Error-Driven Uncertainty Aware Training

Pedro Mendes[a,b,*], **Paolo Romano**[b] **and David Garlan**[a]

[a]Software and Societal Systems Department, Carnegie Mellon University
[b]INESC-ID and Instituto Superior Tecnico, Universidade de Lisboa

Abstract. Neural networks are often overconfident about their predictions, which undermines their reliability and trustworthiness. In this work, we present a novel technique, named Error-Driven Uncertainty Aware Training (EUAT), which aims to enhance the ability of neural models to estimate their uncertainty correctly, namely to be highly uncertain when they output inaccurate predictions and low uncertain when their output is accurate. The EUAT approach operates during the model's training phase by selectively employing two loss functions depending on whether the training examples are correctly or incorrectly predicted by the model. This allows for pursuing the twofold goal of i) minimizing model uncertainty for correctly predicted inputs and ii) maximizing uncertainty for mispredicted inputs, while preserving the model's misprediction rate. We evaluate EUAT using diverse neural models and datasets in the image recognition domains considering both non-adversarial and adversarial settings. The results show that EUAT outperforms existing approaches for uncertainty estimation (including other uncertainty-aware training techniques, calibration, ensembles, and DEUP) by providing uncertainty estimates that not only have higher quality when evaluated via statistical metrics (e.g., correlation with residuals) but also when employed to build binary classifiers that decide whether the model's output can be trusted or not and under distributional data shifts.

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable performance across various domains and are increasingly utilized to automate intricate decision-making processes. However, a critical limitation of current neural models is their tendency to display overconfidence in their predictions [11, 2]. This overconfidence persists even when erroneous predictions are made, ultimately compromising the reliability and trustworthiness of the models.

Recent research efforts [10, 5, 9, 12, 34] have been dedicated to enhancing the trustworthiness of DNNs by estimating the model's predictive uncertainty through various approaches. Bayesian neural networks (BNNs) [34, 43, 44] offer an elegant framework for modeling uncertainty [38]. However, while BNNs provide theoretically sound uncertainty estimates, they incur prohibitive costs, being impractical for large datasets and complex models. To mitigate these challenges, various approximations have been introduced. For instance, Monte Carlo (MC) dropout [10], which leverages dropout regularization during both training and inference stages to approximate the behavior of BNNs.

Furthermore, numerous studies [27, 23, 45, 41, 33] have focused on calibrating the models' predicted uncertainty in different ways. These works can be categorized into two primary groups based on whether they: 1) account for the uncertainty during training by introducing an additional term in the loss function to quantify the model's predictive uncertainty, or ii) implement a post-processing stage to calibrate the model's predicted probabilities using a validation set. Although post-processing methods have empirically proven to be effective and cost-efficient [27], they present additional calibration parameters that are sensitive to the method and data used. On the other hand, despite being more expensive, learning-based methods have achieved better performance for uncertainty estimation [26, 32].

In this work, we mix both approaches by introducing Error-Driven Uncertainty Aware Training (EUAT), a specialized training procedure for classification tasks that aims at improving the model's uncertainty estimation by imposing high uncertainty for erroneous outputs and low uncertainty for accurate predictions. To achieve this twofold goal, during training, EUAT iterates between two loss functions depending on whether the training examples are correctly or incorrectly predicted by the model. More in detail, our approach extends a base loss function, which aims to minimize the classification error rate (e.g., cross-entropy (CE)), with an additional term whose objective is to maximize the model's uncertainty for misclassifications and minimize uncertainty for correct classified inputs. However, to separate the correctly and incorrectly classified inputs and speed up the training procedure, we first consider pre-train the model and then we apply EUAT to conduct a post-learning-based phase to improve its uncertainty.

We conducted an extensive evaluation of EUAT on classification tasks using popular image recognition models and benchmarks, where we compared our approach against several state-of-the-art methods for uncertainty estimation using six different evaluation metrics. Further, we extended our assessment to a binary classification problem, which presents a particularly interesting case involving the class inversion of the high uncertainty outputs that are likely to be wrong classified. We also evaluate our technique in an out-of-distribution detection task, where corrupted inputs are used to evaluate the model, and at last, we integrate our function into adversarial training settings in order to identify possible misclassifications based on uncertainty. We detail the challenges encountered in each domain/task. In general, EUAT presents the best performance in more than 60% of the metrics considered, and in the majority of the cases where the baselines are more competitive, EUAT is still able to achieve similar performance metrics. Further, in all the scenarios considered, we show that our strategy can better separate wrong and accurate predictions based on uncertainty, increasing the reliability and trustworthiness of the models.

* Corresponding Author. Email: pgmendes@andrew.cmu.edu

Table 1: Uncertainty Confusion Matrix

Correctness	Uncertainty: Certain	Uncertainty: Uncertain
Correct	True Certainty TC	False Uncertainty FU
Wrong	False Certainty FC	True Uncertainty TU

2 Related Work

In this section, we first review different formulations of the problem of estimating models' uncertainty, and the corresponding metrics, proposed in the literature. Subsequently, we analyze existing methods to estimate the uncertainty of DNNs. Finally, we discuss methods aimed at improving uncertainty estimation by adjusting the model's outputs via post-processing or uncertainty-aware training techniques.

Problem definition and metrics. The problem of accurately estimating model uncertainty has been formalized using two main theoretical frameworks. One such formalization is based on the notion of calibration, which aims at aligning the probabilities output by the model with the true likelihood of the predicted outcomes [14]. An alternative formalization is based on the Uncertainty Confusion Matrix (UCM) [3, 20], as defined in Table 1. The UCM specializes the concept of confusion matrix to evaluate the ability to leverage the model's uncertainty to discern correct predictions. For example, different metrics such as the expected calibration error (ECE) [33], adaptive calibration error [36], or test-based calibration error [31], have been proposed to measure the calibration error of a model. On the other hand, leveraging the UCM, several works [20, 3] have exploited additional metrics such as uncertainty accuracy (uA) and the uncertainty area under the curve (uAUC) to enhance the reliability of uncertainty estimates.

