

# Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection

Di Feng<sup>1</sup>, Lars Rosenbaum<sup>1</sup>, Klaus Dietmayer<sup>2</sup>

**Abstract**—To assure that an autonomous car is driving safely on public roads, its object detection module should not only work correctly, but show its prediction confidence as well. Previous object detectors driven by deep learning do not explicitly model uncertainties in the neural network. We tackle with this problem by presenting practical methods to capture uncertainties in a 3D vehicle detector for Lidar point clouds. The proposed probabilistic detector represents reliable *epistemic* uncertainty and *aleatoric* uncertainty in classification and localization tasks. Experimental results show that the *epistemic uncertainty* is related to the detection accuracy, whereas the *aleatoric uncertainty* is influenced by vehicle distance and occlusion. The results also show that we can improve the detection performance by 1% – 5% by modeling the aleatoric uncertainty.

## I. INTRODUCTION

Knowing what an object detection model is unsure about is of **paramount** importance for safe autonomous driving. For example, if an autonomous car recognizes a front object as a pedestrian but is uncertain about its location, the system may warn the driver to take over the car at an early stage or slow down to avoid fatal accidents.

Deep learning has been introduced to object detection in the autonomous driving settings that use cameras [1]–[3], Lidar [4]–[10], or both [11]–[17], and has set the benchmark on many popular datasets (e.g. KITTI [18], Cityscapes [19]). However, to the best of our knowledge, none of these methods allow for the estimation of uncertainty in bounding box regression. Moreover, these methods often use a softmax layer to normalize the score vector on the purpose of classifying objects, which does not necessarily represent the classification uncertainty in the model. As a result, these object detections can only tell the human drivers *what* they have seen, but not *how certain* they are about it.

There are two types of uncertainty that we can quantify in the object detection network. *Epistemic uncertainty*, or *model uncertainty*, indicates how uncertain an object detector is to explain the observed dataset. *Aleatoric uncertainty*, conversely, captures the observation noises that are inherent in sensors. For instance, detecting an abnormal object that is different from the training dataset may result in high epistemic uncertainty, while detecting a distant object may result in high aleatoric uncertainty. Capturing both uncertainties

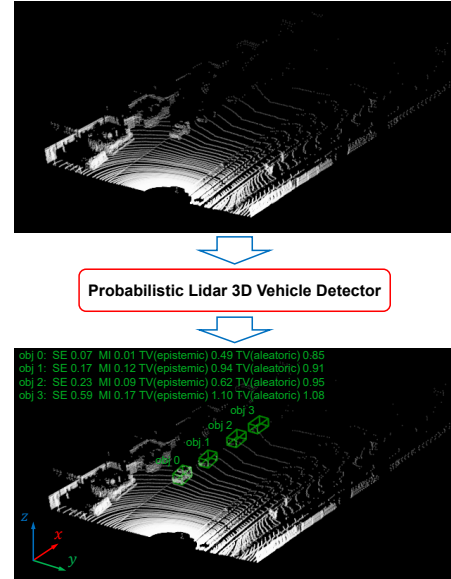


Fig. 1: Our proposed probabilistic Lidar 3D vehicle detection network takes the Lidar point clouds as input. It not only predicts object classes and 3D bounding boxes, but also predicts the model uncertainty and the sensor observation uncertainty. Shannon Entropy (SE) and Mutual Information (MI) quantify the classification uncertainty, and Total Variance (TV) the localization uncertainty. These scores will be described in Sec. IV.

in the object detection network is **indispensable** for safe autonomous driving, as the epistemic uncertainty displays the limitation of detection models, while the aleatoric uncertainty can provide the sensor observation noises for object tracking.

In this work, we develop practical methods to capture epistemic and aleatoric uncertainties in a **3D vehicle detector for Lidar point clouds**. Our contributions are three-fold:

- We extract model uncertainty and observation uncertainty for the vehicle recognition and 3D bounding box regression tasks.
- We show an improvement of vehicle detection performance by modeling the aleatoric uncertainty.
- We study the difference between the epistemic and aleatoric uncertainty. The former is associated with the vehicle detection accuracy, while the latter is influenced by the vehicle distance and occlusion.

The remainder of the paper is structured as follows: Sec. II summarizes related works. Sec. III illustrates the architecture of our probabilistic Lidar 3D vehicle detection

<sup>1</sup> Di Feng, Lars Rosenbaum are with Robert Bosch GmbH, Corporate Research, Driver Assistance Systems and Automated Driving, 71272 Renningen, Germany.

<sup>2</sup> Klaus Dietmayer is with Institute of Measurement, Control and Microtechnology, Ulm University, 89081 Ulm, Germany.

The video to this paper can be found at <https://youtu.be/bQJmssB80oM>.

neural network. Sec. IV presents the proposed methods to capture epistemic and aleatoric uncertainties. Sec. V illustrates the experimental results, followed by a conclusion and a discussion of future research in Sec. VI.

## II. RELATED WORKS

In this section, we first summarize methods for 3D object detection in autonomous driving for Lidar point clouds. We then summarize related works to Bayesian neural network which we use to extract uncertainty in our Lidar vehicle detection network.

### A. 3D Object Detection in Autonomous Driving

1) **3D Object Detection via Lidar Point Clouds:** Many works represent 3D point clouds using **voxel grids**. Li [4] discretizes point clouds into square grids, and then employs a 3D convolution neural network that outputs an objectness map and a bounding box map for each grid. Zhou *et al.* [6] introduce a **voxel feature encoding (VFE) layer** that can learn the unified features directly from the Lidar point clouds. They feed these features to a Region Proposal Network (RPN) that predicts the detections. Engelcke *et al.* [7] use a voting scheme to search possible object locations, and introduce special convolutional layers to tackle with the sparsity of the point clouds. In addition, several works encode 3D point clouds as 2D feature maps. Li *et al.* [5] project range scans into a 2D front-view depth map and use a 2D fully convolutional network (FCN) to detect vehicles. Caltagirone *et al.* [8] and Yang *et al.* [10] project the Lidar point clouds into birds' eye view image and propose a road detector and a car detector, respectively.

