Soft Calibration Objectives for Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Optimal decision making requires that classifiers produce uncertainty estimates consistent with their empirical accuracy. However, deep neural networks are often under- or over-confident in their predictions. Consequently, methods have been developed to improve the calibration of their predictive uncertainty both during training and post-hoc. In this work, we propose differentiable losses to improve calibration based on a soft (continuous) version of the binning operation underlying popular calibration-error estimators. When incorporated into training, these soft calibration losses achieve state-of-the-art single-model ECE across multiple datasets with less than 1% decrease in accuracy. For instance, we observe an 82% reduction in ECE (70% relative to the post-hoc rescaled ECE) in exchange for a 0.7% relative decrease in accuracy relative to the cross entropy baseline on CIFAR-100. When incorporated post-training, the soft-binning-based calibration error objective improves upon temperature scaling, a popular recalibration method. Overall, experiments across losses and datasets demonstrate that using calibration-sensitive procedures yield better uncertainty estimates under dataset shift than the standard practice of using a cross entropy loss and post-hoc recalibration methods.[1]

1 Introduction

Despite the success of deep neural networks across a variety of domains, they are still susceptible to miscalibrated predictions. Both over- and under-confidence contribute to miscalibration, and deep neural networks empirically exhibit high calibration error [Guo et al., 2017]. Calibration error (*CE*) quantifies a model's miscalibration by measuring how much its confidence, i.e. the predicted probability of correctness, diverges from its accuracy, i.e. the empirical probability of correctness. Models with low CE are critical in domains where satisfactory outcomes depend on well-modeled uncertainty, such as autonomous vehicle navigation [Bojarski et al., 2016] and medical diagnostics [Jiang et al., 2012, Caruana et al., 2015, Kocbek et al., 2020]. Calibration has also been shown to be useful for improving model fairness [Pleiss et al., 2017] and detecting out-of-distribution data [Kuleshov and Ermon, 2017, Devries and Taylor, 2018, Shao et al., 2020]. More generally, low CE is desirable in any setting in which thresholds are applied to the predicted confidence of a neural network in order to make a decision.

Methods for quantifying CE typically involve binning model predictions based on their confidence. CE is then computed empirically as a weighted average of the absolute difference in average prediction confidence and average accuracy across different bins [Naeini et al., 2015]. Oftentimes these bins are selected heuristically such as "equal-width" (uniformly binning the score interval) and "equal-mass" (with equal numbers of samples per-bin) [Nixon et al., 2019].

However, these commonly used measures of CE are not differentiable since the binning operation has a zero derivative within a bin and is discontinuous at bin boundaries. As a result, neural networks do

[1] All code to reproduce results will be open-sourced.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

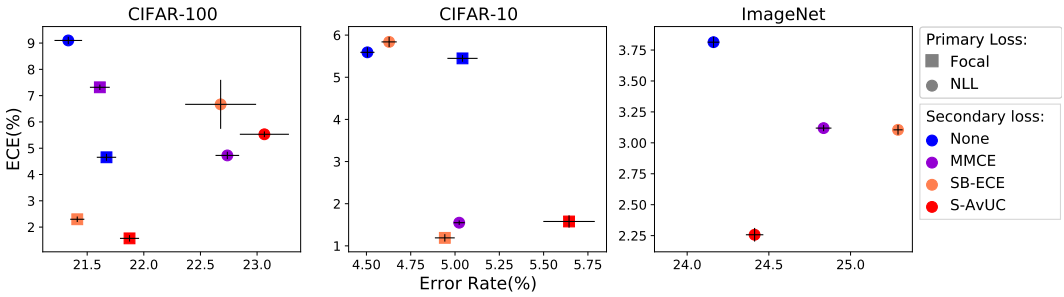

Figure 1: Soft calibration objectives (S-AvUC, SB-ECE), when used as secondary losses with a primary loss, achieve lower ECE (equal-mass binning, ℓ_2 norm) than the corresponding primary loss for CIFAR-10, CIFAR-100, and ImageNet. These statistically significant wins come at the cost of less than 1% accuracy. We report only cross entropy results for ImageNet since focal loss does not work well here. Values reported are mean over 10 runs, and the error bars indicate ± 1 standard error of the mean (SEM). Our methods yield the best ECE for all three datasets.

not directly train their parameters to minimize CE either during training or post-hoc with recalibration. In this paper, we introduce new objectives based on a differentiable binning scheme that can be used to efficiently and directly optimize for calibration.

