Post-hoc Uncertainty Learning using a Dirichlet Meta-Model

Maohao Shen[1], **Yuheng Bu**[2], **Prasanna Sattigeri**[3], **Soumya Ghosh**[3], **Subhro Das**[3], **Gregory Wornell**[1]

[1] Massachusetts Institute of Technology
[2] University of Florida
[3] MIT-IBM Watson AI Lab, IBM Research
maohao@mit.edu

Abstract

It is known that neural networks have the problem of being over-confident when directly using the output label distribution to generate uncertainty measures. Existing methods mainly resolve this issue by retraining the entire model to impose the uncertainty quantification capability so that the learned model can achieve desired performance in accuracy and uncertainty prediction simultaneously. However, training the model from scratch is computationally expensive and may not be feasible in many situations. In this work, we consider a more practical post-hoc uncertainty learning setting, where a well-trained base model is given, and we focus on the uncertainty quantification task at the second stage of training. We propose a novel Bayesian meta-model to augment pretrained models with better uncertainty quantification abilities, which is effective and computationally efficient. Our proposed method requires no additional training data and is flexible enough to quantify different uncertainties and easily adapt to different application settings, including out-of-domain data detection, misclassification detection, and trustworthy transfer learning. We demonstrate our proposed meta-model approach's flexibility and superior empirical performance on these applications over multiple representative image classification benchmarks.

1 Introduction

Despite the promising performance of deep neural networks achieved in various practical tasks (Simonyan and Zisserman 2014; Ren et al. 2015; Hinton et al. 2012; Mikolov et al. 2010; Alipanahi et al. 2015; Litjens et al. 2017), uncertainty quantification (UQ) has attracted growing attention in recent years to fulfill the emerging demand for more robust and reliable machine learning models, as UQ aims to measure the reliability of the model's prediction quantitatively. Accurate uncertainty estimation is especially critical for the field that is highly sensitive to error prediction, such as autonomous driving (Bojarski et al. 2016) and medical diagnosis (Begoli, Bhattacharya, and Kusnezov 2019).

Most state-of-the-art approaches (Gal and Ghahramani 2016; Lakshminarayanan, Pritzel, and Blundell 2017; Malinin and Gales 2018; van Amersfoort et al. 2020) focus on building a deep model equipped with uncertainty quantification capability so that a single deep model can achieve both desired prediction and UQ performance simultaneously. However, such an approach for UQ suffers from practical limitations because it either requires a specific model structure or explicitly training the entire model from scratch to impose the uncertainty quantification ability. A more realistic scenario is to quantify the uncertainty of a pretrained model in a post-hoc manner due to practical constraints. For example, (1) compared with prediction accuracy and generalization performance, uncertainty quantification ability of deep learning models are usually considered with lower priority, especially for profit-oriented applications, such as recommendation systems; (2) some applications require the models to impose other constraints, such as fairness or privacy, which might sacrifice the UQ performance; (3) for some applications such as transfer learning, the pretrained models are usually available, and it might be a waste of resources to train a new model from scratch.

Motivated by these practical concerns, we focus on tackling the post-hoc uncertainty learning problem, i.e., given a pretrained model, how to improve its UQ quality without affecting its predictive performance. Prior works on improving uncertainty quality in a post-hoc setting have been mainly targeted towards improving calibration (Guo et al. 2017; Kull et al. 2019). These approaches typically fail to augment the pre-trained model with the ability to capture different sources of uncertainty, such as epistemic uncertainty, which is crucial for applications such Out-of-Distribution (OOD) detection. Several recent works (Chen et al. 2019; Jain et al. 2021) have adopted the meta-modeling approach, where a meta-model is trained to predict whether or not the pretrained model is correct on the validation samples. These methods still rely on a point estimate of the meta-model parameters, which can be unreliable, especially when the validation set is small.

In this paper, we propose a novel Bayesian meta-model-based uncertainty learning approach to mitigate the aforementioned issues. Our proposed method requires no additional data other than the training dataset and is flexible enough to quantify different kinds of uncertainties and easily adapt to different application settings. Our empirical results provides crucial insights regarding meta-model training: (1) The diversity in feature representations across different layers is essential for uncertainty quantification, especially for out-of-domain (OOD) data detection tasks; (2) Leveraging

the Dirichlet meta-model to capture different uncertainties, including total uncertainty and epistemic uncertainty; (3) There exists an over-fitting issue in uncertainty learning similar to supervised learning that needs to be addressed by a novel validation strategy to achieve better performance. Furthermore, we show that our proposed approach has the flexibility to adapt to various applications, including OOD detection, misclassification detection, and trustworthy transfer learning.

2 Related Work

Uncertainty Quantification methods can be broadly classified as *intrinsic* or *extrinsic* depending on how the uncertainties are obtained from the machine learning models. Intrinsic methods encompass models that inherently provide an uncertainty estimate along with its predictions. Some intrinsic methods such as neural networks with homoscedastic/heteroscedastic noise models (Wakefield 2013) and quantile regression (Koenker and Bassett 1978) can only capture *Data* (aleatoric) uncertainty. Many applications including out-of-distribution detection, requires capturing both *Data* (aleatoric) and *Model* (epistemic) accurately. Bayesian methods such as Bayesian neural networks (BNNs) (Neal 2012; Blundell et al. 2015; Welling and Teh 2011) and Gaussian processes (Rasmussen and Williams 2006) and ensemble methods (Lakshminarayanan, Pritzel, and Blundell 2017) are well known examples of intrinsic methods that can quantify both uncertainties. However, Bayesian methods and ensembles can be quite expensive and require several approximations to learn/optimize in practice (MacKay 1992; Kristiadi, Hein, and Hennig 2021; Welling and Teh 2011). Other approaches attempt to alleviate these issues by directly parameterizing a Dirichlet prior distribution over the categorical label proportions via a neural network (Malinin and Gales 2018; Sensoy, Kaplan, and Kandemir 2018; Malinin and Gales 2019; Nandy, Hsu, and Lee 2020; Charpentier, Zügner, and Günnemann 2020; Joo, Chung, and Seo 2020).

