Calibrating Uncertainty Estimations for Bayesian Deep Object Detectors

Undergraduate Thesis Final Report

Division of Engineering Science, University of Toronto

Student: Yingxue Wang

Student Number: 1002177477

Supervisor: Steven Waslander

April 10, 2024

Abstract

Reliable uncertainty estimation is crucial for deep learning models in safety-critical scenarios such as autonomous driving. Recently, many methods have been proposed to model uncertainties in deep learning based object detectors. In this thesis, we identify such uncertainty miscalibration problems in a Bayesian 2D object detection network, and propose to use two practical methods, temperature scaling and isotonic regression, to significantly reduce errors in uncertainty calibration. Our analysis and experiments provide detailed explanation of the proposed methods and evaluation metrics, and was shown to produce reliable, well-calibrated uncertainties.

Acknowledgements

I would like to thank my supervisor, Steven Waslander, for his guidance and informative feedback through each stage of the thesis process.

I would like to acknowledge Ali Harahkeh, PhD Candidate at UTIAS, for inspiring my interest in the path of my project, and sharing his code, instrumental insights and suggestions. For this I am extremely grateful.

Contents

List of Figures

List of Tables

1 Introduction

In safety-critical machine learning applications such as autonomous driving and medical diagnosis, it is crucial to have high and accurate confidence scores for their predictions. Accurate confidence estimates can assist safe decision-making while inaccurate estimates can be misleading and result in severe consequences. Specifically, in an object detection system, the estimation of both the category to which an object belongs, and its spatial location and extent, often expressed as the tightest fitting bounding box, are required to obtain a reliable prediction. The existing state-of-art uncertainty estimation methods, including **sampling free** [1], **anchor redundancy** [1], **black box** [2] and **Bayesian Inference** [3], may not always produce perfectly calibrated estimates. For example, a 90% credible interval may not contain the true outcome 90% of the time. Therefore, calibrated confidence estimates are also important for model interpretability, as it provide a valuable extra bit of information to establish trustworthiness with the user, especially for neural networks. It is therefore important to develop calibration techniques for uncertainty predictions.

Calibration has been extensively studied in the weather forecasting literature [4]. However, these techniques tend to be specialized and difficult to generalize beyond applications in climate science. **Isotonic regression** [5], **Temperature scaling** [6] and **Standard Deviation Scaling** [7] were showed to be effective in calibrating uncertainty predictions of probabilistic deep learning models for classification tasks by adjusting the variance predictions. However, for regression tasks such as bounding box regression, the problem is harder to define and solve. In this case, we would like to match the output distribution which is continuous over a possible prediction interval, with the ground truths via post-processing. A few attempts of extending recalibration methods for to regression tasks were made by [8] and [5], to improve the accuracy of 3D LiDAR object detection predictions, but experience some defects. Variance scaling methods may fail to achieve well-calibrated uncertainties if the assumed probability distribution significantly differs from the true distribution. Isotonic regression changes the output probability distribution, making it less interpretable and applicable. Therefore, it lacks the capability of differentiating a completely uncorrelated empirical uncertainty from an informative one.

In this thesis, we first identify uncertainty miscalibration problems of a current Bayesian deep neural object detectors, *BayesOD* [3] via calibration plots. Then we attempt to apply the aforementioned temperature scaling and isotonic regression techniques to demonstrate the performance of the re-calibration models and systematically study their robustness on several datasets. The calibration quality was evaluated using *Calibration Plot* or *Reliability Curve* and *Expected Calibration Error (ECE)* [6]. The success of this thesis would contribute to general probabilistic deep learning methods and improve the performance on time series forecasting.

2 Background

Uncertainty modeling and estimation in deep neural network is a critical problem and receives growing attention from machine learning community. There are two major sources of uncertainties one can model, **aleatoric uncertainty** and **epistemic uncertainty** [9]. Aleatoric uncertainty is also referred as statistical uncertainty, and captures noises inherently within the observations. This is due to the fact that we cannot measure sufficiently with our currently available measurement devices, resulting in uncertainty which cannot be reduced even if more data were to be collected. Furthermore, aleatoric uncertainty can be categorized into homoscedastic uncertainty and heteroscedastic uncertainty, corresponding to uncertainty that stays constant for different inputs, and uncertainty depending on inputs. Data-dependent uncertainty like heteroscedastic uncertainty is especially important for computer vision applications. For example, in the object detection cases, obscured/blocked objects in the image are expected to result in less confident prediction, while objects that have clear boundary result in low uncertainty. Epistemic uncertainty accounts for uncertainties within the model parameters, and is sometimes referred as model uncertainty. It accounts for the ignorance about which source generated the collected data. For example, epistemic uncertainty may increase when the model neglects certain effects, or when particular data has been deliberately hidden. Therefore, it can be explained away given sufficient amount of data. Consequently, out-of-distribution examples can be identified with epistemic uncertainty, but not with aleatoric uncertainty alone.

In real-world applications, such as a self-driving car that uses a neural network to detect pedestrians and other obstruction, the network should provide a calibrated confidence measure in addition to its predictions. Control should be passed on human drivers when the confidence of a detected result is low. In other words, the probability associated with the predicted class label should reflect its ground truth correctness likelihood. Note that calibration is an orthogonal concern to accuracy: a network's predictions may be accurate and yet miscalibrated, and vice versa.

This section first gives a brief review of uncertainty estimation techniques and emphasizes the importance of having the uncertainty calibration step. Then it explains some existing uncertainty calibration methods for both classification tasks and regression tasks, and offers an reinterpretation of related literature whose methods were referenced and adopted. All methods are post-processing steps that calibrate uncertainties predicted by a forecaster, and are trained on a hold-out calibration set, which in practice can be the same set used for calibrating hyper-parameters. Metrics for evaluating calibration results, including Calibration Plots (Reliability Diagrams), Expected Calibration Error (ECE), and Negative Log Likelihood (NLL), are explained.

2.1 Uncertainty Estimation for Object Detection

Uncertainty estimation of deep learning networks has been an active field of study in the recent years. The methods to model uncertainty in object detection can be categorized into two groups: the ensemble approach and the direct-modeling approach. The former builds an ensemble of object detectors to approximate an output probability distribution with samples. For example, Lakshminarayanan et al. [10] proposed uncertainty estimation techniques for deep neural networks including ensemble methods, heteroscedastic regression. The direct-modeling approach uses an additional network output layers to learn and predict the parameters of a pre-defined probability distribution, such as a multi-variate Gaussian distribution. It requires only a little additional computation during inference, and can improve the detection accuracy.

The focus of recent research is to adapt deep learning networks to incorporate uncertainty and probabilistic methods. Traditionally it has been difficult to model epistemic uncertainty in computer vision, but with new Bayesian deep learning tools this is now possible. Within such methods, a prior distribution is specified upon the parameters of a NN and then, given the training data, the posterior distribution over the parameters is computed, which is used to quantify predictive uncertainty by integrating over all possible model weights. Exact Bayesian inference is computationally intractable for NNs. Recent advances in variational inference have greatly increased the scalability and usefulness of these approaches [11]. A variety of approximation methods have been developed, and the quality of the predictions are therefore largely dependent on the quality of approximation. Gal and Ghahramani [12] first proposed using Monte Carlo dropout (MC-dropout) to estimate predictive uncertainty by using Dropout [13] at test time. Gal et al.[14] then improved upon this and developed concrete dropout, which allows for automatic tuning of the dropout probability in large models and applied in large vision models and Reinforcement Learning. Later, Miller et al. [2] applied Monte-Carlo Dropout directly to object detectors, treating the deep object detector as a black box, to estimate the epistemic uncertainty in the output of deep object detectors.

Established algorithms for estimating uncertainty can either not be directly applied to object detection networks or result in high inference times. Le et al. also [1] proposed two efficient uncertainty estimation methods, sampling free and anchor redundancy, directly based on the multi-box detection outputs. Sampling-free method employed additional variance output for each model output and changed the loss function accordingly to encompass a loss attenuation and then used for regression tasks or adapted for classification tasks. It is combined with BNNs and calculate both aleatoric and epistemic uncertainty concurrently in a single network using the diagonal covariance matrix of the bounding box output from object detectors. Theses methods had limited success due to information loss at the detector‘s non-maximum suppression (NMS) stage, and fails to take into account the multitask, many-to-one nature of anchor-based object detection.

