Learning an Uncertainty-Aware Object Detector for Autonomous Driving

Gregory P. Meyer[1] and Niranjan Thakurdesai[1,2]

Abstract— The capability to detect objects is a core part of autonomous driving. Due to sensor noise and incomplete data, perfectly detecting and localizing every object is infeasible. Therefore, it is important for a detector to provide the amount of uncertainty in each prediction. Providing the autonomous system with reliable uncertainties enables the vehicle to react differently based on the level of uncertainty. Previous work has estimated the uncertainty in a detection by predicting a probability distribution over object bounding boxes. In this work, we propose a method to improve the ability to learn the probability distribution by considering the potential noise in the ground-truth labeled data. Our proposed approach improves not only the accuracy of the learned distribution but also the object detection performance.

I. INTRODUCTION

A crucial component of autonomous driving is the ability to detect and localize the surrounding objects. To accomplish this task, autonomous vehicles are equipped with various sensors including cameras and LiDARs. A wealth of deep learning based approaches have been proposed to perform 3D object detection using these sensors [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Given the limited sensory information, it is unrealistic to expect any detector to flawlessly classify and localize every actor in all situations. Therefore, it is important for a detector to provide to the autonomous system its level of uncertainty in its predictions and the uncertainties need to be reliable.

Previously proposed detectors provide the uncertainty for the object classification by predicting a categorical distribution over the classes of objects, but the majority of the previous work does not provide an uncertainty for the localization of the object. Following [12], a few methods have been proposed to estimate the localization uncertainty by learning a probability distribution over object bounding boxes given the sensor data [10], [13], [14]. These methods learn the parameters of the probability distribution by maximizing the likelihood of a ground-truth label. In this work, we show that maximizing the likelihood, or equivalently minimizing the negative log likelihood, has undesirable properties for learning uncertainties, which could result in numerical instability and overfitting. This is due to the likelihood being maximized when the distribution becomes a Dirac delta function with zero uncertainty and infinity probability density for a label.

Instead of assuming the labels are samples from a distribution and training the model to maximize their likelihood, we assume each label is itself a distribution, and we train the model by minimizing the Kullback-Leibler (KL) divergence between the predicted distribution and the label distribution. We demonstrate that the KL divergence resolves the issues with the negative log likelihood and improves the performance of the model.

Assuming each label is a distribution implies the labels contain some amount of uncertainty, which is likely to be the case with human annotation. As a result, we need a way to determine the amount of noise in each label, which is a non-trivial problem. In this paper, we propose a heuristic to estimate the uncertainty in a label by considering both the data and how it is annotated.

With our proposed approach, we are able to improve the performance of a state-of-the-art object detector [10] by only modifying the loss function used during training. Our approach significantly improves the performance of rare objects, which we believe is the result of the KL divergence being less prone to overfitting potentially noisy labels. Furthermore, we see an improvement in the accuracy of the predicted distribution.

In the following sections, we discuss the previous work related to estimating uncertainty with deep neural networks (Section II), review an approach for learning the uncertainty in a detection and propose a modification that assumes the ground-truth labels are noisy (Section III-A), propose a way to approximate the uncertainty in a label (Section III-B), and present experimental results for our proposed method on a large-scale autonomous driving dataset (Section IV).

II. RELATED WORK

A. Predicting Uncertainty with Neural Networks

Neural networks tend to make over-confident predictions and do not provide reliable estimates of uncertainty in their predictions until recently. Bayesian modeling provides a theoretically grounded and practical framework for representing uncertainty in neural networks [15], [16]. Applied to computer vision tasks, Kendall and Gal [12] divided uncertainties in the Bayesian framework into two types, aleatoric and epistemic. Aleatoric or data uncertainty is due to noise inherent in the data, so it cannot be reduced by increasing the size of the training set. It arises from sensor noise, incomplete data, class ambiguity, and label noise [12], [17]. Aleatoric uncertainty is modeled by making the outputs of a neural network probabilistic, i.e. predicting a probability distribution instead of a point estimate. Epistemic or model uncertainty is due to uncertainty in the model parameters. It captures our ignorance about the model most suitable to explain our training data [16]. High epistemic uncertainty

[1]Uber Advanced Technologies Group
[2]Georgia Institute of Technology
Correspondence to gmeyer@uber.com

means there may be another model which explains the data better. It is important to model in the case of limited training data, but it can be reduced by collecting more training data [12]. Malinin and Gales [17] modeled a third type of uncertainty called distributional uncertainty arising from a mismatch between the training and test distributions. In this case, the model is unfamiliar with an example if it is out-of-distribution and hence, cannot make confident predictions. They proposed a new framework called Prior Networks to explicitly predict distributional uncertainty and separate it from data uncertainty. In this work, we focus on modeling data uncertainty since it cannot be reduced with a larger dataset. It is important for an autonomous vehicle to understand the uncertainty in its detections due to limited sensor data so that it can plan accordingly, e.g. slow down the vehicle to collect more data.

