Multiclass Alignment of Confidence and Certainty for Network Calibration

Vinith Kugathasan and Muhammad Haris Khan

Mohamed bin Zayed University of Artificial Intelligence, UAE
{muhammad.haris}@mbzuai.ac.ae

Abstract. Deep neural networks (DNNs) have made great strides in pushing the state-of-the-art in several challenging domains. Recent studies reveal that they are prone to making overconfident predictions. This greatly reduces the overall trust in model predictions, especially in safety-critical applications. Early work in improving model calibration employs post-processing techniques which rely on limited parameters and require a hold-out set. Some recent train-time calibration methods, which involve all model parameters, can outperform the post-processing methods. To this end, we propose a new train-time calibration method, which features a simple, plug-and-play auxiliary loss known as multi-class alignment of predictive mean confidence and predictive certainty (MACC). It is based on the observation that a model miscalibration is directly related to its predictive certainty, so a higher gap between the mean confidence and certainty amounts to a poor calibration both for in-distribution and out-of-distribution predictions. Armed with this insight, our proposed loss explicitly encourages a confident (or underconfident) model to also provide a low (or high) spread in the pre-softmax distribution. Extensive experiments on ten challenging datasets, covering in-domain, out-domain, non-visual recognition and medical image classification scenarios, show that our method achieves state-of-the-art calibration performance for both in-domain and out-domain predictions. Our code and models will be publicly released.

Keywords: Network Calibration · Model Calibration · Uncertainty.

1 Introduction

Deep neural networks (DNNs) have displayed remarkable performance across many challenging computer vision problems e.g., image classification [7,13,23,45]. However, some recent works [12,35,40,48] have demonstrated that they tend to make overconfident predictions, and so are poorly calibrated. Consequently, the predicted confidences of classes are higher than the actual likelihood of their occurrences. A key reason behind this DNN behaviour is the supervision from zero-entropy signal which trains them to become over-confident. Poorly calibrated models not only create a general suspicion in the model predictions, but more importantly, they can lead to dangerous consequences in many safety-critical applications, including healthcare [8,44], autonomous vehicles [11], and legal research [49]. In such applications, providing a

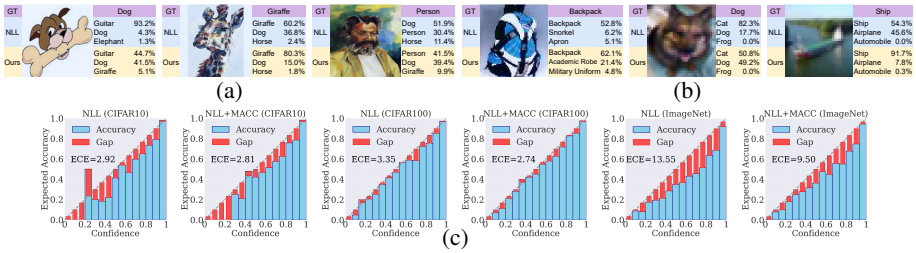


Fig. 1: We propose a new train-time calibration based on a novel auxiliary loss formulation (MACC). We compare between a model trained with NLL loss and ours (NLL+MACC). (a) shows out-of-domain performance (PACS) while (b) displays in-domain performance (Tiny-ImageNet and CIFAR10). NLL+MACC has higher confidence values for correct predictions (Giraffe (PACS)/Backpack (Tiny-ImageNet)) and lower confidence values for incorrect predictions (Dog (PACS)/Dog (CIFAR10)). (c) Reliability diagrams show that NLL+MACC improves bin-wise miscalibration, thereby alleviating under/over-confident predictions.

correct confidence is as significant as providing a correct label. For instance, in automated healthcare, if the control is not shifted to a doctor when the confidence of the incorrect prediction from a disease diagnosis network is high [20], it can potentially lead to disastrous outcomes. We have seen some recent attempts towards improving the network calibration. Among them, a simple technique is based on a post-hoc procedure, which transforms the outputs of a trained network [12]. The parameters of this transformation are typically learned on a hold-out validation set. Such post-hoc calibration methods are simple and computationally efficient, however, they are architecture and data-dependent [31]. Furthermore, in many real-world applications, the availability of a hold-out set is not guaranteed. Another route to reducing miscalibration is train-time calibration which tends to involve all model parameters. A dominant approach in train-time calibration methods proposes auxiliary losses that can be added to the task-specific loss (e.g., NLL) to reduce miscalibration [14,31,35,42]. These auxiliary losses aim at either increasing the entropy of the predictive distribution [31,35,42] or aligning the predictive confidence with the predictive accuracy [14,25].

We take the train-time route to calibration, and propose an auxiliary loss function: Multi-class alignment of predictive mean confidence and predictive certainty (MACC). It is founded on the observation that a model's predictive certainty is correlated to its calibration performance. So, a higher gap between the predictive mean confidence and predictive certainty translates directly to a greater miscalibration both for in-distribution and out-of-distribution predictions. If a model is confident then it should also produce a relatively low spread in the logit distribution and vice versa. Proposed loss function is differentiable, operates on minibatches and is formulated to be used with other task-specific loss functions. Besides showing effectiveness for calibrating in-distribution examples, it is also capable of improving calibration of out-of-distribution examples (Fig. 1). **Contributions: (1)** We empirically observe a correlation between a model's predictive certainty and its calibration performance. **(2)** To this end, we propose a simple, plug-and-play auxiliary loss term (MACC) which attempts to *align the predictive mean confidence with the predictive certainty for all class labels*. It can be used with other task-specific loss functions, such as Cross Entropy (CE), Label Smoothing (LS) [36] and Focal Loss (FL) [35]. **(3)** Besides the predicted class label, it also reduces the

gap between the certainty and mean confidence for non-predicted class labels, thereby improving the calibration of non-predicted class labels. **(4)** We carry out extensive experiments on three in-domain scenarios, CIFAR-10/100[22], and Tiny-ImangeNet[4], a class-imbalanced scenario SVHN [38], and four out-of-domain scenarios, CIFAR-10/100-C (OOD) [17], Tiny-ImangeNet-C (OOD) [17] and PACS [28]. Results show that our loss is consistently more effective than the existing state-of-the-art methods in calibrating both in-domain and out-of-domain predictions. Moreover, we also show the effectiveness of our approach on non-visual pattern recognition task of natural language classification (20 Newsgroups dataset [27]) and a medical image classification task (Mendeley dataset [21]). Finally, we also report results with a vision transformer-based baseline (DeiT-Tiny [47]) to show the applicability of our method.

