A Stitch in Time Saves Nine: A Train-Time Regularizing Loss for Improved Neural Network Calibration

Ramya Hebbalaguppe[1,2§] Jatin Prakash[1§] Neelabh Madan[1§] Chetan Arora[1]
[1]Indian Institute of Technology Delhi, India [2]TCS Research, India

https://github.com/mdca-loss

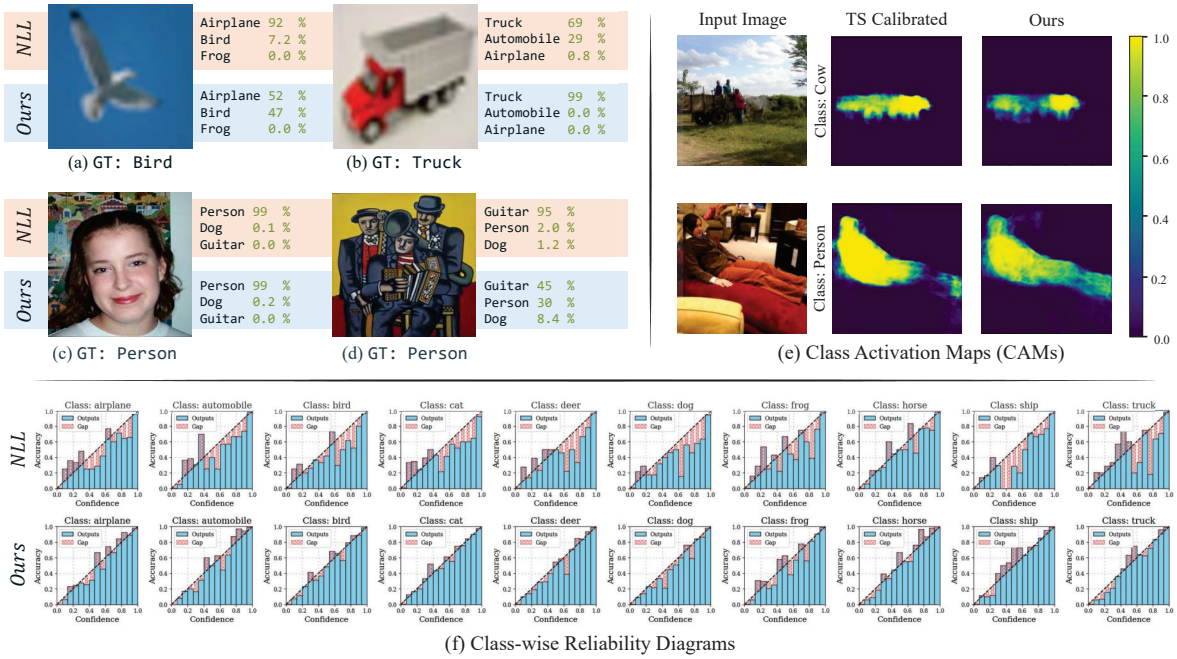


Figure 1. We present a new regularizing loss (MDCA) for train time calibration of Deep Neural Networks (DNN). Figures (a)-(d) shows comparison with a model trained using Cross Entropy loss (NLL), and ours (FL + MDCA). In (a), a DNN trained using NLL makes an incorrect but over-confident prediction. Whereas, training with MDCA reduces the confidence of the mis-predicted label, and increases confidence of the second-highest confident, but correct label. In (b) for a CIFAR 10 minority class, "truck", a model trained with MDCA confidently predicts the correct label as compared to a NLL trained model. In (c) and (d), we show an image from in-domain and out-of-domain dataset. In (c), picture is taken from "Photo" domain (in-domain) of the PACS [30] dataset on which we trained the DNN. Both the models trained with our method as well as NLL predict high confidence score for the correct label. However, in (d), when we change the domain to "Art" (out-of-domain), we see NLL trained model makes highly over-confident mistake on domain shift, whereas, MDCA regularized model remains calibrated. In (e) we show the Class Activation Maps (CAMs) for a model calibrated with Temperature Scaling (TS), and ours for label cow (top row), and person (bottom row). More accurate CAMs show that training with MDCA improves model explainability. (f) shows class-wise reliability diagrams of models trained with NLL and our method. Latter leads to models which are calibrated for all classes.

Abstract

*Deep Neural Networks (DNNs) are known to make over-confident mistakes, which makes their use problematic in safety-critical applications. State-of-the-art (SOTA) calibration techniques improve on the confidence of predicted labels alone, and leave the confidence of non-max classes (e.g. top-2, top-5) uncalibrated. Such calibration is not suitable for label refinement using post-processing. Further, most SOTA techniques learn a few hyper-parameters post-hoc, leaving out the scope for image, or pixel specific calibration. This makes them unsuitable for calibration under domain shift, or for dense prediction tasks like semantic segmentation. In this paper, we argue for intervening at the train time itself, so as to directly produce calibrated DNN models. We propose a novel auxiliary loss function: **M**ulti-class **D**ifference in **C**onfidence and **A**ccuracy (MDCA), to achieve the same.*

§Equal contribution

MDCA *can be used in conjunction with other application/task specific loss functions. We show that training with* MDCA *leads to better calibrated models in terms of Expected Calibration Error (*ECE*), and Static Calibration Error (*SCE*) on image classification, and segmentation tasks. We report* ECE *(*SCE*) score of 0.72 (1.60) on the* CIFAR *100 dataset, in comparison to 1.90 (1.71) by the* SOTA*. Under domain shift, a ResNet-18 model trained on* PACS *dataset using* MDCA *gives a average* ECE *(*SCE*) score of 19.7 (9.7) across all domains, compared to 24.2 (11.8) by the* SOTA*. For segmentation task, we report a* $2\times$ *reduction in calibration error on* PASCAL-VOC *dataset in comparison to Focal Loss [32]. Finally,* MDCA *training improves calibration even on imbalanced data, and for natural language classification tasks.*