2) **3D Object Detection by Combining Lidar and Camera:** Lidar point clouds usually give us accurate spatial information. Camera images provide us with object appearances, which is beneficial for object classification. Therefore, it is natural to combine these two sensors [11]–[17]. For example, Chen *et al.* [11] propose MV3D, a network that generates region proposals from the bird's eye view Lidar features and then combine the proposals with the regional features from the front view Lidar feature maps and RGB camera images for accurate 3D vehicle detection. Qi *et al.* [12] use RGB camera images to draw 2D object bounding boxes, which build **frustums** in the Lidar point cloud. Then, they use these frustums for 3D bounding box regressions.

### B. Bayesian Neural Networks

Bayesian neural networks (BNNs) provide us the probabilistic interpretation of neural networks [20]. Instead of placing deterministic weights in the neural network, BNNs assume a prior distribution over them. By inferring the posterior distribution over the model weights, we can extract how uncertain a neural network is with its predictions. Uncertainty estimation methods include variational inference [21], sampling technique [22], or ensemble [23]. Recently, Y. Gal [24] propose a method that captures the uncertainty in BNNs at test time by sampling the network multiple times with dropout. This dropout sampling method has been

applied to active learning for cancer diagnosis [25], semantic segmentation [26], [27] and image object detection for open-set problem [28].

## III. NETWORK ARCHITECTURE

Our proposed probabilistic Lidar 3D object detection network is shown in Fig. 2. The network takes the features from Lidar **bird's eye view (BEV)** as input and feed it into a pre-processing network. Then, a region proposal network is employed to generate candidates of regions of interest (ROIs), which are processed through the intermediate layers. These layers are built to capture the uncertainties in our object detection network. After the intermediate layers, the features are fed into two fully connected layers for 3D bounding box regression and softmax objectness score.

### A. Input and Output Encoding

To extract the Lidar BEV features, the 3D point clouds are first projected onto a 2D grid with a resolution of 0.1m and then encoded by height, intensity and density maps [11]. The height maps are generated by dividing the point clouds into  $M$  slices. As a result we obtain  $M+2$  channel features. Fig. 2 shows exemplary input features.

The network outputs the softmax objectness score and the oriented 3D bounding boxes. The bounding boxes  $\mathbf{v}$  are encoded by their **8 corner offsets**,  $\mathbf{v} = [x_0, \dots, x_7; y_0, \dots, y_7; z_0, \dots, z_7]$ , and are normalized by the diagonal length of the proposal anchors. Concretely, let  $\hat{\mathbf{v}}$  be the vector of a bounding box in the Lidar coordinate frame and  $\hat{\mathbf{v}}_0$  its corresponding region proposal with diagonal length  $h$ , we encode the 24-dimensional regression outputs by  $\mathbf{v} = \frac{\hat{\mathbf{v}} - \hat{\mathbf{v}}_0}{h}$ .

### B. Feature Pre-processing Network

To process the Lidar feature images, we use the ResNet-8 [29] architecture that contains 4 residual blocks, with the number of kernels increasing from 64, 128, 256 to 512. The last convolution layer is up-sampled with a factor of two for generating region proposals and detecting objects. In this way, the network can detect small vehicles that occupy only a few grid cells (5-40 grid cells [11]).

### C. Region Proposal Networks

We follow the **Faster-RCNN pipeline** [30] to generate BEV 3D region proposals. First, for each feature map pixel we generate nine 2D anchors using 3 scales with box areas of  $16^2$ ,  $32^2$  and  $48^2$  and 3 aspect ratios of 1 : 1, 1 : 2, and 2 : 1. Then, the 2D anchors are projected into 3D space by adding a fixed height value, which is selected large enough to enclose all daily vehicles.

### D. Intermediate Layers

To further process the Lidar features and to extract uncertainties, we design 3 fully connected hidden layers, each of which has 512 hidden units and is followed by a dropout layer.

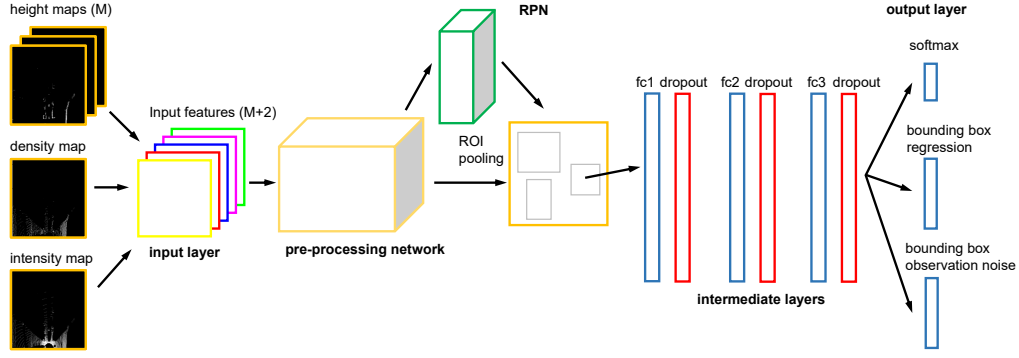


Fig. 2: Network architecture for our proposed probabilistic Lidar 3D vehicle detector.

#### IV. CAPTURING UNCERTAINTY IN LIDAR 3D VEHICLE DETECTION

As introduced in Sec. I, there are two different types of uncertainty that can be captured by our proposed probabilistic Lidar 3D vehicle detector: the epistemic uncertainty that describes the uncertainty in the model, and the aleatoric uncertainty that describes the observation noises.