Harakeh et al.’s uncertainty estimation frameworks for 2D object detection [3] provided a principle

way to reformulate the standard object detector inference and non-maximum suppression components from a Bayesian perspective, and provided uncertainty estimates that are better correlated with the accuracy of detection. In practice, however, Bayesian uncertainty estimates often fail to capture true data distributions due to model bias: A predictor may not be sufficiently expressive to assign the right probability to every credible interval, just as it may not be able to always assign the right label to a datapoint. For example, [15] found that the minimum uncertainty error of the Gaussian Entropy is invariant under additive and multiplicative constants, which indicates a flaw in the Bayesian algorithm.

2.2 Calibrating Uncertainty Estimations

Increased model capacity and lack of regularization in modern deep networks are often closely related to overconfidence in model predictions and results in miscalibration. Overconfident predictions on unseen classes or out-of-distribution cases pose a challenge for reliable deployment of deep learning models. The step of adjusting the output distributions to match the observed empirical ones via a post process is called *uncertainty calibration* [7]. We would like the predictions to exhibit higher uncertainty when the test data is very different from the training data.

Uncertainty calibration for classification is a relatively studied field, and many effective methods have been developed, such as the aforementioned isotonic regression, histogram binning, and temperature scaling. In these cases, we isolate a portion of hold-out data from the dataset and use it as the calibration set. Commonly, this is the same set as we used for parameter tuning and evaluation. In such cases, we train an auxiliary model $R : [0, 1] \rightarrow [0, 1]$ on top of a pre-trained forecaster H such that $R \cdot H$ is calibrated.

As a frequentist notion of uncertainty, calibration measures the discrepancy between subjective forecasts and empirical frequencies. There are several metrics to evaluate the quality of uncertainty predictions. Calibration plots, also referred as Reliability diagrams, provide a visual representation of uncertainty prediction calibration by plotting expected sample accuracy against a function of empirical confidence. The Expected Calibration Error (ECE) summarizes the reliability diagram by weighting the error in each bin and producing a single value measure of the calibration. Similarly, the Maximum Calibration Error (MCE) measures the maximal gap. Negative Log Likelihood (NLL) is a standard measure of a model's fit to the data [6]. Several calibration methods were proposed against these measures. For example, non-parametric transformations include Histogram Binning [16], Bayesian Binning into Quantiles and Isotonic Regression, and parametric transformations include versions of Platt Scaling such as Matrix Scaling and Temperature Scaling. It is demonstrated that the simple Temperature Scaling, consisting of a one scaling-parameter model which multiplies the last layer logits, suffices to produce excellent calibration on many classification datasets.

In comparison with classification, calibration of uncertainty prediction in regression, has received little attention so far. Start with extending the algorithms used for classification tasks, [8] proposed a

practical method for evaluation and calibration based on confidence intervals and isotonic regression. The proposed method is applied in the context of Bayesian neural networks. In recent work [17], the authors followed [8]'s definition and methods of calibration for regression, but used a standard deviation vs MSE scatter plot.

Notations: In our case, we are given a labeled dataset $x_t, y_t \in \mathcal{X}, \mathcal{Y}$ for $t = 1, 2, ..., T$ of i.i.d. realizations of random variables $\mathbf{X}, \mathbf{Y} \sim \mathbb{P}$, where $\mathbb{P}$ is the data distribution. The random variable X can be the category prediction in classification tasks, such as "pedestrian", "car", or one of the $uvhw$ (bounding boxes' center coordinates u, v of the image frame, height h, and width w.) predictions. Meanwhile, the random variable Y is the ground truth of the detected objects' class and bounding boxes. Given a prediction x_t, a forecaster $H : \mathcal{X} \sim (\mathcal{Y} \to [0, 1])$ outputs a probability distribution $F_t(y)$ targeting the ground truth label y_t. The recognition part of the object detection is a classification tasks, and we assume binary distributions for each object class, so that $\mathcal{Y} = \{0, 1\}$. In the bounding box regression tasks, the random variable $\mathbf{Y}$ is continuous, and $F_t(y)$ represents a cumulative probability distribution (CDF).

2.3 Calibration and Sharpness for Classification Tasks

Intuitively, calibration means that whenever a forecaster assigns a probability of 0.8 to an event, that event should occur about 80% of the time. In binary classification, where the expected classes $Y = \{0, 1\}$, we say that a forecaster H is calibrated if

$$\frac{\sum_{t=1}^{T} y_t \mathbb{I}\{H(x_t) = p\}}{\sum_{t=1}^{T} \mathbb{I}\{H(x_t) = p\}} \to p \text{ for all } p \in [0, 1] \tag{1}$$

as $T \to \infty$, $H(x_t)$ is a class prediction and $\mathbb{I}$ is the indicator function. For simplicity, we use $H(x_t)$ to denote the probability of the event $y_t = 1$. Thus, more formally we can define perfect calibration as

$$\mathbb{P}(\hat{Y} = Y | H(x) = p) = p \text{ for all } p \in [0, 1] \tag{2}$$

It is in practice impossible to achieve perfect calibration because we can only pick finite amount of data so the the true distribution is always unknown. This motivates the need for empirical approximations that capture the essence of (2).

It is worth noting that, having a calibrated uncertainty is not enough to guarantee a useful forecast [8]. It is also desirable that the forecaster to be *sharp*, i.e. to output a uncertainty prediction close to 0 or 1. Intuitively this makes sense because, for example, it may not be useful for the forecaster to have predictions of probability 0.5 with 100% confidence because we obtain no insights from this prediction. Therefore, we only wish to calibrate the uncertainties to obtain a more accurate insight about whether accept or reject a uncertainty prediction. Thus knowing whether to trust or not trust an object detection result.

Calibration methods for binary classifiers have been well studied and include logistic calibration, also known as 'Platt scaling' [18]; binning calibration [19] with either equal-width or equal-frequency bins; and isotonic calibration [20]. In the scenario of object detection for autonomous driving, we normally would have more than one category. Calibration in multi-class scenarios has been historically approached by decomposing the problem in to one-vs-rest binary classification tasks, so we calibrate with respect to each individual category in the dataset. The predictions of these calibration models can form combined probability vectors after normalization. Therefore, to extend the classification problem to $K > 2$ classes, we represent the network logits as vectors z, and z_i represent the logit value for $i - th$ data point. Then the predicted class output is $\hat{y} = \arg\max_k z_i^k$, where k represents the prediction classes. The predicted uncertainty $H(x_i)$ by the forcaster is typically derived using the softmax function σ_{SM}:

$$\sigma_{SM}(z_i)^{(k)} = \frac{exp(z_i^{(k)})}{\sum_{j=1}^{K} exp(z_i^{(j)})},\ H(x_i) = \max_k \sigma_{SM}(z_i)^{(k)} \tag{3}$$

The goal of calibration is to produce a calibrated confidence $H(x_i)'$ and (possibly new) class prediction $\hat{y}_i'$ based on $y_i, \hat{y}_i, H(x_i)$ and z_i.

2.4 Calibrated Regression

Current techniques for calibrating regression tasks where the output is continues (i.e. $\mathcal{Y} = \mathbb{R}$) extends the methods for classification tasks. Recall that in regression, the forecaster H outputs at each time step t a CDF F_t targeting the ground truth value y_t. Recall the definition of CDF, which denotes the function that returns probabilities of X being smaller than or equal to some value x, is

$$Pr(X \leq x) = F_t(x) = p \tag{4}$$

This function takes input x and returns values from the $[0, 1]$ interval, which is the probability denoted as p. The inverse of the CDF, or quantile function, tells you what x should make $F(x)$ return some value p. We use $F_t^{-1} : [0, 1] \to \mathbb{Y}$ to denote the quantile function $F_t^{-1}(p) = inf\{y : p \leq F_t(y)\}$, which specifies the value of the random variable such that the probability of the variable being less than or equal to that value equals the given probability. We desire

$$\frac{\sum_{t=1}^{T} \mathbb{I}\{y_t \leq F_t^{-1}(p)\}}{T} \to p \text{ for all } p \in [0, 1] \tag{5}$$

as $T \to \infty$. In other words, the empirical and the predicted CDFs should match as the dataset size goes to infinity. Since we define x_t and y_t as i.i.d realizations of random variables $X, Y \sim \mathbb{P}$, a sufficient condition for this is

$$\mathcal{P}(Y \leq F_t^{-1}(p)) = p \text{ for all } p \in [0, 1] \tag{6}$$

Here we use $F_X = H(X)$ to denote the forecast as X [21].