The methods for predicting uncertainty in neural networks can be broadly divided into two categories, sampling-based and sampling-free. One class of sampling-based approaches uses variational inference over the neural network weights [18], [19]. This involves approximating the posterior distribution of the network weights given the dataset, which is difficult to evaluate, with a more tractable distribution [18], [12]. Epistemic uncertainty is estimated by sampling this approximate distribution in order to draw a set of weights; each sample will result in a set of predictions which are combined to compute the epistemic uncertainty. Gal and Ghahramani [20], [21] introduced a technique for modeling epistemic uncertainty called Monte Carlo dropout which uses multiple stochastic forward passes of the neural network with dropout [22]. Dropout is a popular technique used during training as a form of regularization; however, with Monte Carlo dropout, it is used during inference as well to estimate epistemic uncertainty. All of these methods rely on sampling the output several times which is computationally expensive; as a result, they are unsuitable for real-time autonomous systems. On the other hand, sampling-free methods such as [12], [23] are computationally efficient and able to predict uncertainties in a single forward pass.

B. Uncertainty Estimation in Object Detection

A variety of methods have been proposed for 3D object detection in the context of autonomous driving [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Techniques for predicting localization uncertainty in object detection have been studied recently. Jiang *et al.* [24] predicted the intersection-over-union (IoU) between the 2D ground-truth and predicted bounding boxes as a measure of uncertainty. Along the lines of [12], [13] and [14] captured the epistemic and aleatoric uncertainties in 3D object detection using Monte Carlo dropout [20] and MAP inference respectively. Furthermore, Meyer *et al.* [10] estimated aleatoric uncertainty in 3D object detection by predicting a multimodal distribution over bounding boxes. In this work, we modify the way [10] learns the probability distribution, resulting in better uncertainty estimates and detection performance.

III. PROPOSED METHOD

In this work, we improve upon LaserNet [10] which is a LiDAR-based probabilistic object detector. LaserNet estimates the uncertainty in its detections by predicting a probability distribution over bounding boxes for each object. In the following sections, we will review how the previous work learns a probability distribution and propose an alternative approach.

A. Learning Uncertainty

To estimate the uncertainty in a detection, Meyer *et al.* [10] model the distribution of bounding box corners. In [10], the corners are represented as a D-dimensional vector where D is the number of corners multiplied by the dimensionality of each corner. The dimensions are assumed to be drawn from independent univariate Laplace distributions; as a result, the probability density of a bounding box is defined as follows:

$$\prod_{k=1}^{D} p(y_k|\hat{y}_k, \hat{b}_k) = \prod_{k=1}^{D} \frac{1}{2\hat{b}_k} \exp\left(-\frac{|y_k - \hat{y}_k|}{\hat{b}_k}\right) \quad (1)$$

where $\hat{y}_k \in \mathbb{R}$ and $\hat{b}_k \in \mathbb{R}^+$ specify the mean and scale of the distribution along each dimension, respectively. Given a LiDAR sweep denoted by x, LaserNet is trained to predict a distribution for each of the M objects in the sweep,

$$\left\{\{\hat{y}_{1k}, \hat{b}_{1k}\}_{k=1}^{D}, \cdots, \{\hat{y}_{Mk}, \hat{b}_{Mk}\}_{k=1}^{D}\right\} = F_\theta(x) \quad (2)$$

where F_θ represents a convolutional neural network parameterized by θ, as well as, all pre- and post-processing [10]. Following [12], the network is trained by minimizing the negative log likelihood over the training set,

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{D} \mathcal{L}_{NLL}(y_{ijk}, \hat{y}_{ijk}, \hat{b}_{ijk}) \quad (3)$$

where

$$\mathcal{L}_{NLL}(y, \hat{y}, \hat{b}) = \log 2\hat{b} + \frac{|y - \hat{y}|}{\hat{b}} \quad (4)$$

is the negative log likelihood of a Laplace distribution, y_{ijk} is the ground-truth, $\hat{y}_{ijk}$ is the predicted mean, and $\hat{b}_{ijk}$ is predicted uncertainty for the kth dimension of the jth object bounding box in the ith LiDAR sweep. The model is also trained to predict class probabilities for each object which we omit for brevity. The partial derivatives of the negative log likelihood with respect to the predictions are

