SAND2022-2077C

Evaluating the quality of uncertainty quantification enabled deep learning models

Daniel Ries[1] **Jason Adams**[1] **Tyler Ganter**[1] **Joshua Michalenko**[1]

[1]Sandia National Laboratories Albuquerque, NM, USA

Abstract

Traditional deep learning (DL) models are powerful predictors in both regression and classification problems, but many approaches do not provide uncertainties for their predictions or estimates. Uncertainty quantification (UQ) methods for DL models have received increased attention in the literature due to their usefulness in decision making, particularly for high-consequence decisions. However, there has been little research done on how to evaluate the quality of such methods. We use statistical methods of Frequentist interval coverage and interval width to evaluate the quality of credible intervals, and expected calibration error to evaluate classification predicted confidence. These metrics are evaluated on Bayesian neural networks (BNN) fit using Markov Chain Monte Carlo (MCMC) and variational inference (VI), bootstrapped neural networks (NN), Deep Ensembles (DE), and Gaussian Processes for comparison. Because a true probabilistic data generating mechanism is needed for this assessment, we create two simulated data sets with full probability distributions, one regression and one classification. None of the methods appear to be uniformly best, but they do suggest ordering within like methods. BNN-MCMC performs better than BNN-VI and bootstrap NN perform better than DE. The main contribution of this paper is a direct comparison of UQ quality for DL models.

CONTENTS

1 INTRODUCTION

Traditional deep learning (DL) models are powerful predictors in both regression and classification problems (LeCun et al. [2015]), but many do not provide uncertainties for their predictions or estimates. By uncertainty, we refer to both aleatoric and epistemic uncertainty. The usefulness of uncertainty quantification (UQ) in deep learning (DL) models is being recognized, especially for applications that are

high-consequence, including nuclear stockpile stewardship and safety (Trucano [2004], Stracuzzi et al. [2018]), nuclear energy (Stevens et al. [2016]), national security problems (Ries et al. [2022], Gray et al. [2022]), and medical diagnoses (Begoli et al. [2019], Kompa et al. [2021b]). For example, Kompa et al. [2021b] explains the benefit of using UQ in medical decision making, including models that can report "I don't know" to ensure human experts will further evaluate results.

This increased demand for UQ has led to the development of several methods of providing UQ for DL models including Bayesian neural networks (BNN) (Neal [1996]), ensemble methods (Lakshminarayanan et al. [2017b]), Dirichlet-based (Malinin and Gales [2018]), and dropout (Gal and Ghahramani [2016]). Kabir et al. [2018] and Moloud et al. [2021] give reviews of UQ methods in DL.

Although this broadening of UQ in DL methods is encouraging, there is a lack of understanding for how well some of these approaches measure the uncertainty they are set out to quantify. Unlike evaluating a DL model's predictive performance using metrics like mean squared error (MSE) or accuracy, a commonly accepted UQ quality metric does not exist. A desired metric verifies the uncertainty existing (aleatoric and epistemic) is equal to the uncertainty measured. Because of the large scope of UQ methods for DL, this paper focuses on the popular Bayesian and ensemble approaches. This focus of uncertainty is in a probability framework, so credible intervals (CIs), and predictions of confidence for classification, provide the UQ. The main contribution of this paper is a direct comparison of UQ quality for DL models.

This paper is organized as follows: the remainder of Section 1 reviews previous work in the field. Section 2 introduces UQ from the Bayesian perspective in both regression and classification problems. Section 3 introduces interval coverage, interval width and ECE, the UQ metrics used in this paper to assess UQ quality. Section 4 applies the metrics in Section 3 on DL models to two simulated data sets, one regression and one classification. Finally, Sections 5 and 6 discuss the results and provide conclusions and future research directions, respectively.

1.1 REVIEW OF ASSESSING QUALITY OF UQ IN DL

There has been some previous work assessing the quality of UQ using these metrics. Kabir et al. [2018] reviews the ideas of Frequentist coverage and interval width as tools for UQ evaluation and cites several examples. Yao et al. [2019] evaluates the predictive uncertainty for several BNN training methods and ensembles. The authors found ensembles do not provide the UQ that users believe it provides, and emphasize calibration metrics are not good indicators of posterior approximation. The authors concluded a new metric for assessing predictive uncertainty is needed. Ovadia et al. [2019] gives a large-scale benchmark of current UQ for DL methods using metrics such as negative log likelihood, Brier score, and expected calibration error (ECE). The authors find many methods have trouble in out of distribution (OOD) situations or with dataset shift. Ståhl et al. [2020] evaluated several UQ for DL methods, including BNN and Deep Ensembles (DE) and found they captured the uncertainty differently and correlations between the methods' quantifications were low. Kopetzki et al. [2021] evaluates the UQ from Dirichlet based models and finds these models' UQ has trouble with OOD data and dataset shift. Kompa et al. [2021a] checked empirical Frequentist coverage and interval widths for several DL methods. The authors found dropout and ensembling to have low interval coverages and high variability in results on a regression example. In comparison, BNN and Gaussian Process (GP) provided the expected coverages and low variability in the results. For classification, all methods gave adequate coverages for independent and identically distributed (i.i.d.) data, but methods generally performed poorly in terms of coverage when dataset shift was added. Wenzel et al. [2020] explore the effect of temperature scaling, known as weighted likelihood in statistical literature, on the quality of the estimated posterior distribution. Naeini et al. [2015] developed the Expected Calibration Error (ECE) metric for classification models which assesses the agreement of predicted confidences and model accuracy.

1.2 REVIEW OF BAYESIAN AND ENSEMBLE-BASED UNCERTAINTY QUANTIFICATION

Bayesian neural networks were first popularized by David MacKay (MacKay [1992, 1995]) and his student Radford Neal (Neal [1996]). Neal's dissertation introduced Hamiltonian Monte Carlo (HMC) as a way to sample the posterior distribution of a BNN, providing a practical way of training. To this day, HMC is considered the gold standard for BNN training due to its theoretical backing and lack of approximations.

Variational inference is the most popular method of Bayesian inference for neural networks (NN) (Graves [2011]). Blei et al. [2017] gives an extensive review of VI methods. Blundell et al. [2015] introduced Bayes by Backprop which is a practical stochastic VI algorithm to train a BNN. A common criticism of standard implementations of VI is the mean-field assumption, or assuming posterior independence of all parameters, although alternative exist (Louizos and Welling [2016], Zhang et al. [2018]).

The bootstrap is a simulation-based method that treats the training data as the population and samples with replacement new data sets from the original training set. Uncer-

tainty is measured by creating a large number of these new data sets and then using the distribution of estimates or predictions to quantify uncertainty (Gray et al. [2022]). Deep Ensembles (Lakshminarayanan et al. [2017a]) follow a similar idea to the bootstrap except no resampling is done; the only difference for each model in the ensemble is the set of starting values for the model optimizer.

1.3 EPISTEMIC AND ALEATORIC UNCERTAINTY

Uncertainty can be separated into aleatoric and epsistemic components. A comprehensive introduction to the two types of uncertainties in the context of machine learning is given by Hüllemeier and Waegeman [2021]. In brief, aleatoric uncertainty is the variability due to randomness or noise in the process or measurement. This type of uncertainty is always present and can only be reduced by an improvement in the process of measurement, not by increasing the sample size. Epistemic uncertainty is the uncertainty resulting from imperfect knowledge of the model. Examples of this include uncertainty during model selection and parameter uncertainty during training. Increasing sample sizes will help reduce epistemic uncertainty by either further understanding the mechanism and creating better model architectures, estimating model parameters more precisely, or both.

2 BAYESIAN APPROACH TO UNCERTAINTY

2.1 BAYESIAN PROBABILITY

The Bayesian interpretation of probability relies on the *degree of belief* of an event rather the traditional relative frequency of the event. This allows Bayesian models to be used and easily interpreted in a wider variety of problems, especially rare event problems. The relative ease of the interpretation is also beneficial to non-experts who do not need specialized training to understand the output of Bayesian models. The degrees of belief are initialized through a *prior* distribution, and these beliefs are updated by observing new data, resulting in a *posterior* distribution, containing information on beliefs after taking into account new data.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, be a training data set where $\mathbf{y} = (y_1, ...y_n)'$ and $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)'$. Let $y_i \in \mathbb{R}$ for regression or $y_i \in \mathbb{N}$ for classification and $\mathbf{x}_i \in \mathbb{R}^p$ be a p-dimensional vector of features corresponding to response y_i. For simplicity, we assume each element of $\mathbf{x}_i$ is a scalar, although in general this need not hold. For example, some features can be scalars and some could be more rich information, e.g., functions or surfaces. The posterior distribution for the model parameters θ, given data $\mathcal{D}$ is:

$$p(\theta|\mathcal{D}) = \frac{p(\mathbf{y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{y})}, \tag{1}$$

where $p(\theta)$ represents the prior beliefs of θ, $p(\mathbf{y}|\mathbf{X}, \theta)$ represents the new data available, and $p(\mathbf{y})$ is the marginal likelihood, or normalizing constant. Equation 1 shows the mechanism behind updating degrees of belief. The posterior distribution, $p(\theta|\mathcal{D})$ contains all the information necessary to make inference and predict, including UQ.