2 Related Work

Post-hoc calibration methods: A classic approach for improving model calibration, known as post-hoc calibration, transforms the outputs of a trained model [6,12,46,50]. Among different post-hoc calibration methods, a simple technique is temperature scaling (TS) [12], which is a variant of Platt scaling [12]. It scales the logits (i.e. pre-softmax activations) by a single temperature parameter, which is learned on a hold-out validation set. TS increases the entropy of the predictive distribution, which is beneficial towards improving model calibration. However, it decreases the confidence of all predictions, including the correct one. TS which relies on a single parameter for transformation can be generalized to a matrix transform, where the matrix is also learnt using a hold-out validation set. Dirichlet calibration (DC) employs Dirichlet distributions for scaling the Beta-calibration [24] method to a multi-class setting. DC is incorporated as a layer in a neural network on log-transformed class probabilities, which is learnt using a hold-out validation set. Although TS improves model calibration for in-domain predictions, [40] showed that it performs poorly for out-of-domain predictions. To circumvent this, [46] proposed to perturb the validation set before performing the post-hoc calibration. Recently, [33] proposed a ranking model to improve the post-hoc model calibration, and [6] used a regressor to obtain the temperature parameter at the inference stage.

Train-time calibration methods: Brier score is considered as one of the earliest train-time calibration technique for binary probabilistic forecast [3]. Later, [12] demonstrated that the models trained with negative log-likelihood (NLL) tend to be over-confident, and thus, there is a dissociation between NLL and calibration. Several works proposed auxiliary losses that can be used with NLL to improve miscalibration. For instance, [42] penalized the over-confident predictions by using entropy as a regularization term, and [36] showed that label smoothing (LS) can improve model calibration. A similar insight was reported by [35], that Focal loss (FL) implicitly improves model calibration. It minimizes the KL divergence between the predictive distribution and the target distribution, and at the same time increases the entropy of the predictive distribution. These methods establish that implicit or explicit maximization of entropy improves calibration performance. Based on this observation, [31] proposed a calibration technique based on inequality constraints, which imposes a margin between logit distances. Recently, [29] incorporated the difference between confidence and accuracy (DCA) as an auxiliary

loss term with the the cross-entropy loss. Similarly, [25] developed an auxiliary loss term (MMCE), for model calibration that is computed with a reproducing kernel in a Hilbert space [10]. Prior methods, such as [25,29], only calibrate the maximum class confidence. To this end, [14] proposed an auxiliary loss term, namely MDCA, that calibrates the non-maximum class confidences along with the maximum class confidence. We also take the train-time calibration route, however, different to existing methods, we propose to minimize the gap between the predictive mean confidence and predictive certainty to improve model calibration.

Other calibration methods: Some methods learn to discard OOD samples, either at train-time or post-hoc stage, which reduces over-confidence and leads to improved calibration. Hein et al. [15] demonstrated ReLU makes DNNs provide high confidence for an input sample that lies far away from the training samples. Guo et al. [12] explored the impact of width, and depth of a DNN, batch normalization, and weight decay on model calibration. For more literature on calibrating a DNN through OOD detection, we refer the reader to [5,18,34,41].

Calibration and uncertainty estimation in DNNs: Many probabilistic approaches emerge from the Bayesian formalism [1], in which a prior distribution over the neural network (NN) parameters is assumed, and then a training data is used to obtain the posterior distribution over the NN parameters, which is then used to estimate predictive uncertainty. Since the exact Bayesian inference is computationally intractable, several approximate inference techniques have been proposed, including variational inference [2,32], and stochastic expectation propagation [19]. Ensemble learning is another approach for quantifying uncertainty that uses the empirical variance of the network predictions. We can create ensembles using different techniques. For instance, with the differences in model hyperparameters [48], random initialization of weights and random shuffling of training examples [26], dataset shift [40], and Monte Carlo (MC) dropout [9,51]. In this work, we chose to use MC dropout [9] to estimate predictive mean confidence and predictive uncertainty of a given example for all class labels. It provides a distribution of class logit scores and is simple to implement. However, the conventional implementation of MC dropout can incur high computational cost for large datasets, architectures, and longer training schedules. To this end, we resort to an efficient implementation of MC dropout that greatly reduces this computational overhead.

3 Proposed Methodology

Preliminaries: We consider the task of classification where we have a dataset $\mathcal{D} = \langle(\mathbf{x}_i, y_i^*)\rangle_{i=1}^N$ of N input examples sampled from a joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X}$ is an input space, and $\mathcal{Y}$ is the label space. $\mathbf{x}_i \in \mathcal{X} \in \mathbb{R}^{H \times W \times C}$ is an input image with height H, width W, and number of channels C. Each image has a corresponding ground truth class label $y_i^* \in \mathcal{Y} = \{1, 2, ..., K\}$. Let us denote a classification model $\mathcal{F}_{cls}$, that typically outputs a confidence vector $\mathbf{s}_i \in \mathbb{R}^K$. Since each element of vector $\mathbf{s}_i$ is a valid (categorical) probability, it is considered as the confidence score of the corresponding class label. The predicted class label $\hat{y}_i$ can be computed as: $\hat{y}_i = \arg\max_{y \in \mathcal{Y}} \mathbf{s}_i[y]$. Likewise, the confidence score of the predicted class $\hat{y}_i$ is obtained as: $\hat{s}_i = \max_{y \in \mathcal{Y}} \mathbf{s}_i[y]$.