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \hat{y}} = -\frac{\text{sgn}(y - \hat{y})}{\hat{b}} \quad (5)$$

and

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \hat{b}} = \frac{1}{\hat{b}}\left(1 - \frac{|y - \hat{y}|}{\hat{b}}\right). \quad (6)$$

A problem with minimizing the negative log likelihood is that as $|y - \hat{y}| \to 0$,

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \hat{b}} \to \frac{1}{\hat{b}}, \quad (7)$$

and the derivatives for both $\hat{y}$ and $\hat{b}$ explode as $\hat{b} \to 0$. In practice, gradient clipping can be employed to prevent

(a) Negative Log Likelihood   (b) KL Divergence ($b = 0.2$)   (c) KL Divergence ($b = 0.4$)
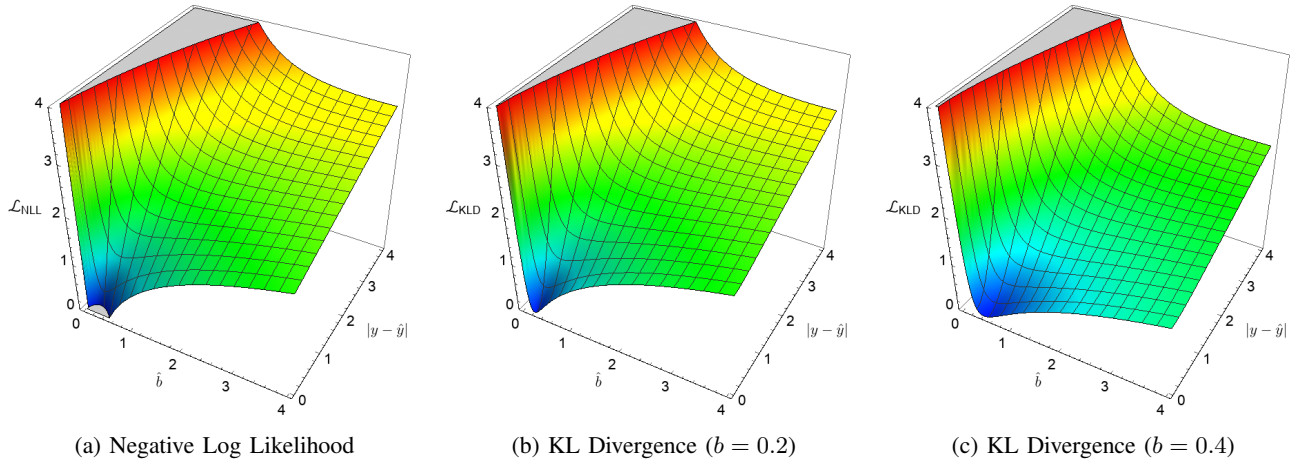
Fig. 1: The negative log likelihood of a Laplace distribution and the KL divergence of Laplace distributions as a function of the prediction error, $|y - \hat{y}|$, and uncertainty, $\hat{b}$. When the error is large, the functions behave similarly. However, near their minimum, the negative log likelihood goes sharply to negative infinity, whereas the KL divergence goes to zero. The shape of the KL divergence is controlled by the label uncertainty, b.

the model from diverging, but the model is still prone to overfitting since the magnitude of the gradient does not tend to zero as the prediction error goes to zero, $|y - \hat{y}| \to 0$.

The issue with the previous approach is that it assumes the labels are free of noise; therefore, (4) is minimized only when the prediction has no error and uncertainty, i.e. $|y - \hat{y}| = 0$ and $\hat{b} = 0$. Instead, let us assume each label contains some amount of noise and is itself a Laplace distribution. As a result, we have the following distributions for each dimension of the bounding box:

$$p(y_k^*|y_k, b_k) = \frac{1}{2b_k} \exp\left(-\frac{|y_k^* - y_k|}{b_k}\right) \quad (8)$$

and

$$q(y_k^*|\hat{y}_k, \hat{b}_k) = \frac{1}{2\hat{b}_k} \exp\left(-\frac{|y_k^* - \hat{y}_k|}{\hat{b}_k}\right) \quad (9)$$

where y_k^* is the unknown true position of the bounding box, $p(y_k^*|y_k, b_k)$ represents our uncertainty in the label, and $q(y_k^*|\hat{y}_k, \hat{b}_k)$ represents our uncertainty in the prediction. Assuming we know the distribution of each label, then we can train the model by minimizing the Kullback-Leibler (KL) divergence between each of the label distributions and the predicted distributions,