Under model misspecification, Bayesian approaches are not well-calibrated and can produce severely miscalibrated uncertainties. In the particular case of BNNs, sparsity-promoting priors have been shown to produce better-calibrated uncertainties, especially in the small data regime (Ghosh, Yao, and Doshi-Velez 2019), and somewhat alleviate the issue. Improved approximate inference methods and methods for prior elicitation in BNN models are active areas of research for forcing BNNs to produce better-calibrated uncertainties. Frequentist methods that approximate the jackknife have also been proposed to construct calibrated confidence intervals (Alaa and Van Der Schaar 2020).

For models without an inherent notion of uncertainty, extrinsic methods are employed to extract uncertainties in a post-hoc manner. The post-hoc UQ problem is still under-explored, and few works focus on tackling this problem. One such approach is to build auxiliary or meta-models, which have been used successfully to generate reliable confidence measures (in classification) (Chen et al. 2019), prediction intervals (in regression), and to predict performance metrics such as accuracy on unseen and unlabeled data (Elder et al. 2020). Similarly, DEUP (Jain et al. 2021) trains an error predictor to estimate the epistemic uncertainty of the pretrained model in terms of the difference between generalization error and aleatoric uncertainty. LULA (Kristiadi, Hein, and Hennig 2021) trains additional hidden units building on layers of MAP-trained model to improve its uncertainty calibration with Laplace approximation. However, many of these methods require additional data samples that are either validation or out of distribution dataset to train or tune the hyper-parameter, which is infeasible when these data are not available. Moreover, they are often not flexible enough to distinguish epistemic uncertainty and aleatoric uncertainty, which are known to be crucial in various learning applications (Hüllermeier and Waegeman 2021). In contrast, our proposed method does not require additional training data or modifying the training procedure of the base model.

3 Problem Formulation

We focus on classification problems in this paper. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the input space, and $\mathcal{Y} = \{1, \cdots, K\}$ denotes the label space. Given a base-model training set $\mathcal{D}_B = \left\{\boldsymbol{x}_i^B, y_i^B\right\}_{i=1}^{N_B} \in \mathcal{Z}^{N_B}$ containing i.i.d. samples generated from the distribution P_Z^B, a pretrained base model $\boldsymbol{h} \circ \Phi : \mathcal{X} \to \Delta^{K-1}$ is constructed, where $\Phi : \mathcal{X} \to \mathbb{R}^l$ and $\boldsymbol{h} : \mathbb{R}^l \to \Delta^{K-1}$ denote two complementary components of the neural network. More specifically, $\Phi(\boldsymbol{x}) = \Phi(\boldsymbol{x}; \boldsymbol{w}_\phi)$ stands for the intermediate feature representation of the base model, and the model output $\boldsymbol{h}(\Phi(\boldsymbol{x})) = \boldsymbol{h}(\Phi(\boldsymbol{x}; \boldsymbol{w}_\phi); \boldsymbol{w}_h)$ denotes the predicted label distribution $P_B(\boldsymbol{y} \mid \Phi(\boldsymbol{x})) \in \Delta^{K-1}$ given input sample $\boldsymbol{x}$, where $(\boldsymbol{w}_\phi, \boldsymbol{w}_h) \in \mathcal{W}$ are the parameters of the pretrained base model.

The performance of the base model is evaluated by a non-negative loss function $\ell_B : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}_+$, e.g., cross entropy loss. Thus, a standard way to obtain the pretrained base model is by minimizing the empirical risk over $\mathcal{D}_B$, i.e., $\mathcal{L}_B(\boldsymbol{h} \circ \Phi, \mathcal{D}_B) \triangleq \frac{1}{N_B} \sum_{i=1}^{N_B} \ell_B\left(\boldsymbol{h} \circ \Phi, (\boldsymbol{x}_i^B, y_i^B)\right)$.

Although the well-trained deep base model is able to achieve good prediction accuracy, the output label distribution $P_B(\boldsymbol{y} \mid \Phi(\boldsymbol{x}))$ is usually unreliable for uncertainty quantification, i.e., it can be overconfident or poorly calibrated. Without retraining the model from scratch, we are interested in improving the uncertainty quantification performance in an efficient post-hoc manner. We utilize a meta-model $\boldsymbol{g} : \mathbb{R}^l \to \tilde{\mathcal{Y}}$ with parameter $\boldsymbol{w}_g \in \mathcal{W}_g$ building on top of the base model. The meta-model shares the same feature extractor from the base model and generates an output $\tilde{\boldsymbol{y}} = \boldsymbol{g}(\Phi(\boldsymbol{x}); \boldsymbol{w}_g)$, where $\tilde{\boldsymbol{y}} \in \tilde{\mathcal{Y}}$ can take any form, e.g., a distribution over Δ^{K-1} or a scalar. Given a meta-model training set $\mathcal{D}_M = \left\{\boldsymbol{x}_i^M, y_i^M\right\}_{i=1}^{N_M}$ with i.i.d. samples from the distribution P_Z^M, our goal is to obtain the meta-model by optimizing a training objective $\mathcal{L}_M(\boldsymbol{g} \circ \Phi, \mathcal{D}_M) \triangleq \frac{1}{N_M} \sum_{i=1}^{N_M} \ell_M\left(\boldsymbol{g} \circ \Phi, (\boldsymbol{x}_i^M, y_i^M)\right)$, where $\ell_M : \mathcal{W}_g \times \mathcal{W}_\phi \times \mathcal{Z} \to \mathbb{R}_+$ is the loss function for the meta-model.

In the following, we formally introduce the post-hoc uncertainty learning problem using meta-model.

Problem 1. *[Post-hoc Uncertainty Learning by Meta-model] Given a base model $\boldsymbol{h} \circ \Phi$ learned from the base-model training set $\mathcal{D}_B$, the uncertainty learning problem by meta-model is to learn the function $\boldsymbol{g}$ using the meta-model training set $\mathcal{D}_M$ and the shared feature extractor Φ, i.e.,*

$$\boldsymbol{g}^* = \arg\min_{\boldsymbol{g}} \mathcal{L}_M(\boldsymbol{g} \circ \Phi, \mathcal{D}_M), \tag{1}$$

such that the output from the meta-model $\tilde{\boldsymbol{y}} = \boldsymbol{g}(\Phi(x))$ equipped with an uncertainty metric function $\boldsymbol{u} : \tilde{\mathcal{Y}} \to \mathbb{R}$ is able to generate a robust uncertainty score $\boldsymbol{u}(\tilde{\boldsymbol{y}})$.