3.1 Definition and Quantification of Calibration

Definition: We can define a perfect calibration if the (classification) accuracy for a given confidence score is aligned with this confidence score for all possible confidence scores [12]: $\mathbb{P}(\hat{y} = y^*|\hat{s} = s) = s \quad \forall s \in [0, 1]$, where $\mathbb{P}(\hat{y} = y^*|\hat{s} = s)$ is the accuracy for a given confidence score $\hat{s}$. The expression only captures the calibration of the predicted label i.e. associated with the maximum class confidence score $\hat{s}$. The confidence score of non-predicted classes, also called as non-maximum class confidence scores, can also be calibrated. It provides us with a more general definition of perfect calibration and can be expressed as: $\mathbb{P}(y = y^*|\mathbf{s}[y] = s) = s \quad \forall s \in [0, 1]$.
Expected calibration error (ECE): ECE is computed by first obtaining the absolute difference between the average confidence of the predicted class and the average accuracy of samples, that are predicted with a particular confidence score. This absolute difference is then converted into a weighted average by scaling it with the relative frequency of samples with a particular confidence score. The above two steps are repeated for all confidence scores and then the resulting weighted averages are summed [37]: $\text{ECE} = \sum_{i=1}^{M} \frac{|B_i|}{N} \left| \frac{1}{|B_i|} \sum_{j:\hat{s}_j \in B_i} \mathbb{I}(\hat{y}_j = y_j^*) - \frac{1}{|B_i|} \sum_{j:\hat{s}_j \in B_i} \hat{s}_j \right|$. Where N is the total number of examples. Since the confidence values have a continuous interval, the confidence range $[0, 1]$ is divided into M bins. $|B_i|$ is the number of examples falling in i^{th} confidence bin. $\frac{1}{|B_i|} \sum_{j:\hat{s}_j \in B_i} \mathbb{I}(\hat{y}_j = y_j^*)$ denotes the average accuracy of examples lying in i^{th} bin, and $\frac{1}{|B_i|} \sum_{j:\hat{s}_j \in B_i} \hat{s}_j$ represents the average confidence of examples belonging to i^{th} confidence bin. The ECE metric for measuring DNN miscalibration has two limitations. First, the whole confidence vector is not accounted for calibration. Second, due to binning of the confidence interval, the metric is not differentiable. See description on Maximum calibration error (MCE) in supplementary material.
Static calibration error (SCE): SCE extends ECE by taking into account the whole confidence vector, thereby measuring the calibration performance of non-maximum class confidences [39], $\text{SCE} = \frac{1}{K} \sum_{i=1}^{M} \sum_{j=1}^{K} \frac{|B_{i,j}|}{N} \left| A_{i,j} - C_{i,j} \right|$. Where K represents the number of classes and $|B_{i,j}|$ is the number of examples from the j^{th} class and the i^{th} bin. $A_{i,j} = \frac{1}{|B_{i,j}|} \sum_{k:\mathbf{s}_k[j] \in B_{i,j}} \mathbb{I}(j = y_k)$ denotes the average accuracy and $C_{i,j} = \frac{1}{|B_{i,j}|} \sum_{k:\mathbf{s}_k[j] \in B_{i,j}} \mathbf{s}_k[j]$ represents the average confidence of the examples belonging to the j^{th} class and the i^{th} bin. Similar to ECE metric, SCE metric is not differentiable, and so it cannot be used as a loss in gradient-based learning methods.

3.2 Proposed Auxiliary Loss: MACC

Our auxiliary loss (MACC) aims at reducing the deviation between the predictive mean confidence and the predictive certainty for predicted and non-predicted class labels.
Quantifying mean confidence and certainty: Our proposed loss function requires the estimation of *class-wise mean confidence and certainty*. We choose to use the MC dropout method [9] to estimate both of these quantities because it provides a distribution of (logit) scores for all possible classes and only requires the addition of a single dropout layer $(\mathcal{M})$, which in our case, is added between the feature extractor $f(\cdot)$ that

generates features and the classifier $g(\cdot)$ that projects the extracted features into class-wise logits vector. The conventional implementation of MC dropout technique requires

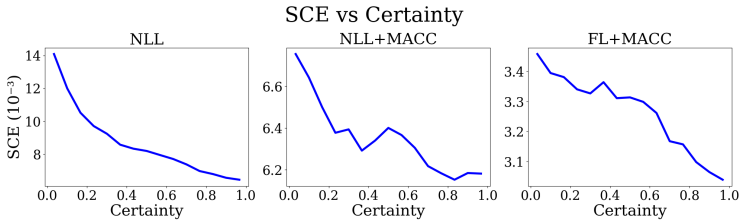


Fig. 2: Left: We investigate if there is a relationship between static calibration error and individual (output) predictive uncertainties. We observe (negative) correlation between a model's predictive certainty and its calibration error (SCE) i.e. as the certainty increases, the calibration error goes down (CIFAR10 trained on ResNet56 model with dropout). Middle & right: Based on this observation, we propose to align predictive mean confidence with the predictive certainty (NLL/FL+MACC), which allow a rapid reduction in the calibration error in comparison to baseline (NLL). See supplementary material for details of the plot.

N MC forward passes for an input example $\mathbf{x}_i$ through the model $\mathcal{F}_{cls}$. From the resulting logits distribution, we can then estimate the mean and variance for each class j, which reflects for $\mathbf{x}_i$, its predictive mean logit score $\bar{\mathbf{z}}_i[j]$ and the predictive uncertainty in logit scores $\mathbf{u}_i[j]$, respectively, where $\bar{\mathbf{z}}_i, \mathbf{u}_i \in \mathbb{R}^K$. To obtain predictive mean confidence $\bar{\mathbf{s}}_i[j]$, we apply softmax to $\bar{\mathbf{z}}_i[j] \forall j$. The certainty $\mathbf{c}_i[j]$ is obtained from the uncertainty $\mathbf{u}_i[j]$ as: $\mathbf{c}_i[j] = 1 - \tanh(\mathbf{u}_i[j])$. The $\tanh$ is used to scale the uncertainty values between 0 and 1.

We resort to an efficient implementation of MC dropout technique aimed at reducing its computational overhead, which is of concern during model training. Specifically, we feed an input example $\mathbf{x}_i$ to the feature extractor network only once and obtain the extracted features $\mathbf{f}_i$. These extracted features are then fed to the combination of dropout layer and classifier ($g \circ \mathcal{M}(\mathbf{f}_i)$) for N number of MC forward passes. Specifically, $\mathbf{u}_i[j] = \frac{1}{N-1} \sum_{m=1}^{M} ([g \circ \mathcal{M}_m(\mathbf{f}_i)]_j - \bar{\mathbf{z}}_i[j])^2$, where $\bar{\mathbf{z}}_i[j]$ represents the mean of the logit distribution given by: $\bar{\mathbf{z}}_i[j] = \frac{1}{N} \sum_{m=1}^{M} [g \circ \mathcal{M}_m(\mathbf{f}_i)]_j$. This so-called architecture-implicit implementation of MC dropout enjoys the benefit of performing only a single forward pass through the feature extractor f as opposed to N forward passes in the conventional implementation. We empirically observe that, for 10 MC forward passes, the efficient implementation reduces the overall training time by 7 times compared to the conventional implementation (see suppl.). Deep ensembles [26] is an alternate to MCDO, however, it is computationally expensive to be used in a train-time calibration approach. On CIFAR10, training deep ensembles with 10 models require around 7.5 hours whereas ours with 10 forward passes, only require around an hour.