For classification BNNs, the uncertainty of interest is on the estimation of the class probabilities, $\pi_c^* = P(y^* = c|\mathbf{x}^*, \theta)$, where y^* and $\mathbf{x}^*$ denotes new data. The estimated probability, $\hat{\boldsymbol{\pi}}^*$, is typically modeled as some parameterized function $g_\theta(\mathbf{x}^*)$ (e.g. $g_\theta(\mathbf{x}^*)$ could represent a DL model). Therefore uncertainty of $\hat{\boldsymbol{\pi}}^*$ can be obtained in the form of $(1-\alpha)\%$ CI for $g_\theta(\mathbf{x}^*)$, denoted by $\mathcal{B}_{g_\theta(\mathbf{x}^*)}(\alpha)$. In this paper we do not distinguish between credible and prediction intervals. These intervals can be constructed in several different ways, the simplest being by computing $\alpha/2^{th}$ and $(1-\alpha/2)^{th}$ quantiles of the posterior distribution $p(g_\theta(\mathbf{x}^*)|\mathcal{D})$. The aleatoric uncertainty comes through the predicted probability itself, $\hat{\boldsymbol{\pi}}^* = g_{\hat{\theta}}(\mathbf{x}^*)$, meaning it must be calibrated for accurate assessment.

In regression problems, predictions are on the *response* space and often need to consider epistemic and aleatoric uncertainties. To do this, the posterior predictive distribution is needed:

$$p(y^*|\mathcal{D}, \mathbf{x}^*) = \int p(\theta|\mathcal{D})p(y^*|\mathbf{x}^*, \theta)d\theta, \tag{2}$$

The posterior predictive distribution represents the degree of belief for future *observations*, given what we have have already seen. Practically, the right side of Equation (2) is marginalizing the data model $p(y^*|\mathbf{x}^*, \theta)$ over the updated posterior distribution $p(\theta|\mathcal{D})$, representing the degree of belief for the model parameters. Uncertainty for a future value y^* can then be obtained in the form of $(1-\alpha)$ CI, denoted by $\mathcal{B}_y(\alpha)$ and computed in a similar way as $\mathcal{B}_{g_\theta(\mathbf{x}^*)}(\alpha)$.

2.2 BNN FOR REGRESSION

Letting $y_i \in \mathbb{R}$, a fully connected, L-layer feed-forward regression BNN can be represented in statistical notation by:

$$y_i|\mathbf{x}_i, \boldsymbol{\theta}, \sigma^2 \overset{iid}{\sim} N(\mu(\mathbf{x}_i; \boldsymbol{\theta}), \sigma^2) \tag{3}$$

$$\mu(\mathbf{x}_i; \boldsymbol{\theta}) = f_{\theta_o}(f_{\theta_L}(\cdots f_{\theta_1}(\mathbf{x}_i))) \tag{4}$$

$$\boldsymbol{\theta} \sim \mathcal{F}(a, b) \tag{5}$$

$$\sigma^2 \sim \mathcal{F}(c, d), \tag{6}$$

where $f_{\theta_o}(\cdot)$ is the output function with parameters θ_o, and $f_{\theta_\ell}(\cdot)$ is a nonlinear activation depending on parameters, θ_ℓ, for $\ell = 1, 2, \ldots, L$. The model's parameters are the NN weights $\boldsymbol{\theta} = (\theta_o, \{\theta\}_{\ell=1}^{L})$ and the data noise, σ^2, which in a Bayesian framework require prior distributions specified by the generic distribution $\mathcal{F}$ with specified hyperparameters a, b, c, d. The distributions do not need to be from the same family. The mean function for observation i, which is often used for point prediction, is represented by $\mu(\mathbf{x}_i; \boldsymbol{\theta})$, showing its dependence only on the features for observation i and the NN weights, $\boldsymbol{\theta}$. Aleatoric uncertainty is measured through σ^2 and epistemic uncertainty is measured in part through the posterior distribution of model parameters, $p(\boldsymbol{\Theta}|\mathcal{D})$, where $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \sigma^2)$. A CI for a new observation y^* is created from computing the posterior predictive distribution of y^*, as in Equation (2).

2.3 BNN FOR CLASSIFICATION

Let $y_i \in \{0, 1\}$, be a binary response. Note this is for simplicity, the model is easily extended to K classes.

$$y_i|\mathbf{x}_i, \boldsymbol{\theta} \overset{iid}{\sim} \text{Bernoulli}(\pi(\mathbf{x}_i; \boldsymbol{\theta})) \tag{7}$$

$$\pi(\mathbf{x}_i; \boldsymbol{\theta}) = f_{\theta_o}(f_{\theta_L}(\cdots f_{\theta_1}(\mathbf{x}_i))) \tag{8}$$

$$\boldsymbol{\theta} \sim \mathcal{F}(a, b), \tag{9}$$

where all the terms mean the same as in the regression case, except the output activation function $f_{\theta_o} : \mathbb{R} \to (0, 1)$. Point predictions of *probability* that observation i is class 1 are given by $\pi(\mathbf{x}_i; \boldsymbol{\theta})$. Because the data space is $\{0, 1\}$, CIs on the data space do not have practical use (the only possible intervals are [0,0], [0,1], and [1,1]), so intervals can instead be constructed and interpreted on the probability space of the output activation. The full posterior distribution is $p(\boldsymbol{\Theta}|\mathcal{D}) = p(\boldsymbol{\theta}|\mathcal{D})$ since there is no σ^2 term. A CI for a new observation y^* is created from the function of the posterior, $\pi(\mathbf{x}^*; \boldsymbol{\theta})$, given by $p(\pi(\mathbf{x}^*; \boldsymbol{\theta})|\mathcal{D}, \mathbf{x}^*)$.

2.4 TRAINING METHODS FOR BNN

Two popular approaches to training BNN, and Bayesian models in general, are MCMC and VI. A review of MCMC methods is given by Gelman et al. [2013] and of VI methods by Blei et al. [2017]. Generic overviews of VI and MCMC algorithms are given in the supplemental material.

Put simply, VI is an approximation to the posterior distribution $p(\theta|\mathcal{D})$ using optimization that improves as the sample size increases, compared to MCMC which is an approximation to $p(\theta|\mathcal{D})$ using sampling that improves as the number of Monte Carlo samples increases. Therefore, VI is constrained in its ability to approximation $p(\theta|\mathcal{D})$ by data, and MCMC is constrained by computation time.

3 UNCERTAINTY QUANTIFICATION QUALITY METRICS

3.1 FREQUENTIST INTERVAL COVERAGE

Credible intervals are contain a set of plausible predictions (for regression) or estimates (for classification), where plausible is defined by the *nominal* rate of the interval itself, typically denoted as $(1 - \alpha)\%$. A $(1 - \alpha)\%$ CI for an estimate should contain the true population parameter about $(1-\alpha)\%$ of the time if the experiment was redone. A $(1 - \alpha)\%$ CI for a prediction should contain the true value of the observation with $(1 - \alpha)\%$ probability. Frequentist coverage (coverage from hereon) is the *actual* rate at which the population parameter is contained in the interval or true value of the observation is contained in the interval, averaged over all observations.

$$\text{CI Coverage (Regression)} = \frac{1}{n}\sum_{i=1}^{n} I\big(y_i \in \mathcal{B}_{y_i}(\alpha)\big) \tag{10}$$

$$\text{CI Coverage (Classification)} = \frac{1}{n}\sum_{i=1}^{n} I\big(\eta_i \in \mathcal{B}_{\eta_i}(\alpha)\big) \tag{11}$$

This empirical value should be as close as possible to the nominal rate of $(1 - \alpha)\%$. Going under or over this value is an indication of poor UQ quality, e.g. a 90% CI with 70% coverage indicates the interval is overly optimistic and not accounting for enough uncertainty, conversely a 90% interval with 99% coverage is overly conservative. Note that Equations (10)-(11) require knowing the *true* value of an observation or model parameter.

3.2 INTERVAL WIDTH

Intervals contain values that are plausible estimates or predictions for a quantity of interest, therefore it would make sense that there is less variability in the data generating mechanism if the interval is smaller. However, it is not quite this simple. The highest UQ quality is given to models that minimize interval width *and* match coverage with nominal rate. The width of intervals is given in equation by

$$\text{Interval Width} = \frac{1}{n}\sum_{i=1}^{n}(\mathcal{B}_{y_i}(\alpha)_{UB} - \mathcal{B}_{y_i}(\alpha)_{LB}). \tag{12}$$

The lower bound and upper bound of the $(1 - \alpha)\%$ interval for y are given by $\mathcal{B}_y(\alpha)_{LB}$ and $\mathcal{B}_y(\alpha)_{UB}$, respectively. The equation for a classification CI width is the same except replacing observation y for model parameter η.

3.3 EXPECTED CALIBRATION ERROR

Naeini et al. [2015] proposed ECE as a metric to check whether a machine learning classifier's confidence scores are calibrated to true probabilities of correctness. Here we use the broader term *predicted confidence* defined as $\hat{\pi}_i \equiv \pi(\mathbf{x}_i, \hat{\theta}) \in [0, 1]$. However, we make no claim that all models are expected to estimate the true probability.