Contributions. We propose estimating CE with soft (i.e., overlapping, continuous) bins rather than the conventional hard (i.e., nonoverlapping, all-or-none) bins. With this formulation, the CE estimate is differentiable, allowing us to use it as (1) a secondary (i.e. auxiliary) loss to incentivize model calibration during training, and (2) a primary loss for optimizing post-hoc recalibration methods such as temperature scaling. In the same spirit, we soften the AvUC loss [Krishnan and Tickoo, 2020], allowing us to use it as an effective secondary loss during training for non-Bayesian neural networks where the AvUC loss originally proposed for Stochastic Variational Inference (SVI) typically does not work. The soft calibration objectives we propose are easy-to-implement and can be added as a secondary loss to the cross entropy (*NLL*) loss that most developers use for model training and as a replacement to the cross entropy loss for post-hoc temperature scaling.

In an extensive empirical evaluation, we compare soft calibration objectives as secondary losses to existing calibration-incentivizing losses. In the process, we find that these losses outperform prior work on in-distribution test sets. Under distribution shift, we find that calibration-sensitive training objectives as a whole (not always the ones we propose) result in better uncertainty estimates compared to the standard cross entropy loss coupled with temperature scaling.

Our contributions can be summarized as follows:

- We propose simple soft calibration objectives as secondary losses which optimize for CE *throughout training*, and we show that such composite losses obtain *state-of-the-art single-model ECE* in exchange for less than 1% reduction in accuracy (Figure 1).
- We improve upon temperature scaling—a popular post-hoc recalibration method—by directly optimizing the temperature parameter for soft calibration error instead of the typical log-likelihood. Our extension consistently beats original temperature scaling across models trained with different loss functions and across calibration error measures on the CIFAR datasets, and we find that the performance is better under dataset shift (Figure 2).
- Overall, our work demonstrates a fundamental advantage of objectives which better incentivize calibration over the standard practice of training with cross entropy loss and then applying post-hoc methods such as temperature scaling. Uncertainty estimates of neural networks trained using these methods generalize better in and out-of-distribution.

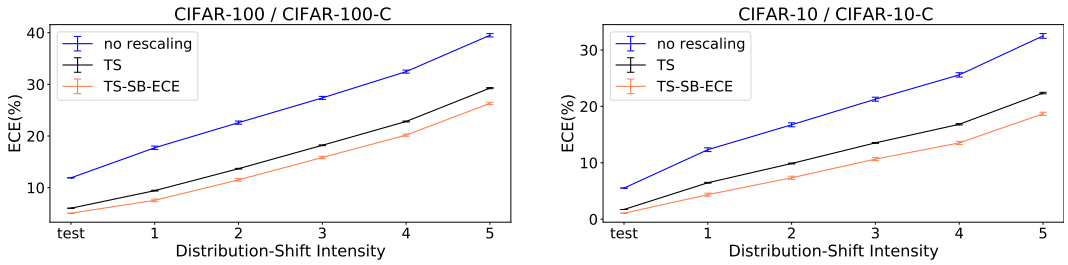

Figure 2: Post-hoc temperature scaling with the soft calibration error objective (TS-SB-ECE) reduces ECE more than standard post-hoc temperature scaling (TS), particularly under distribution shift. This result holds across datasets (left and right panels), distribution shift intensities (along abscissa) and training objectives (not shown). The training objective shown here for both datasets is the most popular one: NLL. The ECE value (equal-mass binning, ℓ_2 norm) shown is the mean ECE across the corruption types that constitute CIFAR-10-C and CIFAR-100-C. Error bars are ± 1 standard error of mean (SEM), corrected for intrinsic variability due to type of corruption [Masson and Loftus, 2003].