Next, the most critical questions are how the meta-model should use the information extracted from the pretrained base model, what kinds of uncertainty the meta-model should aim to quantify, and finally, how to train the meta-model appropriately.

4 Method

In this section, we specify the post-hoc uncertainty learning framework defined in Problem 1. First, we introduce the structure of the meta-model. Next, we discuss the meta-model training procedure, including the training objectives and a validation trick. Finally, we define metrics for uncertainty quantification used in different applications.

The design of our proposed meta-model method is based on three high-level insights: (1) Different intermediate layers of the base model usually capture various levels of feature representation, from low-level features to high-frequency features, e.g., for OOD detection task, the OOD data is unlikely to be similar to in-distribution data across all levels of feature representations. Therefore, it is crucial to leverage the diversity in feature representations to achieve better uncertainty quantification performance. (2) Bayesian method is known to be capable of modeling different types of uncertainty for various uncertainty quantification applications, i.e., total uncertainty and epistemic uncertainty. Thus, we propose a Bayesian meta-model to parameterize the Dirichlet distribution, used as the conjugate prior distribution over label distribution. (3) We believe that the overconfident issue of the base model is caused by over-fitting in supervised learning with cross-entropy loss. In the post-hoc training of the meta-model, a validation strategy is proposed to improve the performance of uncertain learning instead of prediction accuracy.

Before discussing the details of our proposed method, we want to use a toy example of the OOD detection task shown in Figure 1 to elaborate our insights. The goal is to improve the OOD (FashionMNIST) detection performance of a LeNet base model trained on MNIST. The meta-model takes base-model intermediate feature representation as input and parameterizes a Dirichlet distribution over the probability simplex. We train three different meta-models using both intermediate layer features, using either of the two features, respectively, and then visualize and compare the output Dirichlet distribution on the simplex. Specifically, we take the three dominant classes with the largest logits to approximately visualize the Dirichlet distribution over the probability simplex. We observe that the meta-model outputs a much sharper distribution for the in-distribution sample than the OOD sample. Moreover, compared to the meta-model trained with one feature, the meta-model trained with multiple intermediate layers can further distinguish the two distributions on simplex. They generate sharper distribution for the in-distribution sample while exhibiting a more uniform distribution for the OOD sample, which strongly supports our key claims.

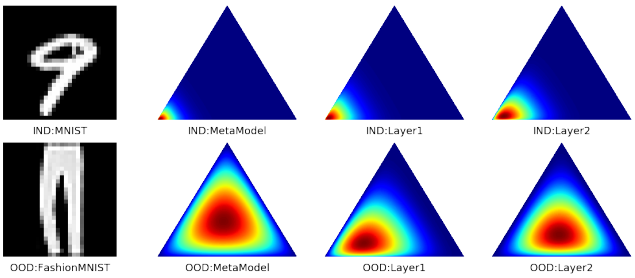

Figure 1: A toy example of our proposed meta-model method in OOD detection application shows the diversity of features in different layers. MetaModel utilizes two intermediate features, while Layer1 and Layer2 only are trained with one individual feature.

4.1 Meta-Model Structure

The proposed meta-model consists of multiple linear layers $\{\boldsymbol{g}_j\}_{j=1}^m$ attached to different intermediate layers from the base model, and a final linear layer $\boldsymbol{g}_c$ that combines all the features and generates a single output. Specifically, given an input sample $\boldsymbol{x}$, denote the multiple intermediate feature representation extracted from the base-model as $\{\Phi_j(\boldsymbol{x})\}_{j=1}^m$. For each intermediate base-feature Φ_j, the corresponding linear layer will construct a low-dimensional meta-feature $\{\boldsymbol{g}_j(\Phi_j(\boldsymbol{x}))\}_{j=1}^m$. Then, the final linear layer of the meta-model takes the multiple meta-features as inputs and generates a single output, i.e., $\tilde{\boldsymbol{y}} = \boldsymbol{g}\left(\{\Phi_j(\boldsymbol{x})\}_{j=1}^m; \boldsymbol{w}_g\right) = \boldsymbol{g}_c\left(\{\boldsymbol{g}_j(\Phi_j(\boldsymbol{x}))\}_{j=1}^m; \boldsymbol{w}_{g_c}\right)$. In practice, the linear layers $\boldsymbol{g}_i$ and $\boldsymbol{g}_c$ only consist of fully connected layers and activation function, which ensures the meta-model has a much simpler structure and enables efficient training.

Given an input sample $\boldsymbol{x}$, the base model outputs a conditional label distribution $P_B(\boldsymbol{y}|\Phi(\boldsymbol{x})) \in \Delta^{K-1}$, corresponding to a single point in the probability simplex. However, such label distribution $P_B(\boldsymbol{y}|\Phi(\boldsymbol{x}))$ is a point estimate, which only shows the model's uncertainty about different classes but cannot reflect the uncertainty due to the lack of knowledge of a given sample, i.e., the epistemic uncertainty. To this end, we adopt the Dirichlet technique commonly used in the recent literature (Malinin and Gales 2018, 2019; Nandy, Hsu, and Lee 2020; Charpentier, Zügner, and Günnemann 2020) in order to better quantify the epistemic uncertainty. Let the label distribution as a random variable over probability simplex, denote as $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_k]$, the Dirichlet prior distribution is the conjugate prior of the cate-