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{D} \mathcal{L}_{KLD}(y_{ijk}, \hat{y}_{ijk}, b_{ijk}, \hat{b}_{ijk}) \quad (10)$$

where

$$\mathcal{L}_{KLD}(y, \hat{y}, b, \hat{b}) = \log\frac{\hat{b}}{b} + \frac{b\exp\left(-\frac{|y-\hat{y}|}{b}\right) + |y - \hat{y}|}{\hat{b}} - 1 \quad (11)$$

is the KL divergence of two Laplace distributions [25]. The partial derivatives of the KL divergence with respect to the predictions are

$$\frac{\partial\mathcal{L}_{KLD}}{\partial\hat{y}} = -\frac{\text{sgn}(y - \hat{y})}{\hat{b}}\left(1 - \exp\left(-\frac{|y - \hat{y}|}{b}\right)\right) \quad (12)$$

and

$$\frac{\partial\mathcal{L}_{KLD}}{\partial\hat{b}} = \frac{1}{\hat{b}}\left(1 - \frac{b\exp\left(-\frac{|y-\hat{y}|}{b}\right) + |y - \hat{y}|}{\hat{b}}\right). \quad (13)$$

As $|y - \hat{y}| \to 0$ and $b > 0$,

$$\frac{\partial\mathcal{L}_{KLD}}{\partial\hat{y}} \to 0 \quad (14)$$

and

$$\frac{\partial\mathcal{L}_{KLD}}{\partial\hat{b}} \to \frac{1}{\hat{b}}\left(1 - \frac{b}{\hat{b}}\right) \quad (15)$$

since the exponential terms in (12) and (13) go to one. As $|y - \hat{y}| \to 0$ and $\hat{b} \to b$, the derivatives for both $\hat{y}$ and $\hat{b}$ go to zero, which is a desirable property for a loss function. However, if the noise in the label is set to zero, $b = 0$, the exponential terms in (12) and (13) become zero regardless of the prediction error, and the derivatives of the KL divergence become identical to those of the negative log likelihood.

A comparison of the loss functions is shown in Fig. 1. The shape of the loss surface is similar for both the negative log likelihood and the KL divergence when the prediction error is large since the exponential terms in (12) and (13) go to zero as the error increases. However, the behavior of the loss functions near the minimum differs significantly. The negative log likelihood goes to negative infinity when the prediction error and uncertainty goes to zero, whereas the KL divergence becomes zero when the predicted distribution matches the label distribution. Moreover, with the KL divergence, increasing the uncertainty of a label reduces the loss for that label as long as the prediction uncertainty does not underestimate the label uncertainty. Refer to the Appendix for a proof. This property of the KL divergence can be leveraged to prevent the model from overfitting to noisy labels. Next, we discuss how we approximate the uncertainty of a label.