Consider a binary decision rule, $\tau(\cdot)$, that generates predictions $\tau(\hat{\pi}_i) = \hat{y}_i \in \{0, 1\}$. Provided a set of true and predicted responses, the accuracy is computed as:

$$acc(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i=1}^{n} I(\hat{y}_i = y_i). \tag{13}$$

The average confidence of the set is

$$conf(\hat{\boldsymbol{\pi}}) = \frac{1}{n} \sum_{i=1}^{n} \hat{\pi}_i. \tag{14}$$

ECE discretizes the interval [0, 1] under equally spaced bins and assigns each predicted confidence to the bin that encompasses it. The calibration error of a bin is the difference between the accuracy and average confidence of the samples assigned to that bin. In other words, calibration error treats predicted confidences as estimated probabilities and measures the disagreement between estimated and true probability of correctness. ECE is a weighted average across all bins:

$$ECE(\mathbf{y}, \hat{\boldsymbol{\pi}}) = \sum_{b=1}^{B} \frac{n_b}{n} \Big| acc\big(\mathbf{y}_b, \tau(\hat{\boldsymbol{\pi}}_b)\big) - conf(\hat{\boldsymbol{\pi}}_b) \Big|. \tag{15}$$

where B is the number of bins, $(\mathbf{y}_b, \hat{\boldsymbol{\pi}}_b)$ is the subset of $(\mathbf{y}, \hat{\boldsymbol{\pi}})$ in the b^{th} bin, and n_b is the number of predictions in bin b, i.e. the rank of $\hat{\boldsymbol{\pi}}_b$.

From a UQ view, this metric assesses the quality of aleatoric uncertainty given by a classification model since the variance of multinomial distributions are determined by class probabilities. Interval coverage and width in regression problems provide an assessment jointly of epistemic and aleatoric uncertainty, while in classification they provide an assessment of only epistemic uncertainty.

4 SIMULATION STUDIES

In this section we evaluate UQ metrics of Section 3 on a simulated regression problem and two class classification (TCC) problem to compare different UQ in DL methods, including BNN trained via MCMC, BNN trained via VI, bootstrapped NN, DE, GP, and in the regression case, an oracle model that knows the true functional form. By having the true underlying data generating mechanism, we are able to assess the quality of the UQ given by these models.

4.1 SIMPLE 1-D REGRESSION

For the regression problem, data is simulated from the following non-linear function:

$$\begin{aligned} f(x) &= .5x - 8x^2 - x^3 + 2x^4 + \epsilon \\ \epsilon &\sim N(0, 1). \end{aligned} \tag{16}$$

Figure 1 shows an example of the training and testing data. We opted to create a block testing set in order to understand small and large interpolation behavior. This provides a way to evaluate how a model provides uncertainties in data rich and data poor environments.

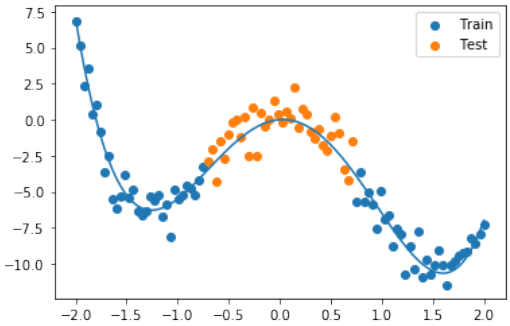

Figure 1: One simulated dataset with mean function overlaid, training/testing split shown.

The BNNs (MCMC and VI), bootstrap NN, and DE had architectures with one hidden layer and each of $(2, 5, 10, 20, 40)$ nodes, to compare UQ across different model complexities. The MCMC model was fit using numpyro, and the VI model was fit using pyro, however, we could not get the model to properly converge for 40 nodes. The two ensemble methods, bootstrap NN and DE, were both fit in pytorch with 100 ensembles each. The GP was fit using Stan with an exponential covariance function with half-Normal(0,1) priors on the nugget and partial sill, and an InverseGamma(1,1) on the partial range. For comparison, an oracle model which knows the true model form from Equation (16) is also fit using ordinary least squares.

We evaluate mean interval coverage and mean interval width for small interpolation (training data) and large interpolation (testing data) over 100 simulated data sets. Figures 2 and 3 show coverages on the testing and training sets, respectively. Figures 4 and 5 show the widths for testing and training data, respectively. We include metrics on the training data in order to compare in-sample vs out-of-sample UQ performance directly. Note the oracle and GP results are constant with respect to nodes.

The oracle model has coverages right at the nominal level for both testing and training as expected. Additionally, its interval widths are the narrowest, except compared to DE and bootstrap NN, whose intervals vastly undercover the nominal level. The orcale model sees few changes between training and testing sets, due to a known model form. The bootstrap NN and DE do not accurately capture the uncertainty with their CI as evidenced by their extremely low coverage. As expected, the bootstrap NN does have slightly wider intervals than the DE. The GP and BNN-MCMC both tend to over cover for the training data and for testing data, when the BNN has at least 10 nodes. Both the BNN-MCMC and GP have interval widths similar to the oracle for the training data. Their widths increase dramatically on the test data, with the BNN-MCMC's widths very high beyond a 10-node model. This shows how a BNN's UQ in regions without data will be affected by increasing its flexibility.

Plots of MSE for training and test is provided in the supplemental material. Overall, the oracle, GP, and BNN-MCMC are similar and the best in terms of predictive performance. Figures with model estimates and UQ for a single simulated data set are also included in the supplemental material.

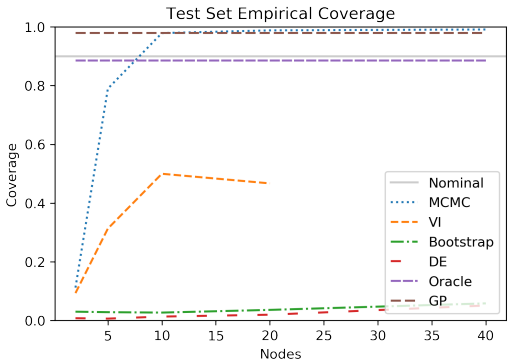

Figure 2: Mean Frequentist coverage of 90% credible intervals on **test** set of regression simulation.

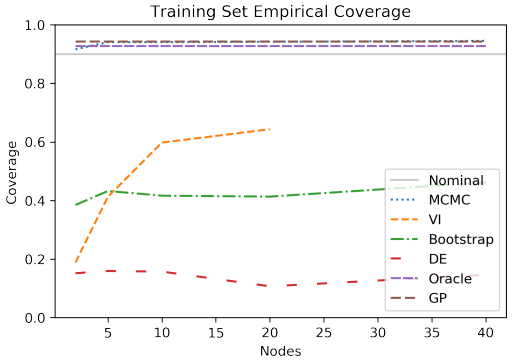

Figure 3: Mean Frequentist coverage of 90% credible intervals on **train** set of regression simulation.

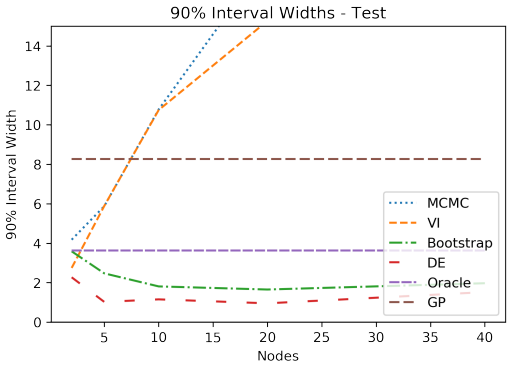

Figure 4: Mean width of 90% credible intervals on **test** set of regression simulation. Limits of y-axis are bounded for easier comparisons. BNN-MCMC width at 40 nodes is 27.4.

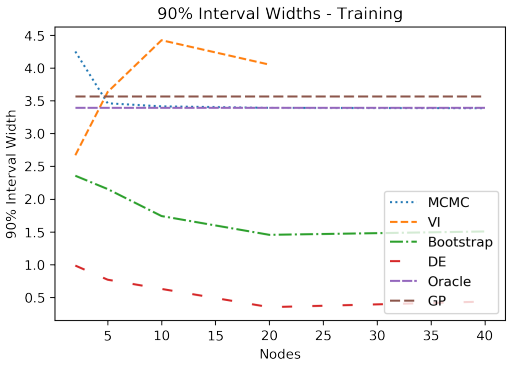

Figure 5: Mean width of 90% credible intervals on **train** set of regression simulation.

4.2 TWO CLASS CLASSIFICATION

The TCC dataset is a fully parameterized generative model with a joint probability that allows direct evaluation of CI coverage. A full probability distribution is needed in classification problems to check CI coverage. The underlying model is a 2-D Gaussian Mixture Model (GMM) with two equally proportioned clusters that undergo a series of transformations and scalings. The result is a data model that can easily generate a large variety of data classification scenarios that arise in quantifying UQ. Figure 6 shows one simulated TCC data set and densities. For more details about how this data was simulated, see the supplemental material.

The architecture for the DL models was a three layer fully connected NN. To further check how the number of nodes per layer affects model UQ, we fit each model type with each of (2,5,10,20,40) nodes per layer. The MCMC-BNN was fit using numpyro, the DE and bootstrap NN used pytorch, and the BNN-VI and GP used Stan. We attempted to train the BNN-VI using pyro, but were unable to get acceptable convergence.