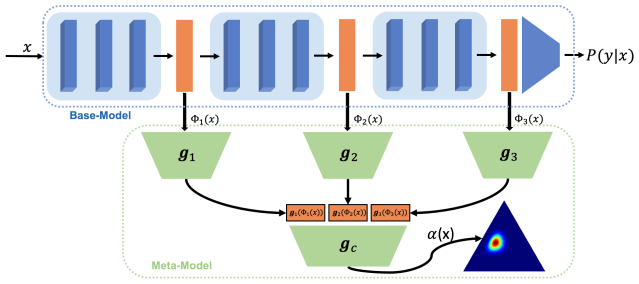

Figure 2: Meta-Model structure

gorical distribution, i.e.,

$$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \triangleq \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)}\prod_{c=1}^{K}\pi_c^{\alpha_c-1}, \alpha_c > 0, \alpha_0 \triangleq \sum_{c=1}^{K}\alpha_c, \tag{2}$$

Our meta-model $\boldsymbol{g}$ explicitly parameterize the posterior Dirichlet distribution, i.e,

$$q(\boldsymbol{\pi}|\Phi(\boldsymbol{x}); \boldsymbol{w}_g) \triangleq \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}(\boldsymbol{x})),\ \boldsymbol{\alpha}(\boldsymbol{x}) = e^{\boldsymbol{g}(\Phi(\boldsymbol{x});\boldsymbol{w}_g)},$$

where the output of our meta-model is $\tilde{\boldsymbol{y}} = \log\boldsymbol{\alpha}(\boldsymbol{x})$, and $\boldsymbol{\alpha}(\boldsymbol{x}) = [\alpha_1(\boldsymbol{x}), \alpha_2(\boldsymbol{x}), ..., \alpha_k(\boldsymbol{x})]$ is the concentration parameter of the Dirichlet distribution given an input $\boldsymbol{x}$. The overall structure of the meta-model is shown in Figure 2.

4.2 Uncertainty Learning

Training Objective From Bayesian perspective, the predicted label distribution using the Dirichlet meta-model is given by the expected categorical distribution:

$$q(y=c|\Phi(\boldsymbol{x}); \boldsymbol{w}_g) = \mathbb{E}_{q(\boldsymbol{\pi}|\Phi(\boldsymbol{x});\boldsymbol{w}_g)}[P(y=c|\boldsymbol{\pi})] = \frac{\alpha_c(\boldsymbol{x})}{\alpha_0(\boldsymbol{x})}, \tag{3}$$

where $\alpha_0 = \sum_{c=1}^{K}\alpha_c$ is the precision of the Dirichlet distribution.

The true posterior of the categorical distribution over sample $(\boldsymbol{x}, y)$ is $P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}), y) \propto P(y|\boldsymbol{\pi}, \Phi(\boldsymbol{x}))P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}))$, which is difficult to evaluate. Instead, we utilize a variational inference technique used in (Joo, Chung, and Seo 2020) to generate a variational distribution $q(\boldsymbol{\pi}|\Phi(\boldsymbol{x}); \boldsymbol{w}_g)$ parameterized by Dirichlet distribution with meta-model to approximate the true posterior distribution $P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}), y)$, then minimize the KL-divergence $\mathrm{KL}\left(q(\boldsymbol{\pi}|\Phi(\boldsymbol{x}); \boldsymbol{w}_g)\|P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}), y)\right)$, which is equivalent to maximize the evidence lower bound (ELBO) loss (derivation is provided in Appendix A), i.e.,

$$\begin{aligned}\mathcal{L}_{\mathrm{VI}}(\boldsymbol{w}_g) = &\frac{1}{N_M}\sum_{i=1}^{N_M}\mathbb{E}_{q(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i);\boldsymbol{w}_g)}[\log P(y_i \mid \boldsymbol{\pi}, \boldsymbol{x}_i)] \\ &- \lambda \cdot \mathrm{KL}\left(q(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i); \boldsymbol{w}_g)\|P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i))\right) \quad (4)\\ = &\frac{1}{N_M}\sum_{i=1}^{N_M}\psi\left(\alpha_{y_i}^{(i)}\right) - \psi\left(\alpha_0^{(i)}\right) \\ &- \lambda \cdot \mathrm{KL}\left(\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}^{(i)})\|\,\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\beta})\right), \quad (5)\end{aligned}$$

where $\boldsymbol{\alpha}^{(i)}$ is the Dirichlet concentration parameter parameterized by the meta-model, i.e., $\boldsymbol{\alpha}^{(i)} = e^{(\boldsymbol{g}(\Phi(\boldsymbol{x}_i);\boldsymbol{w}_g))}$, $\boldsymbol{\psi}$ is the digamma function, and $\boldsymbol{\beta}$ is the predefined concentration parameter of prior distribution. In practice, we simply let $\boldsymbol{\beta} = [1, \cdots, 1]$. The likelihood term encourages the categorical distribution to be sharper around the true class on the simplex, and the KL-divergence term can be viewed as a regularizer to prevent overconfident prediction, and λ is the hyper-parameter to balance the trade-off.

Validation for Uncertainty Learning Validation with early stopping is a commonly used technique in supervised learning to train a model with desired generalization performance, i.e., stop training when the error evaluated on the validation set starts increasing. However, we observe that the standard validation method does not work well for uncertainty learning. One possible explanation is that model achieves the highest accuracy when validation loss is small, but may not achieve the best UQ performance, i.e., the model can be overconfident. To this end, we propose a simple and effective validation approach specifically for uncertainty learning. Instead of monitoring the validation cross-entropy loss, we evaluate a specific uncertainty quantification performance metric. For example, we create another noisy validation set for the OOD task by adding noise to the original validation samples and treating such noisy validation samples as OOD samples (more details are provided in the Appendix B.6). We evaluate the uncertainty score $\boldsymbol{u}(\tilde{\boldsymbol{y}})$ on both the validation set and noisy validation set and stop the meta-model training when the OOD detection performance achieves its maximum based on some predefined metrics, e.g., AUROC score. Unlike most existing literature using additional training data to help achieve desired performance (Hendrycks, Mazeika, and Dietterich 2018; Kristiadi, Hein, and Hennig 2021; Malinin and Gales 2018), we nonetheless do not require any additional data for training the meta-model.

4.3 Uncertainty Metrics

In this section, we show that our meta-model has the desired behavior to quantify different uncertainties and how they can be used in various applications.