We employ the intermediate layers (Fig. 2) as a Bayesian neural network to extract uncertainties. Let us denote  $\mathbf{x}^*$  as a ROI candidate generated by RPN during the test time, and  $\mathbf{y}^*$  the prediction of object labels or bounding boxes. We want to estimate the posterior distribution of the prediction  $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y})$ , where  $\mathbf{X}$  denotes the training dataset and  $\mathbf{Y}$  its ground truth. To do this, we marginalize the prediction over the weights  $\mathbf{W}$  in the fully connected layers through  $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{g}^{\mathbf{W}}(\mathbf{x}^*))p(\mathbf{W}|\mathbf{X}, \mathbf{Y})d\mathbf{W}$ , where  $\mathbf{g}^{\mathbf{W}}(\mathbf{x}^*)$  denotes the network output. By estimating the weight posterior  $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$  and performing Bayesian inference, we can extract epistemic uncertainty; By estimating the observation likelihood  $p(\mathbf{y}^*|\mathbf{g}^{\mathbf{W}}(\mathbf{x}^*))$ , we can obtain the aleatoric uncertainty.

##### A. Capturing Epistemic Uncertainty

In order to extract the epistemic uncertainty (or model uncertainty) in our vehicle detection network, we need to calculate the weight posterior  $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ . However, performing the inference analytically is in general intractable, thus approximation techniques are required. Recently, Gal [24] shows that dropout can be used as approximate inference in a Bayesian neural network: during the training process, performing dropout with stochastic gradient descent is equivalent to optimizing the approximated posterior distribution with Bernoulli distributions over the weights  $\mathbf{W}$ . Gal [24] also shows that by performing the network forward passes several times with dropout during the test time, we can get the samples from the model's posterior distribution, and thus obtain the epistemic uncertainty.

We employ the dropout method to capture the model uncertainty in the object detection network. Our network is trained using dropout with a dropout rate  $p$ . For each region proposal during test time  $\mathbf{x}^*$ , it performs  $N$  forward passes with dropout. The output of the network contains

the softmax score  $\{s_{\mathbf{x}^*}^i\}_{i=1}^N$  and the bounding box regression  $\{\mathbf{v}_{\mathbf{x}^*}^i\}_{i=1}^N$ , with  $s_{\mathbf{x}^*}^i$  and  $\mathbf{v}_{\mathbf{x}^*}^i$  being the softmax score and the regression output for the  $i^{th}$  forward pass, respectively. In the following, we illustrate how to use these outputs to measure the epistemic uncertainty.

1) *Extracting Vehicle Probability and Epistemic Classification Uncertainty:* The vehicle probability  $p(\text{veh}|\mathbf{x}^*)$  - the probability that a sample  $\mathbf{x}^*$  is recognized as a vehicle - is approximated by the mean of softmax score of  $N$  forward passes according to

$$p(\text{veh}|\mathbf{x}^*) = \mathbb{E}_{p(\mathbf{W}|\mathbf{X}, \mathbf{Y})} p(\text{veh}|\mathbf{x}^*, \mathbf{W}) \approx \frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i. \quad (1)$$

Note that if we set  $N = 1$  and  $p = 0$ ,  $p(\text{veh}|\mathbf{x}^*)$  is approximated by a softmax score - the point-wise prediction done by most object detection networks. However, as Gal [24] mentioned, passing the point estimate can underestimate the uncertainty of samples that are different from the training dataset. In order to extract more accurate model uncertainty, the network needs to perform multiple forward passes with dropout (i.e.  $N > 1, p > 0$ ).

We further use the Shannon entropy (SE) and mutual information (MI) to measure the classification uncertainty [24]. For a region proposal  $\mathbf{x}^*$ , its SE score is calculated via

$$\begin{aligned} SE(\mathbf{y}^*|\mathbf{x}^*) &= \mathbb{H}(\mathbf{y}^*|\mathbf{x}^*) \\ &= -p(\text{veh}|\mathbf{x}^*) \log p(\text{veh}|\mathbf{x}^*) - p(\neg\text{veh}|\mathbf{x}^*) \log p(\neg\text{veh}|\mathbf{x}^*) \\ &\approx -\frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i \log \frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i - (1 - \frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i) \log (1 - \frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i). \end{aligned} \quad (2)$$

SE captures the uncertainty in the prediction output  $p(\text{veh}|\mathbf{x}^*)$ . In this work, we use the natural logarithm, and the SE score ranges in  $[0, \ln(2)]$ . When  $p(\text{veh}|\mathbf{x}^*) = 0$  or  $1$ , the network is certain with its prediction, resulting  $SE = 0$ . The SE score reaches its peak when the network is most uncertain with its prediction, namely,  $p(\text{veh}|\mathbf{x}^*) = \frac{1}{2}$ .