The intuition behind calibrating regression is that for any confidence level p, we may estimate from data that the true probability $\mathcal{P}(Y \leq F_t^{-1}(p))$ of a random Y falling in the credible region $(-\infty, F_X^{-1}(p)]$ below the p^{th} quantile of F_X. For example, if $p = 95\%$, by only $80/100$ observed y_t falls below the 95% quantile of F_t, then we should adjust the 95% quantile to 80%. The calibration is achieved through binning the confidence levels similar to the classification tasks. Moreover, the definition of calibration can extend the notion of calibration to general *confidence intervals* by rewriting as

$$\frac{\sum_{t=1}^{T} \mathcal{I}\{F_t^{-1}(p_1) \leq y_t \leq F_t^{-1}(p_2)\}}{T} \to p_2 - p_1 \tag{7}$$

for $p_1, p_2 \in [0, 1]$ being the boundaries, and as $T \to \infty$.

Similar to classification forecasts, regression forecasts also need to be sharp. This means the confidence intervals should all be as tight as possible around a single value [8]. More formally, we want the variance $var(F_t)$ of the random variable whose CDF is F_t to be small.

Moreover, a simple recalibration scheme for regression tasks was proposed by [8], similar to calibrating classification predictions. In such cases, we train an auxiliary model $R : [0, 1] \to [0, 1]$ such that $R \cdot F_t$ is calibrated. (Algorithm 1). This approach was also adopted by the thesis for calibrating the bounding box regression task.

Algorithm 1 Recalibration of Regression Models

Input: Uncalibrated model $H : \mathcal{X} \to (\mathcal{Y} \to [0, 1])$ and calibration set $\mathcal{S} = \{(x_t, y_t)\}_{t=1}^{T}$.
Output: Auxiliary recalibration model $R : [0, 1] \to [0, 1]$.
1. Construct a recalibration dataset:
    $\mathcal{D} = \{([H(x_t)](y_t), \hat{P}([H(x_t)](y_t)))\}_{t=1}^{T}$,
    where,
    $\hat{P}(p) = |\{y_t | [H(x_t)](y_t) \leq p, t = 1, ..., T\}|/T$.
2. Train a model R (e.g. isotonic regression) on D.

2.5 Diagnostic Tools

2.5.1 Calibration Curve (Reliability Diagram)

A calibration curve, as shown in Figure 1, is a visual representation of model calibration. It is a graph where the conditional distribution of the observations, given the forecast probability, is plotted against the forecast probability. If the model is perfectly calibrated, then the diagram should plot the identity function, i.e. along the diagonal line. Any deviation from a perfect diagonal represents miscalibration. As indicated in Figure 1, the model is overconfident when below the diagonal line, and under confident above the diagonal line.

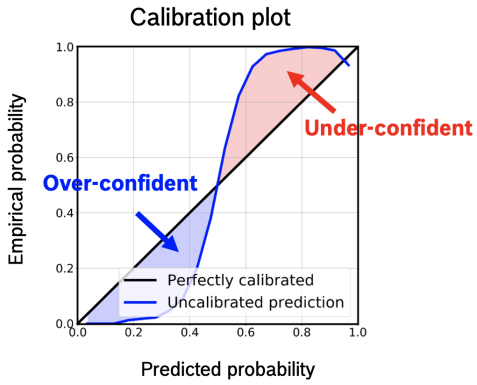


Figure 1: A guide of a reliability diagram. The distributions for perfectly reliable forecasts are plotted along the 45-degree diagonal.

To estimate the expected accuracy from finite amount of samples, we group the samples in to M interval bins, each of size $1/M$ and calculate the accuracy of each bin. Let B_m be the set of indices of samples whose prediction confidence falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. In classification task, the accuracy of B_m can be expressed as

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i) \tag{8}$$

where $\hat{y}_i$ and y_i are the predicted and true class labels for the sample i. $acc(B_m)$ is therefore an unbiased and consistent estimator of $P(\hat{Y} = Y | \hat{P} \in I_m)$. We also define the average confidence within bin B_m as

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{9}$$

where $\hat{p}_i$ is the confidence for sample i. $acc(B_m)$ and $conf(B_m)$ approximate the left-hand and right-hand side of equation (2) respectively for bin B_m. Therefore, a perfectly calibrated model would have $acc(B_m) = conf(B_m)$ for all $m \in 1, ..., M$. The calibration curve is then obtained by plotting $acc(B_m)$ against $conf(B_m)$.

In the case of regression, the accuracy of B_m can be expressed as

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}\{y_t \leq F_t^{-1}(p)\} \tag{10}$$

referring back to the left-hand side of equation (6), which is the definition for calibrating regression problem. The right hand side is then the ground truth probability, which corresponds to the interval bins' boundaries.

2.5.2 Expected Calibration Error (ECE)

While calibration curves are useful visual tools, it is more convenient to have a scalar summary statistic of calibration. Since statistics comparing two distributions cannot be comprehensive, previous works have proposed variants, each with a unique emphasis. One notion of miscalibration is the difference in expectation between confidence and accuracy, i.e.

$$\mathbb{E}_{\hat{P}}[|\mathbb{P}(\hat{Y} = Y|\hat{P} = p) - p|] \tag{11}$$

ECE [22] approximates (6) by partitioning predictions into M equally-spaced bins and taking a weighted average of the bins' accuracy/confidence difference. More precisely,

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)| \tag{12}$$

where N is the total number of samples. The weighted error is a good measure of empirical calibration quality.

In the case of regression, ECE calculates the weighted error between the actual calibration curve and the diagonal line at each uncertainty level. i.e.

$$ECE = \sum_{m=1}^{M} \frac{N_m}{N} |p^m - \hat{p}^m| \tag{13}$$

with N_m being the number of samples in the mth interval.

2.5.3 Negative Log Likelihood

Negative Log Likelihood (NLL) is a standard measure of a probabilistic model's quality, and is also referred as the cross entropy loss [23]. Normally, we would have a softmax function in the uncertainty prediction model, and the NLL loss is often used in tandem with the softmax function in classification tasks.

Let f be a vector containing the class scores for a single example, that is, the output of the network. Thus f_k is an element for a certain class k in all j classes. We can then write the softmax output as

$$p_k = \frac{e^{f_k}}{\sum_j e^{f_k}} \tag{14}$$

the negative log-likelihood can be then defined as

$$L_i = -log(p_{y_i}) \tag{15}$$

This is summed for all correct classes. When training a model, our goal is to find the minima of the loss function given a set of parameters. It is easy to see that whenever the network assigns high confidence at the correct class, meaning p_{y_i} is high, the negative log-likihood is low; when the network assigns low confidence at the correct class, the loss is high. In other words, for a probabilistic model $\hat{\pi}(Y|X)$, NLL is minimized if the model $\hat{\pi}(Y|X)$ recovers the ground truth conditional distribution $\pi(Y|X)$. Therefore, in this experiment, NLL can be used along with temperature scaling to find the optimal temperature T by specifying how well the predicted confidence matches with the ground truth distribution. A low NLL loss suggests a good calibration model. In general, NLL is a standard measure of a model's fit to the data [6] but combines both accuracy of the model and its uncertainty estimation in one measure, which violates the definition of calibration of being independent with model accuracy, and we should be aware of this drawback when making conclusion.

3 Methods

3.1 Temperature Scaling

Temperature Scaling (TS), a variance of Platt scaling [18], is a post-processing approach which rescales the logits of a deep model by parameter T that is called temperature. Temperature scaling is used to soften the output of the softmax layer and makes it more calibrated. With $T = 1$, the original probability is unchanged.

The optimal T can be found by minimizing the NLL loss function respecting to T conditioned by $T > 0$ on validation set V as defined as follows:

$$T^* = \arg\min_x(-\sum_{i=1}^{N} log(S_{y=y_i}(x_i, T))),\ S.t : T > 0, (x_i, y_i) \in \mathcal{V} \tag{16}$$

where $S_{y=y_i}(x_i, T) = exp(\frac{h_i^{y_i}}{T}) / \sum_{j=1}^{K} exp(\frac{h_i^j}{T})$, is the softed version of softmax by applying T, and h_i represents the forecaster's prediction.

[24] showed temperature does not work properly when the validation set used for calibration is too small or contains noisy-labeled samples. Temperature scaling also cannot calibrate highly accurate networks as well as non-highly accurate ones since it basically smooths out the distributions.