Total Uncertainty. Total uncertainty, also known as predictive uncertainty, is a combination of epistemic uncertainty and aleatoric uncertainty. The total uncertainty is often used for misclassification detection problems, where the misclassified samples are viewed as in-distribution hard samples. There are two standard ways to measure total uncertainty: (1) **Entropy** (Ent): The Shannon entropy of expected categorical label distribution over the Dirichlet distribution, i.e., $\mathcal{H}\left(P(y|\Phi(\boldsymbol{x}); \boldsymbol{w}_g)\right) = \mathcal{H}\left(\mathbb{E}_{P(\boldsymbol{\pi}|\Phi(\boldsymbol{x});\boldsymbol{w}_g)}[P(y|\boldsymbol{\pi})]\right)$; (2) **Max Probability** (MaxP): The probability of the predicted class in label distribution, i.e., $\max_c P(y=c|\Phi(\boldsymbol{x}); \boldsymbol{w}_g)$.

Epistemic Uncertainty. The epistemic uncertainty quantifies the uncertainty when the model has insufficient knowledge of a prediction, e.g., the case of an unseen data sample. The epistemic uncertainty is especially useful in OOD detection problems. When the meta-model encounters an

unseen sample during testing, it will output a high epistemic uncertainty score due to a lack of knowledge. We define three metrics to measure the epistemic uncertainties. (1) **Differential Entropy** (Dent): Differential entropy measures the entropy of the Dirichlet distribution, a large differential entropy corresponds a more spread Dirichlet distribution, i.e., $\mathcal{H}\left(P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i);\boldsymbol{w}_g)\right) = -\int P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i);\boldsymbol{w}_g) \cdot \log P(\boldsymbol{\pi}|\Phi(\boldsymbol{x}_i);\boldsymbol{w}_g) d\boldsymbol{\pi}$. (2) **Mutual Information** (MI): Mutual Information is the difference between the Entropy (measures total uncertainty) and the expected entropy of the categorical distribution sampled from the Dirichlet distribution (approximates aleatoric uncertainty), i.e., $\mathcal{I}\left(y, \boldsymbol{\pi}|\Phi(\boldsymbol{x}_i)\right) = \mathcal{H}\left(\mathbb{E}_{P(\boldsymbol{\pi}|\Phi(\boldsymbol{x});\boldsymbol{w}_g)}[P(y|\boldsymbol{\pi})]\right) - \mathbb{E}_{P(\boldsymbol{\pi}|\Phi(\boldsymbol{x});\boldsymbol{w}_g)}[\mathcal{H}\left(P(y|\boldsymbol{\pi})\right)]$. (3) **Precision**(Prec): The precision is the summation of the Dirichlet distribution concentration parameter $\boldsymbol{\alpha}$, larger value corresponds to sharper distribution and higher confidence, i.e., $\alpha_0 = \sum_{c=1}^{k} \alpha_c$.

5 Experiment Results

In this section, we will demonstrate the strong empirical performance of our proposed meta-model-based uncertainty learning method: first, We introduce the UQ applications; then we describe the experiment settings; next, we present the main results of the three aforementioned uncertainty quantification applications; and finally, we discuss our takeaways. More experiment results and implementation details are given in the Appendix B and Appendix C.

5.1 Uncertainty Quantification Applications

We primarily focus on three applications that can be tackled using our proposed meta-model approach: (1) **Out of domain data detection.** Given a base-model $\boldsymbol{h}$ trained using data sampled from the distribution P_Z^B. We use the same base-model training set to train the meta-model, i.e., $\mathcal{D}_B = \mathcal{D}_M$. During testing, there exists some unobserved out-of-domain data from another distribution P_Z^{ood}. The meta-model is expected to identify the out-of-distribution input sample based on epistemic uncertainties. (2) **misclassification Detection.** Instead of detecting whether a testing sample is out of domain, the goal here is to identify the failure or success of the meta-model prediction at test time using total uncertainties. (3) **Trustworthy Transfer Learning.** In transfer learning, there exists a pretrained model trained using source task data $\mathcal{D}_s$ sampled from source distribution P_Z^s, and the goal is to adapt the source model to a target task using target data $\mathcal{D}_t$ sampled from target distribution P_Z^t. Most existing transfer learning approaches only focus on improving the prediction performance of the transferred model, but ignore its UQ performance on the target task. Our meta-model method can be utilized to address this problem, i.e, given pretrained source model $\boldsymbol{h}^s \circ \Phi^s$, the meta-model can be efficiently trained using target domain data by $\boldsymbol{g}^t = \arg\min_{\boldsymbol{g}} \mathcal{L}_E(\boldsymbol{g} \circ \Phi^s, \mathcal{D}_t)$.

5.2 Settings

Benchmark. For both OOD detection and misclassification detection tasks, we employ three standard datasets to train the base model and the meta-model: MNIST, CIFAR10, and CIFAR100. For each dataset, we use different base-model structures, i.e., LeNet for MNIST, VGG-16 (Simonyan and Zisserman 2014) for CIFAR10, and WideResNet-16 (Zagoruyko and Komodakis 2016) for CIFAR100. For LeNet and VGG-16, the meta-model uses extracted feature after each pooling layer, and for WideResNet-16, the meta-model uses extracted feature after each residual black. In general, the total number of intermediate features is less than 5 to ensure computational efficiency. For the OOD task, we consider five different OOD datasets for evaluating the OOD detection performance: Omiglot, FashionMNIST, KMNIST, CIFAR10, and corrupted MNIST as outliers for the MNIST dataset; SVHN, FashionMNIST, LSUN, TinyImageNet, and corrupted CIFAR10 (CIFAR100) as outliers for CIFAR10 (CIFAR100) dataset. For the trustworthy transfer learning task, we use the ResNet-50 pretrained on ImageNet as our pretrained source domain model and adapt the source model to the two target tasks, STL10 and CIFAR10, by training the meta-model.