Different from SE, MI calculates the models' confidence in the output. It measures the information difference between the prediction probability  $p(\text{veh}|\mathbf{x}^*)$  and the posterior of model parameters [24] according to

$$\begin{aligned}
MI(\mathbf{y}^*, \mathbf{W}) &= \mathbb{H}(\mathbf{y}^* | \mathbf{x}^*) - \mathbb{E}_{p(\mathbf{W} | \mathbf{x}, \mathbf{y})} \mathbb{H}(\mathbf{y}^* | \mathbf{x}^*, \mathbf{W}) \\
&= SE(\mathbf{y}^* | \mathbf{x}^*) + \mathbb{E}_{p(\mathbf{W} | \mathbf{x}, \mathbf{y})} \left( p(\text{veh} | \mathbf{x}^*, \mathbf{W}) \log p(\text{veh} | \mathbf{x}^*, \mathbf{W}) \right. \\
&\quad \left. + p(\neg \text{veh} | \mathbf{x}^*, \mathbf{W}) \log p(\neg \text{veh} | \mathbf{x}^*, \mathbf{W}) \right) \\
&\approx -\frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i \log \frac{1}{N} \sum_{i=1}^N s_{\mathbf{x}^*}^i \\
&\quad + \frac{1}{N} \sum_{i=1}^N \left( s_{\mathbf{x}^*}^i \log s_{\mathbf{x}^*}^i + (1 - s_{\mathbf{x}^*}^i) \log (1 - s_{\mathbf{x}^*}^i) \right). \tag{3}
\end{aligned}$$

MI ranges between  $[0, 1]$ , with a large value indicating high epistemic classification uncertainty.

2) *Extracting 3D Bounding Box and Epistemic Spatial Uncertainty*: We estimate the bounding box position of a region proposal in the Lidar coordinate frame, denoted as  $\mathbf{l}_{\mathbf{x}^*}$ . For this purpose, we calculate the mean value of the regression outputs of  $N$  forward passes. To do this, we first transform the bounding box prediction into the Lidar coordinate frame  $\hat{\mathbf{v}}_{\mathbf{x}^*} = h\mathbf{v}_{\mathbf{x}^*} + \hat{\mathbf{v}}_{0_{\mathbf{x}^*}}$ , the mean is calculated accordingly

$$\mathbf{l}_{\mathbf{x}^*} \approx \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{v}}_{\mathbf{x}^*}^i. \tag{4}$$

To estimate the epistemic spatial (regression) uncertainty in the bounding box prediction, we use the total variance of the covariance matrix of  $N$  forward pass regressions. The covariance matrix is calculated via  $C(\mathbf{x}^*) = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{v}}_{\mathbf{x}^*}^i \hat{\mathbf{v}}_{\mathbf{x}^*}^{iT} - \mathbf{l}_{\mathbf{x}^*} \mathbf{l}_{\mathbf{x}^*}^T$ . The total variance  $TV(\mathbf{x}^*)$  is the trace of the covariance matrix according to

$$TV(\mathbf{x}^*) = \text{trace}(C(\mathbf{x}^*)). \tag{5}$$

This score ranges in  $[0, +\infty)$ , with a large value indicating high epistemic spatial uncertainty.

### B. Capturing Aleatoric Uncertainty

So far, we have explained how to use the dropout sampling technique to approximate the posterior distribution and to extract epistemic uncertainty. To capture the aleatoric uncertainty, we need to model the distribution  $p(\mathbf{y}^* | g^{\mathbf{W}}(\mathbf{x}^*))$ . In the classification task we can model it by the softmax function, i.e.  $p(\mathbf{y}^* | g^{\mathbf{W}}(\mathbf{x}^*)) = \text{softmax}(g^{\mathbf{W}}(\mathbf{x}^*)) = s_{\mathbf{x}^*}$ , with  $\mathbf{y}^*$  referring to the object labels. In the 3D bounding box regression task, we model the observation likelihood as a multi-variate Gaussian distribution with diagonal covariance matrix via

$$\begin{aligned}
p(\mathbf{y}^* | g^{\mathbf{W}}(\mathbf{x}^*)) &= \mathcal{N}(g^{\mathbf{W}}(\mathbf{x}^*), \Sigma(\mathbf{x}^*)), \\
\Sigma(\mathbf{x}^*) &= \text{diag}(\sigma_{\mathbf{x}^*}^2). \tag{6}
\end{aligned}$$

Here,  $\mathbf{y}^*$  is the bounding box prediction. The parameter  $\sigma_{\mathbf{x}^*}^2$  is a 24-dimensional vector, with each element representing an observation noise of a prediction in the bounding box regression  $\mathbf{v}_{\mathbf{x}^*}$ . We can obtain the observation noises by adding an output regression layer in our vehicle detection

network. In addition to the softmax scores and bounding box regressions, now it allows us to predict the observation noises  $\lambda_{\mathbf{x}^*}$ , where we use  $\lambda_{\mathbf{x}^*} := \log(\sigma_{\mathbf{x}^*}^2)$  for numerical stability. In the training phase, we modify the cost function for the bounding box regression through

$$L_{\text{out-reg}} = \frac{1}{2} \exp(-\lambda^T) \|\mathbf{v}_{\text{gt}} - \mathbf{v}\| + \frac{1}{2} \lambda^T \mathbf{1}. \tag{7}$$

As [26] mentioned, this loss function can increase the network robustness when learning from noisy dataset. When the training data has high aleatoric uncertainty, the model is penalized by the  $\frac{1}{2} \lambda^T \mathbf{1}$  term and ignores the residual term  $\|\mathbf{v}_{\text{gt}} - \mathbf{v}\|$  since  $\exp(-\lambda^T)$  becomes small. Consequently, the data with high uncertainty contributes less to the loss. Note that instead of performing approximate inference when extracting epistemic uncertainty, here we perform MAP inference.

### C. Implementation

Training the neural network is achieved in a multi-loss end-to-end learning fashion. We use smooth  $l_1$  loss and cross-entropy loss for the oriented 3D box regression ( $L_{\text{out-reg}}$ ) and classification ( $L_{\text{out-cls}}$ ) respectively. We also use the same loss function for region proposals and object category outputs in the Region Proposal Network ( $L_{\text{rpn-cls}}$  and  $L_{\text{rpn-reg}}$ ). The final loss function is formulated as:

$$Loss = \gamma_1 L_{\text{rpn-cls}} + \gamma_2 L_{\text{rpn-reg}} + \gamma_3 L_{\text{out-cls}} + \gamma_4 L_{\text{out-reg}}, \tag{8}$$

where we set  $\gamma_1, \gamma_3 = 1$  and  $\gamma_2, \gamma_4 = 0.05$ . We employ  $L_2$  regularization and dropout with dropout rate  $p = 0.5$  to prevent over-fitting and to perform posterior distribution approximation. The network is trained 70000 steps using Adam optimizer and a learning rate of 0.0001. We then reduce the learning rate to 0.00001 and train another 10000 steps for fine tuning.