1. Introduction

Deep Neural Networks (DNNs) have shown promising results for various pattern recognition tasks in recent years. In a classification setting, with input $\mathbf{x} \in \mathcal{X}$, and label $y \in \mathcal{Y} = \{1, \ldots, K\}$, a DNN typically outputs a *confidence* score vector $\mathbf{s} \in \mathbb{R}^K$. The vector, $\mathbf{s}$, is also a valid probability vector, and each element of $\mathbf{s}$ is assumed to be the predicted confidence for the corresponding label. It has been shown in recent years that the confidence vector, $\mathbf{s}$, output by a DNN is often poorly calibrated [14, 36]. That is:

$$\mathbb{P}\Big(\widehat{y} = y^* \;\Big|\; \mathbf{s}[\widehat{y}]\Big) \;\neq\; \mathbf{s}[\widehat{y}], \tag{1}$$

where $\widehat{y}$, and y^* are the predicted, and true label respectively for a sample. E.g. if a DNN predicts a class "truck" for an image with score 0.7, then a network is calibrated, if the probability that the image actually contains a truck is 0.7. If the probability is lower, a network is said to be over-confident, and under-confident if probability is higher. For a pixel-wise prediction task like semantic segmentation, we would like to calibrate prediction for each pixel. Similarly, we would like calibration to hold not only for the predicted label, i.e. $\widehat{y} = \arg\max_{y \in \mathcal{Y}} \mathbf{s}[y]$, but for the whole vector $\mathbf{s}$ (all labels), i.e., $\forall y \in \mathcal{Y}$.

One of the main reasons for the miscalibration is the specific training regimen used. Most modern DNNs, when trained for classification in a supervised learning setting, are trained using one-hot encoding that have all the probability mass centered in one class; the training labels are thus zero-entropy signals that admit no uncertainty about the input [48]. The DNN is thus trained to become over-confident. Besides creating a general distrust in the model predictions, the miscalibration is especially problematic in safety critical applications, such as self-driving cars [13], legal research [51] and healthcare [10, 46], where giving the correct confidence for a predicted label is as important as the correct label prediction itself.

Researchers have tried to address miscalibration by learning a post-hoc transformation of the output vector so that the confidence of the predicted label matches with the likelihood of the label for the sample [15, 17]. Since such techniques focus on the predicted label only, they could end up calibrating only the label which has maximum confidence for each sample. Hence, in a multi-class setting, the labels with non-maximal confidence scores remain uncalibrated. This makes any post-processing for label refinement, such as posterior inference using MRF-MAP [4], ineffective.

In this paper we argue for the calibration at the train-time. Unlike post-hoc calibration techniques that use limited parameters[1], a train time strategy allows exploiting millions of learnable parameters of DNN itself, thus providing a flexible learning more suited to image and pixel specific transformation for model calibration. Our experiments under domain shift, and for a dense predict task (semantic segmentation) shows the strength of the approach.

Armed with the above insight, we propose a novel auxiliary loss function: **M**ulti-class **D**ifference in **C**onfidence and **A**ccuracy (MDCA). The proposed loss function is designed to be used during the training stage in conjunction with other application specific loss functions, and overcomes the non-differentiablity of the loss functions proposed in earlier methods. Though we do not advocate it, the proposed technique is complimentary to the post-hoc techniques which may still be used after the training, if there is a separate hold-out dataset available for exploitation. Since ours is a train time calibration approach, it implies good regularization for the predictions. We show that models trained using our loss function remain calibrated even under domain shift.

Contributions: We make the following key contributions: **(1)** A trainable DNN calibration method with inclusion of a novel auxiliary loss function, termed MDCA, that takes into account the entire confidence vector in a multi-class setting. Our loss function is differentiable and can be used in conjunction with any existing loss term. We show experiments with Cross-Entropy, Label Smoothing [38], and Focal Loss [32]. **(2)** Our approach is on par with post-hoc methods [14, 23] without the need for hold-out set making the deployment more practical (See Tab. 6). **(3)** Our loss function is a powerful regularizer, maintaining calibration even under domain/dataset drift and dataset imbalance which We demonstrate on PACS [30], Rotated MNIST [29] and imbalanced CIFAR 10 datasets. **(4)** Although the focus is primarily on image classification, our experiments on multi-class semantic segmentation show that our technique outperforms TS based calibration, and Focal Loss [32]. We also show the effectiveness of our approach on natural language classification task on 20Newsgroup dataset [27].

[1]For example, Temperature scaling (TS) calibrates uses a single global scalar, T; and Dirichlet Calibration (DC) uses $\mathcal{O}(K^2)$ hyper-parameters for K classes to calibrate the model output

2. Related Work

Techniques for calibrating DNNs can be broadly classified into train-time calibration, post-hoc calibration, and calibration through Out-Of-Distribution (OOD). Train-time calibration integrate model calibration during the training procedure while a post-hoc calibration method utilizes a hold-out set to tune the calibration measures. On the other hand,learning to reject OOD samples (at train-time or post-hoc) mitigates overconfidence and thus, calibrates DNNs.

Train-Time Calibration: One of the earliest train-time methods proposes Brier Score for the calibrating binary probabilistic forecast [2]. [14] show models trained with Negative-Log-Likelihood (NLL) tend to be over-confident and empirically show a disconnect between NLL and accuracy. Specifically, the overconfident scores necessitates re-calibration. A common calibration approach is to use additional loss terms other than the NLL loss: [44] use entropy as a regularization term whereas Müller et al. [38] propose Label Smoothing (LS) [47] on soft-targets which aids in improving calibration. Recently, [37] showed that focal loss [32] can implicitly calibrate DNNs by reducing the KL-divergence between predicted and target distribution whilst increasing the entropy of the predicted distribution, thereby preventing the model from becoming overconfident. Liang et al. [31] have proposed an auxiliary loss term, DCA, which is added with Cross-Entropy to help calibrate the model. The DCA term penalizes the model when the cross-entropy loss is reduced, but the accuracy remains the same, i.e., when the over-fitting occurs. [26] propose to use MMCE, an auxiliary loss term for calibration, computed using a reproducing kernel in a Hilbert space [12]. Maroñas et al. [33] analyse MixUp [52] data augmentation for calibrating DNNs and conclude Mixup does not necessarily improve calibration.