(a) 0.16 IoU

(b) 0.49 IoU

(c) 0.59 IoU

(d) 0.72 IoU
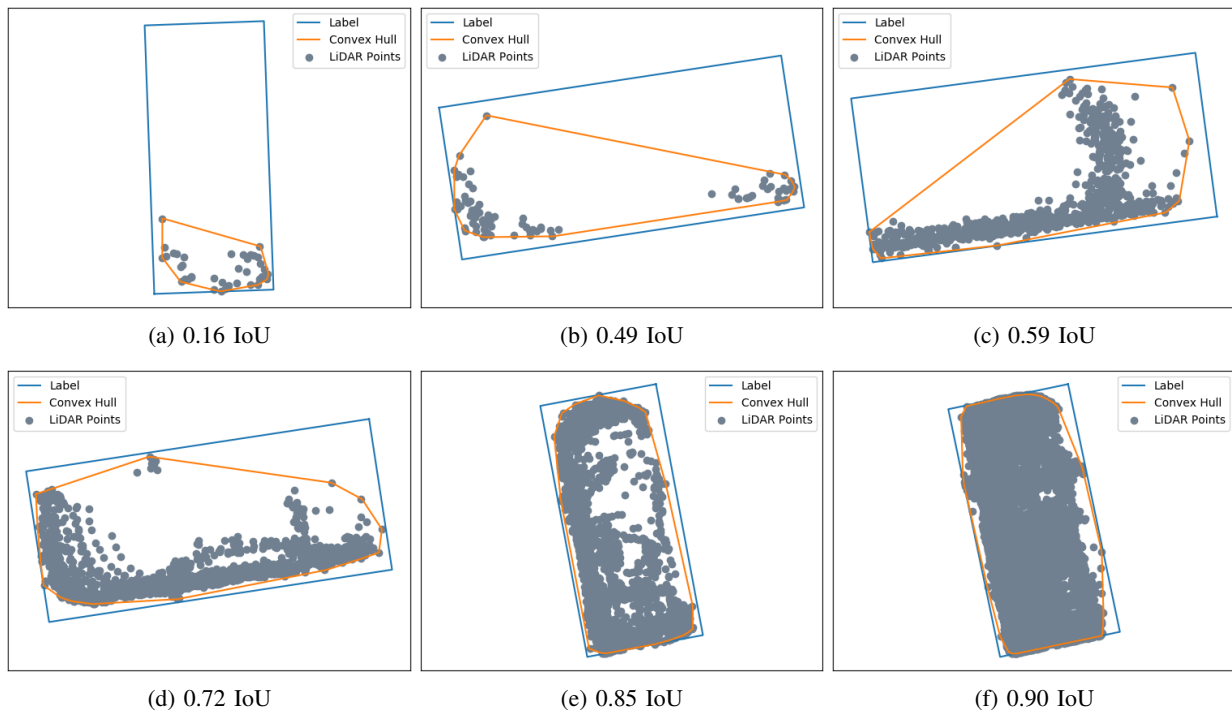
(e) 0.85 IoU

(f) 0.90 IoU

Fig. 2: Examples of labeled vehicles that motivate the use of the IoU between the label bounding box and the convex hull of aggregated LiDAR points to estimate the noise in the annotation. We speculate that there is an inverse non-linear relationship between the uncertainty of a label and the IoU between the label and the convex hull.

B. Estimating Label Uncertainty

To utilize KL divergence, we need to identify the distribution of every label. One approach could be to have an annotator provide the distribution as he/she labels the data. However, directly annotating the uncertainty of the label may be challenging and time-consuming. Another possibility is to draw samples from the underlying label distribution by having multiple annotators provide labels for the same data, and then estimate the distribution from the samples. Unfortunately, this process would be expensive and there is no guarantee that the label samples are independent and identically distributed (i.i.d.). In this work, we explore ways to approximate the uncertainty in the labeled data.

The simplest approximation would be to assume a Laplace distribution with a constant uncertainty for every label, where the label itself defines the mean y_{ijk} and a single scale b_{ijk} is shared across all labels. Depending on the scale we choose, we may significantly underestimate or overestimate the noise in a label. To obtain a better approximation of the label uncertainty, we need to consider the data and how it is annotated. In the case of object detection with LiDAR, annotators label objects within each LiDAR sweep and typically have access to previous and future sweeps. For the dataset used in our experiments (Section IV), each LiDAR sweep is transformed from the sensor's coordinate system into a global coordinate frame which accounts for the ego-motion of the autonomous vehicle. Labels are provided as oriented rectangles in the x-y plane (bird's eye view) of the global coordinate frame. To estimate the noise in a label,

we start by accumulating all the LiDAR points inside the label across all the sweeps where the label is visible. To account for motion of the object across sweeps, we assume the object is rigid and compute its translation and rotation from the current sweep to an arbitrary reference sweep. For example, let p_{ij} be a point inside the ith label and captured during the jth sweep. To transform p_{ij} from the jth to the kth sweep, we perform the following operation:

$$p_{ij}^k = \mathbf{R}_z(\theta_{ik} - \theta_{ij})\left[p_{ij} - c_{ij}\right] + c_{ik} \tag{16}$$

where c_{ij} and c_{ik} are the centers of the label in the jth and kth sweeps, θ_{ij} and θ_{ik} are the orientations of the label in their respective sweeps, and $\mathbf{R}_z(\theta)$ is a rotation matrix about the z-axis parameterized by θ. Afterwards, we determine the convex hull of the accumulated points and compute the intersection-over-union (IoU) between the convex hull and the label. We speculate that the higher the IoU, the smaller the ambiguity in the label, as illustrated in Fig. 2. For each label, we use an exponential function of the form

$$y = \alpha \exp(-\beta \cdot x) + \gamma \tag{17}$$

to map from the IoU between the label and the convex hull to the uncertainty of the label, where α, β, and γ are hyper-parameters to control the mapping. In the next section, we evaluate the effect of KL divergence on the predicted distribution where our heuristic is used to estimate the scale of the label distribution.