MACC: The calibration is a frequentist notion of uncertainty and could be construed as a measure reflecting a network's *overall predictive uncertainty* [26]. So, we investigate if there is a relationship between static calibration error and individual (output) predictive uncertainties. We identify a (negative) correlation between a model's predictive certainty and its calibration error (SCE). In other words, as the certainty increases, the calibration error goes down (Fig. 2). With this observation, we propose to align the

predictive mean confidence of the model with its predictive certainty. Our loss function is defined as:

$$\mathcal{L}_{\text{MACC}} = \frac{1}{K}\sum_{j=1}^{K}\left|\frac{1}{M}\sum_{i=1}^{M}\bar{\mathbf{s}}_i[j] - \frac{1}{M}\sum_{i=1}^{M}\mathbf{c}_i[j]\right|, \tag{1}$$

where $\bar{\mathbf{s}}_i[j]$ denotes the predictive mean confidence of the i^{th} example in the mini-batch belonging to the j^{th} class. Likewise, $\mathbf{c}_i[j]$ represents the certainty of the i^{th} example in the mini-batch belonging to the j^{th} class. M is the number of examples in the mini-batch, and K is the number of classes.

Discussion: Given an example, for which a model predicts high mean confidence, our loss formulation forces the model to also produce relatively low spread in logits distribution and vice versa. This alignment directly helps towards improving the model calibration. Fig. 2 shows that, compared to baseline, a model trained with our loss allows a rapid decrease in calibration error. Moreover, Fig. 3a, 3b show that when there are relatively greater number of examples with a higher gap between (mean) confidence and certainty (i.e. the distribution is more skewed towards right), a model's calibration is poor compared to when there are relatively smaller number of examples with a higher gap (Fig. 3c, 3d). The proposed auxiliary loss is a simple, plug-and-play term. It is differentiable and operates over minibatch and thus, it can be used with other task-specific loss functions to improve the model calibration, $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \beta.\mathcal{L}_{\text{MACC}}$, where β represents the weight with which our $\mathcal{L}_{\text{MACC}}$ is added to the task-specific loss function $\mathcal{L}_{\text{task}}$ e.g., CE, LS [36] and FL [35].

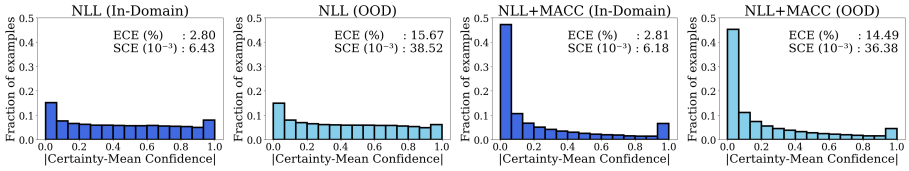


Fig. 3: Empirical distribution of difference between predictive certainty and the predictive mean confidence of all the classes for in-domain examples (CIFAR10), and out-of-domain (OOD) examples (CIFAR10-C). (Left - (a),(b)) When there are relatively greater number of examples with higher gap between the mean confidence and certainty, calibration errors (ECE and SCE) are higher, compared to when there are relatively smaller number of examples (Right - (c),(d)). For (a) and (b), a ResNet56 model with dropout is trained with NLL. For (c) and (d), the same model is trained with NLL+MACC (ours).

4 Experiments

Datasets: To validate in-domain calibration performance, we use four challenging image classification datasets: CIFAR10[22], CIFAR100[22], Tiny-ImageNet[4] and Mendeley V2[21] and a natural language processing dataset: 20 Newsgroups[27]. Tiny-Image -Net is a subset of ImageNet[43] comprising 200 classes. Further, to report calibration performance in out-of-domain scenarios, we show results on four challenging benchmarks: CIFAR10-C[16], CIFAR100-C[16], Tiny-ImageNet-C[16] and PACS[28].

For CIFAR10-C, CIFAR100-C and Tiny-ImageNet-C, we use their corresponding in-domain benchmarks for training and validation. Finally, to evaluate calibration performance under class imbalance, we report results on SVHN[38].

Implementation details and evaluation metrics: We use ResNet[13] and DeiT-Tiny[47] (only for CIFAR10) as the backbone networks in our experiments. For our method, we insert a single dropout layer in between the penultimate feature layer and the final classifier of the ResNet architecture. We also input the predictive mean confidence, obtained for our MACC, to the task-specific loss.We set the number of MC samples to 10 in all experiments. The dropout ratio is sought in the range $p \in \{0.2, 0.3, 0.5\}$ using the validation set. See suppl. material for details. We report the calibration performance with ECE[37] and SCE[39] metrics and the classification performance with top-1 accuracy. The number of bins is $\mathrm{M} = 15$ for both the calibration metrics across all the experiments. Moreover, we plot reliability diagrams and report AUROC scores.

Baselines: We evaluate MACC against models trained with CE, LS[36], FL[30], adaptive sample-dependent focal loss (FLSD)[35], brier score (BS)[3] and MMCE[25]. We also compare against the recent auxiliary loss functions: MbLS[31] and MDCA[14]. Hyper-parameters of the compared methods are set based on the values reported in the literature. For both MDCA and our loss (MACC), the relative weight is chosen from $\beta \in \{1, 5, 10, 15, 20, 25\}$ and the most accurate model on the validation set is used to report the calibration performance, following MDCA[14] implementation. Meanwhile, the scheduled γ in FLSD is set to 5 for $s_k \in [0, 0.2)$ and 3 for $s_k \in [0.2, 1)$, where s_k is the confidence score of the correct class. Refer to the supplementary for the detailed description of these hyperparameters.

