet with Detectron2 [35], a codebase for Faster RCNN object detectors implemented with PyTorch. For the region proposal network (RPN), the anchors are set to 3.68 m × 7.35 m, and orientations in -90°, -45°, 0° and 45°. The matching of positive and negative samples is conducted with thresholds of 0.55 and 0.45, respectively, while the IoU threshold of NMS is set to 0.7. We also kept top 1000 proposals during training while 500 are kept during inference. For the RoI head, the size of the RoI head pooling is set to 7 × 7 while the IoU threshold of NMS is set to 0.2. For the hyperparameters, we set $\lambda_{consist} = 1.0$ and $\lambda_{strong} = 1$ in all the experiments for simplicity. We set the confidence threshold as $\delta = 0.8$. We note that the used historical frame is set as 2 in our model due to the limitation of GPU resources while MVDNet uses 4 ([26] also shows more frames can lead to slightly improved performance). During the initialization stage described in Sec. 3.3, we train the model using the ground-truth labels for 10k iterations with detection loss $\mathcal{L}_{\text{detect}}$. We then copy the weights to both Teacher and Student models in the beginning of mutual learning and train the ST-MVDNet for 80k iterations. We set the learning rate as 0.01 without decaying since we found this can improve the performance. We optimize the network using Stochastic Gradient Descent (SGD). The weight smooth coefficient parameter of the exponential moving average (EMA) for the teacher model is set to 0.9996. Each experiment is conducted on 1 Nvidia 2080 Ti with a batch size of 1.

4.3. Results and Comparisons

Simulated foggy weather In this setting, we train all detectors with randomly foggified Lidar point clouds while fusion detectors are additionally trained with Radar intensity maps. We then test these models on foggy or clear Lidar following [26] while fusion models can take clear Radar as additional input. We note that there is no missing or missing sensor stream in this standard setting where we can benchmark our ST-MVDNet fairly with current state-of-the-art approaches. We compare our ST-MVDNet against existing Lidar-only detectors (PIXOR [40], PointRCNN [28], and PointPillars [16]), and the Lidar-Radar fusion methods (DEF [2], MVDNet [26]). The results are summarized in Table 1. We observe four phenomena. First, all three Lidar-only detectors perform significantly worse than the fusion methods. This indicates Radar plays an important role for improved performance in either clear or foggy weather condition. Second, in both clear or foggy weather condition, our ST-MVDNet show significant advantages over other detectors, justifying the generalization of our model in the foggy setting with clear sensor streams. The performance gain (around 4% in each setting) between our model and MVDNet can be credited to the proposed mutual learning of our model, which makes the learning procedure stable

Table 2. The average precision (AP, in %) on different experimental settings of the **missing sensor** with different methods. The number in bold indicates the best score.

Method	Train	Clear + Foggy (Lidar)						Clear					
	Test	Missing Radar			Missing Lidar			Missing Radar			Missing Lidar		
	IoU	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8	0.5	0.65	0.8
MVDNet [26]		82.3	80.7	67.8	73.4	68.3	43.3	80.5	77.1	64.8	71.0	65.9	40.1
MVDNet [26] + Strong Lidar aug.		77.4	74.6	62.5	75.2	70.1	47.4	77.6	73.5	61.3	72.1	67.7	43.6
MVDNet [26] + Strong Radar aug.		83.2	80.9	68.9	68.7	63.2	40.1	81.0	77.9	65.1	66.5	60.2	39.2
MVDNet [26] + Strong aug.		82.5	81.2	68.4	73.6	68.7	44.5	80.1	77.8	64.5	71.6	66.0	40.8
ST-MVDNet (w/o strong aug.)		85.7	83.5	70.4	75.1	72.5	51.6	83.9	80.2	67.8	74.5	70.1	51.5
ST-MVDNet (w/ strong Lidar aug.)		85.4	83.1	72.5	82.5	**78.6**	70.4	83.4	80.1	67.7	79.1	**78.0**	61.5
ST-MVDNet (w/ strong Radar aug.)		88.5	86.2	**74.1**	75.0	71.6	52.0	89.1	83.2	72.5	74.0	70.7	50.6
ST-MVDNet (Ours)		**88.7**	**86.9**	73.2	**82.6**	78.1	**70.6**	**89.7**	**84.3**	**73.1**	**79.3**	77.4	**61.7**

Table 3. Ablation studies on the proposed ST-MVDNet. The average precision (AP, in %) on different experimental settings.