Baselines. For OOD and misclassification tasks, except the naive base-model trained with cross-entropy loss, we mainly compare with the existing post-hoc UQ methods as baselines: (1) The meta-model based method (Whitebox) (Chen et al. 2019); (2) The post-hoc uncertainty quantification using Laplace Approximation (LULA) (Kristiadi, Hein, and Hennig 2021). In order to further validates our strong empirical performance, we also compare with other SOTA intrinsic UQ methods in the Appendix C: (1) The standard Bayesian method Monte-Carlo Dropout (Gal and Ghahramani 2016); (2) The Dirichlet network with variational inference (Be-Bayesian) (Joo, Chung, and Seo 2020); (3) The posterior network with density estimation (Charpentier, Zügner, and Günnemann 2020); (4) The robust OOD detection method ALOE (Chen et al. 2020).

For the trustworthy transfer learning task, since there is no existing work designed for this problem, we compare our method with two simple baselines: (1) Fine-tune the last layer of the source model. (2) Train our proposed meta-model on top of the source model using standard cross-entropy loss.

Performance Metrics. We evaluate the UQ performance by measuring the area under the ROC curve (AUROC) and the area under the Precision-Recall curve (AUPR). The results are averaged over five random trails for each experiment. For the OOD task, we consider the in-distribution test samples as the negative class and the outlier samples as the positive class. For the misclassification task, we consider correctly classified test samples as the negative class and miss-classified test samples as the positive class.

5.3 OOD Detection

The OOD detection results for the three benchmark datasets, MNIST, CIFAR10, and CIFAR100, are shown in Table 1. Additional baseline comparisons are provided in Table 7 in Appendix. Our proposed Dirichlet meta-model method consistently outperforms all the baseline methods in terms of AUROC score (AUPR results are shown in Appendix),

Table 1: **OOD Detection AUROC score.** MI, Dent, and Prec stand for different epistemic uncertainty metrics, i.e., Mutual Information, Differential Entropy, and precision. Settings stand for post-hoc or traditional, i.e., training the entire model from scratch. Additional Data stands for if using additional training data or not.

ID Data & Model	Methods	Settings	Additional Data	Omniglot	FMNIST	KMNIST	CIFAR10	Corrupted
MNIST	Base Model(Ent)	Traditional	No	98.9±0.5	97.8±0.8	95.8±0.8	99.4±0.2	99.5±0.3
LeNet	Base Model(MaxP)	Traditional	No	98.7±0.6	97.6±0.8	95.6±0.8	99.3±0.2	99.4±0.4
	Whitebox	Post-hoc	Yes	98.5±0.3	97.7±0.6	96.0±0.2	99.5±0.1	99.5±0.1
	LULA	Post-hoc	Yes	99.8±0.0	99.4±0.0	**99.3**±0.1	99.9±0.0	99.6±0.1
	Ours(Ent)	Post-hoc	No	99.7±0.1	99.5±0.2	98.2±0.3	**100.0**±0.0	**100.0**±0.0
	Ours(MaxP)	Post-hoc	No	99.3±0.2	99.3±0.2	98.0±0.2	**100.0**±0.0	**100.0**±0.0
	Ours(MI)	Post-hoc	No	**99.9**±0.0	**99.6**±0.2	97.7±0.4	**100.0**±0.0	**100.0**±0.0
	Ours(Dent)	Post-hoc	No	99.8±0.0	99.5±0.2	97.6±0.4	**100.0**±0.0	**100.0**±0.0
	Ours(Prec)	Post-hoc	No	**99.9**±0.0	**99.5**±0.2	97.7±0.5	**100.0**±0.0	**100.0**±0.0
ID Data & Model	**Methods**	**Settings**	**Additional Data**	**SVHN**	**FMNIST**	**LSUN**	**TinyImageNet**	**Corrupted**
CIFAR10	Base Model(Ent)	Traditional	No	86.4±4.6	90.8±1.3	89.0±0.5	87.5±1.1	85.9±8.2
VGG16	Base Model(MaxP)	Traditional	No	86.3±4.4	90.4±1.2	88.7±0.5	87.3±1.1	85.7±8.1
	Whitebox	Post-hoc	Yes	96.9±0.9	95.2±1.2	89.3±2.2	88.9±2.5	96.4±1.0
	LULA	Post-hoc	Yes	97.1±1.7	94.3±0.0	92.8±0.1	90.0±0.0	97.7±2.0
	Ours(Ent)	Post-hoc	No	96.3±3.0	89.0±5.2	89.6±3.4	89.4±3.5	95.9±4.3
	Ours(MaxP)	Post-hoc	No	95.6±3.6	87.8±4.4	89.1±2.4	88.2±2.6	94.0±7.3
	Ours(MI)	Post-hoc	No	**100.0±0.0**	**98.8±0.5**	95.2±0.9	**98.1±0.3**	**100.0±0.0**
	Ours(Dent)	Post-hoc	No	**100.0±0.0**	98.4±0.9	**95.7**±0.8	97.7±0.5	**100.0±0.0**
	Ours(Prec)	Post-hoc	No	**100.0±0.0**	**98.8±0.5**	95.1±0.5	**98.1±0.3**	**100.0±0.0**
CIFAR100	Base Model(Ent)	Traditional	No	76.2±5.2	77.8±2.4	80.1±0.5	79.7±0.3	65.8±11.4
WideResNet	Base Model(MaxP)	Traditional	No	73.9±4.3	76.4±2.3	78.7±0.5	78.0±0.2	63.8±10.4
	Whitebox	Post-hoc	Yes	89.0±0.7	82.4±1.1	80.5±0.7	79.0±1.1	83.1±1.6
	LULA	Post-hoc	Yes	84.2±1.0	83.2±0.1	79.6±0.3	78.5±0.1	80.6±1.0
	Ours(Ent)	Post-hoc	No	92.6±2.0	80.8±3.0	81.1±1.2	84.9±1.2	85.6±3.9
	Ours(MaxP)	Post-hoc	No	88.6±3.2	78.4±3.0	79.7±0.6	82.4±0.6	79.3±4.5
	Ours(MI)	Post-hoc	No	94.3±1.0	**84.4**±1.8	81.9±4.7	85.5±3.6	**90.8**±3.3
	Ours(Dent)	Post-hoc	No	93.3±1.4	84.0±2.3	79.5±3.0	84.6±2.8	89.8±2.6
	Ours(Prec)	Post-hoc	No	**94.4**±1.0	**84.4**±1.7	**82.1**±4.9	**85.6**±3.6	**90.8**±3.4

including the recent proposed SOTA post-hoc uncertainty learning method LULA. We also evaluate the performance of all the uncertainty metrics defined in section 4.3, as it can be observed that compared to total uncertainty (Ent and MaxP), epistemic uncertainties (MI, Dent, Prec) can achieve better UQ performance for the OOD detection task. Moreover, our proposed method does not require additional data to train the meta-model. In contrast, Whitebox requires an additional validation set to train the meta-model, and LULA also needs an additional OOD dataset during training to distinguish the in-distribution samples and outliers, which imposes practical limitations.