Post-Hoc Calibration: Post-hoc calibration techniques calibrate a model using a hold-out training set, which is usually the validation set. Temperature scaling (TS) smoothes the logits to calibrate a DNN. Specifically, TS is a variant of Platt scaling [45] that works by dividing the logits by a scalar $T > 0$, learnt on a hold-out training set, prior to taking a softmax. The downside of using TS during calibration is reduction in confidence of every prediction, including the correct one. A more general version of TS transforms the logits using a matrix scaling. The matrix M is learnt using the hold-out set similar to TS. Dirichlet calibration (DC) uses Dirichlet distributions to extend the Beta-calibration [24] method for binary classification to a multi-class one. DC is easy to implement as an extra layer in a neural network on log-transformed class probabilities, which is learnt on a hold-out set. Meta-calibration propose differentiable ECE-driven calibration to obtain well-calibrated and high-accuracy models [1]. Islam et al. [18] propose class-distribution-aware TS and LS that can be used as a post-hoc calibration. They use a class-distribution aware vector for TS/LS to fix the overconfidence. Ding et al. [9] propose a spatially localized calibration approach for semantic segmentation.

Calibration Through OOD Detection: Hein et al. [34] show that one of the main reasons behind the overconfidence in DNNs is the usage of ReLu activation that gives high confidence predictions when the input sample is far away from the training data. They propose data augmentation using adversarial training, which enforces low confidence predictions for samples far away from the training data. Guo et al. [14] analyze the effect of width, and depth of a DNN, batch normalization, and weight decay on the calibration. Karimi et al. [19] use spectral analysis on initial layers of a CNN to determine OOD sample and calibrate the DNN. We refer the reader to [8, 16, 35, 43] for other representative works on calibrating a DNN through OOD detection.

3. Proposed Methodology

Calibration: A calibrated classifier outputs confidence scores that matches the empirical frequency of correctness. If a calibrated model predicts an event with 0.7 confidence, then 70% of the times the event transpires. If the empirical occurrence of the event is $< 70\%$ then the model is overconfident, and if the empirical probability $> 70\%$ then the model is under-confident. Formally, we define calibration in a classical supervised setting as follows. Let $\mathcal{D} = \langle (x_i, y_i) \rangle_{i=1}^{N}$ denote a dataset consisting of N samples from a joint distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$, where for each sample $x_i \in \mathcal{X}$ is the input and $y_i^* \in \mathcal{Y} = \{1, 2, ..., K\}$ is the ground-truth class label. Let $\mathbf{s} \in \mathbb{R}^K$, and $\mathbf{s}_i[y] = f_\theta(x_i)$ be the confidence that a DNN, f, with model parameters θ predicts for a class y on a given input x_i. The class, $\widehat{y}_i$, predicted by f for a sample x_i is computed as:

$$\widehat{y}_i = \arg\max_{y \in \mathcal{Y}} \mathbf{s}_i[y]. \tag{2}$$

The confidence for the predicted class is correspondingly computed as $\widehat{s}_i = \max_{y \in \mathcal{Y}} s_i[y]$. A model is said to be *perfectly calibrated* [14] when, for each sample $(x, y) \in \mathcal{D}$:

$$\mathbb{P}(y = y^* \mid \mathbf{s}[y] = s) = s. \tag{3}$$

Note that the perfect calibration requires each score value (and not only the $\widehat{s}$) to be calibrated. On the other hand, most calibration techniques focus only on the predicted class. That is, they only ensure that: $\mathbb{P}(\widehat{y}_i = y_i^* \mid \widehat{s}_i) = \widehat{s}_i$.

Expected Calibration Error (ECE): ECE is calculated by computing a weighted average of the differences in the confidence of the predicted class, and the accuracy of the samples, predicted with a particular confidence score [39]:

$$\text{ECE} = \sum_{i=1}^{M} \frac{B_i}{N} \Big| A_i - C_i \Big|. \tag{4}$$

Here N is the total number of samples, and the weighting is done on the basis of the fraction of samples in a given confidence bin/interval. Since the confidence values are in a continuous interval, for the computation of ECE, we divide the confidence range $[0, 1]$ into M equidistant bins, where i^{th} bin is the interval $(\frac{i-1}{M}, \frac{i}{M}]$ in the confidence range, and B_i, represents the number of samples in the i^{th} bin. Further, $A_i = \frac{1}{|B_i|}\sum_{j\in B_i}\mathbb{I}(\hat{y}_j = y_j)$, denotes accuracy for the samples in bin B_i, and $C_i = \frac{1}{|B_i|}\sum_{j:\widehat{s}_j\in B_i}\widehat{s}_j$, is the average predicted confidence of the samples, such that $\widehat{s}_j \in B_i$. The evaluation of DNN calibration via ECE suffers from the following shortcomings: (a) ECE does not measure the calibration of all score values in the confidence vector, and (b) the metric is not differentiable, and hence can not be incorporated as a loss term during training procedure itself. Specifically, non-differentiablity arises due to binning samples into bins B_i.

Maximum Calibration Error (MCE): MCE is defined as the maximum absolute difference between the average accuracy and average confidence of each bin:

$$\text{MCE} = \max_{i\in 1,\ldots,M}\big|A_i - C_i\big|.$$

The *max* operator ends up pruning a lot of useful information about calibration, making the metric not-so-popular. However, it does represent a statistical value that can be used to discriminate large differences in calibration.

Static Calibration Error (SCE): SCE is a recently proposed metric to measure calibration by [41]:

$$\text{SCE} = \frac{1}{K}\sum_{i=1}^{M}\sum_{j=1}^{K}\frac{B_{i,j}}{N}\big|A_{i,j} - C_{i,j}\big|, \tag{5}$$

where, K denotes the number of classes, and $B_{i,j}$ denotes number of samples of the j^{th} class in the i^{th} bin. Further, $A_{i,j} = \frac{1}{B_{i,j}}\sum_{k\in B_{i,j}}\mathbb{I}(j = y_k)$ is the accuracy for the samples of j^{th} class in the i^{th} bin, and $C_{i,j} = \frac{1}{B_{i,j}}\sum_{k\in B_{i,j}}\mathbf{s}_k[j]$ or average confidence for the j^{th} class in the i^{th} bin. **Classwise-ECE** [23] is another metric for measuring calibration in a multi-class setting, but is identical to Static Calibration Error (SCE). It is easy to see that SCE is a simple class-wise extension to ECE. Since SCE takes into account the whole confidence vector, it allows us to measure calibration of the non-predicted classes as well. Note that, similar to ECE, the metric SCE is also non-differentiable, and can not be used as a loss term during training.