Uncertainty Estimation. One important foundation of these works lies in the computation of uncertainty. Uncertainty in DNNs plays a key role in quantifying the reliability and robustness of their predictions. There are two main types of uncertainty: epistemic uncertainty, associated with the model's lack of knowledge or data, and aleatoric uncertainty, linked to the inherent randomness and unpredictability within the training data. Various metrics are employed to measure these types of uncertainty such as predictive entropy (PE) and mutual information (MI) [1]. However, quantifying uncertainty with DNNs is a challenging task. Bayesian methods [34, 28] can directly provide an estimate of the uncertainty by parameterizing the parameters of the network with distributions. However, training BNNs usually comes with a prohibitive cost. Thus, several approximations have been developed. Monte Carlo (MC) dropout [10], which is one of the most popular techniques for uncertainty quantification on DNNs [1, 4], proposed as a Bayesian approximation to estimate the uncertainty by sampling multiple dropout masks during the training and inference phases and aggregating the predictions, providing a probabilistic estimate that is used to quantify uncertainty. One can also approximate and estimate model uncertainty by computing the variance of the output predictions [5, 9, 25]. Additionally, Markov Chain Monte Carlo (MCMC) methods [12] offer another avenue for drawing the posterior distribution (albeit, those require a long time to converge to the final distribution [1, 34]). Further, Variational Inference (VI) is a technique used to estimate the uncertainty of BNNs by approximating the posterior distribution over the model's weights, which are treated as random variables with associated probability distributions. Training the network to approximate these distributions enables the capture of inherent uncertainty in the model's predictions. Moreover, one can also resort to models that can directly output uncertainty estimations, such as Deep Gaussian Processes [6], or use Laplace approximations for uncertainty [29], or deep ensemble methods that offer yet another approach to estimate model uncertainty [24, 1], by aggregating the outputs of each learner in the ensemble and computing relevant metrics such as the entropy or MI. At last, DEUP [23] trains a new simple estimator to predict the uncertainty of the base model exploiting its error/loss, and DUN [2] leverages the outputs of different layers of a DNN to measure the uncertainty.

In this work, we resort to MC dropout to estimate the uncertainty of our models and compute the PE using the outputted distributions. Distinctly, we explicitly aim to increase the uncertainty of misclassifications by maximizing the PE of the wrong prediction, while minimizing the global error rate.

Post-processing Calibration Methods. Model calibration becomes especially critical in tasks where estimated uncertainties play a fundamental role in risk assessment. Thus, a first set of works aims at calibrating a fully trained model by applying a post-processing phase to align the output probabilities with the true likelihood of the predicted outcomes of events occurring [14]. DNNs are not inherently calibrated, and several techniques, such as Platt Scaling [39], Isotonic Regression [45], Temperature Scaling [14], or Beta Calibration [22], can be employed to fine-tune the probabilities outputted by the model ensuring a more accurate alignment with true outcome frequencies. Moreover, Krishnan et al. [20] introduced an accuracy versus uncertainty calibration (AvUC) loss function devised to obtain well-calibrated uncertainties while simultaneously preserving or enhancing model accuracy, and they extended their approach by proposing a post-hoc calibration phase that extends Temperature Scaling using AvUC. In addition, Karandikar et al. [18] proposed an extension of ECE and AvUC by developing a soft version of the binning operation underlying these calibration-error estimators, and also extended their approach for post-processing calibration by optimizing the temperature parameter in the temperature scaling method based on the soft calibration error. Complementary, Gupta et al. [15] presented a binning-free calibration approach. However, these calibration methods can be very sensitive to both the model and the validation set [27] and perform suboptimally when faced with shifts in data distribution [37].

Uncertainty Aware Training Methods. One fundamental aspect of training DNNs is the choice of a loss function. Although CE stands out as a common choice for addressing classification problems, it tends to increase the over-confidence of the resulting model [14, 32]. Thus, since accounting for predictive uncertainty during training improves model calibration [20], several loss functions have been developed to extend existing ones, aiming to incorporate the model's uncertainty. These learning-based methods such as focal loss [26] or label smoothing [26] introduce an additional term addressing model uncertainty. These terms are typically balanced using corresponding weights introducing as a consequence additional hyper-parameters. Similarly, Shamsi et al. [41] proposed two loss functions that extend the CE by adding a new term to address the model's uncertainty that can be determined through the PE or the ECE. Additionally, Einbinder et al [8] introduced an uncertainty-aware confor-

mal loss function by adding a new term that quantifies uncertainty via conformal prediction. CALS [27] exploits the Augmented Lagrangian Multiplier method to adaptively learn the weights of the penalties to balance each term in the new loss function. Separately, class uncertainty-aware (CUA) loss [19] tailored object detection introduces the uncertainty of each class to augment the loss value when prediction results are uncertain. Further, Ding et al. [7] developed an uncertainty-aware loss for selective medical image segmentation that considers uncertainty in the training process to directly maximize the accuracy on the confident segmentation subset, rather than the accuracy on the whole dataset. Differently from the aforementioned works, our approach takes a distinct path by focusing on leveraging a pre-trained model to deliberately increase the uncertainty associated with incorrectly classified inputs without degrading the overall error rate.