2 Related Work

Many techniques have been proposed to train neural networks for calibration. These can be organized into three categories. One category augments or replaces the training loss with a term to explicitly incentivize calibration. These include the AvUC loss [Krishnan and Tickoo, 2020], MMCE loss [Kumar et al., 2018] and Focal loss [Mukhoti et al., 2020, Lin et al., 2018]. The Mean Squared Error loss also compares favourably [Hui and Belkin, 2021] to Cross Entropy loss. We show that our methods outperforms all these calibration-incentivizing training objectives, applied across multiple primary losses. Label smoothing [Müller et al., 2020] has been shown to improve calibration and can be interpreted as a modified primary loss.

A second category of methods are post-hoc calibration methods, which rescale model predictions after training. These methods optimize additional parameters on a held-out validation set [Platt, 1999, Zadrozny and Elkan, 2002, Kull et al., 2019, Zadrozny and Elkan, 2001, Naeini and Cooper, 2016, Allikivi and Kull, 2019, Kull et al., 2017, Naeini et al., 2015, Wenger et al., 2020, Gupta et al., 2020]. The most popular technique is temperature scaling [Guo et al., 2017], which maximizes a single temperature parameter on held-out NLL. We examine temperature scaling and propose an improvisation that directly optimizes temperature for a soft calibration objective instead of NLL. Temperature scaling has shown to be ineffective under distribution shift in certain scenarios [Ovadia et al., 2019]. We show that uncertainty estimates of methods which train for calibration generalize better than temperature scaling under distribution shift.

A third category of methods examines model changes such as ensembling multiple predictions [Lakshminarayanan et al., 2017, Wen et al., 2020] or priors [Dusenberry et al., 2020]. Similar to previous work [Kumar et al., 2018, Lin et al., 2018], we focus on the choice of loss functions for improving calibration of a single neural network—whether during training or post-hoc—and do not compare against ensemble models or Bayesian neural networks. These techniques are complementary to ours and can be combined with our techniques to further improve performance.

Recent works have investigated issues [Nixon et al., 2019, Kumar et al., 2019, Roelofs et al., 2020] with the originally proposed ECE [Guo et al., 2017] and suggested new ones. Debiased calibration error [Kumar et al., 2019] and mean-sweep calibration error [Roelofs et al., 2020] have been shown to have lesser bias and more consistency across the number of bins parameter than ECE. We report these metrics in the appendix.

3 Background

3.1 The Task and the Model

Consider a classification task over K classes with a dataset of N samples $D = \langle(\boldsymbol{x}_i, y_i)\rangle_{i=1}^{N}$ drawn from the joint probability distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ over the input space $\mathcal{X}$ and label space $\mathcal{Y} = \{1, 2, \ldots K\}$. The task is modelled using a deep neural network with parameters $\boldsymbol{\theta}$ whose top layer is interpreted as a softmax layer. The top layer consists of K neurons which produce logits $\boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle g_{\boldsymbol{\theta}}(y|\boldsymbol{x})\rangle_{y\in\mathcal{Y}}$. The predictive probabilities for a given input are:

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle f_{\boldsymbol{\theta}}(y|\boldsymbol{x})\rangle_{y\in\mathcal{Y}} = \mathrm{softmax}(g_{\boldsymbol{\theta}}(\boldsymbol{x})).$$

The parameters $\boldsymbol{\theta}$ of the neural network are trained to minimize $\mathbb{E}_{(\boldsymbol{x},y)}\mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), y)$ using stochastic gradient descent where $(\boldsymbol{x}, y)$ is sampled from $\mathcal{D}(\mathcal{X}, \mathcal{Y})$. Here $\mathcal{L}$ is a trainable loss function which incentivizes the predictive distribution $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ to fit the label y. The model's prediction on datapoint $(\boldsymbol{x}, y)$ is denoted by $q_{\boldsymbol{\theta}}(\boldsymbol{x}) = \arg\max(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))$. We denote by $c_{\boldsymbol{\theta}}(\boldsymbol{x}) = \max(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}))$ the confidence of this prediction and by boolean quantity $a_{\boldsymbol{\theta}}(\boldsymbol{x}, y) = \mathbf{1}_{q_{\boldsymbol{\theta}}(\boldsymbol{x})=y}$ the accuracy of this prediction.