Figures 7 shows mean coverages for 90% CIs. The BNN-

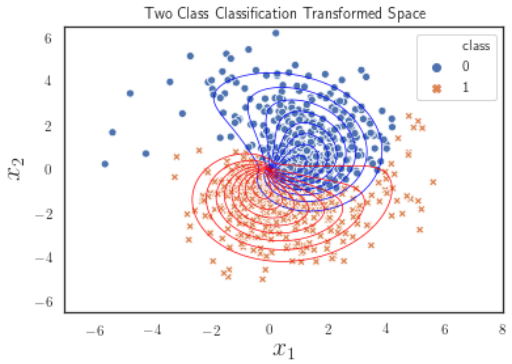

Figure 6: TCC transformed space with 10% contours for $P(Y = y|x_1, x_2)$.

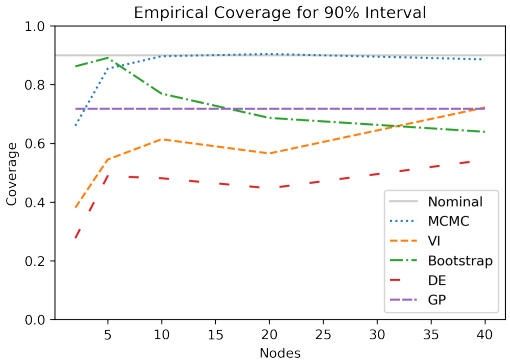

Figure 7: Coverage on TCC.

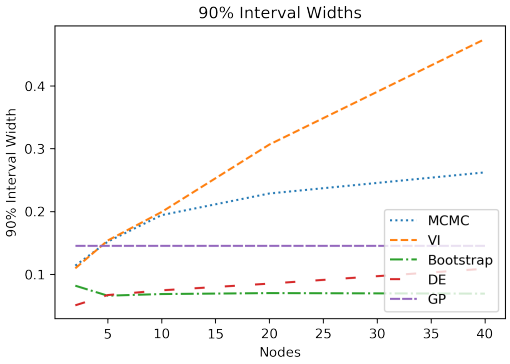

Figure 8: Interval width on TCC.

MCMC is right at the nominal value for nodes per layer greater than two, and the bootstrap NN has nominal coverage for 5 nodes, but then decreases as nodes increase. Both DE and BNN-VI undercover significantly. The GP's coverage is about 0.75.

Figure 8 shows the interval width as a function of number of nodes. BNN-MCMC's widths increase as the nodes per layer increase, to a point where the CI would not be practically useful. Conversely, the bootstrap NN's widths aren't majorly affected by the model complexity. DE's width are also not affected by model complexity, but its coverages are too low. The width of BNN-VI increases dramatically as model complexity grows. The GP's widths are wider than the bootstrap, but typically narrower than the BNN-MCMC.

Figure 9 shows the ECE as a function of nodes. The bootstrap has the best overall ECE at five nodes per layer, which is also where its coverage reached the nominal level. BNN-MCMC, DE, and GP have similar ECE, and the bootstrap NN's ECE tracks with them at higher model complexity. BNN-VI again sees much worse results compared to other models. Figures with model estimates and UQ for a single simulated data set are also included in the supplemental material. These plots clearly differentiate the DL models which find a decision boundary compared to the GP which models the underlying class distributions.

5 DISCUSSION

There are several results from the simulations that are worth further discussion. First, DE failed to provide useful UQ in either simulated example. As already argued, this is not surprising since DE creates an ensemble by simply using different starting values for each model in the ensemble. Practically this means the uncertainty the ensemble is capturing is the optimization uncertainty. Although this may be of interest in some scenarios, we do not believe this is the case for most users. However, DE is a simple way to understand the complexity of the training procedure. In Lakshminarayanan et al. [2017a], the authors even mention, for problems with large sample sizes, the fact that DE doesn't resample with replacement like a bootstrap, did not make a difference. However, in cases where we are not data-rich, as in many high-consequence national security problems, we do not have the luxury of an abundance of data. Therefore, for high-consequence problems, we recommend to proceed with caution when using DE, and urge users to understand theoretically which types of uncertainty DE will measure, and which it will not.

Simply resampling data with replacement (bootstrap) for each model in the ensemble gives a theoretically plausible solution to the simplicity of DE. Surprisingly, the bootstrapped NN did not perform as expected on the regression problem, even though it did give slightly higher uncertainty than DE. On the classification problem, it performed as expected providing adequate coverage, reasonable interval widths, and the best ECE, comparable with BNN-MCMC.

Bayesian neural networks fit using MCMC significantly outperformed BNN fit using VI. Although MCMC is the gold standard for Bayesian estimation, we hoped VI would have given better results given the theoretical guarantees it has. We do note that BNN fit with VI is still a difficult process, and we believe it is possible better results could be obtained using different software or VI algorithms. But in

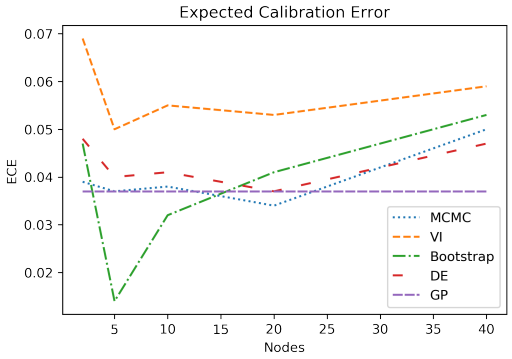

Figure 9: Expected calibration error on TCC.

light of this, we recommend caution for non-experts using BNN fit via VI. VI provides a significant speedup that should not be ignored, therefore future work should continue to develop VI algorithms and continue to make them more user-friendly. More research and applications of BNN fit using VI will help understanding of how to diagnose common training issues.

There are ample opportunities for future work in the assessment of the quality of UQ for DL models. New metrics should be created that assess the quality of UQ given by DL models, preferably ones that are more well suited to the DL framework. Although the traditional statistical metrics used in this paper are adequate, there are certainly better ways. And we argue for metrics beyond simply combining the two, such as with the coverage width criterion of Khosravi et al. [2011] or evaluating coverages and a large number of nominal rates such as with the continuous ranked probability score from Zamo and Naveau [2018]. We recognize these metrics are useful in evaluation too, but they still require knowing the underlying *true* probability distribution, which for classification problems is only possible with simulated data. New metrics will be able to be used in regression and classification on real data to compare which UQ method to use for that specific data set, much like model selection is currently done (where it only considers predictive performance of point estimates). Furthermore, more general methods that can compare models across paradigms, such as comparing the UQ models considered in this paper to Dirichlet-based models. A metric analogue to the AIC, which allows simple comparison of model fits, is desired to measure the quality of UQ.

6 CONCLUSION

Uncertainty quantification of DL models is an active area of research since researchers and users of DL models have realized point predictions are not always enough, especially in high consequence problems. Many different approaches to UQ for DL models have been proposed, some of which have a probabilistic interpretation. However, there has been little research into the *quality* of those UQ methods. This paper explores the quality of UQ given by several probabilistic UQ models, including BNN, DE, and bootstrapped NN, using traditional statistical metrics of frequentist coverage and CI width, as well as ECE for classification problems. Two simulated data sets, one regression and one classification, for which complete knowledge of the data generating mechanism was known, were used to quantitatively assess the UQ qualities. Although there was not a clear winner, BNN trained via MCMC tended to give the best overall results, but this is not without caveats. However, this paper only explores two specific cases and therefore more research is this area is needed, and better UQ metrics need to be developed to definitively compare UQ in DL methods.

Author Contributions

Daniel Ries conceived the idea, created code, and wrote the paper. Jason Adams conceived the idea, created code, and reviewed the paper. Tyler Ganter created the code and wrote the section for ECE. Joshua Michalenko created the code and plots for TCC.

Acknowledgements

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1:20–3, 2019.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–77, 2017.

Charles Blundell, Julien Cornebise, Kavukcuoglu, and Daan Wiestra. Weight uncertainty in neural networks. *Proceedings of the International Conference on Machine Learning*, 37, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep

learning. *Proceedings of the International Conference on Machine Learning*, 2016.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 3 edition, 2013.

Alex Graves. Practical variational inference for neural networks. *Conference on Neural Information Processing Systems*, 2011.

Kathryn Gray, Daniel Ries, and Joshua Zollweg. Low-shot, semi-supervised, uncertainty quantification enabled model for high consequence hsi data. *Accepted to IEEE Aerospace*, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *Proceedings of the International Conference on Machine Learning*, 2017.

Eyke Hüllemeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110: 457–506, 2021.

H. M. Dipu Kabir, Abbas Khosravi, Mohammad Anwar Hosen, and Saeid Nahavandi. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE Access*, 6, 2018.

Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F. Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, 22(9), 2011.

Benjamin Kompa, Jasper Snoek, and Andrew Beam. Empirical frequentist coverage of deep learning uncertainty quantification procedures. *arXiv preprint arXiv:2010.03039*, 2021a.

Benjamin Kompa, Jasper Snoek, and Andrew Beam. Second opinion needed: communicating uncertainty in medical machine learning. *npj Digital Medicine*, 4, 2021b.

Anna-Kathrin Kopetzki, Bertrand Charpentier, Daniel Zügner, Sandhya Giri, and Stephan Günnemann. Evaluating robustness of predictive uncertainty estimation: Are dirichlet-based models reliable? *Proceedings of the International Conference on Machine Learning*, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Conference on Neural Information Processing Systems*, 2017a.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Conference on Neural Information Processing System*, 2017b.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521, 2015.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. *Proceedings of the International Conference on Machine Learning*, 48, 2016.

David J.C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–72, 1992.