Further, through our novel method, the objective is to empower Machine Learning (ML) systems to recognize possible model misclassification in production and take customized actions accordingly. This idea can be further extended to adversarial training scenarios, where the deliberate increase in uncertainty for misclassified adversarial inputs enhances the system's ability to detect and respond to potential attacks in production.

3 Error-Driven Uncertainty Aware Training

DNNs often exhibit overconfidence in their predictions [11, 2]. Thus, in this work, we address this problem by proposing a loss function called Error-Driven Uncertainty Aware Training that shifts the focus toward refining the uncertainty guarantees of a pre-trained model by increasing the uncertainty associated with misclassifications while reducing the error rate and uncertainty of correct predictions. We assert that if a model reaches a stagnation phase in terms of quality and the pursuit of further training offers negligible benefit, it should, at the very least, produce high uncertainty values for incorrect predictions.

In order to achieve our design goals, we start the process by querying a pre-trained model to determine which inputs of the training dataset are wrong and correctly classified. Subsequently, we create two sets, one with incorrect classified inputs $\mathcal{W}$ and the other with the correct ones $\mathcal{C}$. We resize the correct classified inputs' set by randomly selecting samples until the two sets have the same size. Then, in order to reduce overfitting, we mix wrong and correct classified inputs. Our approach employs distinct loss functions for each set. Since our objective is to deliberately increase the uncertainty of misclassifications, we minimize the CE and maximize the uncertainty (determined via MC dropout and PE) for the wrong-classified inputs while, for the correct-classified inputs, we minimize the CE and the uncertainty, i.e.,

$$L_{\text{EUAT}}(f_\theta(\mathbf{x}),\mathbf{y})=\begin{cases} L_{\text{CE}}(f_\theta(\mathbf{x}),\mathbf{y})\text{-}L_U(f_\theta(\mathbf{x}),\mathbf{y}) & \forall(x,y)\in\mathcal{W} \\ L_{\text{CE}}(f_\theta(\mathbf{x}),\mathbf{y})\text{+}L_U(f_\theta(\mathbf{x}),\mathbf{y}) & \forall(x,y)\in\mathcal{C} \end{cases} \tag{1}$$

where the cross-entropy loss is given by

$$L_{\text{CE}}(f_\theta(x),y) = -\frac{1}{K}\sum_{i=1}^{K} t(x_i)\log(f_\theta(x_i)), \tag{2}$$

($t(x)$ denotes the true label given the input x), and the uncertainty loss is measured by resorting to predictive entropy H

$$L_U(f_\theta(x),y) = H[P(y|x)] = -\sum_{y\in\mathcal{Y}} P(y|x)\log P(y|x), \tag{3}$$

Algorithm 1 Pseudo-code to train a model with EUAT loss function.

Input: model f, training set $\mathcal{S}$, optimizer opt
Output: model f
no_inputs = 0
while True **do** ▷ *stop condition*
  $\mathcal{W},\mathcal{C} \leftarrow f(\mathcal{S})$ ▷ *Wrong $\mathcal{W}$ & correct $\mathcal{C}$ classified sets*
  $\mathcal{C} \leftarrow$ Random($\mathcal{C}$, $|\mathcal{W}|$)
  $L_{\mathcal{W}}(f(\mathbf{x}),\mathbf{y})$=$L_{CE}(f(\mathbf{x}),\mathbf{y})$-$L_U(f(\mathbf{x}),\mathbf{y})$ $\forall(\mathbf{x},\mathbf{y})\in\mathcal{W}$
  ▷ *Minimize CE-PE for wrong outputs*
  $L_{\mathcal{C}}(f(\mathbf{x}),\mathbf{y})$=$L_{CE}(f(\mathbf{x}),\mathbf{y})$+$L_U(f(\mathbf{x}),\mathbf{y})$ $\forall(\mathbf{x},\mathbf{y})\in\mathcal{C}$
  ▷ *Minimize CE+PE for correct outputs*
  $L(f(\mathbf{x}),\mathbf{y})$=$L_{\mathcal{W}}(f(\mathbf{x}),\mathbf{y})$+$L_{\mathcal{C}}(f(\mathbf{x}),\mathbf{y})$ ▷ *Add losses*
  L.backward() ▷ *Gradient estimation*
  opt.step() ▷ *Back-propagation computation*
  no_inputs += ($|\mathcal{W}|$ + $|\mathcal{C}|$)
  if no_inputs > epoch_size **then**
    f.test() ▷ *Test f if trained with the same amount of data as the dataset*
  end if
end while
return f

where $P(y|x)$ is the model's output distribution over the set of possible outcomes $\mathcal{Y}$ obtained via MC dropout by approximating the model's output predictions using the average across parameters θ_i sampled from a dropout distribution

$$p(y|\mathcal{D},x) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|\theta_i,x). \tag{4}$$

We decided to also minimize uncertainty for correct-classified inputs in order to avoid a peak in the global model's uncertainty. By using this approach, the model is able to better separate the erroneous and correct predictions using the uncertainty of each individual forecast. We also implement an early stopping policy based on the different evaluation metrics considered. It should also be noted that this proposed loss function is differentiable and continuous. The pseudo-code of the method proposed is described in Algorithm 1.

By resorting to EUAT, which effectively distinguishes between the uncertainty of accurate and erroneous predictions, we enhance the quality of uncertainty estimates and ultimately improve the reliability and trustworthiness of the model. Operationalizing the model in production involves assessing the uncertainty associated with each prediction. When the prediction falls below an uncertainty threshold, the model outcome can be trusted. Otherwise, when the uncertainty is above the threshold, the prediction is untrustworthy, warranting further scrutiny or review by human evaluators. Consequently, tuning the uncertainty threshold becomes fundamental. Akin to the implementation of the early stopping policy, we also explore various evaluation metrics to select the threshold.