3.2 Expected Calibration Error (ECE)

Given a distribution $\hat{\mathcal{D}}(\mathcal{X}, \mathcal{Y})$ on datapoints, there are two standard notions of Ideal Calibration Error that we refer to as Ideal Binned Expected Calibration Error (of order p) $\mathrm{IECE}_{\mathrm{bin},p}(\hat{\mathcal{D}}, \boldsymbol{\theta}) = \mathrm{IECE}_{\mathrm{bin},p}$ and Ideal Expected Label-Binned Calibration Error (of order p) $\mathrm{IECE}_{\mathrm{lb},p}(\hat{\mathcal{D}}, \boldsymbol{\theta}) = \mathrm{IECE}_{\mathrm{lb},p}$. This nomenclature is consistent with that introduced by Roelofs et al. [2020]. Both these measure, in slightly different ways, the pth root of the pth moment of the absolute difference between model confidence and the empirical accuracy given that confidence. They are defined as follows:

$$\mathrm{IECE}_{\mathrm{bin},p}(\hat{\mathcal{D}}, \boldsymbol{\theta}) = \left(\mathbb{E}_{c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_0)}\left[|\mathbb{E}[a_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_1, \hat{y}_1)|c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_1) = c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_0)] - c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_0)|^p\right]\right)^{1/p} \tag{1}$$

$$\mathrm{IECE}_{\mathrm{lb},p}(\hat{\mathcal{D}}, \boldsymbol{\theta}) = \left(\mathbb{E}_{(\hat{\boldsymbol{x}}_0,\hat{y}_0)}\left[|\mathbb{E}[a_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_1, \hat{y}_1)|c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_1) = c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_0)] - c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_0)|^p\right]\right)^{1/p}. \tag{2}$$

Note that the critical dependence on $\hat{\mathcal{D}}(\mathcal{X}, \mathcal{Y})$ is implicit in both definitions since the datapoint $(\hat{\boldsymbol{x}}_0, \hat{y}_0)$ from the outer expectation and the datapoint $(\hat{\boldsymbol{x}}_1, \hat{y}_1)$ from the inner expectation are both sampled from $\hat{\mathcal{D}}(\mathcal{X}, \mathcal{Y})$.

We cannot compute $\mathrm{IECE}_{\mathrm{bin},p}$ and $\mathrm{IECE}_{\mathrm{lb},p}$ in practice since the number of datapoints are finite. Instead we consider a dataset $\hat{D} = \langle(\hat{\boldsymbol{x}}_i, \hat{y}_i)\rangle_{i=1}^{\hat{N}}$ drawn from $\hat{\mathcal{D}}(\mathcal{X}, \mathcal{Y})$ and partition the confidence interval $[0, 1]$ into bins $\mathcal{B} = \langle B_i\rangle_{i\in\{1,2,\ldots M\}}$, each of which also corresponds to a confidence interval. We will use c_i as a shorthand for $c_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_i)$ and a_i as a shorthand for $a_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_i, \hat{y}_i)$. We denote by $b_i(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = b_i$ the bin to which c_i belongs. We define the size of bin j as $S_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = S_j$, the average confidence of bin j as $C_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = C_j$ and the average accuracy of bin j as $A_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = A_j$. These are expressed as follows:

$$S_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = |\{i|b_i = j\}| \tag{3}$$

$$C_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \tfrac{1}{S_j}\Sigma_{i|b_i=j}c_i \tag{4}$$

$$A_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \tfrac{1}{S_j}\Sigma_{i|b_i=j}a_i. \tag{5}$$

We are now in a position to define the Expected Binned Calibration Error of order p which we denote by $\mathrm{ECE}_{bin,p}$ and Expected Label-Binned Calibration Error of order p we denote by $\mathrm{ECE}_{lb,p}$. These serve as empirical approximations to the corresponding intractable ideal notions from equations 1 and 2. They are defined as follows:

$$\mathrm{ECE}_{bin,p}(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \left(\Sigma_{i=1}^{M}\tfrac{S_j}{\hat{N}}\cdot|A_j - C_j|^p\right)^{1/p} \tag{6}$$

$$\mathrm{ECE}_{lb,p}(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \left(\tfrac{1}{\hat{N}}\Sigma_{i=1}^{\hat{N}}|A_{b_i} - c_i|^p\right)^{1/p}. \tag{7}$$

It follows from Jensen's inequality that $\mathrm{ECE}_{lb,p}(\mathcal{B}, D, \boldsymbol{\theta}) \geq \mathrm{ECE}_{bin,p}(\mathcal{B}, D, \boldsymbol{\theta})$ [Roelofs et al., 2020].