To solely extract epistemic uncertainty, the network performs  $N$  number of forward passes with dropout during the test time. We fix the dropout rate at  $p = 0.5$ , though it can be tuned by grid-searching to maximize the validation log likelihood [24] or by using gradient methods [31]. A dropout rate of 0.5 introduces the highest epistemic uncertainty in the model. We also fix  $N = 40$ . To solely extract the aleatoric uncertainty, the network is trained with the loss function modified according to Eq. 7. It performs the feed-forward pass only once without dropout during the test time. To extract both epistemic and aleatoric uncertainty, the network trained with Eq. 7 needs to perform multiple feed-forward passes with dropout.

## V. EXPERIMENTAL RESULTS

In this section, we experimentally evaluate our proposed probabilistic Lidar 3D object detection network. We show that modeling the aleatoric uncertainty improves the vehicle detection performance. We also show that the network captures model uncertainty and observation uncertainty, which behave differently from each other: The model (epistemic) uncertainty is influenced by the detection accuracy, but is



unaffected by the vehicle distance. Conversely, the observation (aleatoric) uncertainty is associated with the vehicle distance and occlusion, and has little relationship to detection accuracy.

### A. Experimental Setup

Our proposed method was evaluated on the **KITTI** raw dataset [18]. Among the dataset, we randomly chose 29 drives for training, and another 6 drives for testing (drive 0001, 0015, 0039, 0052, 0070, 0086). This corresponds to 9918 training frames and 2010 testing frames. We considered the Lidar point clouds in the ranges  $x = 0 - 100$  m,  $y = -30 - 30$  m,  $z = -3.5 - 0.6$  m (The lidar coordinate system is illustrated in Fig. 1). With a discretization resolution of 0.1 m, each bird's eye view input channel was a 2D image with  $1000 \times 600$  pixels. We set the height slices  $M$  to be 13, and only evaluated the detections that were visible in the front-view of the camera.

The 3D vehicle proposals were considered to be detected when their probability scores were larger than 0.5. Their performance was evaluated using Intersection Over Union (IoU) threshold ranging from 0.1 to 0.8. The IoU scores rate the similarity between a predicted vehicle and its ground truth. A higher IoU value with the ground truth indicates a more accurate prediction.

### B. 3D Vehicle Detection Performance

We first evaluated the 3D vehicle detection performance of our proposed network that captures epistemic and aleatoric uncertainty. As a baseline method, we trained a vehicle detector called *Non-Bayesian* which did not explicitly model the observation noise  $P(\mathbf{v}|\mathbf{g}^W(\mathbf{x}))$  and the weight posterior distribution  $P(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ . We then compared the baseline method with the vehicle detectors that captured epistemic uncertainty (*Epistemic*), aleatoric uncertainty (*Aleatoric*), or both (*Epistemic+Aleatoric*). We used  $F_1$  scores ( $F_1 = \frac{2PR}{P+R}$ ) at different IoU threshold to evaluate the detection performance, where  $P$  indicates the 3D bounding box precision, and  $R$  the recall value. Results are illustrated in Tab. I. The vehicle detectors that captured aleatoric uncertainty (*Aleatoric* and *Epistemic+Aleatoric*) consistently outperformed the baseline method, with an improvement of  $F_1$  scores by 1%–5%. This is because modeling the aleatoric uncertainty has increased the model robustness when dealing with noisy input data. The network *Epistemic*, conversely, slightly underperformed the baseline method, as network capability was reduced by dropping out some of its hidden units. We leave finding the optimal dropout rate (e.g. concrete dropout [31]) as an interesting future work.

TABLE I:  $F_1$  score comparison for different vehicle detectors

Network	IoU threshold							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Non-Bayesian (baseline)	0.737	0.724	0.697	0.675	0.634	0.508	0.247	0.043
Epistemic	0.732	0.717	0.691	0.668	0.625	0.498	0.245	0.042
Aleatoric	<b>0.750</b>	<b>0.736</b>	<b>0.716</b>	<b>0.688</b>	<b>0.651</b>	<b>0.533</b>	<b>0.253</b>	<b>0.044</b>
Epistemic+Aleatoric	0.745	0.730	0.704	0.679	0.636	0.517	0.248	0.043

### C. Understanding the Epistemic Uncertainty in 3D Object Detection

We then evaluated how our proposed network captured the epistemic uncertainty in 3D object detection task. Fig. 3(a) and Fig. 3(b) demonstrate the mean values of Shannon Entropy (SE) and Mutual Information (MI) averaged over all predicted samples that lay within different IoU intervals. Predictions with higher IoU scores were more accurate. Results show that SE behaved similar to the MI score. Both scores were associated well with the prediction accuracy, as they became smaller with increasing IoU. We also evaluated the epistemic uncertainty in the bounding box regression. Following Eq. 5, we calculated the mean values of total variance in  $x$ ,  $y$ , and  $z$  axes. Fig. 3(c) shows that the total variance in the  $x$ -axis was the largest, indicating that the network was most uncertain to estimate the vehicles' positions in the  $x$  direction. The total variance in the  $z$ -axis was the smallest, as there was little variance in vehicles' height. Fig. 3(c) also shows that the regression uncertainty was affected by the prediction accuracy. The total variance decreased for larger IoU values.