In this work, we have considered classification tasks using different image recognition models. Thus, given the model's prediction, we can promptly categorize it as either accurate or erroneous, and place it in the respective set. The early stopping policy and threshold selection are individually optimized for each metric.

We also consider a binary classification problem where the high-uncertainty output predictions that are likely to be wrongly classified can be inverted to the opposite class. Given the particularities of this problem, we tune the uncertainty threshold for EUAT differently by maximizing the number of correct predictions (i.e., true positives and true negatives) that are below the threshold and the number of incor-

rect predictions that are above the threshold (false positives and false negatives). Then, in production, when the outputted uncertainty is larger than the threshold, we can directly flip the prediction for the opposite class, and this way improve the model's quality.

At last, we can also use EUAT to perform adversarial training. It should be noted that our approach does not aim to identify adversarial attacks using uncertainty; rather, when applied to AT focuses on identifying misclassifications based on the output uncertainty, regardless of whether they result from an attack or not. Equivalently, we can separate the wrong and correct predictions (clean and adversarial) in the two sets and train directly with our loss function. Since the function is differentiable, we can solve the optimization problem to find the perturbation using adversarial training methods like Fast Gradient Descent Method (FGSM) [13] or Projected Gradient Descent (PGD) [30]. Similar to Smith and Gal [42], we acknowledge that neither dropout nor our method alone can be considered a reliable adversarial defense. Rather, we advocate for the joint use of all these methods when training a model adversarially in order to achieve better guarantees and trade-offs of accuracy, robustness, and uncertainty, yielding resilient models harder to attack.

4 Evaluation

In this section, we report the evaluation of the EUAT on a variety of domains and tasks.

4.1 Experimental Setup, Benchmarks, and Baselines

To evaluate EUAT, we employed four models and datasets widely used in the image recognition domain namely, ResNet50 [16] with ImageNet [40], Wide-ResNet-28x10 [46] with Cifar100 [21], ResNet18 with Cifar10, and ResNet18 with SVHN [35]. We also considered a binary classification model (using ResNet18 with Cifar10 to verify if there is a cat in an image), and an out-of-distribution (OOD) detection task, where corrupted inputs using distributional data shifts are used to evaluate the model. Finally, we evaluate our approach in adversarial training settings using three models/datasets mentioned above in the image recognition domain.

All the models and training procedures were implemented in Python3 via the Pytorch framework and trained using a single Nvidia RTX A4000.

To train the models, we used a dropout rate of 0.3 and resorted to stochastic gradient descent to minimize the loss function using a momentum of 0.9 and a batch size of 64 for all the models, a learning rate of 0.01 and weight decay of 10^{-5} for ResNet50/ImageNet and 0.1 and 0 for the remaining ones, respectively. Before training the model using EUAT, we pre-trained the models using CE loss and then decreased the learning rate by $10^3\times$ when applying EUAT. Additionally, we exploited automatic mixed precision to train the ResNet50/ImageNet and Wide-ResNet/Cifar100. The models were trained during 60 epochs (except in the binary classification problem where it was trained for 200 epochs). More in detail, we pre-trained the models for 30 epochs (100 epochs in the binary classification problem) before starting the second phase of training where we applied the EUAT. The implementation of the training pipeline and additional information to ensure the reproducibility of results are provided in the supplemental material.

We compared EUAT against the CE loss, model calibration, DEUP [23], an ensemble of five learners [24], CALS [27], and a loss function incorporating both CE and PE (CE+PE) [41]. To calibrate the model and train DEUP's additional error predictor, we created a validation set comprising 10% randomly selected samples from the test set. Further, we resort to Isotonic regression [45] to calibrate the model, achieving superior results compared to other methods like Platt scaling [39], temperature scaling [14], and beta calibration [22]. Although we experimented DEUP with different validation set sizes, to ensure fairness, we maintained consistency by employing the same validation set size in both cases.

We evaluate the different baselines using six different metrics. First, we report the uncertainty accuracy (uA) (Eq. 5) and the uncertainty area under the curve (uAUC), which are computed based on the Uncertainty Confusion Matrix [3] defined in Table 1.

$$uA = \frac{TC + TU}{TC + TU + FC + FU} \tag{5}$$

We also evaluate the models using the correlation between the residuals of the model and predicted uncertainties (Corr. w/ res.) [23], and the Wasserstein distance of the uncertainty between the sets of correct and wrong predictions (Wasser. dist.). At last, we report the ECE and the model's error. The uncertainty/confidence of the models is always computed via MC dropout using the normalized PE, except for DEUP, which resorts to the loss values of the base model to estimate the quality of its predictions (for a fair comparison, after testing the model trained with DEUP, we had to normalize the loss values). Moreover, we optimize each metric considered independently by employing an early stopping policy and tuning the uncertainty threshold, and we report the best values obtained for each metric.

4.2 Experimental results

Next, we report the results obtained using EUAT in the different domains evaluated.

4.2.1 Image Recognition Models

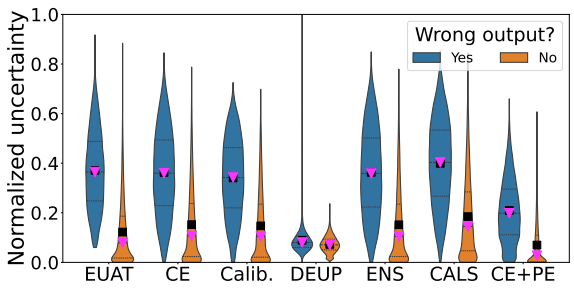
We start by reporting in Table 2 the results obtained using four models/datasets for image recognition: ResNet50/ImageNet, Wide-ResNet/Cifar100, ResNet18/Cifar10, and ResNet18/SVHN. Across all baselines and metrics, it is evident that EUAT consistently demonstrates superior performance, outperforming others in 16 out of 24 cases. Notably, in the cases where the baselines are more competitive, the differences are marginal (e.g., the error using EUAT increases by 3.4%, 13.8%, and 14.6% compared to the best baseline training a Wide-ResNet/Cifar100, ResNet18/Cifar10, and ResNet18/SVHN, respectively).