Method	Train	Clear + Foggy (Lidar)					
	Test	Missing Radar			Missing Lidar		
	IoU	0.5	0.65	0.8	0.5	0.65	0.8
Ours		**88.7**	**86.9**	**73.2**	**82.6**	**78.1**	**70.6**
Ours w/o $\mathcal{L}_{\text{strong}}$		85.7	83.5	70.4	75.1	72.5	51.6
Ours w/o $\mathcal{L}_{\text{consist}}$		87.9	86.1	72.8	81.0	76.5	67.8
Ours w/o $\mathcal{L}_{\text{consist}}$ & $\mathcal{L}_{\text{strong}}$		82.7	81.4	68.7	73.4	70.5	48.8
MVDNet [26]		82.3	80.7	67.8	73.4	68.3	43.3

and regularized. If referred to "ST-MVDNet (w/o strong aug.)" of Table 1, our model outperforms MVDNet without strong augmentation. Third, we also observe performance drop in the model "MVDNet + Strong aug.", which infers that the strong augmentation can hurt the model if trained naively with ground-truth detection loss. Lastly, comparing to training using both clear and foggy Lidar point clouds, we can observe performance drop on training using only clear Lidar. Also noted in [26], the result also indicates that augmenting foggy Lidar is crucial for improved model performance.

Missing sensor Since the purpose of our model design is to address the issue of missing sensors during inference, we evaluate our model in the settings with either missing Lidar or Radar. We compare our model with the MVDNet [26] and summarize the results in Table 2. We note that "strong Lidar aug." or "strong Radar aug." indicate only one augmentation is applied during the training, while "strong aug." means the missing augmentation for both sensors is applied during the training (the default setting for our final model in the last line). We also train all detectors with clear or foggy Lidar and Radar intensity maps. First, we observe that for MVDNet, adding strong augmentation on each sensor leads to improved performance when the corresponding sensor is missing during testing, but leads to performance drop when testing on the other sensor with noise. For example, MVDNet using strong Lidar augmentation leads to performance gain on missing Lidar but has significant performance drop on missing Radar. We attribute this drop to the bias towards (over-reliance on) the clear sensor caused by the strong augmentation when trained directly with the ground-truth detection loss. Nevertheless, we can still observe slight performance gains on MVDNet if it is trained with both augmentations. On the other hand, our proposed model, when trained with strong augmentation on a single sensor, improves performance when that sensor is missing and maintains performance when the other sensor is faulty. Further, applying augmentations to both sensors leads to performance gains over MVDNet and partial strong augmentation settings.

4.4. Ablation studies

To further analyze the the losses in our proposed framework, we conduct ablation studies in as shown in Table 3.

Strong consistency loss $\mathcal{L}_{strong}$. To analyze how significant the strong consistency loss is, we exclude the loss $\mathcal{L}_{strong}$ and report the performance on two settings with missing sensors. It can be observed that the 3% and 7% performance drop appears on each of the settings. This shows that strong augmentation plays an important role in mitigating the issue of missing sensors.

Consistency loss $\mathcal{L}_{consist}$. To further analyze the consistency loss, we remove the loss $\mathcal{L}_{consist}$ and observe around 1-2% performance drop. This implies that the consistency loss helps with the regularization of the mutual learning between the Teacher and the Student.

Exponential moving average (EMA). As discussed in the earlier section, the Teacher model can be seen as an ensemble of the Student model's current and earlier versions. That is, its generated predictions are more robust and stable than the Student employing architecture of MVDNet. To substantiate this, we exclude both of the losses: $\mathcal{L}_{strong}$ & $\mathcal{L}_{consist}$, and report performance of the model using only EMA updating for Teacher: "Ours w/o $\mathcal{L}_{\text{consist}}$ & $\mathcal{L}_{\text{strong}}$". We can observe the performance gain between this model and MVDNet can be credited to the EMA ensemble.