4 Soft Calibration Objectives

In this section, we define quantities that can be used to better incentivize calibration during training.

4.1 Soft-Binned ECE (SB-ECE)

The quantities in the definitions of $\text{ECE}_{bin,p}$ and $\text{ECE}_{lb,p}$ can be written in terms of a formal definition of the bin membership function. Let us denote the bin-membership function for a given binning $\mathcal{B} = \langle B_i \rangle_{i \in \{1,2,\ldots M\}}$ by $\boldsymbol{u}_{\mathcal{B}} : [0,1] \to \mathcal{U}_M$, where $\mathcal{U}_M = \{\boldsymbol{v} \in [0,1]^M | \Sigma_j v_j = 1\}$ is the set of possible bin membership vectors over M bins. The membership function for bin j is denoted by $u_{\mathcal{B},j} : [0,1] \to [0,1]$ and is defined by $u_{\mathcal{B},j}(c) = \boldsymbol{u}_{\mathcal{B}}(c)_j$. The size, average accuracy, and average confidence of bin j from equations 3, 4, and 5 can now be written in terms of $\boldsymbol{u}_{\mathcal{B}}$ as follows:

$$S_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \Sigma_{i=1}^{\hat{N}} u_{\mathcal{B},j}(c_i) \tag{8}$$

$$C_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \tfrac{1}{S_j} \Sigma_{i=1}^{\hat{N}} (u_{\mathcal{B},j}(c_i) \cdot c_i) \tag{9}$$

$$A_j(\mathcal{B}, \hat{D}, \boldsymbol{\theta}) = \tfrac{1}{S_j} \Sigma_{i=1}^{\hat{N}} (u_{\mathcal{B},j}(c_i) \cdot a_i). \tag{10}$$

The quantities $\text{ECE}_{bin,p}$ and $\text{ECE}_{lb,p}$ can further be written in terms of the quantities S_j, C_j and A_j using equation 6 and a modification of equation 7 (see equation 12 below). We know that the differentials $\partial \text{ECE}_{bin,p}/\partial \boldsymbol{\theta}$ and $\partial \text{ECE}_{lb,p}/\partial \boldsymbol{\theta}$ are non-trainable since they are zero within bin boundaries and undefined at bin boundaries. The formulation above makes it clear that this is precisely because $\partial u_{\mathcal{B},j}/\partial c$ also has the same undesirable property. Moreover, this observation implies that if we could come up with a trainable soft bin-membership function $\boldsymbol{u}^*_{\mathcal{B}}$ then we could use it in place of the usual hard bin-membership function $\boldsymbol{u}_{\mathcal{B}}$ to obtain a trainable version of $\text{ECE}_{bin,p}$ and $\text{ECE}_{lb,p}$.

With this motivation we define the soft bin-membership function that has a well-defined non-zero gradient in $(0,1)$. It is parameterized by the number of bins M and a temperature parameter T. We consider equal-width binning for simplicity and so we represent it as $\boldsymbol{u}^*_{M,T}$ rather than $\boldsymbol{u}^*_{\mathcal{B}}$. We desire unimodality over confidence: if ξ_j denotes the center of bin j then we want $\partial u^*_{M,T,j}/\partial c$ to be positive for $c < \xi_j$ negative for $c > \xi_j$. Similarly, we also desire unimodality over bins: if $c < \xi_i < \xi_j$ or $c > \xi_i > \xi_j$, then we want that $u^*_{M,T,i}(c) > u^*_{M,T,j}(c)$. Finally, we also want the aforementioned temperature parameter T to control how close the binning is to hard binning (i.e. how steeply membership drops off). This would give us the nice property of hard-binning being a limiting condition of soft-binning. With this motivation, we define the soft bin-membership function as

$$\begin{aligned} \boldsymbol{u}^*_{M,T}(c) &= \text{softmax}(\boldsymbol{g}_{M,T}(c)), \\ \text{where } g_{M,T,i}(c) &= -(c - \xi_i)^2/T \qquad \forall i \in \{1, 2, \ldots M\}. \end{aligned}$$