TABLE I: Loss Function and Label Uncertainty Experimental Results

Loss Function	Label Uncertainty (m)			Average Precision (%)			Mean Average Precision (%)
	Vehicle	Bike	Pedestrian	Vehicle	Bike	Pedestrian	
$\mathcal{L}_{NLL}$	0.00	0.00	0.00	85.34	61.93	80.37	75.88
$\mathcal{L}_{KLD}$	0.50	0.50	0.50	69.64	55.35	76.73	67.24
$\mathcal{L}_{KLD}$	0.25	0.25	0.25	80.23	60.10	80.66	73.66
$\mathcal{L}_{KLD}$	0.10	0.10	0.10	85.20	63.32	82.08	76.87
$\mathcal{L}_{KLD}$	0.05	0.05	0.05	85.91	62.81	81.90	76.87
$\mathcal{L}_{KLD}$	0.01	0.01	0.01	85.48	63.78	82.37	77.21
$\mathcal{L}_{KLD}$	$1.00 \rightarrow 0.05 \rightarrow 0.01$	$1.00 \rightarrow 0.05 \rightarrow 0.01$	$1.00 \rightarrow 0.05 \rightarrow 0.01$	**86.05**	63.71	82.02	77.26
$\mathcal{L}_{KLD}$	$0.50 \rightarrow 0.05 \rightarrow 0.01$	$0.25 \rightarrow 0.05 \rightarrow 0.01$	$0.10 \rightarrow 0.05 \rightarrow 0.01$	86.04	64.29	82.43	77.59
$\mathcal{L}_{KLD}$	$1.00 \rightarrow 0.05 \rightarrow 0.01$	$0.50 \rightarrow 0.05 \rightarrow 0.01$	$0.25 \rightarrow 0.05 \rightarrow 0.01$	85.87	64.58	82.41	77.62
$\mathcal{L}_{KLD}$	$2.00 \rightarrow 0.05 \rightarrow 0.01$	$1.00 \rightarrow 0.05 \rightarrow 0.01$	$0.50 \rightarrow 0.05 \rightarrow 0.01$	85.74	**64.82**	**82.61**	**77.72**

IV. EXPERIMENTS

Our proposed method is evaluated on the ATG4D dataset which contains 5,000 sequences for training and 500 for validation. The training sequences are sampled at 10 Hz, and the validation sequences are sampled at 0.5 Hz. The training set contains 1.2 million sweeps, and the validation set has 5,969 sweeps. A Velodyne 64E LiDAR was used to capture all of the sweeps in the dataset.

For all of our experiments, we use LaserNet [10] as our base detector and only modify the loss function used to learn the probability distribution over bounding boxes (Equation (9) in [10]). We change the loss function from the negative log likelihood of a Laplace distribution to the KL divergence of Laplace distributions. All other settings remain unchanged; please refer to [10] for details.

We experimented with two ways to approximate the label uncertainty: a constant uncertainty for all labels and a heuristically obtained label uncertainty. In either case, the label distribution is assumed to be a Laplace distribution, and the mean of the distribution, y_{ijk}, is defined by the label. When using a fixed uncertainty, a single scale parameter, b_{ijk}, is used for all labels. We performed a variety of experiments with different values for the scale. When utilizing the heuristic, we estimate the noise in each label. For each label in the training set, we compute the IoU between the label and the convex hull of the LiDAR points observed within the label using the approach described in Section III-B. A histogram of IoU values is depicted in Fig. 3 and shows there is a diverse set of labeled data in the dataset. Equation (17) is used to map from the IoU to b_{ijk} for each label, and we experimented with different values for α, β, and γ.

A. Detection Evaluation

Following the previous work, we evaluate detections within the front $90°$ field of view of the LiDAR and up to 70 meters away from the sensor. The average precision (AP) metric is used to measure detection performance. To be considered a true positive, a vehicle detection needs to achieve an IoU of 0.7 with a ground-truth bounding box, and for bike and pedestrian detections, an IoU of 0.5 is required.

Detection performance for the various experiments are presented in Table I. The values of b_{ijk} for each experiment are listed in the table under "Label Uncertainty." When a constant uncertainty is used for all labels, only a single value
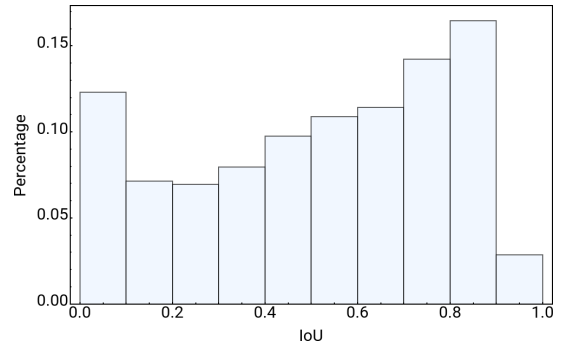

Fig. 3: The distribution of IoU values in the ATG4D dataset.