Class-j-ECE: [23] has proposed to evaluate calibration error of each class independent of other classes. This allows one to capture the contribution of a single class j to the overall SCE (or classwise-ECE) error. We refer to this metric as class-j-ECE in our results/discussion.

3.1. Proposed Auxiliary loss: MDCA

We propose a novel multi-class calibration technique using the proposed auxiliary loss function. The loss function is inspired from SCE [41] but avoids the non-differentiability caused due to binning $B_{i,j}$ as shown in Eq. (5) [31]. Our calibration technique is **independent** of the binning scheme/bins. This is important, because as [50] and [25] have also highlighted, binning scheme leads to underestimated calibration errors. We name our loss function, *Multi-class Difference of Confidence and Accuracy (*MDCA*)*, and apply it for each **mini-batch** during training. The loss is defined as follows:

$$\mathcal{L}_{\text{MDCA}} = \frac{1}{K}\sum_{j=1}^{K}\Big|\frac{1}{N_b}\sum_{i=1}^{M}\mathbf{s}_i[j] - \frac{1}{N_b}\sum_{i=1}^{M}q_i[j]\Big|, \tag{6}$$

where $q_i[j] = 1$ if label j is the ground truth label for sample i, i.e. $j = y_i^*$, else $q_i[j] = 0$. Note the second term inside $|\cdot|$ corresponds to average count of samples in a mini-batch containing N_b training samples. Since the average count is a constant value so learning gradients solely depends on the first term representing confidence assigned by the DNN. K denotes number of classes. $\mathcal{L}_{\text{MDCA}}$ is computed on a mini-batch, and the modulus operation ($|\cdot|$) implies that the summations are not interchangeable [2]. Further, $\mathbf{s}_i[j]$ represents the confidence score by a DNN for the j^{th} class, of i^{th} sample in the mini-batch.

Note that $\mathcal{L}_{\text{MDCA}}$ is differentiable, whereas, the loss given by DCA [31] involves accuracy over the mini-batch, and is non-differentiable. The differentiablity of our loss function ensures that it can be easily used in conjunction with other application specific loss functions as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_C + \beta\cdot\mathcal{L}_{\text{MDCA}}, \tag{7}$$

where β is a hyperparameter to control the relative importance with respect to application specific losses, and is typically found using a validation set. $\mathcal{L}_C$ is a standard classification loss, such as Cross Entropy, Label Smoothing [47], or Focal loss [32]. Our experiments indicate that the proposed MDCA loss in conjunction with focal loss gives best calibration performance.

Ideally to achieve *confidence calibration*, we want the average prediction confidence to be same as accuracy of the model. However, in *multiclass calibration*, we want average prediction confidence of every class k_i to match with its average occurrence in the data-distribution. In $\mathcal{L}_{MDCA}$, we explicitly capture this idea for every mini-batch i.e. we

[2]Note that $\mathcal{L}_{\text{MDCA}}$ may appear similar to $\mathcal{L}_1$ loss due to the usage of the modulus in both. However, the two loss functions are very different. Mathematically, $\mathcal{L}_1 = \frac{1}{K\cdot N_b}\sum_{j=1}^{K}\sum_{i=1}^{N_b}\Big|\mathbf{s}_i[j] - q_i[j]\Big|$ whereas $\mathcal{L}_{\text{MDCA}}$ is as given in Eq. (6). The two terms inside the modulus of $\mathcal{L}_{\text{MDCA}}$ loss represent mean statistic for a particular class, j (motivated by our objective of class-wise calibration), whereas, in the case of $\mathcal{L}_1$ the modulus operate on a single sample.

intuitively want that $\tilde{s}[k_i] \approx \tilde{q}[k_i]$ (where $\tilde{s}[k_i], \tilde{q}[k_i]$ is the average prediction confidence and the average count class k_i in a mini-batch respectively). Any deviation from this leads DNN to be penalized by $\mathcal{L}_{MDCA}$.

4. Dataset and Evaluation

Datasets: We validate our technique on well-known benchmark datasets for image classification, semantic segmentation and natural language processing (NLP). For each of the datasets: CIFAR10/100 [22], SVHN [40], Mendeley V2 [20], Tiny-ImageNet [7] and 20-Newsgroups [28], we have a separate train and test set. The train set is further split into 2 mutually exclusive sets (a) training set containing 90% of the samples, and (b) the validation set containing 10%. We use validation set as the hold-out set for post-hoc calibration. This division has been consistent throughout our experimentation. See Supplementary material for detailed description of datasets, DNN architectures, and training procedure.

Evaluation: We report calibration measures, SCE, ECE, and class-j-ECE along with test error for studying calibration performance. We observe that we achieve superior calibration using our technique without any significant drop in the accuracy. We also visualize the calibration using reliability diagrams (please see supplementary material for detailed description of reliability diagrams).

Compared Techniques: We compare our method against models trained with Cross-Entropy (NLL), Label Smoothing (LS) [47], DCA [31], Focal Loss (FL) [32], Brier Score (BS) [2], FLSD [37] as well as MMCE [26]. For details on individual methods and their training specifics, please refer to the supplementary.