The uncertainties of regression tasks are represented by the predicted covariance matrix, diagonal or full, output from the predictor. Existing temperature scaling method cannot incorporate the correlation

across different attributes. Therefore, we assume statistical independence of the regression parameters and only utilize the diagonal components, σ_i^2 where $i \in 1, ..., N$ for N parameters. By scaling the variances by $\hat{\sigma} \leftarrow \sigma^2/T$, we obtain a calibrated probability prediction. [5] attempted to calibrate variance prediction of a 3D LiDAR object detector with temperature scaling successfully.

3.2 Isotonic Regression

Isotonic Regression is a more powerful calibration method that can correct any monotonic distortion. The isotonic regression fits a free-form line to a sequence of observations such that the fitted line is non-decreasing everywhere while minimizing the mean squared error.

We train an auxiliary model based on the isotonic regression $p \rightarrow g(p)$, which is a non-parametric monotonically increasing function to fit the true probability. In our case, we tried to fit the predicted confidence $\mathbb{P}(\hat{Y} = Y|H(X) = p)$ to the truth probability p calculated with the real data using a monotonically increasing function by minimizing the error between them. This is the case for both classification tasks and regression tasks. During the test time, the object detector produces an uncalibrated uncertainty, which was then corrected by the recalibration model $g(\cdot)$ as the final output.

Unfortunately, this extra power comes at a price. A learning curve analysis shows that Isotonic Regression is more prone to overfitting, and thus performs worse than Platt Scaling when data is scarce [25]. In the context of object detection, this should not be a problem because we usually have sufficient amount of data. Moreover, isotonic regression was recommended by [8] for calibrating regression because it accounts for the fact that the true distribution $\mathbb{P}(Y \leq F_X^{-1}(p))$ is monotonically increasing (a CDF, because we defined so). Moreover, it is also non-parametric so we can learn the true distribution given enough i.i.d data.

4 Experiments

4.1 Network Architecture and Baselines

In this experiment, we are aiming to calibrate the uncertainty predictions output by BayesOD [3], which is an uncertainty estimation approach that reformulates the standard object detector, RetinaNet's [26] inference and Non-Maximum suppression components from a Bayesian perspective. The categorical uncertainty prediction outputs of the network are parameters of a categorical distributions $\hat{S}_i \sim Cat([\hat{p}_1 \ ... \ \hat{p}_K])$, which are computed as:

$$\hat{p}_k = \frac{1}{T} \sum_{t=1}^{T} SoftMax(g(x_i, \theta_t))_k \tag{17}$$

where $Softmax(\cdot)$ is the soft max function, and $g(x_i, \theta_t)_k$ is the output logit of the k^{th} category estimated at t^{th} MC-Dropout run of the Network. The aleatoric classification uncertainty is contained within the

estimated parameters $[\hat{p}_1 \ ... \ \hat{p}_K]$ [27]. The per-anchor bounding box regression outputs of the BayesOD model is a probabilistic forecast $F_t(x_t)$ assumes Gaussian marginal probability $\hat{B}_i \sim \mathcal{N}(\mu(x_t), \sigma^2(x_t))$. where

$$\begin{aligned} \mu(x_i) &= \frac{1}{T}\sum_{i=1}^{T} f(x_i, \theta_t) \\ \sum\nolimits_e(x_i) &= \frac{1}{T}(\sum_{t=1}^{T} f(x_i, \theta_t) f(x_i|\theta_t)^T) - \mu(x_i)\mu(x_i)^T \\ \sum(x_i) &= \sum\nolimits_e(x_i) + \frac{1}{T}\sum_{t=1}^{T}\sum\nolimits_a(x_i, \theta_t) \end{aligned} \tag{18}$$

where T is the number of times MC-dropout sampling is performed, $f(x_i, \theta_t)$ is the bounding box regression output of the neural network for the t^{th} MC-Dropout run. The covariance matrix, $\sum_e$, captures the epistemic uncertainty in the estimated bounding box $\hat{B}_i$, and the final per-anchor output covariance is the sum of the epistemic uncertainty and the aleatoric uncertainty $\sum_a(x_i)$.

In addition to the BayesOD methods, to test the generalizability of the calibration algorithms, we also applied them on the predictions of other uncertainty estimation algorithms, including black-box, sample free, and anchor redundancy methods. All baseline uncertainty estimation methods used in comparison are integrated into the inference process of RetinaNet, trained using regression loss function to estimate a diagonal bounding box covariance matrix.

4.2 Experiments and Datasets

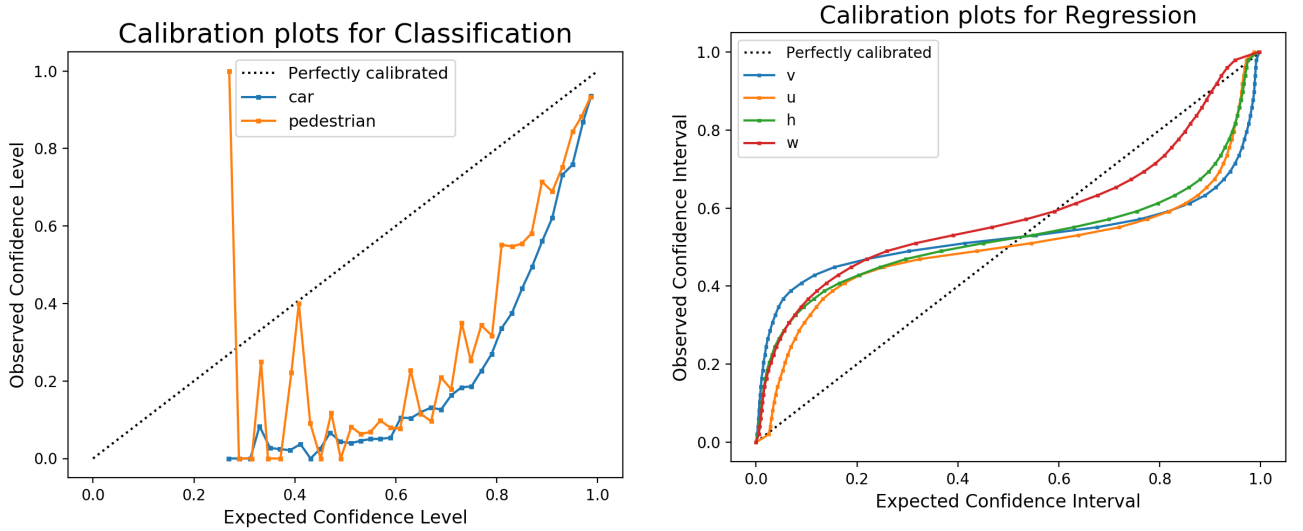
To reduce the computational complexity, we train our model on KITTI 2D object detection and orientation estimation benchmark dataset, which consists of 7481 training images and 7518 testing images. The benchmark uses 2D bounding box overlap to compute precision-recall curves for detection and computes orientation similarity to evaluate the orientation estimates in bird's eye view [28]. We split the training images according to the standard 50% train/validation parts, and use it as the calibration dataset to train and test our calibration model. To evaluate the performance of calibration on KITTI dataset, we trained on 3,712 frames training set, and test on 3,769 frames of validation set. We also tried to vary the train-to-test ratio to evaluate the robustness of calibration models.

4.3 Identifying Uncertainty Miscalibration

We first visualize the existing miscalibration of the existing uncertainty estimation methods in both classification tasks and regression tasks by examing the calibration curves and calibration errors.

In practice, to draw calibration plot for classification, we group the softmax scores into M intervals using the probability thresholds $0 < p_c^1 < ... < p_c^m < ... < 1$, and calculate the empirical probability

following (8) for each interval, denoted as $\hat{p}_c^m$, where the subscript c stands for "classification". In the case of regression, we group predictions into different confidence levels p_r^m , calculated by $F_r(y_r)$, and estimate the corresponding empirical frequency $\hat{p}_r^m$ by (7) and (10). When drawing the regression plot, we assume independence and draw a calibration plot for each bounding box regressor separately. The output of the detectors is in the $vuhw$ format, representing $center_x, center_y, height, width$ respectively. Moreover, we only care about the true-positive object prediction results because it would be meaning less to calibrate a wrong detection. In the experiment, we chose number of bins to be 50 for calculating both calibration plots and ECE losses. ECE of uncalibrated predictions is listed in Table 1.



(a) Calibration plot for the classification tasks.

(b) Calibration plot for the marginal probability distributions for regression tasks.