David J.C. MacKay. Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Computation in Neural Systems*, 6:469–505, 1995.

Audrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Conference on Neural Information Processing System*, 2018.

Abdar Moloud, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Caom Xiaochun, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications, and challenges. *Information Fusion*, 76:243–297, 2021.

M. P. Naeini, G. F. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *AAAI Conference on Artificial Intelligence*, 2015.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Conference on Neural Information Processing System*, 2019.

Daniel Ries, Joshua Zollweg, and Jason Adams. Target detection on hyperspectral images using mcmc and vi trained bayesian neural networks. *Accepted to IEEE Aerospace*, 2022.

Garrison Stevens, Sez Atamturktur, Ricardo Lebensohn, and George Kaschner. Experiment-based validation and uncertainty quantification of coupled multi-scale plasticity models. *Multidiscipline Modeling in Materials and Structures*, 12(1):151–176, 2016.

David J. Stracuzzi, Michael C. Darling, Matthew G. Peterson, and Maximillian G. Chen. Quantifying uncertainty to improve decision making in machine learning. Sand 2018-111666, Sandia National Laboratories, Albuquerque, NM, 2018. URL `https://www.osti.gov/servlets/purl/1481629/`.

Niclas Ståhl, Göran Falkman, Alexander Karlsson, and Gunnar Mathiason. Evaluation of uncertainty quantification in deep learning. *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1237: 556–568, 2020.

Timothy G. Trucano. Uncertainty quantification and the department of homeland security. Sand 2004-2411p, Sandia National Laboratories, Albuquerque, NM, 2004. URL `https://cfwebprod.sandia.gov/cfdocs/CompResearch/docs/SAND2004-2411P.pdf`.

Yixin Wang and David M. Blei. Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114:1147–61, 2019.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *Proceedings of the International Conference on Machine Learning*, 2020.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for bayesian neural network inference. *Proceedings of the International Conference on Machine Learning*, 2019.

Michael Zamo and Philippe Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Mathematical Geosciences*, 50:209–234, 2018.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. *Proceedings of the International Conference on Machine Learning*, 2018.

SUPPLEMENTAL MATERIAL

DETAILS ON BAYESIAN COMPUTATION

Markov Chain Monte Carlo is a generic algorithm applied to Bayesian statistics whose stationary distribution is the posterior distribution of interest (Equation (1)). The Markov chain is usually formulated as irreducible and aperiodic, which will ensure that the stationary distribution exists. Simulation from this chain gives an approximation to the posterior distribution.Although other approaches are possible, Hamiltonian Monte Carlo (HMC) was applied specifically to BNN (Neal [1996]) for efficiency reasons.

HMC augments each parameter (weight) with its own momentum variable, ϕ, which evolves as a function of the gradient of the log posterior, ∇_θ. From Gelman et al. [2013]: Draw samples of θ by repeating:

1. Initialize ϕ^t, θ^t (use previous draw for θ for $t > 0$)
2. For L steps:
   (a) $\phi^* = \phi^{t/*\text{ for L} > 1} + \epsilon\nabla_\theta$
   (b) $\theta^* = \theta^{t/*\text{ for L} > 1} + \epsilon M^{-1}\phi^*$
3. $r = \frac{p(\theta^*|\mathcal{D})p(\phi^*)}{p(\theta^{t-1}|\mathcal{D})p(\phi^{t-1})}$
4. Set $\theta^t = \theta^*$ wp min(r,1), θ^{t-1} otherwise,

and ϵ, L, and M are hyperparameters used to tune the acceptance rate.

VI takes a different approach, and seeks to find an approximation distribution, $q_\phi(\theta)$, parameterized by ϕ, that is close to the posterior distribution $p(\theta|y)$. A common measure of distance between distributions is the Kullback-Leibler distance, and VI's approximation to the posterior, $q_\phi(\theta)$, is obtained by:

$$\arg\min_\phi KL(q_\phi(\theta)||p(\theta|\mathcal{D}) = -E_q \log\left(\frac{p(\theta|\mathcal{D})}{q_\phi(\theta)}\right) \quad (17)$$

$$\ldots = -E_q \log\left(\frac{p(\theta, \mathbf{y}|\mathbf{X})}{q_\phi(\theta)}\right). \quad (18)$$

It is a straightforward exercise to show the equivalence in Equation (18). In practice, $q(\cdot)$ is taken to be a parameterized family, often Normal. Wang and Blei [2019] proved Frequentist consistency and asymptotic Normality of VI in different cases, providing an asymptotic theoretical justification for taking $q(\cdot)$ to be Normal. After optimizing with respect to ϕ, an approximation to the $p(\theta|\mathcal{D})$ is available, and CIs can be computed.

MODEL ESTIMATES AND UQ FOR REGRESSION SIMULATION

Figures 10, 11, 12, 13, 14, 15 show the fitted models with 90% credible intervals for the oracle, BNN-MCMC, BNN-VI, DE, bootstrap NN, and GP, respectively. The oracle model has fairly consistent uncertainty in the training and testing areas due to the known model form. The BNN-MCMC and GP have high uncertainty in the area without training data, reflecting the model's flexibility. BNN-VI interestingly, has fairly constant uncertainty across regions with and without training data. This causes concern that the estimation method will not adapt its uncertainty estimates based on data availability and proximity. Both the DE and bootstrap NN are relatively unaffected by the lack of training data in the middle, and neither produces predictions with much uncertainty. This is not particularly surprising for the DE since it is only accounting for optimization uncertainty, but the bootstrap NN should be able to account for sampling uncertainty by resampling the data with replacement.

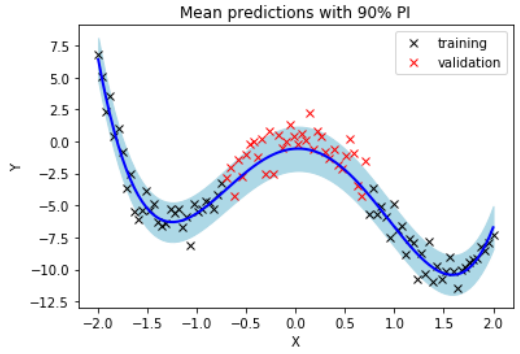

Figure 10: Oracle model predictions and uncertainties to simulated data in Equation (1).

Figures 16 and 17 show the testing and training MSE, respectively, as functions of number of nodes. Overall, the BNN-MCMC and GP are most aligned with the oracle model while bootstrap NN and DE do not perform as well.

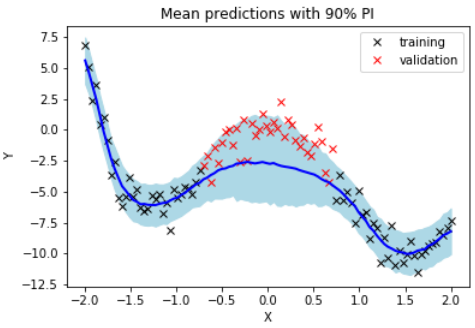

Figure 11: BNN model with with MCMC predictions and uncertainties to simulated data in Equation (1).

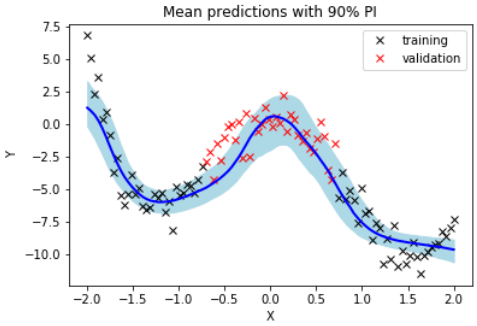

Figure 12: BNN model fit with VI predictions and uncertainties to simulated data in Equation (1).

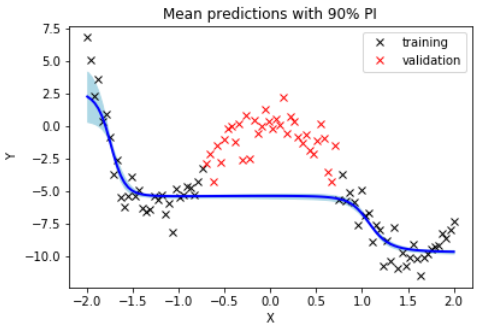

Figure 13: Deep Ensemble model fit with VI predictions and uncertainties to simulated data in Equation (1).

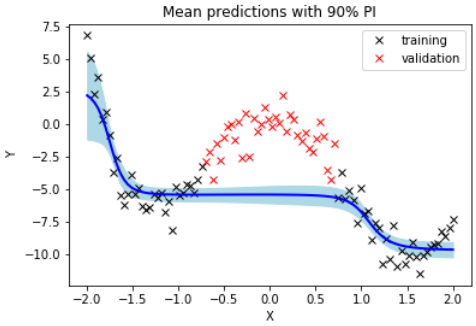

Figure 14: Bootstrap NN model fit with VI predictions and uncertainties to simulated data in Equation (1).

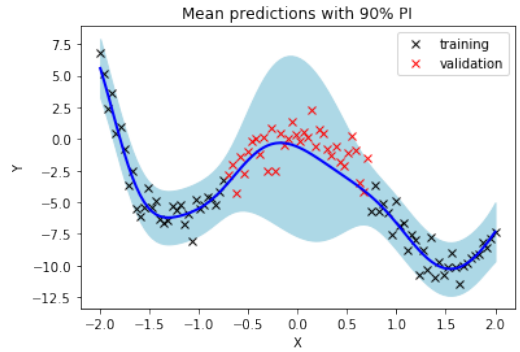

Figure 15: Gaussian Process model fit with VI predictions and uncertainties to simulated data in Equation (1).