5. Results

Experiments with Application Specific Loss Functions: Our loss is meant to be used in conjunction with another application specific loss function to help improve the calibration performance of a model. Common application specific loss include cross entropy loss (NLL) which in turn minimizes negative log likelihood score of the ground truth label in the predicted confidence vector. Focal Loss (FL) [32] has been proposed to improve training in the presence of many easy negatives, and fewer hard negatives. Whereas Label Smoothing (LS) [47] introduces another term in the NLL to smoothen the prediction of a model. We add the proposed MDCA with each of these loss terms, and measure the calibration performance of a model (in terms of ECE, and SCE scores), before and after adding our loss. Tab. 1 shows the result. We refer to configurations using our technique as **"*+MDCA"**, where * refers to NLL/LS/FL. For each of the combination we use relative weight of $\beta \in \{1, 5, 10, 15, 20, 25\}$ for $\mathcal{L}_{\text{MDCA}}$, and report the calibration performance of the most accurate model on the validation set. Our experiments suggest that setting $\beta < 1$ did not have strong regularizing effect). For $\mathcal{L}_{\text{LS}}$ we use $\alpha = 0.1$, and for $\mathcal{L}_{\text{FL}}$ we use $\gamma \in \{1, 2, 3\}$. Please refer to [47] and [32] for interpretation of α, and γ respectively. Tab. 1 shows that the proposed MDCA loss improves calibration performance of all the above application specific loss functions, across multiple datasets, and architectures. We also note that FL+MDCA gives best calibration performance. We will use this loss configuration in our experiments hereafter.

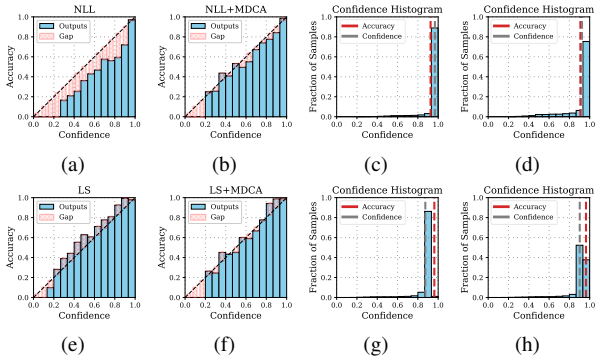


Figure 2. First row shows Reliability diagrams (a,b) and confidence histograms (c,d) of NLL trained model compared against MDCA regularized version (NLL+MDCA). We use ResNet-32 trained on CIFAR10 dataset for comparison. Second row shows corresponding plots for ResNet-20 network trained with Label Smoothing(LS) vs. MDCA regularized LS on SVHN dataset. Please refer to the text for the interpretation of the plots. We show a similar comparison with FL, and FL+DCA in the supplementary.

Calibration Comparison with SOTA: Tab. 2 compares calibration performance of our method with all recent SOTA methods. We note that calibration using our method improves both SCE as well as ECE score on all the datasets, and different architectures.

Class-Conditioned Calibration Error: Current state-of-the-art focuses on calibrating the predicted label only, which leaves some of the minority class un-calibrated. One of the benefits of our calibration approach is better calibration for all and not only the predicted class. To demonstrate the effectiveness of our method, we report class-j-ECE % values of all the competing methods against our method, using ResNet-20 model trained on the SVHN dataset. Tab. 3 shows the result. Our method gives best scores for all but 3 out of 10 classes, where it is second-best. Class-wise reliability diagrams (c.f. Fig. 1) reinforce a similar conclusion. We show results on CIFAR 10 dataset in the supplementary.

Test Error: Tab. 2 also shows the Test Error (TE) obtained by a model trained using our method and other SOTA approaches. We note that using our proposed loss, a model is able to achieve best calibration performance without sacrificing on the prediction accuracy (Test Error).

Dataset	Model	NLL		NLL+MDCA		LS [38]		LS+MDCA		FL [32]		FL+MDCA	
		SCE(10^{-3})	ECE (%)	SCE(10^{-3})	ECE (%)	SCE(10^{-3})	ECE (%)	SCE(10^{-3})	ECE (%)	SCE(10^{-3})	ECE (%)	SCE(10^{-3})	ECE (%)
CIFAR10	ResNet32	8.68	4.25	4.63	1.69	14.08	6.28	10.39	4.31	4.60	1.76	**3.22**	**0.93**
	ResNet56	7.11	3.27	6.87	3.15	12.54	5.38	9.88	3.97	4.18	1.11	**2.93**	**0.70**
CIFAR100	ResNet32	3.03	12.45	2.59	9.94	1.99	2.09	1.74	2.29	1.83	1.62	**1.72**	**1.49**
	ResNet56	2.50	9.32	2.41	8.95	1.73	8.94	1.68	1.48	1.66	2.29	**1.60**	**0.72**
SVHN	ResNet20	3.43	1.64	**1.46**	**0.43**	18.80	8.88	13.91	6.46	2.54	0.89	1.90	0.47
	ResNet56	3.84	1.82	**1.47**	0.53	21.08	10.00	17.62	8.43	7.85	3.89	1.51	**0.23**
Mendeley V2	ResNet50	131.2	4.78	88.14	3.63	103.8	**2.68**	97.38	5.03	108.3	8.17	**85.68**	4.81
Tiny-ImageNet	ResNet34	1.91	14.91	1.87	14.22	1.38	5.96	1.36	5.90	1.19	2.26	**1.17**	**1.99**
20 Newsgroups	Global-Pool CNN	602.68	14.78	559.50	16.53	988.42	**3.45**	520.50	17.30	729.39	13.35	**487.82**	16.55

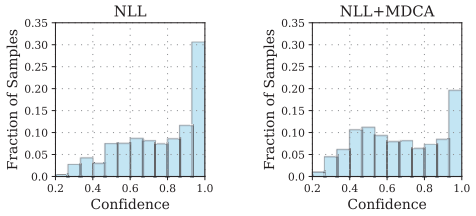
Table 1. Our loss is meant to be used in addition to another application specific loss. The table compares the calibration performance improvement using MDCA over three commonly used loss functions (NLL/LS/FL). Our loss improves calibration performance across multiple datasets and architectures.