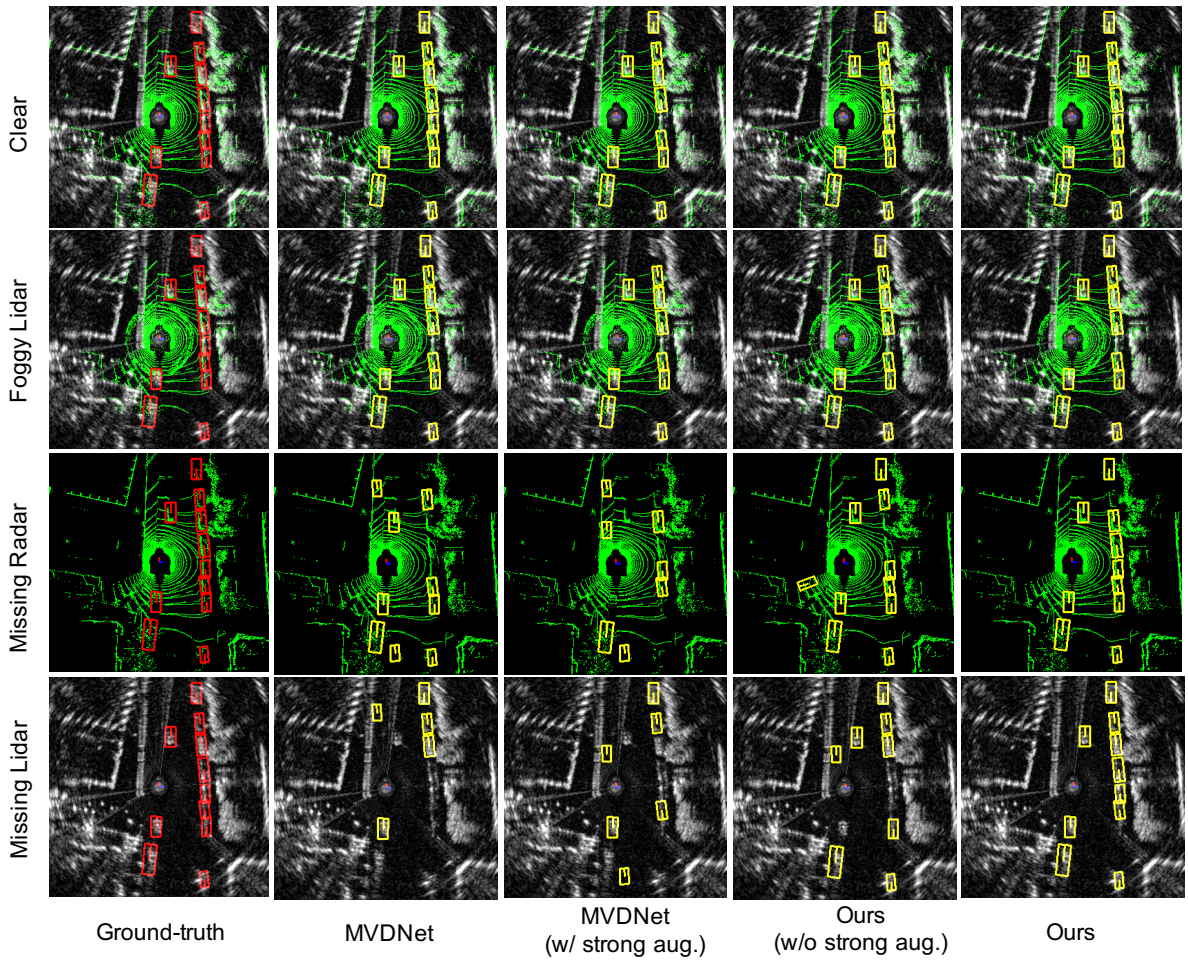

Figure 3. **Qualitative comparisons and analysis.** All of the compared models are trained on clear Lidar and foggy point clouds and Radar intensity maps. The white map indicates Radar intensity while the green one represents the Lidar BEV projection map. We visualize both maps in the same figure. Each row indicates the testing environments.