Figure 2: Calibration plots on KITTI $eval$ set using BayesOD Uncertainty Estimation.

As we can observe from the calibration plots for classification tasks in Figure 2a, the curve lies under the diagonal line. According to 1, we conclude that the BayesOD uncertainty prediction model is miscalibrated, and generally overconfident about its predictions. This suggest we should not have trusted the uncertainty prediction that much, especially when confidence is relatively high.

Figure 2b shows the calibration plot for prediction of bounding box parameters. We observe that the regressor is under-confident when the expected confidence is below than 0.5 and over-confidence when the expected confidence level is above 0.5.

4.4 Calibration with Temperature Scaling

Calibrating Object Classification

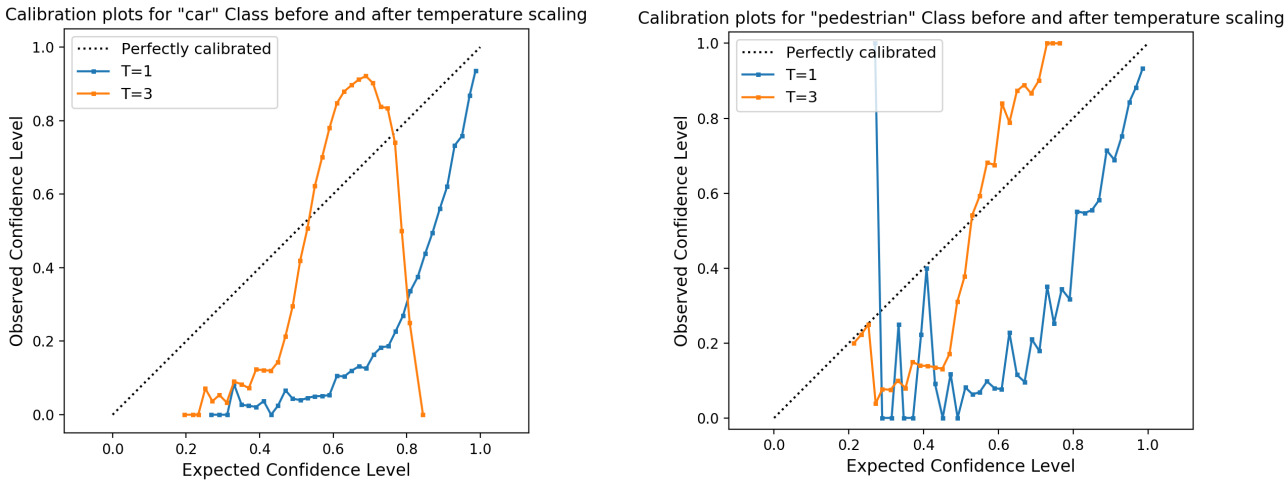
We trained two different calibration models using temperature scaling for classification task and regression task, as the two tasks are inherently different. For both tasks, we experiment with a wide range of values for the hyperparameter, temperature T, on the training set, and obtain the best temperature T

that results in the lowest ECEs and NLLs. We obtain the optimal temperature scales T for each variables independently because they may have independent distributions. The complete tuning results of various temperature scales can be found in Appendix A.

In the before and after comparison calibration plot of the calibration test set is shown in Figure 3a for the "car" category, and Figure 3b for "pedestrian" category. The two plots are generated with the same temperature scales T, because we would desire a general calibration model that performs well on the classification task.

From Figure 3a, we can clearly identify the effect of temperature scaling: stretching the distribution to smooth it out. In calibration plot of the "Car" category(Figure 3 left), we observed that the neural network becomes less confidence after calibration. Due to the smoothing effect, the uncertainty predictions becomes weirdly distributed: the observed confidences surge when they are expected to be near 0.7. The possible explanation is that we are scaling in the logit domain of the network before the softmax function instead linearly scale the probability distribution, so we observe some edge effect near of the end of the probability interval $[0, 1]$.

In the plot for "pedestrian" category, we see the curve moving towards the diagonal line, meaning the predictions are better calibrated. However, the calibration performance is still significantly deviant from the diagonal line and has an "S-shape". This suggests the regressor is still not perfectly calibrated after temperature scaling.



(a) **(b)**

Figure 3: Calibration plots for "car" and "pedestrian" Class with various temperature scaling factors T. When $T = 1$, the model is uncalibrated.

Calibrating Bounding Box Regression

Different from the performance in the classification tasks, the temperature scaling did fairly well in the regression tasks. The best calibration performance on the training set is when $T = 13$, which gives

a minimal calibration error. The calibration plots of the regressor on test calibration set before and after calibration is shown in Figure 4. (Figure 2b is repeated below for ease of comparison.)

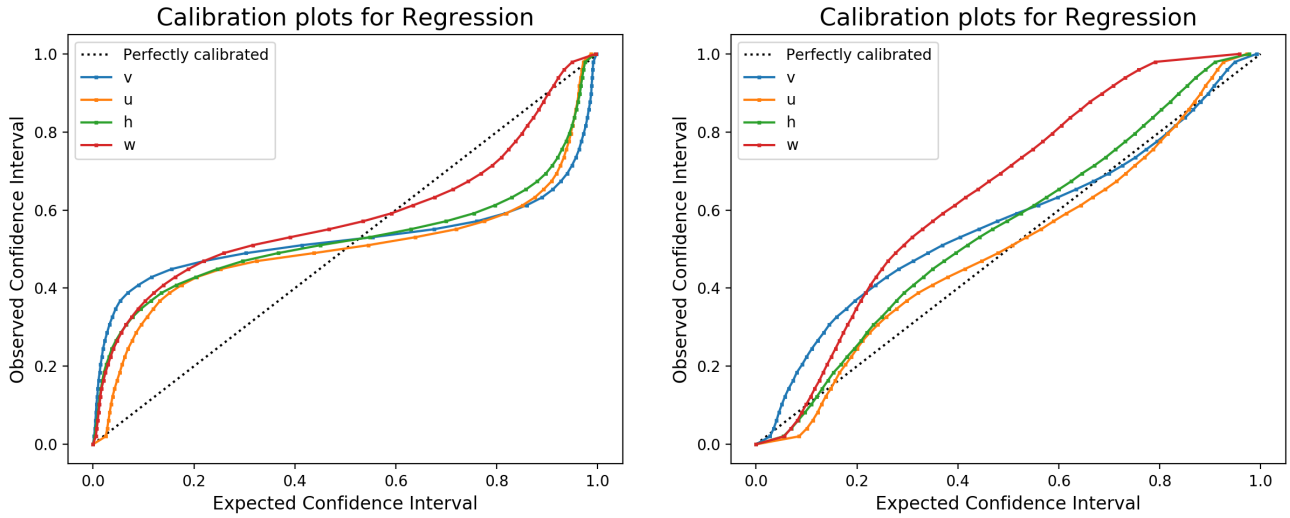


Figure 4: Calibration plots for regression tasks before(left) and after(right) temperature scaling.

Through temperature scaling, all of the regressors are successfully calibrated to have calibration curves near or slightly above the diagonal. The red curve, representing the "width" of the bounding box, has the largest deviant, meaning the object detect has the least accuracy in predicting the confidence of the with of the objects. This may due to the dataset's inherent labeling error, or the deep detector's special property.

4.5 Calibration with Isotonic Regression

We train an isotonic regression model for each uncertainty variable using the training data from the calibration set because we assume they all have independent distribution. Thus, we obtained 6 models in total, 2 for each category in classification tasks, and 4 for each of the bounding box uncertainty predictions.

As describe in Section 3, we train an auxiliary model to solve the problem of minimizing the weighted difference, by

$$\text{minimize} \sum_i w_i(y_i - \hat{y}_i)^2 \tag{19}$$

$$\text{subject to } \hat{y}_{min} = \hat{y}_1 \leq \hat{y}_2 ... \hat{y}_n = \hat{y}_{max} \tag{20}$$

where each weight w_i is strictly positive and each y_i is an arbitrary real number representing different "levels" of magnitudes. It yields the vector which is composed of non-decreasing elements the closest in terms of mean squared error. In practice, the list of element forms a function that is piece-wise linear. The non-decreasing properties is desirable for calibrating uncertainties for regression tasks, since we defined the predictions in the form of a CDF F_t targeting the ground truth value y_t for each time step t. Therefore, we can see the trained isotonic model analogously as a "look-up" table, where when we have a uncalibrated

predictions in the testing step and simply refer to the location near the isotonic function to obtain the transformed calibrated prediction.