Dataset	Model	BS [2]			DCA [31]			MMCE [26]			FLSD [37]			Ours (FL+MDCA)		
		SCE	ECE	TE	SCE	ECE	TE	SCE	ECE	TE	SCE	ECE	TE	SCE	ECE	TE
CIFAR10	ResNet32	6.60	2.92	7.76	8.41	4.00	**7.06**	8.17	3.31	8.41	9.48	4.41	7.87	**3.22**	**0.93**	7.18
	ResNet56	5.44	2.17	7.75	7.59	3.38	**6.53**	9.11	3.71	8.23	7.71	3.49	7.04	**2.93**	**0.70**	7.08
CIFAR100	ResNet32	1.97	5.32	33.53	2.82	11.31	29.67	2.79	11.09	31.62	1.77	1.69	32.15	**1.72**	**1.49**	**31.58**
	ResNet56	1.86	4.69	30.72	2.77	9.29	43.43	2.35	8.61	**28.75**	1.71	1.90	29.11	**1.60**	**0.72**	29.8
SVHN	ResNet20	2.12	**0.45**	**3.56**	4.29	2.02	3.83	9.18	4.34	4.12	18.98	9.37	4.10	**1.90**	0.47	3.92
	ResNet56	2.18	0.66	**3.25**	2.16	0.49	3.32	9.69	4.48	4.26	26.15	13.23	3.65	**1.51**	**0.23**	3.85
Mendeley V2	ResNet50	117.6	3.75	18.43	145.1	8.29	17.47	130.4	**3.45**	**15.06**	104.3	9.64	19.71	**85.68**	4.81	17.95
Tiny-ImageNet	ResNet34	1.53	7.79	43.00	2.11	17.40	**36.68**	1.62	9.71	40.75	1.18	**1.91**	37.01	**1.17**	1.99	37.49
20 Newsgroups	Global-Pool CNN	725.82	13.71	**25.93**	719.83	15.30	28.07	731.31	12.69	28.63	940.70	**4.52**	30.80	**487.82**	16.55	27.88

Table 2. Calibration measures SCE (10^{-3}) and ECE (%) score) and Test Error (TE) (%) in comparison with various competing methods. We use $M = 15$ bins for SCE and ECE calculation. We outperform all the baselines across various popular benchmark datasets, and architectures in terms of calibration, while maintaining a similar accuracy.

Method	Classes									
	0	1	2	3	4	5	6	7	8	9
Cross Entropy	**0.20**	0.62	0.33	0.65	0.23	0.36	0.25	0.26	**0.21**	0.41
Focal Loss [32]	0.30	0.48	0.41	**0.18**	0.38	0.19	0.33	0.36	0.32	0.30
LS [38]	1.63	2.60	2.54	1.90	1.91	1.74	1.73	1.75	1.63	1.58
Brier Score [2]	0.23	0.28	0.40	0.45	0.25	0.26	0.25	0.27	0.21	0.37
MMCE [26]	1.78	2.35	2.12	2.00	1.74	1.87	1.65	1.76	1.70	1.84
DCA [31]	0.31	0.70	0.40	0.72	0.31	0.46	0.35	0.35	0.37	0.36
FLSD [37]	1.52	3.24	2.74	2.15	1.79	1.82	1.84	1.62	1.54	1.38
Ours (FL+MDCA)	0.22	**0.16**	**0.24**	0.25	**0.22**	**0.16**	**0.16**	**0.17**	0.25	**0.20**

Table 3. Class-j-ECE (%) score on all 10 classes for ResNet-20 model trained on the SVHN dataset with different learnable calibration methods. Our method gives best calibration for 7 out of 10 classes, and is second-best on 3 classes.



(a) (b)

Figure 3. Confidence value histogram for misclassified predictions. MDCA regularized NLL makes less confident incorrect predictions as compared to the uncalibrated method trained using NLL.

Mitigating Under/Over-Confidence: Tab. 1 and Tab. 2 already show that our method improves over SOTA in terms of SCE, and ECE scores. However the tables do not highlight whether they correct for over-confidence or under-confidence. We show the reliability diagram (Fig. 2) for a ResNet-32/20 model trained on CIFAR 10/SVHN. The uncalibrated model is overconfident (Fig. 2a) which gets rectified after calibrating with our method (Fig. 2b). We also show confidence plots in the picture, and the colored dashed lines to indicate average confidence of the predicted label, and the accuracy. It can be seen that accuracy is lower than average confidence in the uncalibrated confidence plot (Fig. 2c), which indicates the overconfident model. After calibrating with our method, the two dashed lines almost overlap indicating the perfect calibration achieved (Fig. 2d). Similarly, second row of Fig. 2 show that the model trained with LS solely is under-confident; and a model trained with LS along with MDCA is confident and calibrated.

Confidence Values for Incorrect Predictions: The focus of the discussion so far has been on the fact that confidence value for a class should be consistent with the likelihood of the class for the sample. Here, we analyze our method for the confidence values it gives when the prediction is incorrect. Fig. 3 shows the confidence value histogram for

Method	Art	Cartoon	Sketch	Average
NLL	6.33	17.95	15.01	13.10
LS [38]	7.80	11.95	**10.88**	10.21
FL [32]	8.61	16.62	10.94	12.06
Brier Score [2]	6.55	13.19	15.63	11.79
MMCE [26]	6.35	15.70	17.16	13.07
DCA [31]	7.49	18.01	14.99	13.49
FLSD [37]	8.35	13.39	13.86	11.87
Ours (FL+MDCA)	**6.21**	**11.91**	11.08	**9.73**

Table 4. Calibration performance (SCE (10^{-3})) under domain shift on PACS dataset [30]. For each column we train on the other two subsets, and then test on the subset listed as the column heading.

Method	CIFAR10 IF-10	CIFAR10 IF-50	CIFAR10 IF-100	SVHN IF-2.7
NLL	18.44	32.21	31.04	3.43
FL [32]	14.65	29.67	28.89	2.54
LS [38]	14.88	26.30	**20.79**	18.80
BS [2]	15.74	33.57	29.01	2.12
MMCE [26]	15.10	29.05	21.56	9.18
FLSD [37]	16.05	31.35	30.28	18.98
DCA [31]	18.57	32.81	35.53	4.29
Ours (FL+MDCA)	**11.83**	**22.97**	26.89	**1.90**

Table 5. Our calibration technique works best even when there is a significant class imbalance in the dataset. For this experiment we created imbalance of various degrees in CIFAR 10 as suggested in [6]. Original SVHN has a Imbalance Factor(IF) of 2.7. Hence we show calibration performance (SCE (10^{-3})) on original SVHN.