TABLE II: Comparisons to State-of-the-Art Methods

| Method	Input	Average Precision (%)		
		Vehicle	Bike	Pedestrian
Proposed Method	LiDAR	85.74	64.82	82.61
PIXOR [7]	LiDAR	80.99	-	-
PIXOR++ [26]	LiDAR	82.63	-	-
ContFuse [8]	LiDAR	83.13	57.27	73.51
LaserNet [10]	LiDAR	85.34	61.93	80.37
ContFuse [8]	LiDAR+RGB	85.17	61.13	76.84
LaserNet++ [11]	LiDAR+RGB	**86.23**	**65.68**	**83.42**

is specified for each class. When using the heuristic to estimate the uncertainty for each label, three values are shown with the notation $b_{0.0} \rightarrow b_{0.5} \rightarrow b_{1.0}$ which correspond to the value of b_{ijk} when the IoU between the label and the convex hull is zero, one-half, and one, respectively. These values are used to determine the parameters α, β, and γ.

By replacing the negative log likelihood with the KL divergence, we obtain an improvement in performance, as long as, the label uncertainty is not overestimated (row two and three in Table I). By approximating the label uncertainty with our proposed heuristic, we see a considerable improvement in the rarest class, the bike class. We believe the gain in performance is due to our method reducing the amount of overfitting to potentially noisy labels, specifically, vehicle labels which are by far the most commonly occurring class.

A comparison between our proposed method (last row in Table I) and recent state-of-the-art object detectors is shown in Table II. Our approach outperforms the previous methods that only utilize LiDAR data. Furthermore, by only changing the loss function, we observe a similar gain in performance as adding an additional sensing modality (LaserNet++ [11]).
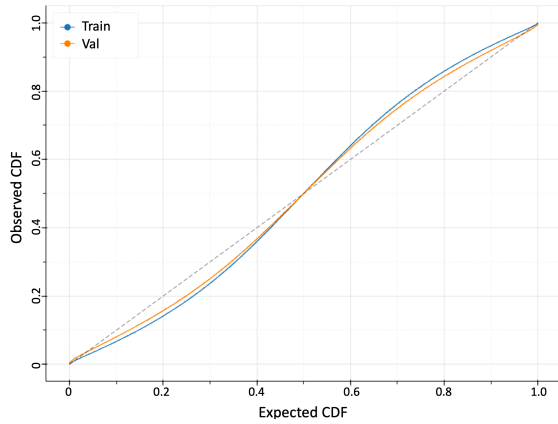
(a) LaserNet [10]
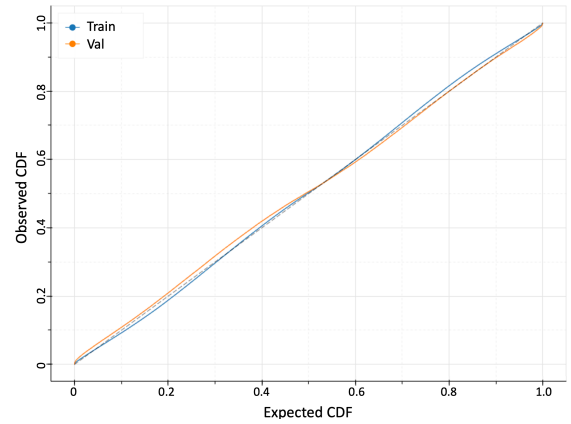
(b) Proposed Method

Fig. 4: Calibration plots showing the reliability of the predicted distribution from LaserNet [10] and our proposed method. A perfectly calibrated model would follow the dashed line (the observed CDF matches the expected CDF at all probabilities).

B. Uncertainty Evaluation

To evaluate the predicted probability distribution, we compare the expected cumulative distribution function (CDF) to the observed CDF. For each ground-truth label from a particular class, we calculate its standard score given the parameters of the predicted distribution. A cumulative histogram is created from the standard scores, and it is compared to the CDF of a standard Laplace distribution. The resulting plots for the vehicle class are shown in Fig. 4. Compared to [10], our proposed approach is capable of learning a better calibrated probability distribution.