EUAT presents the best uA when training a ResNet50 with ImageNet and a Wide-ResNet with Cifar100, and yields minimal reduction of 0.5% and 0.3% in the uA when training a ResNet18 with Cifar10 and SVHN compared with the best baselines (namely, ensemble and CALS, respectively). Additionally, EUAT consistently outperforms existing approaches in terms of uAUC and the correlation between the residuals of the model and predicted uncertainties. EUAT achieves an improvement on the uAUC by up to 15.1%, 8.9%, 64.3%, 28.1%, 8.5%, and 20.4% compared to CE, calibration, DEUP, deep ensemble, CALS, and CE+PE, respectively. Moreover, the correlation between the model's residuals and predicted uncertainties improves by 20%, 20.7%, 8.6%, and 11.5% when using EUAT for training ResNet50 with ImageNet, Wide-ResNet with Cifar100, and ResNet18 with Cifar10 and SVHN compared to the best-performing baseline in each scenario.

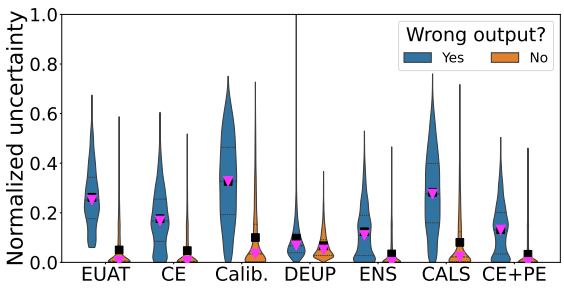
Next, the effectiveness of EUAT in distinguishing correct predictions from misclassifications based on the predicted uncertainty is

Table 2: Comparison of EUAT against the baselines using different evaluation metrics and considering four benchmarks.

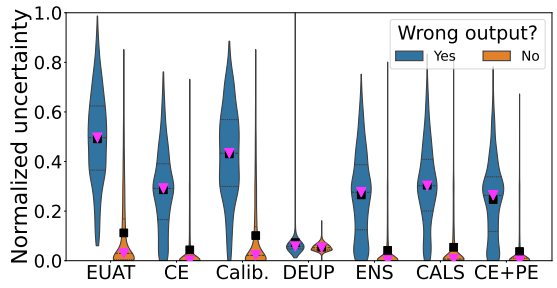
Benchmark	Baseline	uA	uAUC	Corr. w/ res.	Wasser. dist.	ECE	Error
ResNet50/ImageNet	**EUAT**	**0.804**	**0.878**	**0.655**	**0.301**	**0.223**	**0.439**
	CE	0.749	0.812	0.546	0.220	0.224	0.513
	Calibration	0.743	0.807	0.538	0.193	0.274	0.535
	DEUP	0.580	0.591	0.291	0.030	0.429	0.522
	Ensemble	0.745	0.809	0.536	0.213	0.245	0.506
	CALS	0.746	0.809	0.542	0.214	0.235	0.525
	CE+PE	0.755	0.786	0.530	0.141	0.382	0.524
Wide-ResNet/Cifar100	**EUAT**	**0.858**	**0.891**	**0.711**	0.216	0.162	0.273
	CE	0.794	0.774	0.546	0.128	0.235	0.296
	Calibration	0.787	0.836	0.580	**0.226**	0.146	0.312
	DEUP	0.699	0.601	0.300	0.052	0.252	0.328
	Ensemble	0.742	0.696	0.466	0.089	0.294	0.332
	CALS	0.813	0.831	0.589	0.233	**0.114**	**0.264**
	CE+PE	0.779	0.740	0.518	0.099	0.252	0.300
ResNet18/Cifar10	**EUAT**	0.914	**0.921**	**0.626**	**0.410**	0.018	0.099
	CE	0.905	0.866	0.576	0.273	0.025	0.103
	Calibration	0.898	0.893	0.533	0.329	0.030	0.108
	DEUP	0.917	0.563	0.297	0.032	0.039	0.094
	Ensemble	**0.919**	0.840	0.545	0.224	0.039	**0.087**
	CALS	0.907	0.875	0.563	0.283	**0.012**	0.101
	CE+PE	0.907	0.837	0.566	0.213	0.052	0.102
ResNet18/SVHN	**EUAT**	0.960	**0.927**	**0.638**	**0.479**	**0.011**	0.047
	CE	0.956	0.841	0.572	0.232	0.021	0.047
	Calibration	0.953	0.902	0.537	0.346	0.026	0.051
	DEUP	0.960	0.564	0.312	0.040	0.024	0.044
	Ensemble	0.960	0.756	0.515	0.162	0.029	0.043
	CALS	**0.963**	0.867	0.569	0.264	**0.011**	**0.041**
	CE+PE	0.959	0.799	0.547	0.184	0.029	0.045