TCC DATA GENERATION MODEL

As mentioned in Section 4.2, the TCC dataset is generated from a fully parameterized GMM model that undergoes a particular set of rotation and scaling transformations. The end result is a dataset that allows direct coverage and interval length metric calculations on ground truth data. For the specific parameterization of this paper, we used a GMM model with 2 clusters in a 50/50 cluster distribution probability. Figure 18 shows the density of the TCC simulated data before the transformation shown in Figure 6 of the main paper.

GMM sampling occurs in a 2 step process where a cluster label y is initially assigned, followed by sampling from the associated Gaussian distribution. More specifically, a cluster label $y = \{0, 1\}$ is sampled under a uniform distribution, then a 2-d vector $\mathbf{z} = (z_1, z_2)$ is sampled from the associated Gaussian distribution. For the TCC transformation, the $\mathbf{z}$ vector is passed through an initial scaling transformation followed by a rotation transformation centered upon the origin. The entire transformation is defined by a cascade of equations 22 followed by 23. The effect of the scaling transformation is designed to simulate a 'black-hole' effect, where samples closer to the origin are pulled in faster than samples with larger $\|\mathbf{z}\|_2$. Similarly, the rotation transformation is designed to create a 'spiral' effect, where samples further from the origin (i.e. large $\|\mathbf{z}\|_2$) are rotated more than samples closer to the origin. All together, these two simple transformations convert the original data and label space shown in Figure 18 to that of Figure 6. Two important parameters that define the transformation are λ_{scale} and λ_{rot}, for which we use values of 0.2 and 0.4 respectively.

$$\mathbf{S}(\mathbf{z}; \lambda_{\text{scale}}) = r_{\text{new}} \begin{bmatrix} \cos(\tan^{-1}(\mathbf{z})) & 0 \\ 0 & \sin(\tan^{-1}(\mathbf{z})) \end{bmatrix} \quad (19)$$

$$\text{where} \quad \mathrm{r}_{\text{new}} = \|\mathbf{z}\|_2 (1 - e^{-\lambda_{\text{scale}} * \|\mathbf{z}\|_2}) \quad (20)$$

$$\mathbf{R}(\mathbf{z}; \lambda_{\text{rot}}) = \begin{bmatrix} \cos(\lambda_{\text{rot}} \|\mathbf{z}\|_2) & -\sin(\lambda_{\text{rot}} \|\mathbf{z}\|_2) \\ \sin(\lambda_{\text{rot}} \|\mathbf{z}\|_2) & \cos(\lambda_{\text{rot}} \|\mathbf{z}\|_2) \end{bmatrix} \quad (21)$$

$$\mathbf{z}_{\text{scaled}} = \mathbf{S}(\mathbf{z}; \lambda_{\text{scale}}) \mathbf{z} \quad (22)$$

$$\mathbf{x} = \mathbf{R}(\mathbf{z}_{\text{scaled}}; \lambda_{\text{rot}}) \mathbf{z}_{\text{scaled}} \quad (23)$$

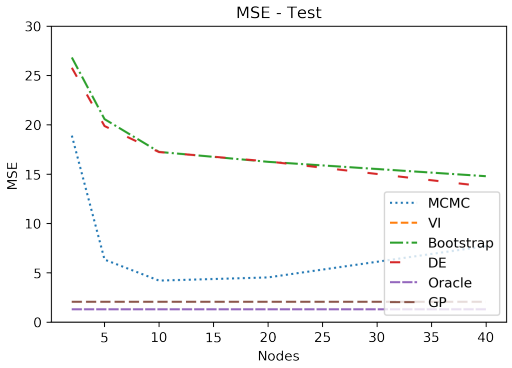

Figure 16: MSE for **test** set of regression simulation. Y-limits exclude BNN-VI for ease of comparison

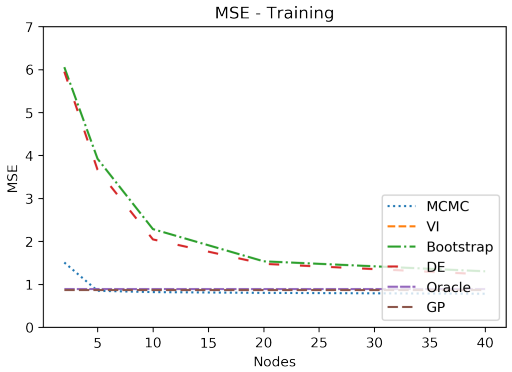

Figure 17: MSE for **train** set of regression simulation. Y-limits exclude BNN-VI for ease of comparison

MODEL ESTIMATES AND UQ FOR TCC SIMULATION

Figures 19, 21, 23, 25, and 27 show the estimation surfaces, $\pi(\mathbf{x}^*, \hat{\theta})$, for BNN-MCMC, BNN-VI, DE, bootstrap NN, and GP, respectively. For the DL models, the results in these plots are from the 10-node models. Figures 20, 22, 24, 26, and 28 show the interval widths of 90% CIs on the domain surface, respectively. These interval widths provide an easy way to understand the uncertainties for TCC since estimated class probabilities must lie $\in [0, 1]$.

The estimation surfaces for all methods except the GP are similar. The GP appears to also be measuring the density of the domain as well as class probabilities, potentially giving it an OOD measure. The interval widths among all the methods except GP are also similar. The main difference is that the DE's uncertainty doesn't fan out as quickly as it departs from training data. This behavior is expected for the same reasons as described in the regression case; DE don't account for sampling variation. Unlike in the regression simulation, the bootstrap NN appears to capture model parameter uncertainties better in TCC as it resembles the uncertainty surface of BNN-MCMC more closely. The GP

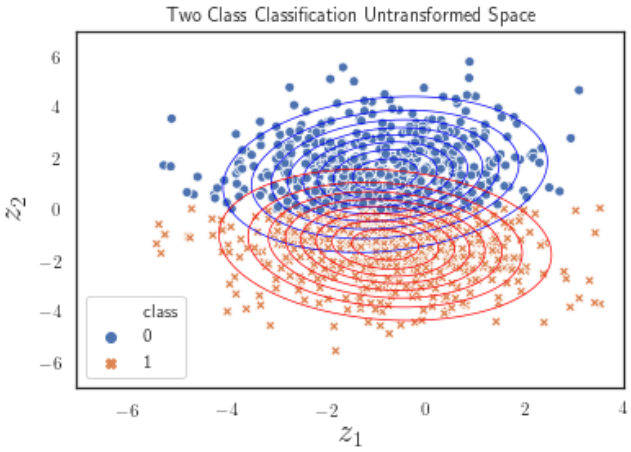

Figure 18: TCC untransformed space with 10% contours for $P(Y = y|z_1, z_2)$.

uncertainty surface looks similar to its estimation surface in that it is not estimating a decision boundary, but rather each class density. Points far from either central density have high uncertainty. This again could be advantageous for UQ in OOD problems.

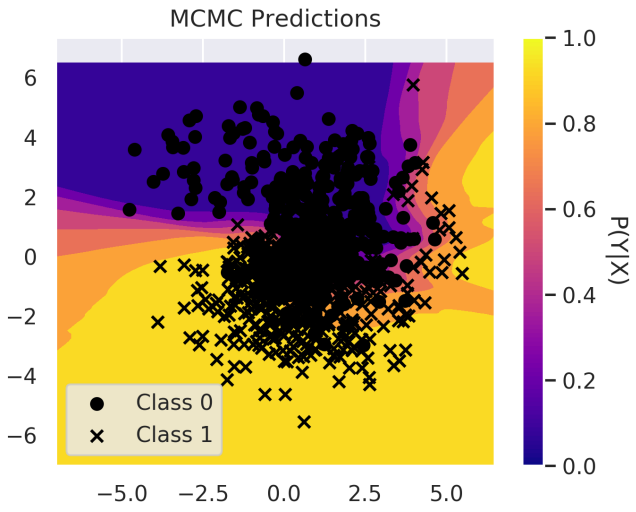

Figure 19: BNN MCMC preditions on TCC.

CALIBRATION PLOTS

Figures 29, 30, 31, 32, 33 show confidence histograms (top) and reliability diagrams (bottom) for models with 10 nodes per layer fit to the TCC data. These diagrams mimic those seen in Guo et al. [2017], from which ECE is directly calculated.

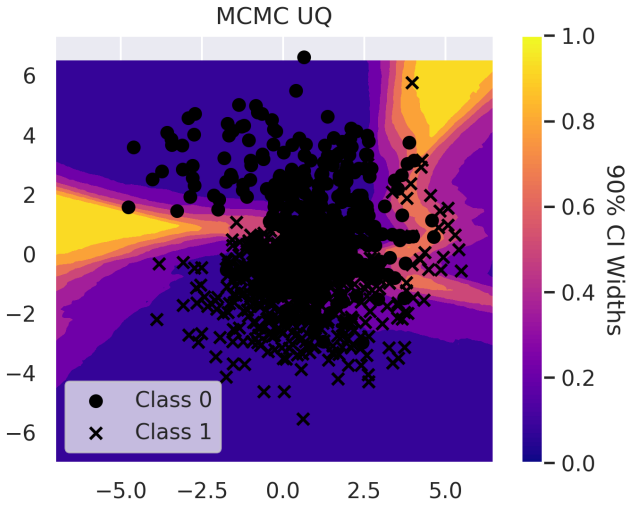

Figure 20: BNN MCMC UQ on TCC.