Experiments with task-specific loss functions: Our loss (MACC) is developed to be used with a task-specific loss function. We consider CE (NLL), LS and FL as the task-specific losses and report the calibration performance with and without incorporating our MACC. For LS we use $\alpha \in \{0.05, 0.1\}$ and for FL we use $\gamma \in \{1, 2, 3\}$ and the most accurate model on the validation set is used to report the performance. Table 1 shows that our auxiliary loss function (MACC) consistently improves the calibration performance of all tasks-specific losses across six datasets. We also note that FL is a much stronger task-specific loss function in calibration performance in all datasets, except SVHN and 20 Newsgroups. The CE loss performs relatively better than FL loss on SVHN and LS performs better on 20 Newsgroups. We choose to report performance with FL+MACC on all datasets, except SVHN (for which we use CE loss), in all subsequent experiments.

Comparison with state-of-the-art (SOTA): We compare the calibration performance against recent SOTA train-time calibration methods (Table 2, Table 3). We use NLL+ MbLS to report the performance as it provides better results than FL+MbLS (see suppl.). Our method achieves lower calibration errors in ECE, SCE and AUROC metrics across six datasets. To demonstrate the effectiveness of MACC on natural language classification, we conduct experiments on the 20 Newsgroups dataset (Table 2). Our FL+MACC outperforms others in both SCE and ECE metrics. Experiments with vision-transformer based backbone architecture, namely DeiT-Tiny [47] show that our FL+MACC is capable of improving the calibration performance of DeiT. Note that, DeIT is a relatively stronger baseline in calibration performance compared to ResNet

(see Table 2). For training DeiT models, we use the hyperparameters specified by the authors of DeiT.

Table 1: Calibration performance in SCE (10^{-3}) and ECE (%) metrics of our auxiliary loss (MACC) when added to three task-specific losses: CE, LS, and FL. Throughout, the best results are in bold, and the second best are underlined.

Dataset	Model	NLL		NLL+MACC		LS[36]		LS+MACC		FL[30]		FL+MACC	
		SCE	ECE	SCE	ECE	SCE	ECE	SCE	ECE	SCE	ECE	SCE	ECE
CIFAR10	ResNet56	6.50	2.92	6.18	2.81	5.90	1.85	5.51	1.57	<u>3.79</u>	<u>0.64</u>	**3.04**	**0.59**
CIFAR100	ResNet56	2.01	3.35	<u>1.99</u>	2.74	2.08	<u>0.86</u>	2.07	0.92	<u>1.99</u>	0.89	**1.97**	**0.64**
Tiny-ImageNet	ResNet50	2.06	13.55	1.72	9.50	1.50	2.04	**1.37**	<u>1.37</u>	1.50	3.52	<u>1.44</u>	**1.33**
SVHN	ResNet56	<u>1.70</u>	<u>0.43</u>	**1.50**	**0.27**	11.70	4.95	7.72	3.10	7.79	3.55	<u>1.70</u>	0.49
20 Newsgroups	GP CNN	23.05	21.11	20.30	18.32	<u>10.35</u>	<u>5.80</u>	**9.61**	**2.17**	21.30	19.54	13.84	11.28
Mendeley	ResNet50	206.98	16.05	77.12	7.59	133.25	5.32	**57.70**	<u>4.52</u>	160.58	5.13	<u>66.97</u>	**4.14**

Temperature Scaling (TS): MACC outperforms NLL/FL + TS (Table 3). We report the best calibration obtained for TS with the primary losses of NLL and FL. For TS we follow the same protocol used by the MDCA [14] where 10% of the training data is set aside as the hold-out validation set and a grid search between the range of 0 to 10 with a step-size of 0.1 is performed to find the optimal temperature value that gives the least NLL on the hold-out set. In CIFAR10/100 and SVHN, the obtained metric scores are similar to that of MDCA [14]. For Tiny-ImageNet, MDCA [14] does not report results, and our results are better than MbLS [31], which uses the same protocol as ours.

Class-wise calibration performance and test accuracy: Table 4 reports class-wise ECE (%) scores of competing calibration approaches, including MDCA and MbLS, on SVHN and CIFAR10 datasets, with ResNet56. In SVHN, NLL+MACC (ours) achieves the lowest ECE(%) in three classes while demonstrating the second best score in other four. FL+ MACC provides the best values in two classes and the second best values in another two classes. NLL+MbLS also performs well, being the best in five classes. In CIFAR10, FL+MACC (ours) provides the best ECE(%) scores in five classes while showing the second best in all others. Table 2 shows the discriminative performance (top-1 accuracy %) of our loss (MACC) along with the other competing approaches. Our loss shows superior accuracy than most of the existing losses, including MDCA, in CIFAR100 and Tiny-ImageNet. Moreover, it provides the best accuracy in SVHN, Mendeley and 20 Newsgroups.

Out-of-distribution performance: Table 5 reports the out-of-domain calibration performance on the CIFAR10-C, CIFAR100-C and Tiny-ImageNet-C benchmarks. In both CIFAR10-C and CIFAR100-C datasets, our loss records the best calibration performance in ECE and SCE metrics. In Tiny-ImageNet-C, our loss shows the lowest ECE score and reveal the second lowest SCE score. We plot calibration performance as a function of corruption level in CIFAR10-C dataset (see Fig. 4 suppl.). Our loss consistently obtains lowest ECE and SCE across all corruption levels. For the OOD evaluation, including CIFAR10-C, CIFAR100- C, and Tiny-ImageNet-C, we follow the same protocol and train/val splits as used for in-domain evaluation. Specifically, we train a model using the training split, and optimize parameters using the validation split and the trained model is then evaluated on the in-domain test set or the corrupted test set.

Table 2: Comparison of calibration performance in SCE (10^{-3}) and ECE (%) metrics with the SOTA train-time calibration methods.