5.4 Misclassification Detection

The misclassification detection results for the three benchmark datasets, MNIST, CIFAR10, and CIFAR100, are shown in Table 2. Additional baseline comparisons are provided in Table 9. LULA turns out to be a strong baseline for the misclassification detection task. Although our proposed method performs slightly worse than LULA in terms of the AUROC, it outperforms all the baselines in terms of AUPR.

5.5 Trustworthy Transfer Learning

We use ImageNet pretrained ResNet-50 as our source domain base model and adapt the pretrained model to the target task by training the meta-model using the target domain training data. Unlike traditional transfer learning, which only focuses on testing prediction accuracy on the target task, we also evaluate the UQ ability of the meta-model in terms of OOD detection performance. We use Fashion-MNIST as OOD samples for both target tasks STL10 and CIFAR10 and evaluate the AUROC score. The results are shown in Table 3. Our proposed meta-model method can achieve comparable prediction performance to the baseline methods and significantly improve the OOD detection performance, which is crucial in trustworthy transfer learning.

5.6 Discussion

In this section, we further investigate our proposed method through an ablation study using the CIFAR10 OOD task. Based on our insights and the empirical results, we conclude the following four key factors in the success of our meta-model based method:

Feature Diversity. We replace our proposed meta-model structure with a simple linear classifier attached to only the final layer. The ablation results are shown in Table 4 as "**Linear-Meta**". It can be observed here and in Figure 1 that the performance degrades without using features from all intermediate layers, which further justifies the importance of feature diversity and the effectiveness of our meta-model structure.

Dirichlet Technique. Instead of using a meta-model to parameterize a Dirichlet distribution, we train the meta-model using the standard cross-entropy loss, which simply

Table 2: **Misclassification Results.** Ent and MaxP stand for Entropy and Max Probability, respectively. Settings stand for post-hoc or traditional, i.e., training the entire model from scratch.

Methods	Settings	Metric	MNIST	CIFAR 10	CIFAR 100	Metric	MNIST	CIFAR 10	CIFAR 100
Base Model(Ent)	Traditional	AUROC	96.7±0.9	92.1±0.2	87.2±0.2	AUPR	37.4±4.0	47.5±1.2	67.0±1.2
Base Model(MaxP)	Traditional	AUROC	96.7±0.9	92.1±0.2	86.8±0.2	AUPR	39.4±3.6	46.6±1.1	65.7±1.0
Whitebox	Post-hoc	AUROC	94.9±0.2	90.2±0.1	80.3±0.1	AUPR	30.4±0.3	45.7±0.2	52.5±0.3
LULA	Post-hoc	AUROC	**98.8**±0.1	**94.5**±0.0	**87.5**±0.1	AUPR	40.7±4.2	47.3±0.7	66.0±0.4
Ours(Ent)	Post-hoc	AUROC	96.9±0.6	91.1±0.2	83.4±0.1	AUPR	35.6±4.5	50.0±3.1	66.3±0.4
Ours(MaxP)	Post-hoc	AUROC	97.4±0.4	92.2±0.7	85.8±0.2	AUPR	**44.5**±5.1	**54.2**±3.2	**68.2**±0.5

Table 3: **Trustworthy Transfer Learning Results.** Ent, MaxP, MI, and Dent stand for different uncertainty measurements, i.e., Entropy, Max Probability, Mutual Information, and Differential Entropy, respectively. We use ResNet-50 pretrained with ImageNet as the source model and FashionMNIST as OOD samples.

Methods	Target	Test Acc	AUROC	AUPR	Target	Test Acc	AUROC	AUPR
FineTune(Ent)	**STL10**	48.1±0.5	89.2±0.9	89.3±1.2	**CIFAR10**	65.0±0.4	74.8±1.6	71.6±1.8
FineTune(MaxP)		48.1±0.5	81.8±1.2	83.2±1.7		65.0±0.4	72.7±1.4	69.4±1.5
CrossEnt Loss(Ent)		48.0±0.2	88.9±1.0	87.4±1.5		86.3±0.1	85.0± 1.0	82.2±1.9
CrossEnt Loss(MaxP)		48.0±0.2	84.9±0.7	84.7±0.8		86.3±0.1	83.1±0.9	79.0±1.6
Ours(Ent)		47.2±0.3	**91.8±0.8**	**91.3±0.7**		86.6±0.3	89.9±1.3	88.8±1.5
Ours(MaxP)		47.2±0.3	87.3±1.7	87.9±1.5		86.6±0.3	87.6±1.6	85.5±2.0
Ours(MI)		47.2±0.3	90.2±2.0	88.7±2.8		86.6±0.3	90.8±0.9	89.9±1.2
Ours(Dent)		47.2±0.3	91.7±1.0	90.3±1.7		86.6±0.3	**92.0±0.8**	**91.1±0.8**

outputs a categorical label distribution. The ablation results are shown in Table 4 as "**Cross-Ent**". It can be shown that performance degrades again because it cannot quantify epistemic uncertainty, which justifies the effectiveness of using Bayesian techniques.

Validation for Uncertainty Learning. We retrain the last layer of the base model using the cross-entropy loss with the proposed validation trick. The results are shown in Table 4 as "**LastLayer**". It turns out even such a naive method can achieve improved performance compared to the base model, which further justifies the effectiveness of the post-hoc uncertainty learning setting, i.e., the benefit of solely focusing on UQ performance at the second stage. This interesting observation inspires us to conjecture that efficiently retraining the classifier of the base model at the second stage will lead to better UQ performance. A theoretical investigation of this observation can be interesting for future work.