The calibration results on the train and test data of the calibration set are shown in Figure 5 and 6 for classification and regression tasks respectively.

Calibrating Object Classification

The adjusted calibration plot for classification task is still noisy, even for the training set. However, an obvious trend of the curve approaching the perfect calibration line can be seen. Recall from Figure 2a that, the uncalibrated curve for pedestrian category is very noisy, and this coarse input data clearly affects the performance of the regression model.

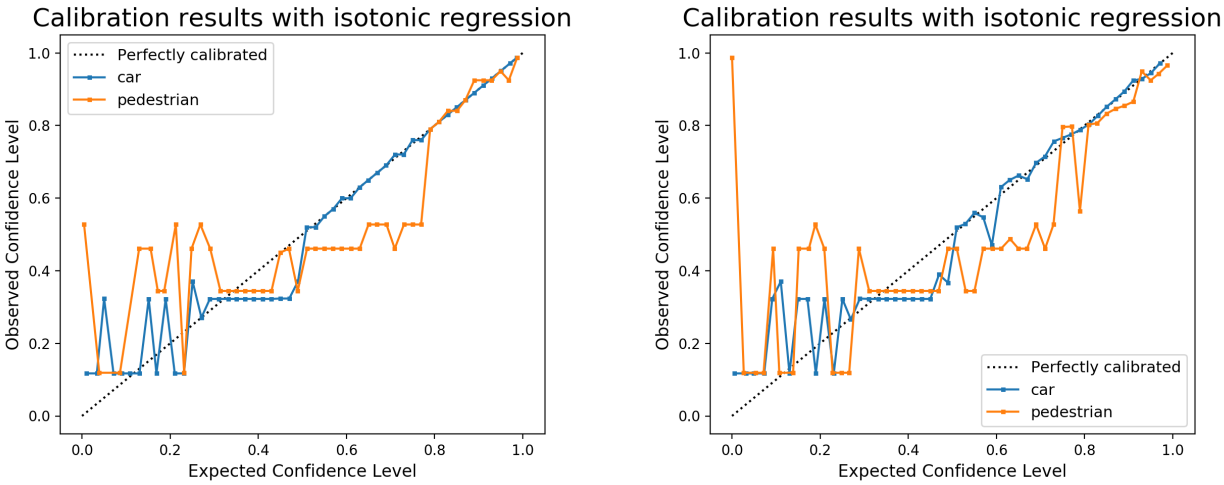


Figure 5: Calibration results of classification tasks on training and test data of calibration set using isotonic regression.

Calibrating Bounding Box Regression

On the other hand, as suggested by [8], isotonic regression performs very well in calibrating uncertainties predicted in regression tasks. We see from Figure 6 that the calibration model achieves perfect result on training set (the left figure), and the expected calibration error is zero. It also performs great on the test set, which suggests that it captures the correct distribution of the variables. Indeed we have less noise in calibrating regression parameters because we only considered the true-positive predictions.

Table 1: Expected calibration errors (ECEs) on the calibration test set.

Method	car	pedestrian	v	u	h	w
Uncalibrated (bayesOD)	0.327	0.391	0.174	0.143	0.142	0.108
Temp. Scaling	0.205	0.215	0.073	0.036	0.057	0.148
Isotonic Regr.	0.074	0.149	0.0018	0.0039	0.0019	0.0055

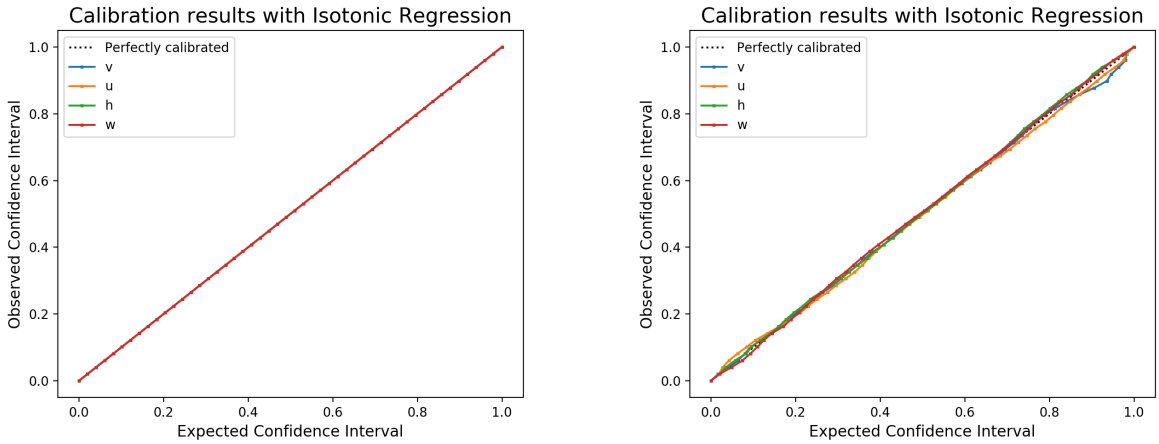


Figure 6: Calibration results of regression tasks on training (left) and test (right) data of calibration set using isotonic regression.

5 Discussions

5.1 Comparison of Calibration Performance

The performance of both calibration method perform mediocre on calibrating the object classification task. One of the important reason could attribute to the aforementioned "one-vs-rest" binary categorization in the classification predictions, which introduce significant amount of noise in the classification model. The performance of both methods on regression tasks are good, and this may due the fact that we only considered true positive detection outputs and discard all the rest. For example, we may desire a high confidence for bounding box predictions for false positive objects in some cases, to tell us the detection results are not trust worthy. However, this is impossible with our setup.

If we were to include the false predictions, the calibration plot would look like the ones shown in Figure 7, where the plot on the left is uncalibrated and the plot on the right is calibrated by temperature scaling.

5.2 Noise and Assumptions in Calibration Models

To properly define and evaluate calibration results, a few assumptions was made. To start with, we need to note that the outputs of the deep object detector forms a categorical distribution. To visualize the predictions for a specific category, we need to marginalize the categorical distribution to obtain a binary distribution. This implicitly assumes that the neural network will perform consistently when performing binary and multi-class predictions for each categories, which is usually not the case. Similarly, we marginalize the full-covariance of the regression outputs by assuming the predictions, v, u, h, w are independently distributed.

Furthermore, we had to assume independence of each individual prediction during the calibration

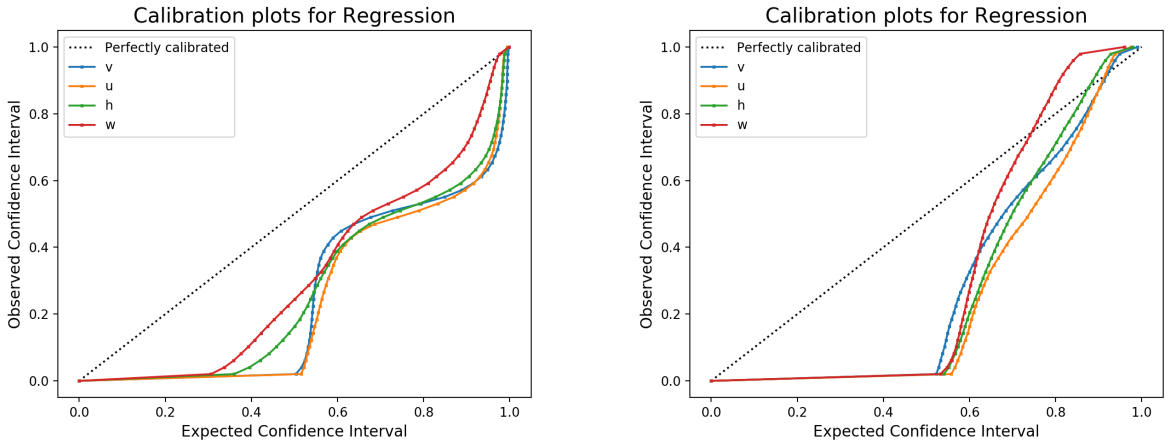


Figure 7: Calibration plot for regression task including false positive predictions, before (left) and after (right) temperature scaling.

process. Recall that calibrating an uncertainty prediction was to have its empirical uncertainty match with the actual one. In 2D object detection, it is fairly common to encounter the case where two or more objects locating on different image depths overlapping on the image plane. As shown in Figure 3, there is a row of cars parking alongside the road with overlaps, the object detector outputs many predictions (in blue) with large IOUs. The network is not sure about the boundaries between them, so just gives many valid predictions. On the right image, the network sees the fence of as cars and outputs many false-positive results. The result of this defect is that the ground truth and predictions are not of an one-to-one correspondence. We have to duplicate the ground truth labels so that the distribution can be matched correctly. However, this is lead the network to a large bias towards closely-clustered objects when calibrating the bounding boxes.