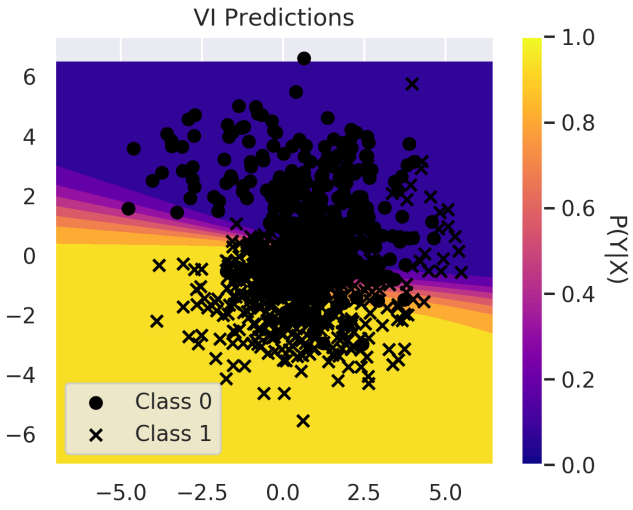

Figure 21: BNN VI preditions on TCC.

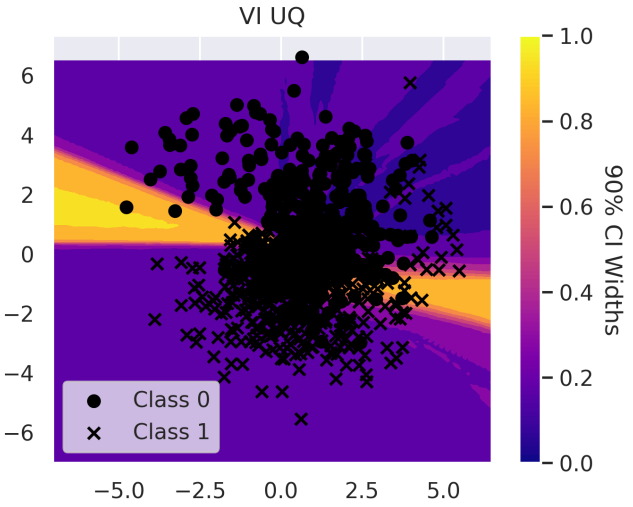

Figure 22: BNN VI UQ on TCC.

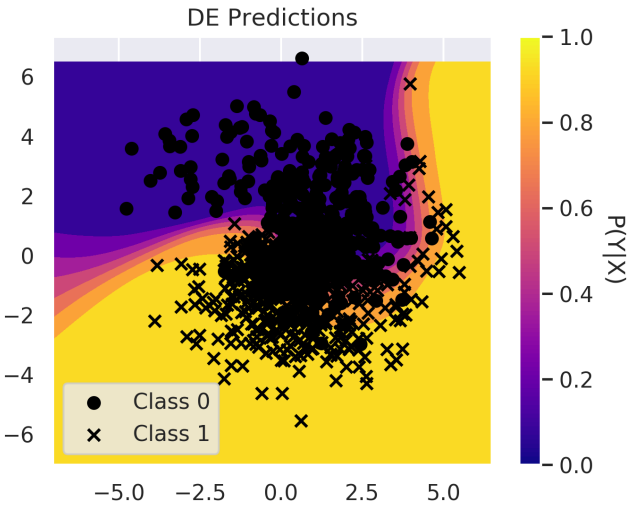

Figure 23: DE preditions on TCC.

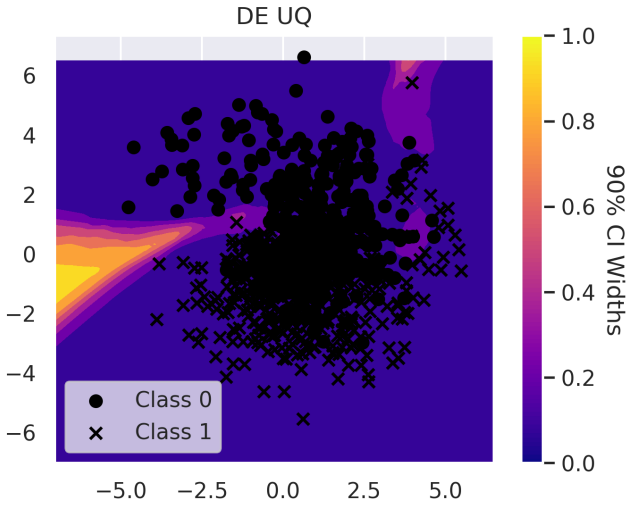

Figure 24: DE UQ on TCC.

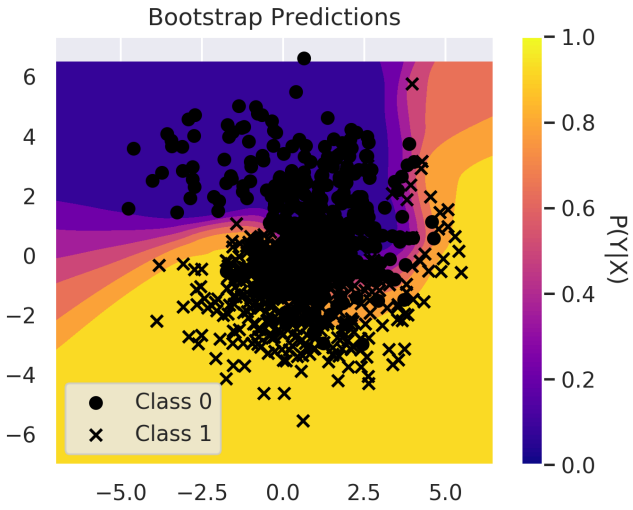

Figure 25: Bootstrap preditions on TCC.

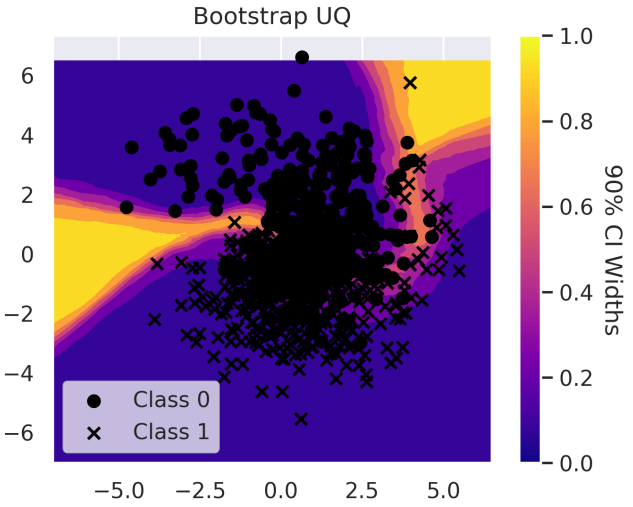

Figure 26: Bootstrap UQ on TCC.

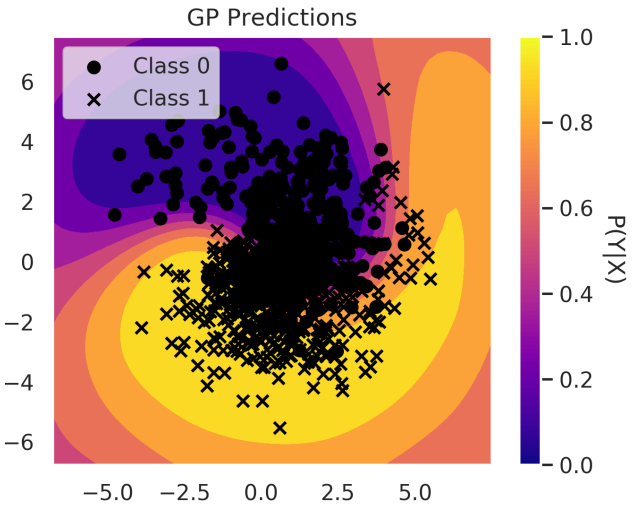

Figure 27: GP preditions on TCC.

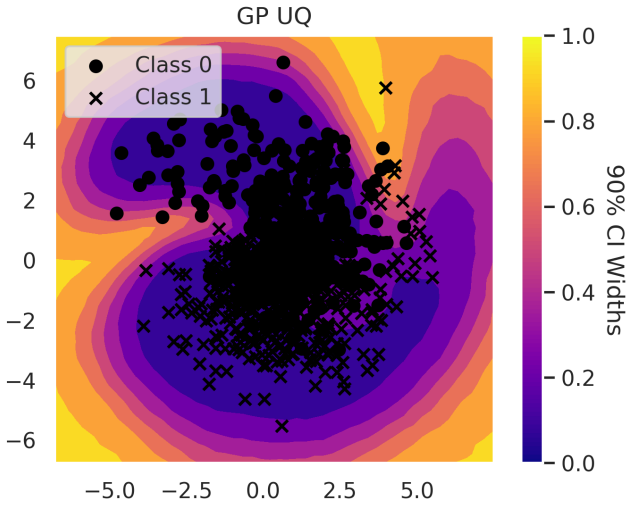

Figure 28: GP UQ on TCC.