Figure 3 visualizes soft bin-membership. We can now formulate trainable calibration error measures. We define the Expected Soft-Binned Calibration Error $\text{SB-ECE}_{\text{bin},p}(M, T, \hat{D}, \boldsymbol{\theta})$ and Expected Soft-Label-Binned Calibration Error $\text{SB-ECE}_{\text{lb},p}(M, T, \hat{D}, \boldsymbol{\theta})$:

$$\text{SB-ECE}_{\text{bin},p}(M, T, \hat{D}, \boldsymbol{\theta}) = \left(\Sigma_{i=1}^{M} (\tfrac{S_j}{\hat{N}} |A_j - C_j|^p) \right)^{1/p}, \tag{11}$$

$$\text{SB-ECE}_{\text{lb},p}(M, T, \hat{D}, \boldsymbol{\theta}) = \left(\tfrac{1}{\hat{N}} \Sigma_{i=1}^{\hat{N}} \Sigma_{j=1}^{M} (u^*_{\mathcal{M},T,j}(c_i) \cdot |A_j - c_i|^p) \right)^{1/p}. \tag{12}$$

The quantities S_j, C_j and A_j in these expressions are obtained by using the soft bin membership function $\boldsymbol{u}^*_{M,T}$ in place of the hard bin membership function $\boldsymbol{u}_{\mathcal{B}}$ in equations 8, 9 and 10 respectively.

We use these trainable calibration error measures as: (1) part of the training loss function and (2) the objective that is minimized to tune the temperature scaling parameter T_{ts} during post-hoc calibration.

4.2 Soft AvUC (S-AvUC)

The accuracy versus uncertainty calibration (AvUC) loss [Krishnan and Tickoo, 2020] categorizes each prediction that a model with parameters $\boldsymbol{\theta}$ makes for a labelled datapoint $d_i = (\boldsymbol{x}_i, y_i)$ from dataset $\hat{D}$ according to two axes: (1) accurate [A] versus inaccurate [I], based on the value of the boolean quantity $a_{\boldsymbol{\theta}}(\boldsymbol{x}_i, y_i) = a_i$ (2) certain [C] versus uncertain [U], based on whether the entropy

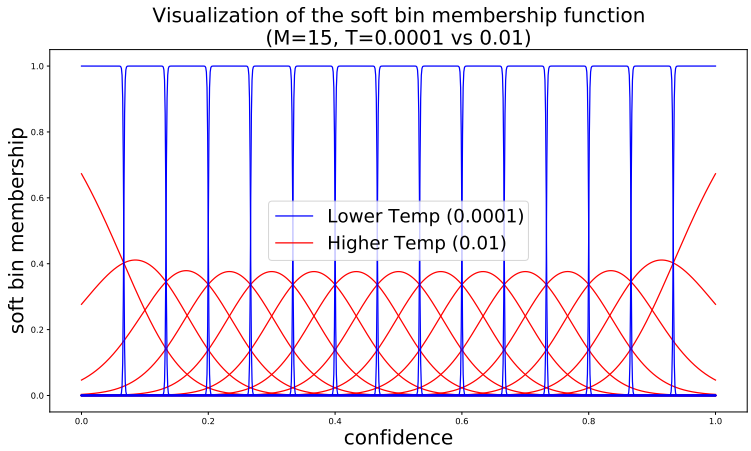

Figure 3: Visualization of the soft bin membership function which shows that the temperature parameter determines the sharpness of the binning. Soft binning limits to hard binning as temperature tends to zero.