To better understand the epistemic uncertainty, we plot the SE and MI values for each predicted vehicle with  $\text{IoU} > 0.8$ , referring to the most accurate detections (Fig. 4(a)), and  $\text{IoU} < 0.1$  – the worst detections (Fig. 4(b)). The SE and MI values for detections with  $\text{IoU} > 0.8$  mostly lay on the bottom left of the figure, showing the low epistemic classification uncertainty. In contrast, detections with  $\text{IoU} < 0.1$  were widely spread in Fig. 4(b), demonstrating higher uncertainty. We also analyzed the distribution of total variance for  $\text{IoU} > 0.8$  and  $\text{IoU} < 0.1$ . Fig. 4(c), Fig. 4(d) show large epistemic regression uncertainties, when the network made inaccurate detections. In conclusion, our proposed Lidar 3D object detector captured reliable epistemic classification and regression uncertainty. The objects with low classification and regression uncertainty were more likely to be detected accurately.

*Qualitative Observations.* We observed that big vehicles such as vans and trucks as well as the “ghost” objects (i.e. False Positives) often showed high levels of epistemic classification uncertainty (e.g. obj 5 in Fig. 5(a) and obj 0 in Fig. 5(b)). Besides, detections with abnormal bounding boxes, e.g. boxes with unusual small length and large height (e.g. obj 6 in Fig. 5(a)), often showed high epistemic spatial uncertainty. This is because that these detections differed from our training dataset, which contained no ghost objects, no objects with abnormal shapes, and a few big vehicles. Therefore, our Lidar 3D object detector was uncertain with them, and showed high epistemic uncertainty scores. Such information can be used to efficiently improve the vehicle detector in an active learning paradigm: the detector actively queries the unseen samples with high epistemic uncertainty. More detection results can be found in the supplementary video.

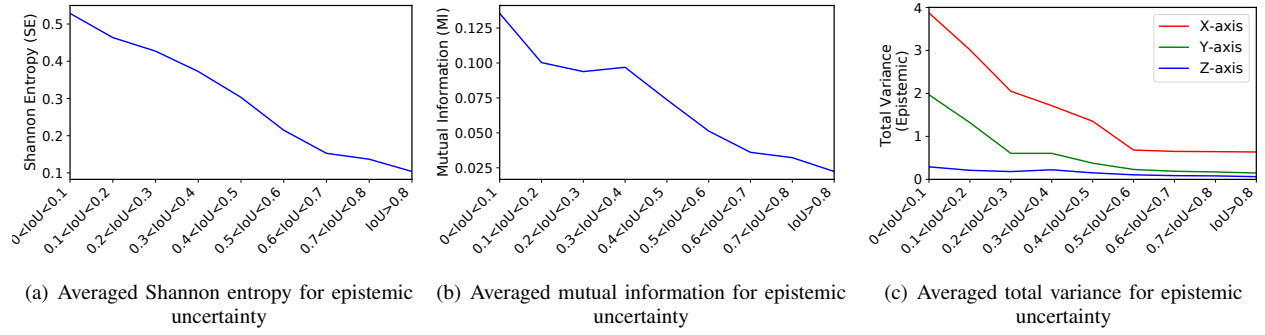


Fig. 3: Averaged epistemic uncertainty of predicted samples at different IoU intervals. The horizontal axis represents the increasing IoU values, and the vertical axis represents the uncertainty measurement. (a), (b): Shannon Entropy and Mutual Information estimates for epistemic classification uncertainty. (c): Total Variance for epistemic spatial uncertainty. The total variance in  $x$ ,  $y$ , and  $z$  axes were calculated.

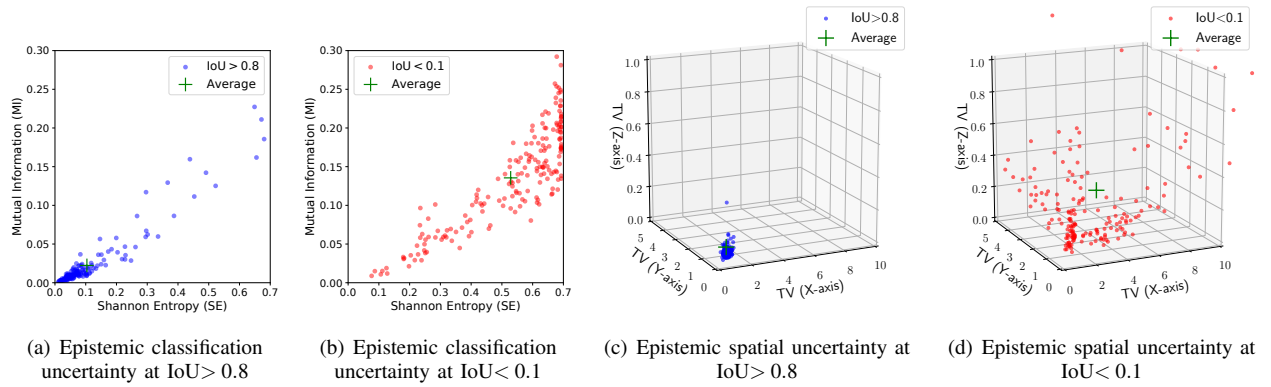


Fig. 4: The epistemic uncertainty estimates for each detection at  $\text{IoU} < 0.1$  and  $\text{IoU} > 0.8$ . (a), (b): classification uncertainty; (c), (d): 3D bounding box spatial uncertainty. The total variances in  $x$ ,  $y$ , and  $z$  axes were calculated.