Dataset	Model	BS[3]			MMCE[25]			FLSD[35]			FL+MDCA[14]			NLL+MbLS[31]			FL/NLL+MACC		
		SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.
CIFAR10	ResNet56	4.78	1.67	92.46	5.87	1.74	91.75	7.87	3.17	92.37	3.44	0.79	92.92	4.63	1.48	93.41	**3.04**	**0.59**	92.86
	DeiT-Tiny	–	–	–	–	–	–	–	–	–	3.63	1.52	97.11	**3.01**	0.56	96.78	3.02	**0.48**	96.43
CIFAR100	ResNet56	2.08	4.75	69.64	1.98	2.76	69.71	2.05	1.76	70.97	**1.92**	0.68	70.34	1.99	0.96	71.33	1.97	**0.64**	70.50
Tiny-ImageNet	ResNet50	–	–	–	–	–	–	1.50	2.75	60.39	1.44	2.07	60.24	**1.42**	1.59	62.69	1.44	**1.33**	61.60
SVHN	ResNet56	2.41	0.51	96.57	12.34	5.88	95.51	17.49	8.59	95.87	1.77	0.32	96.10	**1.43**	0.37	96.59	1.50	**0.27**	96.74
20 Newsgroups	GP CNN	21.44	18.64	66.08	17.32	14.76	67.54	14.78	11.62	66.81	17.40	15.47	67.04	17.59	15.55	67.74	**13.84**	**11.28**	67.87
Mendeley	ResNet50	224.34	15.73	76.28	199.16	10.98	78.69	146.19	4.16	79.17	177.72	7.85	78.69	176.93	9.70	78.85	**66.97**	**4.14**	80.93

Table 3: Calibration performance with Temperature Scaling (TS) & comparison of calibration performance in AUROC metric with SOTA train-time calibration methods.

Dataset	Comparison with TS										SOTA Comparison		
	NLL+MACC		NLL+TS			FL+MACC		FL+TS			MDCA	MbLS	MACC
	SCE	ECE	SCE	ECE	T	SCE	ECE	SCE	ECE	T	AUROC Score		
CIFAR10	6.18	2.81	4.12	0.87	1.4	3.04	0.59	3.79	0.64	1.0	**0.9966**	0.9958	**0.9966**
CIFAR100	1.99	2.74	1.84	1.36	1.1	1.97	0.64	1.99	0.89	1.0	**0.9922**	0.9916	**0.9922**
Tiny-ImageNet	1.72	9.50	2.06	13.55	1.0	1.44	1.33	2.42	18.05	0.6	0.9848	0.9811	**0.9858**
SVHN	1.50	0.27	2.80	1.01	1.2	1.70	49	3.00	0.91	0.8	0.9973	**0.9977**	**0.9977**

Table 4: Class-wise calibration performance in ECE(%) of competing approaches on SVHN and CIFAR10 benchmarks (ResNet56).

Loss	Classes																			
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
	SVHN										CIFAR10									
NLL+MDCA	0.17	0.20	0.34	0.22	0.13	0.16	0.15	0.17	0.16	0.14										
NLL+MACC	0.12	0.16	**0.14**	0.20	0.14	**0.14**	**0.10**	0.16	0.19	0.14										
FL+MDCA	0.13	0.22	0.21	0.20	0.16	0.18	0.16	0.14	0.16	0.22	0.33	**0.32**	0.42	0.72	**0.21**	0.39	0.30	**0.21**	**0.21**	0.33
NLL+MbLS	**0.07**	**0.14**	0.18	0.22	**0.11**	0.17	0.13	**0.09**	0.18	**0.13**	**0.24**	0.51	0.33	0.68	0.44	0.51	0.48	0.57	0.46	0.41
FL+MACC	0.17	0.23	0.18	**0.18**	0.17	0.19	0.17	0.14	**0.10**	0.17	0.33	0.34	**0.31**	**0.45**	0.30	**0.33**	**0.25**	0.28	0.23	**0.23**

Table 5: Out of Domain (OOD) calibration performance of competing approaches across CIFAR10-C, CIFAR100-C and Tiny-ImageNet-C.

Dataset	Model	FL+MDCA			NLL+MbLS			FL+MACC		
		SCE (10^{-3})	ECE (%)	Acc. (%)	SCE (10^{-3})	ECE (%)	Acc. (%)	SCE (10^{-3})	ECE (%)	Acc. (%)
CIFAR10 (In-Domain)	ResNet56	3.44	0.79	92.92	4.63	1.48	93.41	**3.04**	**0.59**	92.86
CIFAR10-C (OOD)		29.01	11.51	71.30	27.61	12.21	73.75	**23.71**	**9.10**	72.85
CIFAR100 (In-Domain)	ResNet56	**1.92**	0.68	70.34	1.99	0.96	71.33	1.97	**0.64**	70.5
CIFAR100-C (OOD)		4.09	12.21	44.74	4.03	12.48	45.60	**4.01**	**12.11**	44.90
Tiny-ImageNet (In-Domain)	ResNet50	1.44	2.07	60.24	**1.42**	1.59	62.69	1.44	**1.33**	61.60
Tiny-ImageNet-C (OOD)		3.87	22.79	20.74	**3.04**	18.17	23.70	3.46	**17.82**	21.29

We also show the OOD calibration performance on the PACS dataset under two different evaluation protocols. In first, following [14], a model is trained on **Photo** domain while **Art** domain is used as the validation set, and the trained model is then tested on the rest of domains. Table 6 shows that the proposed loss obtains the best calibration performance in ECE score, while the second best in SCE. In second, a model is trained on each domain and then tested on all other domains. In this protocol, 20% of images corresponding to the training domain is randomly sampled as the validation set, and the remaining 3 domains form the test set. Table 7 shows that FL+MACC provides improved calibration than all other competing approaches.

Mitigating Under/Over-Confidence: We plot reliability diagrams to reveal the effectiveness of our method in mitigating under/over-confidence (Fig. 1c & 4 (top)). Furthermore, we plot confidence histograms to illustrate the deviation between the overall

Table 6: OOD calibration performance (SCE (10^{-2}) & ECE (%)) on PACS when ResNet18 model is trained on **Photo**, validated on **Art**, and tested on **Sketch** and **Cartoon** [14].

Domain	FL+MDCA			NLL+MbLS			FL+MACC		
	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE
Cartoon	25.34	15.62	44.05	27.69	12.82	30.88	32.17	**10.19**	**22.18**
Sketch	29.14	14.82	40.16	26.57	**11.40**	**18.03**	24.94	12.59	28.14
Average	27.24	15.22	42.10	27.13	12.11	**24.45**	28.55	**11.39**	25.16

Table 7: Out of Domain (OOD) calibration performance (SCE (10^{-2}) & ECE (%)) on PACS when ResNet18 model is trained on each domain and tested on other 3 domains. Validation set comprises 20% randomly sampled images from the training domain.