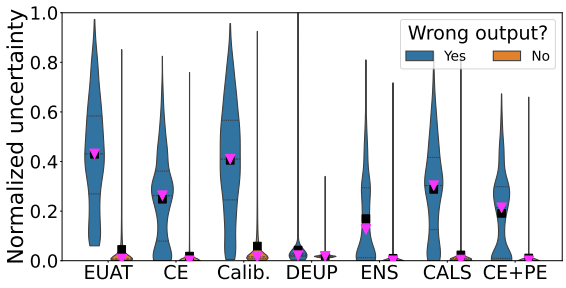
(a) ResNet50/ImageNet



(b) Wide-ResNet/Cifar100



(c) ResNet18/Cifar10



(d) ResNet18/SVHN

Figure 1: Normalized uncertainty distribution of correct and incorrect predictions for the different baselines (the average value of each distribution is marked with a black square, and the median with a pink triangle).

assessed using the Wasserstein distance of the uncertainty between the sets of correct and wrong predictions. More in detail, on average across all models/datasets, the Wasserstein distance increases by up to 1.7×, 1.3×, 9.7×, 2.3×, 1.4×, and 2.2× using EUAT compared to CE, calibration, DEUP, ensemble, CALS, and CE+PE, respectively. At last, we evaluate the impact of using EUAT on the ECE and models' misclassification rate. EUAT results in lower ECE in two cases (ResNet50/ImageNet and ResNet18/SVHN), while a slight increase is observed in the other two (Wide-ResNet/Cifar100 and ResNet18/Cifar10). Importantly, the error rate remains consis-

Table 3: Comparison of EUAT against the baselines using different evaluation metrics and considering a binary classification problem.

Baseline	uA	uAUC	Corr. w/ res.	Wasser. dist.	ECE	Error w/o flip	Error w/ flip	F1	Precision	TPR	TNR
EUAT	**0.861**	**0.816**	**0.446**	**0.405**	**0.112**	0.152	**0.139**	**0.860**	**0.863**	**0.858**	0.864
CE	0.845	0.759	0.435	0.350	0.113	0.152	0.221	0.763	0.823	0.712	0.847
Calib.	0.783	0.779	0.366	0.356	0.313	0.159	0.290	0.733	0.670	0.809	0.603
DEUP	0.844	0.591	0.323	0.049	0.124	0.172	0.178	0.812	0.859	0.771	**0.874**
Ensemble	**0.861**	0.683	0.358	0.216	**0.112**	**0.147**	0.163	0.834	0.847	0.822	0.852
CALS	0.831	0.748	0.407	0.326	0.119	0.164	0.163	0.834	0.847	0.822	0.852
CE+PE	0.811	0.690	0.360	0.274	0.153	0.184	0.225	0.759	0.817	0.709	0.842

Table 4: Comparison of EUAT against the baselines using different evaluation metrics and tested with out-of-distribution samples with Cifar10.

Baseline	uA	uAUC	Corr. w/ res.	Wasser. dist.	ECE	Error
EUAT	**0.754**	**0.796**	**0.529**	**0.255**	**0.143**	0.489
CE	0.691	0.676	0.311	0.126	0.292	0.539
Calib.	0.619	0.553	0.110	0.036	0.497	0.619
DEUP	0.617	0.509	0.072	0.004	0.551	0.617
Ensemble	0.753	0.734	0.426	0.177	0.237	0.555
CALS	0.686	0.716	0.383	0.147	0.216	**0.464**
CE+PE	0.681	0.663	0.299	0.103	0.418	0.568

tent across all benchmarks for all baselines, with a notable improvement observed in the ResNet50/ImageNet benchmark, showcasing a reduction of 13.2% (compared to the best baseline, namely, a deep ensemble). Furthermore, the large gains of EUAT were obtained using larger models and datasets, where the model is less accurate, and thus, by prioritizing the training of incorrect predictions, the EUAT is able to yield significant improvements.

Further, in Figure 1, we plot the distribution of the normalized uncertainty of the correctly and incorrectly predicted sets using the different baselines. By visualizing these distributions, we verify an improvement in the separation of the uncertainty of these two sets using EUAT (which is confirmed by the computation of the Wasserstein distance in Table 2). These results confirm the trustworthiness of the EUAT to improve the model's uncertainty to separate accurate and incorrect predictions.

4.2.2 Binary Classification Problem

Next, we proceed by assessing the effectiveness of the EUAT through a binary classification scenario, wherein high-uncertainty predictions can be inverted to the opposite class. We compared the models obtained using the different baselines and tuned the uncertainty threshold as described in Section 3. Additionally, we conducted an extensive evaluation utilizing supplementary metrics such as F1-score, precision, True Positive Rate (TPR), True Negative Rate (TNR), and error rates concerning the inversion or retention of high uncertainty predictions.

Table 3 presents a comprehensive overview of these performance metrics. Remarkably, the EUAT outperforms other baselines across 9 out of 11 evaluated metrics. More in detail, employing EUAT leads to significant enhancements in the F1-score, precision, and TPR, on average, by up to 9.2%, 11.3%, and 11.3%, respectively, with a corresponding 29.9% reduction in error rates through high uncertainty prediction inversion. Furthermore, our method improves the uA, uAUC, and the correlation with model residuals by up to 9.9%, 52.5%, and 113.3%, while the Wasserstein distance of the uncertainty between the correct and wrong predicted sets is enhanced by up to 17.5× (averaging at 4.1×). Lastly, it is noteworthy that the ECE achieved using the EUAT aligns closely with other baselines, namely CE, DEUP, Ensemble, and CALS, and reduces on average across baselines by 18.1%, while the misclassification rate lowers on average by 6%. These findings underscore the importance of using EUAT to improve the model quality in a binary classification task.