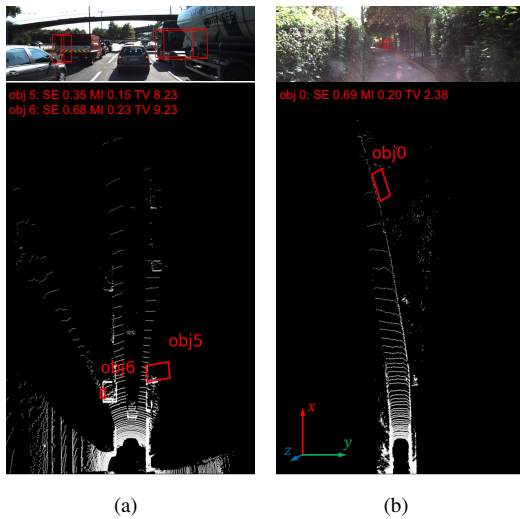


Fig. 5: Illustration of detections with high epistemic uncertainty: (a) big vehicles such as trucks; (b) ghost objects. A larger value indicates higher uncertainty. The camera images are used only for visualization purpose.

#### D. Understanding the Aleatoric Uncertainty in 3D Object Detection

Finally, we evaluated the aleatoric uncertainty, i.e. the uncertainty that captures the observation noises in Lidar point clouds. Here, we focused on the 3D bounding box regression whose uncertainty is quantified by  $\sigma^2$  (Sec. IV-B). In this work, we did not explicitly model the parameters to quantify the aleatoric classification uncertainty. We leave it as an interesting topic for the future research.

We demonstrated the mean value of total variance of aleatoric spatial uncertainty with regard to different IoU intervals, similar to the experiment in Sec. V-C. To do this, we summed up the variance  $\sigma^2$  of observation noises of an object in  $x$ ,  $y$ ,  $z$  axis, respectively. Fig. 6 shows that the aleatoric spatial uncertainty was little related to the detection accuracy, which was different from the epistemic uncertainty shown in Fig 3(c).

We further analyzed the behavior of aleatoric uncertainty with regard to the distance between the ego-vehicle and the detected vehicles. We used the **Pearson Correlation Coefficient (PCC)** to quantify the linear correlation between the distance and the total variance in  $x$ ,  $y$ ,  $z$  axes as well as the total variance of the whole covariance matrix, for detections in the test dataset. Tab. II shows the results. The aleatoric uncertainty was positively correlated with distance,

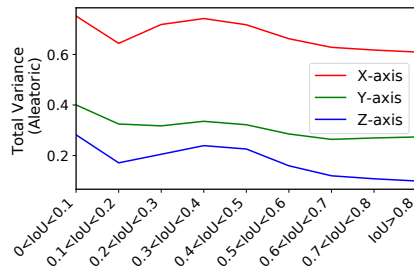


Fig. 6: Averaged total variance for aleatoric spatial uncertainty of predicted samples at different IoU intervals. The total variance in  $x$ ,  $y$ , and  $z$  axes were calculated.

indicating that a more distant vehicle was more difficult to be localized. This is due to the fact that the point clouds of an object become increasingly sparse at a large distance. Contrarily, the epistemic uncertainty showed little relationship with the distance. This conclusion was also supported by an exemplary sequence of vehicle detection shown in Fig. 7. As the car was leaving from the ego-vehicle, the aleatoric uncertainties in  $x$ ,  $y$ ,  $z$  axes were continuously increasing, whereas the epistemic uncertainty showed no tendency.

TABLE II: Pearson correlation coefficient between distance and spatial uncertainty

Network	X axis	Y axis	Z axis	All
Epistemic	-0.046	-0.030	0.202	-0.029
Aleatoric	<b>0.569</b>	<b>0.412</b>	<b>0.497</b>	<b>0.537</b>

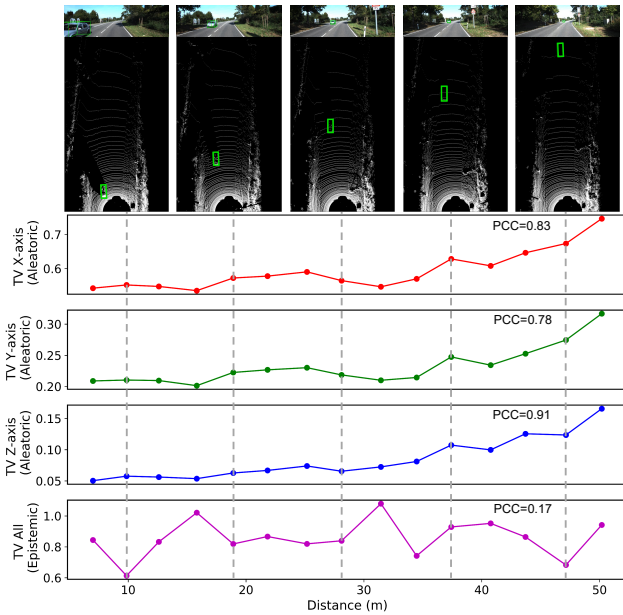


Fig. 7: An exemplary evolution of aleatoric and epistemic spatial uncertainty in sequential detections. PCC: Pearson correlation coefficient.

The observation noises were not only affected by distance, but by occlusion as well. We found that the corners of the bounding boxes which face directly towards the Lidar sensor consistently display smaller aleatoric spatial uncertainty than

the occluded corners. For instance, the red scores in Fig. 8(a) and Fig. 8(b) represent the sum of the spatial uncertainty of the vehicles' corners that face towards the Lidar sensor, and blue scores represent the uncertainty of occluded corners. These values were calculated by summing the observation noises  $\sigma^2$  of the front corners and back corners separately. For all detections in Fig. 8(a) and Fig. 8(b), the aleatoric spatial uncertainties of occluded corners were *consistently* higher than the ego-vehicle facing corners.