FL+MDCA			NLL+MbLS			FL+MACC			FL+MDCA			NLL+MbLS			FL+MACC		
Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE	Acc.	SCE	ECE
Photo Domain									**Art Domain**								
38.48	13.59	38.93	35.32	13.44	33.77	34.75	**9.09**	**14.63**	56.87	9.33	24.81	57.46	**8.06**	**15.21**	51.45	9.70	16.28
Cartoon Domain									**Sketch Domain**								
62.68	**5.79**	13.91	59.43	6.38	9.62	57.47	6.58	**5.95**	18.31	17.99	54.37	11.57	19.50	57.17	13.24	**16.68**	**45.23**
Average																	
						44.09	11.68	33.00	40.94	11.83	28.94	39.23	**10.51**	**20.52**			

confidence (dotted line) and accuracy (solid line) of the predictions (Fig. 4 (bottom)). Fig. 4a & 4b show that our method can effectively mitigate the under-confidence of a model trained with LS loss. Fig. 4e & 4f illustrate that our method notably reduces the gap between the overall confidence and accuracy, thereby mitigating the under-confident behaviour. Likewise, Fig. 4c, 4d, 4g & 4h display the capability of our method in mitigating the over-confidence of an uncalibrated model.

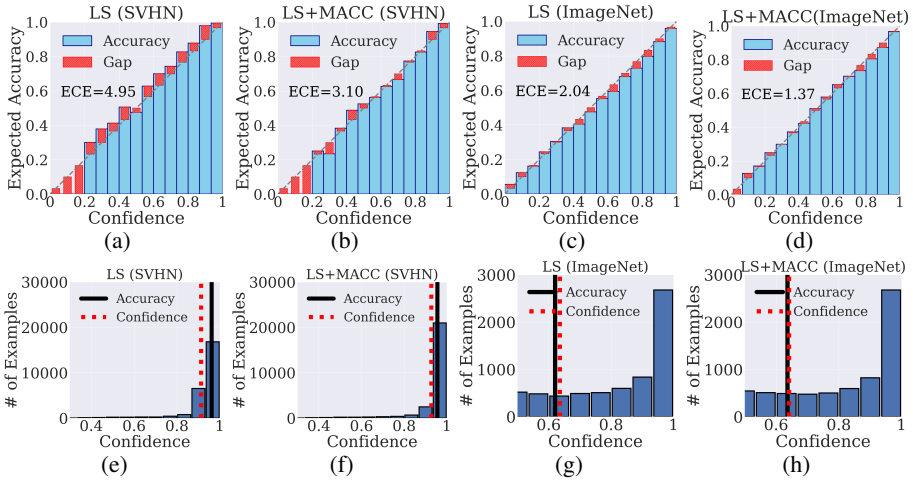
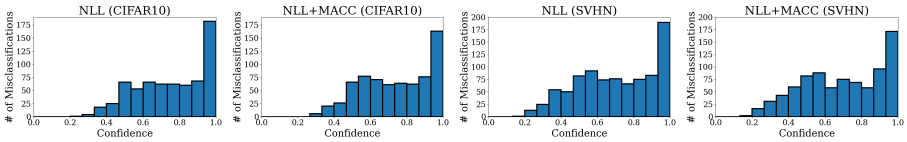


Fig. 4: Reliability diagrams (a,b,c,d) and confidence histograms (e,f,g,h) of (ResNet) models trained with LS and LS+MACC. (a,b,e,f) show that our method is effective in reducing under-confidence, while (c,d,g,h) reveal that it can reduce over-confidence. We refer to the supplementary for similar plots with NLL and FL.

Confidence of incorrect predictions: Fig. 5 shows the histogram of confidence values for the incorrect predictions. After adding our auxiliary loss (MACC) to CE loss, the confidence values of the incorrect predictions are decreased (see also Fig. 1a & 1b).

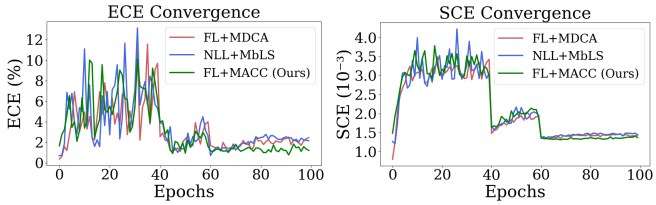


(a) (b) (c) (d)

Fig. 5: Confidence histogram of incorrect predictions from CIFAR10 and SVHN (ResNet56).

Calibration performance under class imbalance: We use SVHN to validate the calibration performance under class imbalance, which has a class imbalance factor of 2.7 [14]. Both Tables 2 and 4 show that, compared to competing approaches, MACC not only improves calibration performance over the (whole) dataset, but also displays competitive calibration performance in each class. This is largely because MACC considers whole confidence/certainty vector, which calibrates even the non-predicted classes.

Comparison of ECE and SCE Convergence with SOTA: Fig. 6 shows that the proposed loss function MACC is optimizing both ECE and SCE, better than the current SOTA methods of MbLS and MDCA. Although MACC does not explicitly optimize ECE and SCE, it achieves better ECE and SCE convergence. Moreover, compared to others, it consistently decreases both SCE and ECE throughout the evolution of training.



(a) (b)

Fig. 6: ECE and SCE convergence plot while training ResNet50 on the Tiny-ImageNet for MDCA, MbLS and ours (MACC). We use the learning rate decay factor of 0.1 and 0.02 at epochs 50 and 70, respectively, while for MDCA and MbLS the factor is 0.1.

Impact of our model and training settings: Table 8 shows the performance of the task-specific loss functions and the SOTA calibration losses with the same architecture and training settings as in our loss. i.e., ResNet model with dropout is used and the learning rate at the last stage of the learning rate scheduler is reduced. Upon comparing Table 8 with Table 1 and Table 2, we note that with our model architecture and learning rate setting, as such, the calibration of competing losses is poor than our loss. So, the effectiveness of our method is not due to the model architecture or some specific training settings but because of our loss formulation.

Table 8: Calibration performance of different losses with our model (ResNet model as in Table 2 with dropout) and training settings.