V. CONCLUSION

In this work, we presented an approach to learn the uncertainty of a detection given the sensor data while being aware of the uncertainty in the labeled data. Our proposed method improves a state-of-the-art probabilistic object detector [10] in terms of both object detection performance and the accuracy of the learned probability distribution. The sole difference is that [10] makes the implicit assumption that labels are noise-free and uses the negative log likelihood to learn the distribution, whereas our proposed method uses a heuristic to estimate the uncertainty in a label and learns the distribution by minimizing the KL divergence. We believe the improvement is due to the KL divergence being more well-behaved during training than the negative log likelihood.

APPENDIX

In this section, we provide a proof for the claim made in Section III-A.

Claim 1. *The following inequality holds for all $y, \hat{y} \in \mathbb{R}$:*

$$\log \frac{\hat{b}}{b_1} + \frac{b_1 \exp\left(-\frac{|y-\hat{y}|}{b_1}\right) + |y - \hat{y}|}{\hat{b}} - 1 > \\ \log \frac{\hat{b}}{b_2} + \frac{b_2 \exp\left(-\frac{|y-\hat{y}|}{b_2}\right) + |y - \hat{y}|}{\hat{b}} - 1 \quad (18)$$

when $\hat{b} \geq b_2 > b_1 > 0$.

Proof. The inequality is equivalent to

$$f_1(x) = \log \frac{b_2}{b_1} + \frac{b_1 \exp\left(-\frac{1}{b_1}x\right) - b_2 \exp\left(-\frac{1}{b_2}x\right)}{\hat{b}} > 0 \quad (19)$$

where $x = |y - \hat{y}| \geq 0$. To prove the inequality, we will utilize the mean value theorem; therefore, we need to show that $f_1(0) > 0$ and $f_1'(x) \geq 0$ when $x \geq 0$. The first derivative of $f_1(x)$ is

$$f_1'(x) = \frac{1}{\hat{b}} \left(\exp\left(-\frac{1}{b_2}x\right) - \exp\left(-\frac{1}{b_1}x\right) \right). \quad (20)$$

The function $\exp(-x)$ monotonically decreases; therefore, $\exp(-x_1) > \exp(-x_2)$ when $x_2 > x_1$. Since $1/b_1 > 1/b_2$ and $\hat{b} > 0$, $f_1'(x) \geq 0$ when $x \geq 0$. Next, we need to demonstrate that

$$f_1(0) = \log \frac{b_2}{b_1} + \frac{b_1 - b_2}{\hat{b}} > 0. \quad (21)$$

Since $\hat{b} \geq b_2 > b_1 > 0$,

$$0 > \frac{b_1 - b_2}{\hat{b}} \geq \frac{b_1 - b_2}{b_2}. \quad (22)$$

Therefore, if

$$\log \frac{b_2}{b_1} + \frac{b_1 - b_2}{b_2} > 0 \quad (23)$$

or equivalently

$$f_2(\rho) = \log \rho + \frac{1}{\rho} - 1 > 0 \quad (24)$$

where $\rho = b_2/b_1 > 1$, then $f_1(0) > 0$. Again leveraging the mean value theorem, we can show that $f_2(\rho) > 0$ when $\rho > 1$. The first derivative of $f_2(\rho)$ is

$$f_2'(\rho) = \frac{1}{\rho} - \frac{1}{\rho^2}. \quad (25)$$

When $\rho > 1$, $\rho^2 > \rho$ and $f_2'(\rho) > 0$. Since $f_2(1) = 0$, $f_2(\rho) > 0$ when $\rho > 1$. As a result, $f_1(0) > 0$ which completes the proof. □

REFERENCES

[1] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proceedings of Robotics: Science and Systems (RSS)*, 2016.

[2] 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 (CVPR)*, 2017.

[3] Y. Zhou and O. 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)*, 2018.

[4] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[7] 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 (CVPR)*, 2018.

[8] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[9] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3D bounding box estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "LaserNet: An efficient probabilistic 3D object detector for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3D object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[12] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[13] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3D vehicle detection," in *Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[14] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time LiDAR 3D object detection," *arXiv preprint arXiv:1809.05590*, 2018.

[15] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.

[16] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[17] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.

[18] A. Graves, "Practical variational inference for neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

[19] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[20] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[21] ——, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2016.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] S. Choi, K. Lee, S. Lim, and S. Oh, "Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[24] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[25] G. P. Meyer, "An alternative probabilistic interpretation of the Huber loss," *arXiv preprint arXiv:1911.02088*, 2019.

[26] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.