Data Efficiency. Instead of using all the training samples, we randomly choose only 10% samples to train the meta-model. The results are shown in Table 4 as "10%**data**". It can be observed that our meta-model requires a small amount of data to achieve comparable performance due to the smaller model complexity. Therefore, our proposed method is also more computationally efficient than the approaches that retrain the whole model from scratch.

Table 4: **Ablation Study of Meta-model (CIFAR10 AUROC score).** The results are reported as mean over five experiment trails. Error bars are provided in Table 10 in the Appendix.

Methods	SVHN	FMNIST	LSUN	TIM	Corrupted
Base Model(Ent)	86.4	90.8	89.0	87.5	85.9
Base Model(MaxP)	86.3	90.4	88.7	87.3	85.7
Linear-Meta(Ent)	90.4	91.3	91.5	89.5	90.4
Linear-Meta(MaxP)	90.1	91.5	91.4	89.6	90.1
Linear-Meta(MI)	90.6	90.1	91.2	88.8	90.5
Linear-Meta(Dent)	90.4	90.7	91.4	89.2	90.3
Linear-Meta(Prec)	90.6	90.0	91.2	88.8	90.6
Cross-Ent(Ent)	94.2	91.2	91.2	90.3	94.7
Cross-Ent(MaxP)	93.3	91.1	90.9	90.0	94.0
LastLayer(Ent)	93.0	90.2	91.9	89.9	93.1
LastLayer(MaxP)	92.9	90.5	91.9	90.1	93.1
10%**data(Ent)**	90.0	89.1	88.7	88.2	90.2
10%**data(MaxP)**	90.9	88.1	86.9	86.5	91.7
10%**data(MI)**	99.9	98.1	95.4	97.2	99.9
10%**data(Dent)**	96.7	97.4	94.3	95.7	96.7
10%**data(Prec)**	99.9	98.0	95.4	97.3	99.9
Ours(Ent)	96.3	89.0	89.6	89.4	95.9
Ours(MaxP)	95.6	87.8	89.1	88.2	94.0
Ours(MI)	**100.0**	**98.8**	95.2	**98.1**	**100.0**
Ours(Dent)	**100.0**	98.4	**95.7**	97.7	**100.0**
Ours(Prec)	**100.0**	**98.8**	95.1	**98.1**	**100.0**

6 Concluding Remarks

We provide a novel solution for the uncertainty quantification problem via our proposed post-hoc uncertainty learning framework and the Dirichlet meta-model approach. Our method turns out to be both effective and computationally efficient for various UQ applications. We believe our meta-model approach not only has the flexibility to tackle other applications relevant to uncertainty quantification, such as quantifying transfer-ability in transfer learning and domain adaptation, but also can adapt to other model architecture such as transformer and language model. Exploring these potential applications and offering a theoretical interpretation of the meta-model can be interesting future work.

Acknowledgement

This work was supported, in part, by the MIT-IBM Watson AI Lab under Agreement No. W1771646, and NSF under Grant No. CCF-1816209.

References

Alaa, A.; and Van Der Schaar, M. 2020. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. In *International Conference on Machine Learning*, 165–174. PMLR.

Alipanahi, B.; Delong, A.; Weirauch, M. T.; and Frey, B. J. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8): 831–838.

Begoli, E.; Bhattacharya, T.; and Kusnezov, D. 2019. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1): 20–23.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight Uncertainty in Neural Network. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1613–1622. Lille, France: PMLR.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Charpentier, B.; Zügner, D.; and Günnemann, S. 2020. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems*, 33: 1356–1367.

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2020. Robust out-of-distribution detection for neural networks. *arXiv preprint arXiv:2003.09711*.

Chen, T.; Navrátil, J.; Iyengar, V.; and Shanmugam, K. 2019. Confidence scoring using whitebox meta-models with linear classifier probes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1467–1475. PMLR.

Elder, B.; Arnold, M.; Murthi, A.; and Navratil, J. 2020. Learning prediction intervals for model performance. *arXiv preprint*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Ghosh, S.; Yao, J.; and Doshi-Velez, F. 2019. Model Selection in Bayesian Neural Networks via Horseshoe Priors. *Journal of Machine Learning Research*, 20(182): 1–46.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6): 82–97.

Hüllermeier, E.; and Waegeman, W. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3): 457–506.

Jain, M.; Lahlou, S.; Nekoei, H.; Butoi, V.; Bertin, P.; Rector-Brooks, J.; Korablyov, M.; and Bengio, Y. 2021. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*.

Joo, T.; Chung, U.; and Seo, M.-G. 2020. Being bayesian about categorical probability. In *International Conference on Machine Learning*, 4950–4961. PMLR.

Koenker, R.; and Bassett, G., Jr. 1978. Regression Quantiles. *Econometrica*, 46(1): 33–50.

Kristiadi, A.; Hein, M.; and Hennig, P. 2021. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence*, 344–353. PMLR.

Kull, M.; Perello Nieto, M.; Kängsepp, M.; Silva Filho, T.; Song, H.; and Flach, P. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Litjens, G.; Kooi, T.; Bejnordi, B. E.; Setio, A. A. A.; Ciompi, F.; Ghafoorian, M.; Van Der Laak, J. A.; Van Ginneken, B.; and Sánchez, C. I. 2017. A survey on deep learning in medical image analysis. *Medical image analysis*, 42: 60–88.

MacKay, D. J. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472.

Malinin, A.; and Gales, M. 2018. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31.

Malinin, A.; and Gales, M. 2019. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in Neural Information Processing Systems*, 32.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 1045–1048. Makuhari.

Nandy, J.; Hsu, W.; and Lee, M. L. 2020. Towards maximizing the representation gap between in-domain & out-of-distribution examples. *Advances in Neural Information Processing Systems*, 33: 9239–9250.

Neal, R. M. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Rasmussen, C. E.; and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems*, 31.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

van Amersfoort, J.; Smith, L.; Teh, Y. W.; and Gal, Y. 2020. Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network. *arXiv preprint arXiv:2003.02037*.

Wakefield, J. 2013. *Bayesian and frequentist regression methods*. Springer Science & Business Media.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.