Dataset	**CE**		**FL**		**NLL+MbLS**		**FL+MDCA**	
	SCE	ECE	SCE	ECE	SCE	ECE	SCE	ECE
CIFAR10	6.43	2.80	3.59	**0.59**	4.62	1.35	**3.12**	0.86
CIFAR100	2.01	4.00	**1.87**	**0.79**	2.12	0.85	2.00	1.04
SVHN	1.99	**0.27**	7.30	3.30	1.92	0.34	**1.86**	0.38

5 Conclusion

We propose a new train-time calibration method which is based on a novel auxiliary loss term (MACC). Our loss attempts to align the predictive mean confidence with the predictive certainty and is based on the observation that a greater gap between the two translates to higher miscalibration. It is differentiable, operates on minibatches, and acts as a regularizer with other task-specific losses. Extensive experiments on ten challenging datasets show that our loss consistently shows improved calibration performance over the SOTA calibration methods across in-domain and out-of-domain scenarios.

References

1. Bernardo, J.M., Smith, A.F.: Bayesian Theory, vol. 405. John Wiley & Sons (2009)
2. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: International Conference on Machine Learning. pp. 1613–1622 (2015)
3. Brier, G.W., et al.: Verification of forecasts expressed in terms of probability. Monthly Weather Review **78**(1), 1–3 (1950)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
5. DeVries, T., Taylor, G.W.: Learning confidence for out-of-distribution detection in neural networks. arXiv preprint arXiv:1802.04865 (2018)
6. Ding, Z., Han, X., Liu, P., Niethammer, M.: Local temperature scaling for probability calibration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6889–6899 (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Dusenberry, M.W., Tran, D., Choi, E., Kemp, J., Nixon, J., Jerfel, G., Heller, K., Dai, A.M.: Analyzing the role of model uncertainty for electronic health records. In: Proceedings of the ACM Conference on Health, Inference, and Learning. pp. 204–213 (2020)
9. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. pp. 1050–1059 (2016)
10. Gretton, A.: Introduction to rkhs, and some simple kernel algorithms. Adv. Top. Mach. Learn. Lecture Conducted from University College London **16**, 5–3 (2013)
11. Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. Journal of Field Robotics **37**(3), 362–386 (2020)
12. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. pp. 1321–1330 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
14. Hebbalaguppe, R., Prakash, J., Madan, N., Arora, C.: A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16081–16090 (2022)
15. Hein, M., Andriushchenko, M., Bitterwolf, J.: Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 41–50 (2019)

16. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019)
17. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
18. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. In: International Conference on Learning Representations (2018)
19. Hernández-Lobato, J.M., Adams, R.: Probabilistic backpropagation for scalable learning of bayesian neural networks. In: International Conference on Machine Learning. pp. 1861–1869 (2015)
20. Jiang, X., Osl, M., Kim, J., Ohno-Machado, L.: Calibrating predictive model estimates to support personalized medicine. Journal of the American Medical Informatics Association **19**(2), 263–274 (2012)
21. Kermany, D., Zhang, K., Goldbaum, M., et al.: Labeled optical coherence tomography (oct) and chest x-ray images for classification. Mendeley data **2**(2), 651 (2018)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Master Thesis (2009)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
24. Kull, M., Silva Filho, T., Flach, P.: Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In: Artificial Intelligence and Statistics. pp. 623–631 (2017)
25. Kumar, A., Sarawagi, S., Jain, U.: Trainable calibration measures for neural networks from kernel mean embeddings. In: International Conference on Machine Learning. pp. 2805–2814 (2018)
26. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
27. Lang, K.: Newsweeder: Learning to filter netnews. In: Machine learning proceedings 1995, pp. 331–339. Elsevier (1995)
28. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5542–5550 (2017)
29. Liang, G., Zhang, Y., Wang, X., Jacobs, N.: Improved trainable calibration method for neural networks on medical imaging classification. arXiv preprint arXiv:2009.04057 (2020)
30. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2980–2988 (2017)
31. Liu, B., Ben Ayed, I., Galdran, A., Dolz, J.: The devil is in the margin: Margin-based label smoothing for network calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 80–88 (2022)
32. Louizos, C., Welling, M.: Structured and efficient variational deep learning with matrix gaussian posteriors. In: International Conference on Machine Learning. pp. 1708–1716 (2016)
33. Ma, X., Blaschko, M.B.: Meta-cal: Well-controlled post-hoc calibration by ranking. In: International Conference on Machine Learning. pp. 7235–7245 (2021)
34. Meronen, L., Irwanto, C., Solin, A.: Stationary activations for uncertainty calibration in deep learning. In: Advances in Neural Information Processing Systems. vol. 33, pp. 2338–2350 (2020)
35. Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., Dokania, P.: Calibrating deep neural networks using focal loss. In: Advances in Neural Information Processing Systems. vol. 33, pp. 15288–15299 (2020)

36. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Advances in Neural Information Processing Systems. vol. 32 (2019)
37. Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
38. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning (2011)
39. Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D.: Measuring calibration in deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. vol. 2 (2019)
40. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
41. Padhy, S., Nado, Z., Ren, J., Liu, J., Snoek, J., Lakshminarayanan, B.: Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. arXiv preprint arXiv:2007.05134 (2020)
42. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)
43. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
44. Sharma, M., Saha, O., Sriraman, A., Hebbalaguppe, R., Vig, L., Karande, S.: Crowdsourcing for chromosome segmentation and deep classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 34–41 (2017)
45. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
46. Tomani, C., Gruber, S., Erdem, M.E., Cremers, D., Buettner, F.: Post-hoc uncertainty calibration for domain drift scenarios. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10124–10132 (2021)
47. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
48. Wenzel, F., Snoek, J., Tran, D., Jenatton, R.: Hyperparameter ensembles for robustness and uncertainty quantification. In: Advances in Neural Information Processing Systems. vol. 33, pp. 6514–6527 (2020)
49. Yu, R., Alì, G.S.: What's inside the black box? AI challenges for lawyers and researchers. Legal Information Management **19**(1), 2–13 (2019)
50. Zhang, J., Kailkhura, B., Han, T.Y.J.: Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In: International Conference on Machine Learning. pp. 11117–11128 (2020)
51. Zhang, Z., Dalca, A.V., Sabuncu, M.R.: Confidence calibration for convolutional neural networks using structured dropout. arXiv preprint arXiv:1906.09551 (2019)