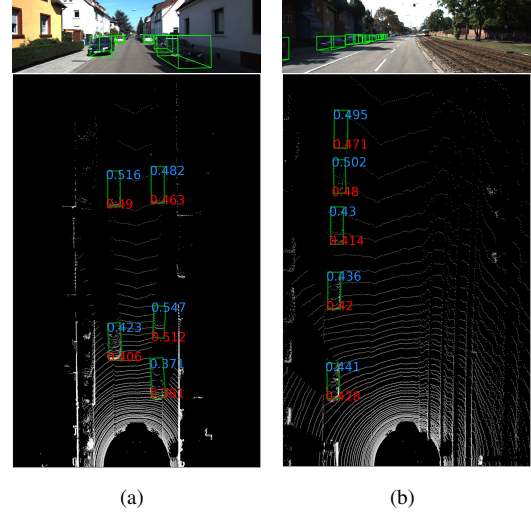


Fig. 8: Illustration of aleatoric spatial uncertainty in the corners of the bounding boxes that face towards ego-vehicle (marked in red) and occluded corners (marked in blue). The detections of the range  $x = 0 - 50$  m are shown.

## VI. CONCLUSIONS AND DISCUSSIONS

We have presented a probabilistic Lidar 3D vehicle detection network that captures reliable uncertainties in vehicle recognition and 3D bounding box regression tasks. Our proposed network models the epistemic uncertainty, i.e. the model uncertainty to describe data, by performing predictions several times with dropout. Our network also models the aleatoric uncertainty, i.e. the observation noises inherent in Lidar by adding an auxiliary output layer.

Epistemic and aleatoric uncertainties behave differently from each other. Experimental results showed that the epistemic uncertainty is associated with the detection accuracy. The network showed high epistemic uncertainty with samples that were different from the training dataset, such as ghost objects, big vehicles, or vehicles with abnormal bounding box regressions. Since the epistemic uncertainty displays the limitations of the vehicle detection network, it is highly valuable to be applied to efficiently querying the unseen samples to improve the model during the offline training phase (active learning). For example, when a vehicle detector trained on highways is deployed to urban areas, it is necessary to adapt the object detector in this new environment. By employing the epistemic uncertainty, the vehicle detector can be efficiently improved by actively querying objects such as pedestrians and cyclists which do not exist on highways.



Conversely, experiments showed that the aleatoric uncertainty is influenced by the detection distance and occlusion rather than detection accuracy, as distant vehicles or the occluded parts of vehicles contain high observation noises. In this way, the aleatoric uncertainty shows the sensor limitations and can be applied to improve the tracking of a vehicle position. Furthermore, we have showed that modeling the aleatoric uncertainty improved the detection performance by 1%–5%, indicating that it increased the model robustness to noisy data. Finally, computing the aleatoric uncertainty of a sample requires only one-time inference. Thus, modeling aleatoric uncertainty is useful for online deployment.

One limitation of our method is the computation cost when extracting the epistemic uncertainty, where the network needs to perform multiple feed-forward passes with dropout (0.3 fps on a Titan X GPU). This makes epistemic uncertainty infeasible for online autonomous driving. Finding the trade-off between the performance of epistemic uncertainty and the number of feed-forward passes is an open question for the further research.

In the future, we intend to explore uncertainties in different Lidar based object detection architectures such as one-stage detection pipeline [32]. Furthermore, we plan to investigate more factors that may influence aleatoric uncertainties, such as Lidar reflection rates and different bounding box encodings. Finally, we plan to apply our uncertainty estimation to active learning and object tracking to improve bounding box predictions, or to incorporate these uncertainty estimations as additional knowledge to the training phase (e.g. [33]).

#### ACKNOWLEDGMENT

We thank Zhongyu Lou and Florian Faion for their suggestions and inspiring discussions.

#### REFERENCES

- [1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.
- [2] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.
- [3] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3d bounding box estimation using deep learning and geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 5632–5640.
- [4] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 1513–1518.
- [5] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *Robotics: Science and Systems*, 2016.
- [6] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *arXiv preprint arXiv:1711.06396*, 2017.
- [7] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1355–1361.
- [8] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1019–1024.
- [9] A. Asvadi, L. Garrote, C. Premebeda, P. Peixoto, and U. J. Nunes, "Depthcn: vehicle detection using 3d-lidar and convnet," in *IEEE International Conference on Intelligent Transportation Systems*, 2017.
- [10] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.
- [11] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," *arXiv preprint arXiv:1711.08488*, 2017.
- [13] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," *arXiv preprint arXiv:1711.10871*, 2017.
- [14] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *arXiv preprint arXiv:1712.02294*, 2017.
- [15] Z. Wang, W. Zhan, and M. Tomizuka, "Fusing bird view lidar point cloud and front view camera image for deep object detection," *arXiv preprint arXiv:1711.06703*, 2017.
- [16] X. Du, M. H. Ang Jr, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles," *arXiv preprint arXiv:1803.00387*, 2018.
- [17] A. Pfeuffer and K. Dietmayer, "Optimal sensor data fusion architecture for object detection in adverse weather conditions," in *Proceedings of International Conference on Information Fusion*, 2018, pp. 2592 – 2599.
- [18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.
- [20] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.
- [21] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the sixth annual conference on Computational learning theory*. ACM, 1993, pp. 5–13.
- [22] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.
- [23] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," in *Advances in neural information processing systems*, 2016, pp. 4026–4034.
- [24] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.
- [25] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *International Conference on Machine Learning*, 2017, pp. 1183–1192.
- [26] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, 2017, pp. 5580–5590.
- [27] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.
- [28] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," *arXiv preprint arXiv:1710.06677*, 2017.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.
- [30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [31] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in Neural Information Processing Systems*, 2017, pp. 3584–3593.
- [32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [33] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.