4.2.3 Out-Of-Distribution Detection Task

Predicted uncertainty can be leveraged to reject difficult examples with high uncertainty, even when these samples present huge distribution shifts from the dataset used to train. Further, the OOD detection task varies significantly when using different methods and datasets [18]. Thus, in this section, we evaluate the effectiveness of EUAT to detect and reject OOD examples based on the predicted uncertainty. For each baseline, we trained a ResNet18 using the Cifar10 dataset and then tested it using a corrupted version with Gaussian noise of this dataset (called Cifar10-C [17]). For a fair comparison, when calibrating the model and training using DEUP, we use an additional validation set containing 10% of the clean inputs of the original test set.

In Table 4, we compared EUAT against the other baselines considering the previous metrics. Notably, in 83% of the metrics considered, EUAT performs better than the baselines. More in detail, EUAT improves the uA by 9.2%, 21.9%, 22.2%, 0.25%, 9.9%, and 10.7% compared to CE, calibration, DEUP, ensembles, CALS, and CE+PE, respectively, while the uAUC increases by 17.7%, 43.9%, 56.7%, 8.4%, 11.2%, and 20.1%. We also verify an enhancement in the correlation between model residuals and the predicted uncertainty, Wasserstein distance of the uncertainty between the correct and wrong predicted sets, and ECE of 3.8×, 16.9×, 2.8× on average compared to the other baselines. Finally, all the baselines, except CALS, yielded a model with a larger error rate than EUAT.

4.2.4 Adversarial Training

Finally, we evaluate the EUAT in adversarial training settings. We opted to exclusively train our models with adversarial examples, utilizing the FGSM [13] to generate perturbations, with a predefined perturbation bound ϵ set to 4/255. Due to resource constraints and the overhead introduced by adversarial training, we did not deploy the ResNet50/ImageNet benchmark in adversarial settings.

We report the results of the different baselines on standard and adversarial settings in Table 5. Despite the challenges posed by adversarial scenarios, overall, we see similar trends compared to the standard training. As expected the adversarial error increases in all baselines and benchmarks considered. In half of the scenarios/metrics assessed, EUAT outperforms the baselines. Quantitatively, employing EUAT yields an average increase in uAUC of 20.2%, 10.9%, and 17.0%, when training a Wide-ResNet on Cifar100, a ResNet18 on Cifar10, and a ResNet18 on SVHN datasets, respectively. Moreover, while the gains in uA are slightly smaller, they still present

Table 5: Comparison of EUAT against the baselines considered using different evaluation metrics in the adversarial training scenario using three benchmarks.

		Standard Settings						Adversarial Settings					
Benchmark	**Baseline**	**uA**	**uAUC**	**Corr. w/ res.**	**Wasser. dist.**	**ECE**	**Error**	**uA**	**uAUC**	**Corr. w/ res**	**Wasser. dist.**	**ECE**	**Error**
Cifar100	**EUAT**	**0.850**	**0.902**	**0.709**	0.254	0.154	0.303	0.791	**0.853**	0.643	**0.224**	0.268	**0.436**
	CE	0.786	0.793	0.565	0.154	0.241	0.329	0.694	0.716	0.430	0.104	0.377	0.460
	Calibration	0.771	0.839	0.576	**0.271**	**0.087**	0.335	0.734	0.801	0.529	0.222	**0.202**	0.456
	DEUP	0.677	0.585	0.222	0.036	0.235	0.337	0.577	0.584	0.214	0.033	0.391	0.493
	Ensemble	0.800	0.817	0.597	0.161	0.183	**0.288**	**0.865**	**0.852**	**0.750**	0.194	0.334	**0.436**
	CALS	0.788	0.787	0.567	0.159	0.245	0.332	0.689	0.714	0.430	0.105	0.383	0.466
	CE+PE	0.773	0.750	0.535	0.113	0.286	0.354	0.625	0.633	0.346	0.059	0.422	0.482
Cifar10	**EUAT**	0.905	**0.921**	**0.635**	**0.414**	0.031	0.117	0.845	**0.883**	0.601	**0.353**	0.031	0.196
	CE	0.895	0.898	0.571	0.306	0.011	0.116	0.826	0.847	0.542	0.259	0.028	0.198
	Calibration	0.890	0.893	0.528	0.339	0.051	0.119	0.819	0.845	0.513	0.278	0.019	0.201
	DEUP	0.904	0.591	0.235	0.044	0.018	0.104	0.804	0.603	0.237	0.034	0.092	0.205
	Ensemble	**0.923**	0.894	0.535	0.310	**0.007**	**0.081**	**0.901**	0.881	**0.780**	0.348	0.083	**0.179**
	CALS	0.895	0.898	0.579	0.317	0.009	0.117	0.833	0.842	0.542	0.264	**0.015**	0.196
	CE+PE	0.895	0.850	0.564	0.232	0.046	0.119	0.819	0.789	0.505	0.177	0.104	0.197
SVHN	**EUAT**	0.944	**0.947**	**0.649**	**0.447**	0.030	0.072	0.794	**0.848**	0.588	**0.307**	**0.020**	0.281
	CE	0.931	0.893	0.590	0.335	**0.009**	0.080	0.749	0.773	0.473	0.196	0.134	0.307
	Calibration	0.929	0.916	0.521	0.428	0.138	0.082	0.758	0.815	0.516	0.261	**0.020**	0.299
	DEUP	0.933	0.560	0.258	0.035	0.029	0.071	0.667	0.535	0.302	0.022	0.313	0.354
	Ensemble	**0.951**	0.893	0.545	0.333	**0.009**	**0.053**	**0.881**	0.827	**0.681**	0.221	0.234	0.308
	CALS	0.940	0.894	0.592	0.340	**0.009**	0.069	0.820	0.757	0.481	0.206	0.117	**0.200**
	CE+PE	0.932	0.874	0.596	0.274	0.021	0.078	0.823	0.736	0.449	0.145	0.127	0.187

significant improvements of 15.2%, 1.4%, and 2.1% across the same models/datasets. Additionally, the error rates across baselines exhibit negligible variance, and we verify a larger separation of the uncertainty of the incorrect and correct predictions when performing AT with EUAT, which highlights the robustness of our method across different tasks and domains, reaffirming its efficacy in the challenging context of adversarial attacks.