4.5. Qualitative results

To further analyze the effectiveness of our model with strong augmentation, we compare our model with MVDNet and present the qualitative results in Figure 3. Here, we train all of the models with clear Lidar and foggy point clouds and Radar intensity maps and test on different settings (different rows). Some phenomena can be summarized as follows. First, in either the clear or the foggy testing settings, both MVDNet and our model exhibit similar detection results, which implies that augmentation does not affect the model testing under clear streams in either clear or foggy weather. Second, when testing on either missing Radar or missing Lidar, MVDNet seems to have missing detection and some false positives whenever the strong augmentation is applied or not. Our model without the strong augmentation also suffers from missing detection when tested with missing sensors. Yet, when the strong augmentation is added, our model produces accurate detections in both of the settings with missing sensors.

5. Conclusion

In this paper, we proposed a framework named ST-MVDNet to address the issue of missing sensors in multi-modal vehicle detection. Our model leveraging Mean Teacher and an off-the-shelf fusion model demonstrates significant robustness in performance even when a modality is missing. We attribute the success of our model to our proposed mutual learning pipeline with strong augmentations, which prevents our model from biasing to single sensor. Extensive experimental results on multiple settings also demonstrates the effectiveness of our framework both in adverse weather and with missing sensors. Our model outperforms existing state-of-the-art by a large margin (5%) on ORR dataset in several experimental settings.

Acknowledgement: We thank DENSO CORPORATION for the project sponsorship. In particular, we thank Prasanna Sivakumar, Shawn Hunt, Hironobu Akita, and Daiji Watanabe for the project discussion.

References

[1] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6433–6438. IEEE, 2020. 2, 3, 5

[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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11682–11692, 2020. 1, 2, 3, 5, 6

[3] Mario Bijelic, Tobias Gruber, and Werner Ritter. A benchmark for lidar sensors in fog: Is detection breaking down? In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 760–767. IEEE, 2018. 1

[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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1, 2

[5] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317. IEEE, 2019. 3

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1907–1915, 2017. 1

[7] Xiaozhi Chen, Huimin Ma, Jixiang Wan, B. Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6526–6534, 2017. 2, 3

[8] C. Choy, JunYoung Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012. 1

[10] Yosef Golovachev, Ariel Etinger, Gad A Pinhasi, and Yosef Pinhasi. Millimeter wave high resolution radar accuracy in fog conditions—theory and experimental verification. *Sensors*, 18(7):2148, 2018. 1

[11] Benjamin Graham, Martin Engelcke, and L. V. D. Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018. 2

[12] Tengteng Huang, Zhe Liu, Xiwu Chen, and X. Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[13] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 1

[14] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2018. 3

[15] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object detection in rgb-d images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4632–4640, 2017. 3

[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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019. 2, 6

[17] Ming Liang, B. Yang, Shenlong Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 6

[19] Ramin Nabati and Hairong Qi. Rrpn: Radar region proposal network for object detection in autonomous vehicles. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3093–3097. IEEE, 2019. 3

[20] Jinhyung D. Park, Xinshuo Weng, Yunze Man, and Kris Kitani. Multi-modality task cascade for 3d object detection. *ArXiv*, abs/2107.04013, 2021. 3

[21] C. Qi, Xinlei Chen, O. Litany, and L. Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4403–4412, 2020. 3

[22] C. Qi, O. Litany, Kaiming He, and L. Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9276–9285, 2019. 2

[23] C. Qi, W. Liu, Chenxia Wu, Hao Su, and L. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, 2018. 1, 3

[24] C. Qi, Hao Su, Kaichun Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2

[25] C. Qi, L. Yi, Hao Su, and L. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[26] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 444–453, 2021. 2, 3, 4, 5, 6, 7

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 4

[28] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2, 6

[29] 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. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2

[30] V. Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282, 2019. 3

[31] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 1

[32] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 4

[33] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4603–4611, 2020. 3

[34] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749, 2019. 3

[35] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[36] Liang Xie, Chao Xiang, Zhengxu Yu, Guodong Xu, Zheng Yang, Deng Cai, and Xiaofei He. Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module. In *ArXiv*, volume abs/1911.06084, 2020. 3

[37] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2018. 1

[38] Yan Yan, Yuxing Mao, and B. Li. Second: Sparsely embedded convolutional detection. In *Sensors (Basel, Switzerland)*, volume 18, 2018. 2

[39] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 496–512. Springer, 2020. 3

[40] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7652–7660, 2018. 2, 3, 6

[41] Zetong Yang, Y. Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11037–11045, 2020. 2

[42] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1951–1960, 2019. 2

[43] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and J. W. Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[44] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 2