Method	Post Hoc	SCE(10^{-3}) ↓ CIFAR10	CIFAR100	SVHN
NLL	None	7.12	2.50	3.84
	TS	3.25	1.49	4.16
	DC	4.98	1.91	2.69
LS [38]	None	12.55	1.73	21.08
	TS	4.49	1.67	3.12
	DC	5.34	1.98	2.81
FL [32]	None	4.19	1.89	7.85
	TS	4.19	1.62	2.72
	DC	5.48	2.02	3.36
BS [2]	None	5.44	1.86	2.18
	TS	3.94	1.68	3.88
	DC	4.83	1.80	2.11
MMCE [26]	None	9.12	2.35	9.69
	TS	4.05	1.61	3.74
	DC	6.26	1.95	5.11
DCA [31]	None	7.60	2.87	2.16
	TS	3.00	1.56	4.29
	DC	4.20	2.06	2.95
FLSD [37]	None	7.71	1.71	26.15
	TS	3.27	1.71	4.41
	DC	5.62	2.01	4.31
Ours (FL+MDCA)	None	**2.93**	**1.60**	**1.51**
	TS	2.93	1.60	5.00
	DC	3.81	1.87	2.72

Table 6. Results after combining various trainable calibration methods including ours with two post-hoc calibration methods (TS: Temperate scaling [45], and DC: Dirichlet Calibration [23]) on SCE (10^{-3}). We use ResNet56 model on CIFAR 10, CIFAR 100, and SVHN datasets for this experiment. Though other methods benefit by post-hoc calibration, our method outperforms them without using any post-hoc calibration.

all the incorrect predictions made by the ResNet-32 model trained on CIFAR 10 dataset using NLL vs. MDCA regularized NLL. It is clear that our calibration reduces the confidence for the mis-prediction. The same is also evident from the Fig. 1 shown earlier.

Calibration Performance under Dataset Drift: Tomani et al. [49] show that DNNs are over-confident and highly uncalibrated under dataset/domain shift. Our experiments shows that a model trained with MDCA fairs well in terms of calibration performance even under non-semantic/natural domain shift. We use two datasets (a) PACS [30] and (b) Rotated MNIST inspired from [42]. The datasets are benchmarks for synthetic non-semantic shift and natural rotations respectively. Dataset specifics and training procedure are provided in the supplementary. Tab. 4 shows that our method achieves the best average SCE value across all the domains in PACS. A similar trend is observed on Rotated MNIST dataset as well (see supplementary), where our method achieves the least average SCE value across all rotation angles.

Calibration Performance on Imbalanced Datasets: The real-world datasets are often skewed and exhibit long-tail distributions, where a few classes dominate over the rare classes. In order to study the effect of class imbalance on the calibration quality, we conduct the following experiment, where we introduce a deliberate imbalance on CIFAR 10 dataset to force a long-tail distribution as detailed in [6]. Tab. 5 shows that a model trained with our method has best calibration performance in terms of SCE score across all imbalance factors. We observe that SVHN dataset already has a imbalance factor of 2.7, and hence create no artificial imbalance in the dataset for this experiment. The efficacy of our approach on the imbalanced data is due to the regularization provided by MDCA which penalizes the difference between average confidence and average count even for the non-predicted class, hence benefiting minority classes.

Our Approach + Post-hoc Calibration: We study the performance of combined effect of post-hoc calibration methods, namely Temperature Scaling (TS) [45], and Dirichlet Calibration (DC) [23], applied over various train-time calibration methods including ours (FL+MDCA). Tab. 6 shows the results. We observe that while TS, and DC improve the performance of other competitive methods, our method outperforms them even without using any of these methods. On the other hand, the performance of our method seems to either remain same or slightly decrease after application of post-hoc methods. We speculate that this is because our method already calibrates the model to near perfection. For example, on performing TS, we observe the optimal temperature values are $T \approx 1$ implying that it leaves little scope for the TS to improve on top of it. Thus, any further attempt to over-spread the confidence prediction using TS or

Method	Pixel Acc. (%)	mIoU (%)	SCE (10^{-3})	ECE (%)
NLL	94.81	79.49	6.4	7.77
NLL+TS	94.81	79.49	6.26	6.1
FL	92.85	77.22	11.8	7.69
Ours (FL+MDCA)	94.47	78.66	**5.8**	**4.66**

Table 7. Segmentation results on Xception65 [5] backbone DeeplabV3+ model [3] on PASCAL-VOC 2012 validation dataset.

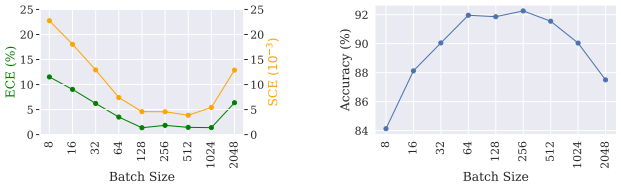


Figure 4. Effect of different batch sizes on Calibration performance metrics (ECE/SCE/Accuracy) while training with MDCA on a ResNet-32 model on CIFAR 10 dataset. The calibration performance drops with larger batch size because SGD optimization is more effective in a small-batch regime [21]. A larger batch results in a degradation in the quality of the model, as measured by its ability to generalize. The performance degradation is also consistent with the model trained using solely on FL on a large batch size.

DC negatively affects the confidence quality.

Calibration Results for Semantic Segmentation: One of the major advantages of our technique is that it allows to use billions of weights of a DNN model to be used for the calibration. This is in contrast to other calibration approaches which are severely constrained in terms of parameters available for tuning. For example in TS one has a single temperature parameter to tune. This makes it hard for TS to provide image and pixel specific confidence transformation for calibration. To highlight pixel specific calibration aspect of our technique we have done experiments on semantic segmentation task, which can be seen as pixel level classification. For the experiment, we train a DeepLabV3+ [3] model with a pre-trained Xception65 [5] backbone on the PASCAL-VOC 2012 [11] dataset. We compare the performance of our method against NLL, FL and TS (post-hoc calibration). Please refer to the supplementary for more details on the training. Tab. 7 shows the results. We see a significant drop in both SCE/ECE in case of our method (FL+MDCA) as compared to FL ($2\times$ drop in SCE and a 40% decrease in ECE). Our method also outperforms TS (after training with NLL) by 23.6%.