Figure 8: Example frame showing overlaps of target objects which causes the network to generate redundant predictions. Green: ground truth. Blue: predictions. Circles: Uncertainties drawn at top-left and bottom-right corners.

5.3 Varying Training Data Size

Since isotonic regression prunes to be over-fitted to training data, we conducted experiment to examine the calibration performance using various training set sizes and measured the expected calibration error of the test data. As an example, we tested on the regression variables because it seemed to fit the training set perfectly and performed relatively well on the test set. Therefore it may have a great chance to over-fit to the training data. Figure 9 shows the change of expected calibration errors with a few different training set sizes varying in log-scale. We trained individual isotonic regression models with training set

sizes of $50\%, 10\%, 1\%, 0.1\%,$ and 0.05% of the entire calibration set, and test the rest of the data against the trained models. We can see that as the training set size decreases, calibration error increases. The effect becomes significant, i.e. the "elbow", happens when training set size is about 10^{-1}, or 1% of the entire calibration set. Even so, the calibration plot can always capture the trend of the diagonal line when the training set size becomes significantly smaller, as shown by the examples in Figure A.2, where we only used 0.1% and 0.05% (18 true positive detection results) of the calibration set to train the data. We thus see the isotonic regression calibration model is not very likely to over-fit, but this good performance may simply due to the simplicity of the training set's data distribution.

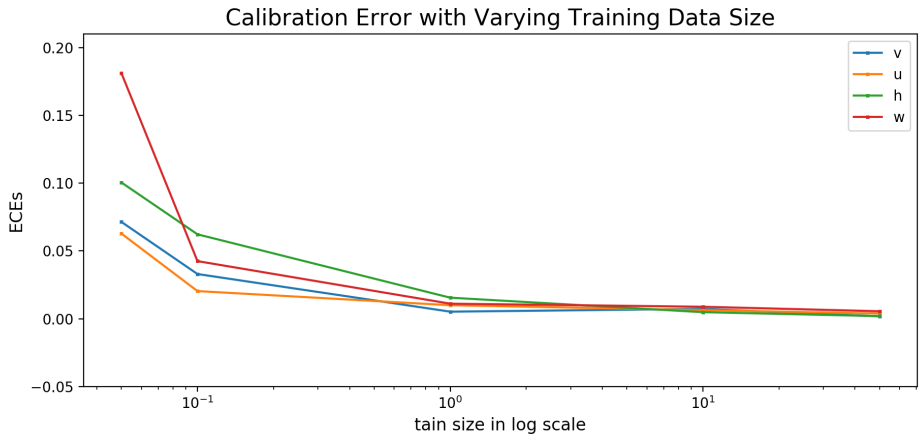
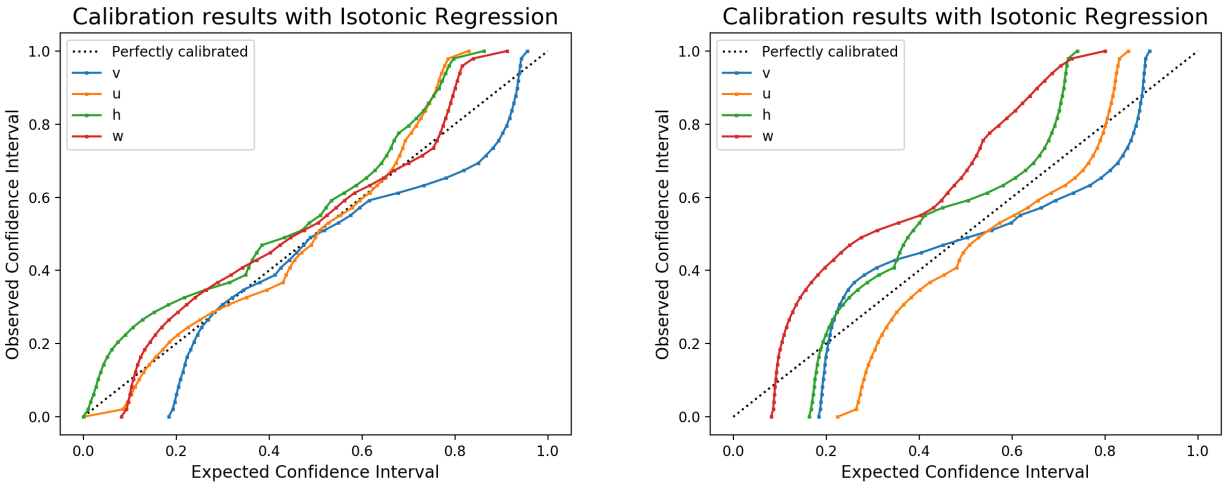


Figure 9: Calibration Error with Varying Training Data Size, plotted in log scale on the x-axis



(a) Calibration result with model trained with 0.1% data **(b)** Calibration result with model trained with 0.05% data

Figure 10: Examples of calibration curves with different training set ratios.

6 Conclusion

Having accurate uncertainty predictions is of equal importance as having high accuracy in safety-critical applications of deep learning models. Many existing state-of-art uncertainty estimation methods, including sampling-free, anchor redundancy, black box, and the Bayesian-based approaches provide principle ways to provide uncertainty estimations. However, as we have identified in Section 4.3, the predicted uncertainties deviates from the expected confidence level. We adopted some existing uncertainty calibration methods, including Temperature Scaling and Isotonic Regression, used in domains such as weather forecasting and general deep learning to 2D object detection scenario. We utilized the standard evaluation metric such as Calibration Curve (or Reliability Diagram), Expected Calibration Error (ECEs) and Negative Log Likelihood to measure the performance of the calibration model. We used the training and validation set from the KITTI dataset to train and evaluate our models.

As we saw in the experiment and discussion session, we conclude that both methods were generally able to calibrate uncertainties in both classification and regression tasks. However, we saw a large amount of noisy data in the object category classification confidence predictions, and temperature scaling failed to calibrate one of the two object classes. Isotonic regression model performed significantly better than temperature scaling method, resulting in a good fit towards the perfect calibration curve and a minimal calibration error. This performance is consistent with most of the relevant research work that were previously published. We also experimented on the effect of training set size on model over-fitting, and showed isotonic regression can over-fit to the training data, but the effect is small and negligible in the case of object detection where we normally have sufficient amount of data.

However, we have not yet tested the model performance on out-of distribution data, for example, evaluate on another dataset such as BDD. This experiment would be a meaningful work that can be conducted in the future. Moreover, some possible ways to improve the performance of calibrating data of categorical distribution can be explored. A recent research [29] proposed a derived method from Dirichlet distributions and suggest to generalize the beta calibration method from binary classification to achieve multi-class calibration. It may serve as a better solution for calibrating the object classification task in deep object detectors.

To sum up, this thesis proved the validity of migrating uncertainty calibration methods from other domain to object detection tasks in autonomous driving scenario. The calibration model is one-pass, simple, require minimal training and fast to apply during the calibration stage. It will adjust the uncertainty predictions of the outputs of the deep object detectors, and correctly indicate the trust-worthiness of a predicted result. This is essential for applications like autonomous navigation, but also useful for other safety-critical applications such as medical diagnosis.