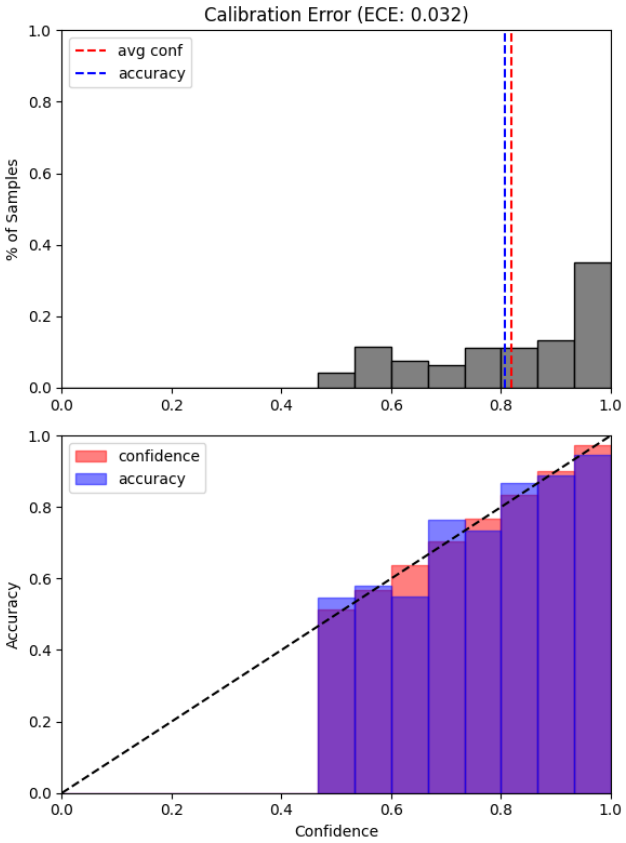

Figure 29: Calibration plot for Bootstrapped NN on TCC with 10 nodes per layer.

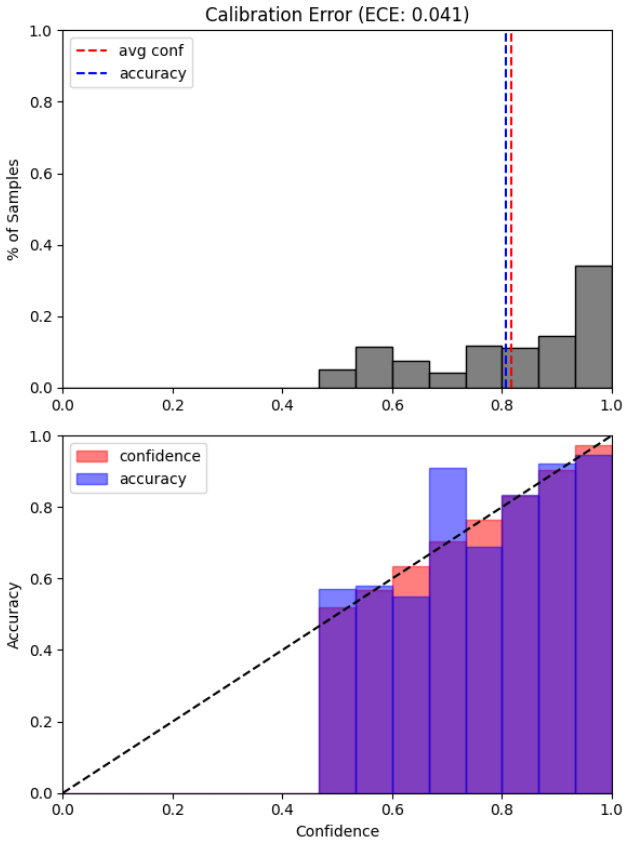

Figure 30: Calibration plot for DE on TCC with 10 nodes per layer.

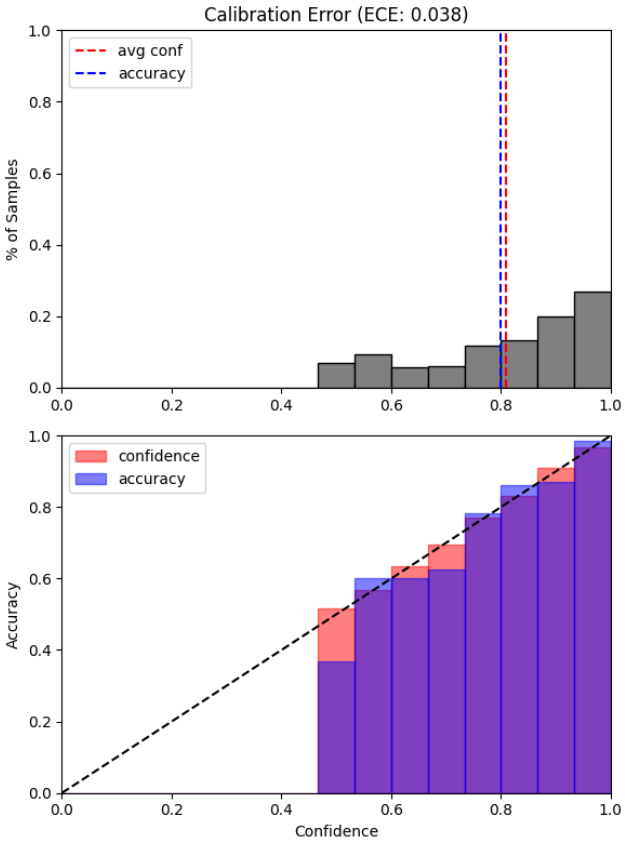

Figure 31: Calibration plot for BNN-MCMC on TCC with 10 nodes per layer.

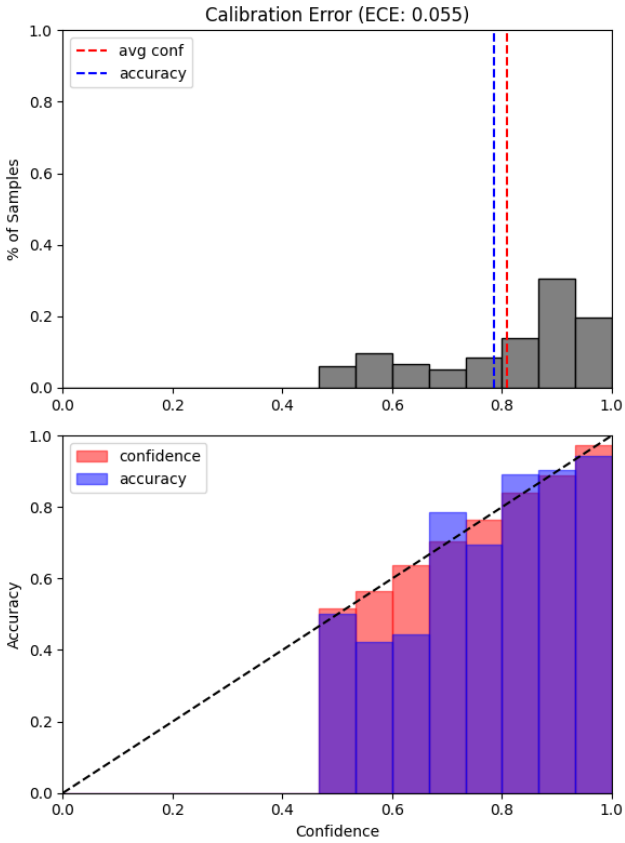

Figure 32: Calibration plot for BNN-VI on TCC with 10 nodes per layer.

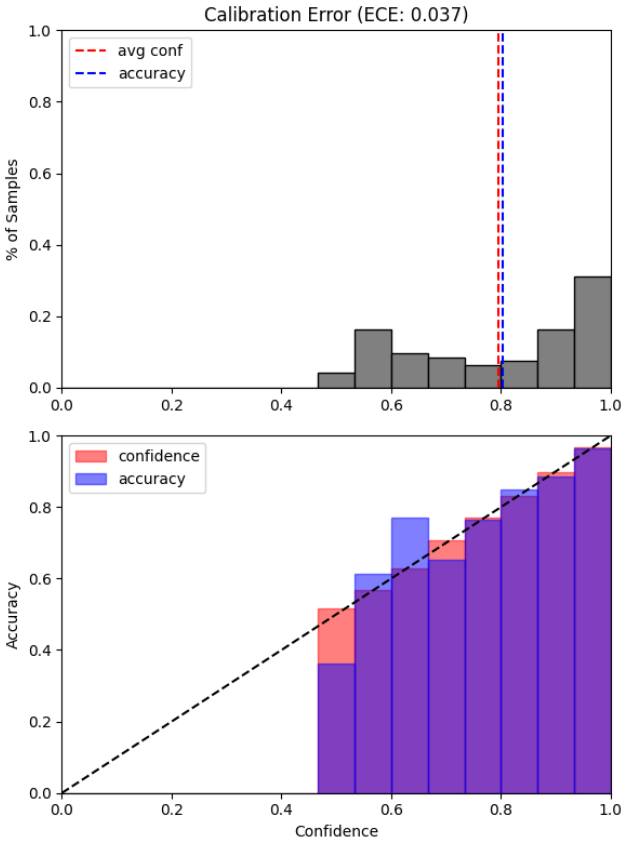

Figure 33: Calibration plot for GP on TCC with 10 nodes per layer.