6. Ablation Study

Effect of Batch Size: Fig. 4 shows the effect of different batch sizes on the calibration performance. We vary the batch sizes exponentially and observe that a model trained with MDCA achieves best calibration performance around batch size of 64 or 128. As we decrease (or increase) the batch size, we see a degradation in calibration, though the drop is not significant.

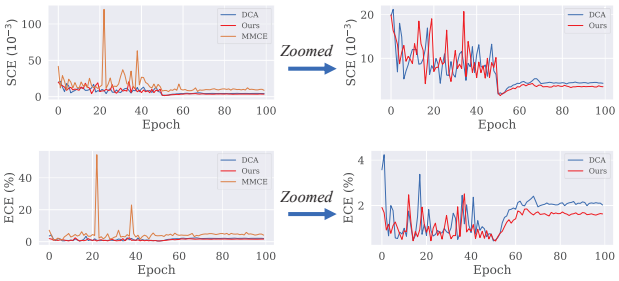


Figure 5. Comparison of ECE/SCE at various epochs for MDCA, MMCE, and DCA. Though, MMCE, and DCA directly optimize for ECE, their loss function is not differentiable and hence the techniques are not able to reduce ECE as much as MDCA. Differentiability of loss function allows MDCA to reduce ECE better even when it does not directly optimize it. We use a learning rate decay of $1/10$ at epochs 50 and 70. Please refer to the supplementary for the details of the experiment.

Comparison of ECE/SCE Convergence with SOTA: In previous sections, we compared the ECE scores of MDCA with other contemporary trainable calibration methods like MMCE [26] and DCA [31]. Many of these methods explicitly aim to reduce ECE scores. While MDCA does not directly optimize ECE, yet we see in our experiments that MDCA manages to get better ECE scores at convergence. We speculate that this is due to the differentiablity of MDCA loss which helps optimize the loss better using backpropagation. To verify the hypothesis, we plot the ECE convergence for various methods in Fig. 5.

7. Conclusion & Future work

We have presented a train-time technique for calibrating the predicted confidence values of a DNN based classifier. Our approach combines standard classification loss functions with our novel auxiliary loss named, Multi-class Difference of Confidence and Accuracy (MDCA). Our proposed loss function when combined with focal loss yields the least calibration error among both trainable and post-hoc calibration methods. We show promising results in case of long tail datasets, natural/synthetic dataset drift, semantic segmentation and a natural language classification benchmark too. In future we would like to investigate the role of class hierarchies to develop cost-sensitive calibration techniques.

8. Acknowledgments

Thanks to Mayank Baranwal and Harshad Khadilkar for helpful discussions and suggestions.

References

[1] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Meta-calibration: Meta-learning of model calibration using differentiable expected calibration error. 2021. iii

[2] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. iii, v, vi, vii

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. viii

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. ii

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. viii

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2019. vii

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. v

[8] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. iii

[9] Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. Local temperature scaling for probability calibration. *CoRR*, abs/2008.05105, 2020. iii

[10] Michael W. Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M. Dai. Analyzing the role of model uncertainty for electronic health records. *Proceedings of the ACM Conference on Health, Inference, and Learning*, Apr 2020. ii

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. viii

[12] Arthur Gretton. Introduction to rkhs, and some simple kernel algorithms. *Adv. Top. Mach. Learn. Lecture Conducted from University College London*, 16:5–3, 2013. iii

[13] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. ii

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. 2019. ii, iii

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016. ii

[16] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2018. iii

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. ii

[18] Mobarakol Islam, Lalithkumar Seenivasan, Hongliang Ren, and Ben Glocker. Class-distribution-aware calibration for long-tailed visual recognition. *ICML Uncertainity in Deep Learning Workshop*, 2021. iii

[19] Davood Karimi and Ali Gholipour. Improving calibration and out-of-distribution detection in medical image segmentation with convolutional neural networks. *CoRR*, abs/2004.06569, 2020. iii

[20] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018. v

[21] N. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017. viii

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. v

[23] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*, 2019. ii, iv, vii

[24] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017. iii

[25] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. *arXiv preprint arXiv:1909.10155*, 2019. iv

[26] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2805–2814. PMLR, 10–15 Jul 2018. iii, v, vi, vii, viii

[27] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995. ii

[28] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995. v

[29] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. ii

[30] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. i, ii, vii

[31] Gongbo Liang, Yu Zhang, Xiaoqin Wang, and Nathan Jacobs. Improved trainable calibration method for neural networks on medical imaging classification. *CoRR*, abs/2009.04057, 2020. iii, iv, v, vi, vii, viii

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. ii, iii, iv, v, vi, vii

[33] Juan Maroñas, Daniel Ramos, and Roberto Paredes. On calibration of mixup training for deep neural networks. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 67–76. Springer, 2021. iii

[34] H. Matthias, A. Maksym, and B. Julian. 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*, pages 41–50, 2019. iii

[35] Lassi Meronen, Christabella Irwanto, and Arno Solin. Stationary activations for uncertainty calibration in deep learning. *arXiv preprint arXiv:2010.09494*, 2020. iii

[36] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *arXiv preprint arXiv:2106.07998*, 2021. ii

[37] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss, 2020. iii, v, vi, vii

[38] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019. ii, iii, vi, vii

[39] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. iii

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. v

[41] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019. iv

[42] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019. vii

[43] Shreyas Padhy, Zachary Nado, Jie Ren, Jeremiah Liu, Jasper Snoek, and Balaji Lakshminarayanan. Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *arXiv preprint arXiv:2007.05134*, 2020. iii

[44] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions, 2017. iii

[45] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. iii, vii

[46] Monika Sharma, Oindrila Saha, Anand Sriraman, Ramya Hebbalaguppe, Lovekesh Vig, and Shirish Karande. Crowdsourcing for chromosome segmentation and deep classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 786–793, 2017. ii

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. iii, iv, v

[48] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019. ii

[49] Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. *CoRR*, abs/2012.10988, 2020. vii

[50] David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *arXiv preprint arXiv:1910.11385*, 2019. iv

[51] Ronald Yu and Gabriele Spina Alì. What's inside the black box? ai challenges for lawyers and researchers. *Legal Information Management*, 19(1):2–13, 2019. ii

[52] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. iii