References

[1] M. T. Le, F. Diehl, T. Brunner, and A. Knol, "Uncertainty Estimation for Deep Neural Object Detectors in Safety-Critical Applications," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI: IEEE, Nov. 2018, pp. 3873–3878. [Online]. Available: https://ieeexplore.ieee.org/document/8569637/

[2] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout Sampling for Robust Object Detection in Open-Set Conditions," *arXiv:1710.06677 [cs]*, Apr. 2018, arXiv: 1710.06677. [Online]. Available: http://arxiv.org/abs/1710.06677

[3] A. Harakeh, M. Smart, and S. L. Waslander, "BayesOD: A Bayesian Approach for Uncertainty Estimation in Deep Object Detectors," *arXiv:1903.03838 [cs]*, Mar. 2019, arXiv: 1903.03838. [Online]. Available: http://arxiv.org/abs/1903.03838

[4] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian Model Averaging to Calibrate Forecast Ensembles," *Monthly Weather Review*, vol. 133, no. 5, pp. 1155–1174, May 2005. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/MWR2906.1

[5] D. Feng, L. Rosenbaum, C. Glaeser, F. Timm, and K. Dietmayer, "Can We Trust You? On Calibration of a Probabilistic Object Detector for Autonomous Driving," *arXiv:1909.12358 [cs]*, Sep. 2019, arXiv: 1909.12358. [Online]. Available: http://arxiv.org/abs/1909.12358

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," *arXiv:1706.04599 [cs]*, Jun. 2017, arXiv: 1706.04599. [Online]. Available: http://arxiv.org/abs/1706.04599

[7] D. Levi, L. Gispan, N. Giladi, and E. Fetaya, "Evaluating and Calibrating Uncertainty Prediction in Regression Tasks," *arXiv:1905.11659 [cs, stat]*, May 2019, arXiv: 1905.11659. [Online]. Available: http://arxiv.org/abs/1905.11659

[8] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate Uncertainties for Deep Learning Using Calibrated Regression," *arXiv:1807.00263 [cs, stat]*, Jun. 2018, arXiv: 1807.00263. [Online]. Available: http://arxiv.org/abs/1807.00263

[9] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" *arXiv:1703.04977 [cs]*, Mar. 2017, arXiv: 1703.04977. [Online]. Available: http://arxiv.org/abs/1703.04977

[10] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," *arXiv:1612.01474 [cs, stat]*, Nov. 2017, arXiv: 1612.01474. [Online]. Available: http://arxiv.org/abs/1612.01474

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," *arXiv:1505.05424 [cs, stat]*, May 2015, arXiv: 1505.05424. [Online]. Available: http://arxiv.org/abs/1505.05424

[12] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *arXiv:1506.02142 [cs, stat]*, Oct. 2016, arXiv: 1506.02142. [Online]. Available: http://arxiv.org/abs/1506.02142

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[14] Y. Gal, J. Hron, and A. Kendall, "Concrete Dropout," *arXiv:1705.07832 [stat]*, May 2017, arXiv: 1705.07832. [Online]. Available: http://arxiv.org/abs/1705.07832

[15] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, "Probabilistic Object Detection: Definition and Evaluation," *arXiv:1811.10800 [cs]*, Apr. 2019, arXiv: 1811.10800. [Online]. Available: http://arxiv.org/abs/1811.10800

[16] C. E. Brodley and ICML, Eds., *Machine learning: proceedings of the eighteenth international conference*. San Francisco, Calif: Kaufmann, 2001, oCLC: 248558527.

[17] B. Phan, R. Salay, K. Czarnecki, V. Abdelzad, T. Denouden, and S. Vernekar, "Calibrating Uncertainties in Object Localization Task," *arXiv:1811.11210 [cs, stat]*, Nov. 2018, arXiv: 1811.11210. [Online]. Available: http://arxiv.org/abs/1811.11210

[18] J. C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[19] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *ICML*, 2001.

[20] ——, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2002. [Online]. Available: https://doi.org/10.1145/775047.775151

[21] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, Apr. 2007. [Online]. Available: http://doi.wiley.com/10.1111/j.1467-9868.2007.00587.x

[22] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining Well Calibrated Probabilities Using Bayesian Binning," *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 2015, pp. 2901–2907, Jan. 2015.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://www.nature.com/articles/nature14539

[24] A. S. Mozafari, H. S. Gomes, W. Leão, S. Janny, and C. Gagné, "Attended Temperature Scaling: A Practical Approach for Calibrating Deep Neural Networks," *arXiv:1810.11586 [cs, stat]*, May 2019, arXiv: 1810.11586. [Online]. Available: http://arxiv.org/abs/1810.11586

[25] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 625–632. [Online]. Available: https://doi.org/10.1145/1102351.1102430

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *arXiv:1708.02002 [cs]*, Feb. 2018, arXiv: 1708.02002. [Online]. Available: http://arxiv.org/abs/1708.02002

[27] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging Heteroscedastic Aleatoric Uncertainties for Robust Real-Time LiDAR 3d Object Detection," *arXiv:1809.05590 [cs]*, May 2019, arXiv: 1809.05590. [Online]. Available: http://arxiv.org/abs/1809.05590

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI: IEEE, Jun. 2012, pp. 3354–3361. [Online]. Available: http://ieeexplore.ieee.org/document/6248074/

[29] M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration," *arXiv:1910.12656 [cs, stat]*, Oct. 2019, arXiv: 1910.12656. [Online]. Available: http://arxiv.org/abs/1910.12656

A Temperature Scaling Parameter Tuning

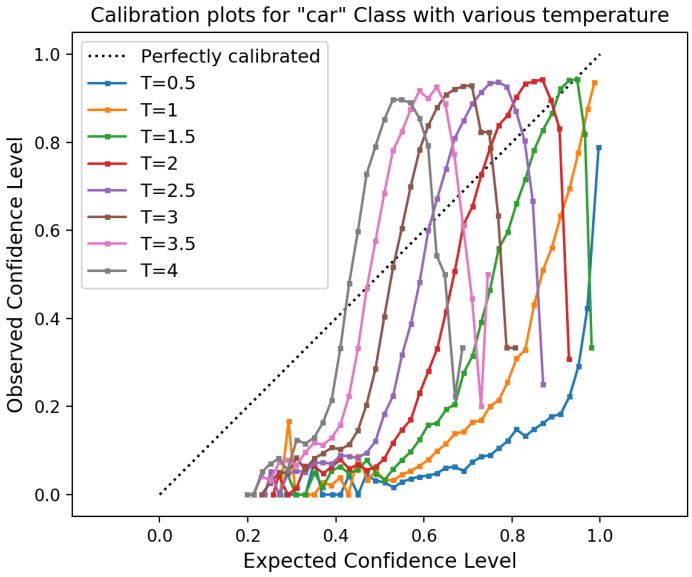


Figure A.1: Calibration with different temperature scales T on training set, "car" category.

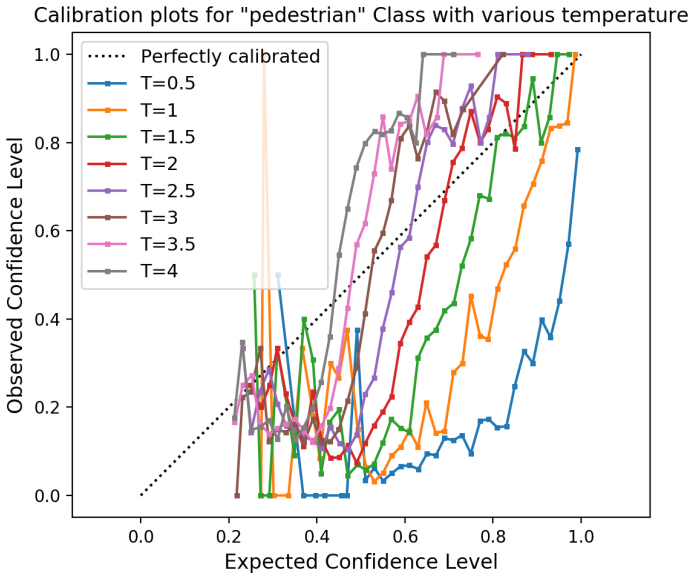


Figure A.2: Calibration with different temperature scales T on training set, "car" category.

Table 2: Expected calibration errors (ECEs) and negative log likelihood (NLL) loss on training set for various temprature scales for classification task.

Temperature	ECE		NLL	
	car	pedestrian	car	pedestrian
0.5	0.412	0.478	1.376	1.236
1(original)	0.329	0.373	0.803	0.835
1.5	0.251	0.300	0.651	0.723
2	0.218	0.257	0.601	0.669
2.5	0.211	0.227	**0.589**	**0.637**
3	**0.207**	**0.205**	0.595	0.618
3.5	0.206	0.188	0.609	0.607
4	0.209	0.174	0.627	0.602

Table 3: Expected calibration errors (ECEs) on training set for various temprature scales for regression task.

Temperature	u	v	h	w
1	0.174	0.143	0.142	0.108
2	0.149	0.115	0.112	**0.094**
5	0.110	0.074	0.066	0.116
10	0.078	0.044	**0.050**	0.128
13	0.071	0.035	0.057	0.135
15	**0.070**	**0.032**	0.061	0.140
20	0.077	0.034	0.070	0.150
30	0.086	0.048	0.087	0.165