5 Conclusion and Future Work

This paper introduces Error-Driven Uncertainty Aware Training, a novel approach designed to refine the estimation of model uncertainty. EUAT is engineered to achieve two primary objectives: first, to heighten uncertainty when models generate inaccurate predictions, and second, to output low uncertainty when predictions are correct. This dual-purpose strategy is achieved through the usage of two loss functions, which adapt based on whether training examples are correctly or incorrectly predicted by the model. By minimizing uncertainty for accurate predictions and maximizing it for mispredictions while retaining error rates, EUAT aims to enhance model reliability.

We evaluate EUAT against six different baselines and using six metrics, and the results consistently demonstrate EUAT's superior performance across the majority of the considered cases. Even when faced with competitive baselines, EUAT is still able to achieve comparable performance. Furthermore, we extend our evaluation to encompass diverse problems, including binary classification, out-of-distribution detection, and adversarial training settings. Across all evaluated domains, EUAT demonstrates an enhanced ability to differentiate between erroneous and accurate predictions based on uncertainty levels, thereby increasing model trustworthiness.

Given our current focus on evaluating EUAT solely within classification tasks using image recognition models, we envision broadening our scope to encompass other domains such as regression models or language models for machine translation or summarization, which present new and diverse challenges.

Acknowledgements

This work was supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under grant SFRH/BD/151470/2021 via projects with reference UIDB/50021/2020 and C645008882-00000055.PRR, by the NSA grant H98230-23-C-0274, and by the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, where we used the Bridges-2 GPU and Ocean resources at the Pittsburgh Supercomputing Center through allocation CIS220073, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] J. Antorán, J. U. Allingham, and J. M. Hernández-Lobato. Depth uncertainty in neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[3] H. Asgharnezhad, A. Shamsi, R. Alizadehsani, S. Khosravi, Abbas and-Nahavandi, Z. A. Sani, D. Srinivasan, and S. M. S. Islam. Objective evaluation of deep uncertainty predictions for covid-19 detection. *Scientific Reports*, 12, 2022.

[4] K. Brach, B. Sick, and O. Dürr. Single shot mc dropout approximation. *ArXiv*, abs/2007.03293, 2020. URL https://api.semanticscholar.org/CorpusID:220381176.

[5] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, page 3–14, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352024. URL https://doi.org/10.1145/3128572.3140444.

[6] A. Damianou and N. D. Lawrence. Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 207–215, Scottsdale, Arizona, USA, 2013. PMLR.

[7] Y. Ding, J. Liu, X. Xu, M. Huang, J. Zhuang, J. Xiong, and Y. Shi. Uncertainty-aware training of neural networks for selective medical image segmentation. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 156–173. PMLR, 2020.

[8] B.-S. Einbinder, Y. Romano, M. Sesia, and Y. Zhou. Training uncertainty-aware classifiers with conformalized deep learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, 2022.

[9] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. In *International Conference on Machine Learning*, 2017.

[10] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48. PMLR, 2016.

[11] J. Gawlikowski, C. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. Zhu. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56, 2023.

[12] S. J. Godsill. On the relationship between markov chain monte carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10(2):230–248, 2001.

[13] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70. PMLR, 2017.

[15] K. Gupta, A. Rahimi, T. Ajanthan, T. Mensink, C. Sminchisescu, and R. Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[18] A. Karandikar, N. Cain, D. Tran, B. Lakshminarayanan, J. Shlens, M. C. Mozer, and B. Roelofs. Soft calibration objectives for neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, 2021.

[19] J. U. Kim, S. T. Kim, H. J. Lee, S. Lee, and Y. M. Ro. Cua loss: Class uncertainty-aware gradient modulation for robust object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9), 2021.

[20] R. Krishnan and O. Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[22] M. Kull, T. S. Filho, and P. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, 2017.

[23] S. Lahlou, M. Jain, H. Nekoei, V. I. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=eGLdVRvvfQ.

[24] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[25] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1), 2017.

[26] B. Liu, I. B. Ayed, A. Galdran, and J. Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *Computer Vision and Pattern Recognition Conference*, 2022.

[27] B. Liu, J. Rony, A. Galdran, J. Dolz, and I. Ben Ayed. Class adaptive network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16070–16079, June 2023.

[28] D. J. C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472, 1992.

[29] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[31] T. Matsubara, N. Tax, R. Mudd, and I. Guy. TCE: A test-based approach to measuring calibration error. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2023.

[32] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[33] M. P. Naeini, G. F. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[34] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[36] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[37] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[38] T. Pearce, F. Leibfried, and A. Brintrup. Uncertainty in neural networks: Approximately bayesian ensembling. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.

[39] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[41] A. Shamsi, H. Asgharnezhad, A. Tajally, S. Nahavandi, and H. Leung. An uncertainty-aware loss function for training neural networks with calibrated predictions. *ArXiv*, abs/2110.03260, 2023.

[42] L. Smith and Y. Gal. Understanding measures of uncertainty for adversarial example detection. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2018.

[43] H. Wang and D.-Y. Yeung. A survey on bayesian deep learning. *ACM Comput. Surv.*, 53(5), sep 2020.

[44] K.-C. Wang, P. Vicol, J. Lucas, L. Gu, R. Grosse, and R. Zemel. Adversarial distillation of Bayesian neural network posteriors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018.

[45] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 2001.

[46] S. Zagoruyko and N. Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016*, 2016.