$h(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) = h_i$ of the predictive distribution is above or below a threshold κ. Denote the number of elements from $\hat{D}$ that fall in each of these 4 categories by $\hat{n}_{\text{AC}}$, $\hat{n}_{\text{AU}}$, $\hat{n}_{\text{IC}}$ and $\hat{n}_{\text{IU}}$ respectively. The AvUC loss incentivizes the model to be certain when accurate and uncertain when inaccurate:

$$\text{AvUC}(\kappa, \hat{D}, \boldsymbol{\theta}) = \log\left(1 + \frac{n_{\text{AU}} + n_{\text{IC}}}{n_{\text{AC}} + n_{\text{IU}}}\right), \tag{13}$$

where the discrete quantities are relaxed to be differentiable:

$$\begin{aligned} n_{\text{AU}} &= \Sigma_{i|(\boldsymbol{x}_i, y_i) \in S_{\text{AU}}}(c_i \tanh h_i) & n_{\text{IC}} &= \Sigma_{i|(\boldsymbol{x}_i, y_i) \in S_{\text{IC}}}((1 - c_i)(1 - \tanh h_i)) \\ n_{\text{AC}} &= \Sigma_{i|(\boldsymbol{x}_i, y_i) \in S_{\text{AC}}}(c_i(1 - \tanh h_i)) & n_{\text{IU}} &= \Sigma_{i|(\boldsymbol{x}_i, y_i) \in S_{\text{IU}}}((1 - c_i) \tanh h_i). \end{aligned} \tag{14}$$

Krishnan and Tickoo [2020] have showed good calibration results using the AvUC loss in SVI settings. However, in our experiments we found that the addition of the AvUC loss term resulted in poorly calibrated models in non-SVI neural network settings (see Appendix). One reason for this seems to be that minimizing the AvUC loss results in the model being incentivized to be even more confident in its inaccurate and certain predictions (via minimizing n_{IC}) and even less confident in its accurate and uncertain predictions (via minimizing n_{AU}). This conjecture is validated by experimental observations: when we stopped the gradients flowing through the c_i terms in equation 14, we were able to obtain calibrated models. Fixing this incentivization issue in a more principled manner than stopping gradients is desirable. Another desirable improvisation is replacing the hard categorization into the certain/uncertain bins with a soft partitioning scheme. We meet both these objectives by defining a notion of soft uncertainty.

We want a limiting case of the soft uncertainty function to be the hard uncertainty function based on an entropy threshold κ. This implies that we will continue to have a parameter κ despite getting rid of the hard threshold. As before, we desire a temperature parameter T that will determine how close the function is to the hard uncertainty function. The soft uncertainty function $t_{\kappa,T} : [0, 1] \to [0, 1]$ takes as input the $[0, 1]$-normalized entropy $h_i^* = h_i / \log(K)$ of the predicted posterior where K is the number of classes. We also need $\partial t_{\kappa,T} / \partial h^*$ to be positive in $[0, 1]$ and would like $t_{\kappa,T}$ to satisfy $\lim_{h^* \to 0} t_{\kappa,T}(h^*) = 0$ and $\lim_{h^* \to 1} t_{\kappa,T}(h^*) = 1$. Finally, it would be good to have the $[0, 1]$ identity mapping as a special case of $t_{\kappa,T}$-family for some value of κ and T. We now define the soft-uncertainty function the following way so that it meets all stated desiderata:

$$t_{\kappa,T}(h^*) = \text{logistic}\left(\frac{1}{T} \log \frac{h^*(1 - \kappa)}{(1 - h^*)\kappa}\right).$$

Finally, we define Soft AvUC in terms of soft uncertainty by modifying equations 13 and 14. In our experiments, we use Soft AvUC as part of the loss function to obtain calibrated models.

$$\text{S-AvUC}(\kappa, T, \hat{D}, \boldsymbol{\theta}) = \log\left(1 + \frac{n'_{\text{AU}} + n'_{\text{IC}}}{n'_{\text{AC}} + n'_{\text{IU}}}\right), \tag{15}$$

where

$$\begin{aligned} n'_{\text{AU}} &= \Sigma_{i|a_i=1}(t_{\kappa,T}(h_i^*) \tanh h_i) & n'_{\text{IC}} &= \Sigma_{i|a_i=0}((1 - t_{\kappa,T}(h_i^*))(1 - \tanh h_i)) \\ n'_{\text{AC}} &= \Sigma_{i|a_i=1}((1 - t_{\kappa,T}(h_i^*))(1 - \tanh h_i)) & n'_{\text{IU}} &= \Sigma_{i|a_i=0}(t_{\kappa,T}(h_i^*) \tanh h_i). \end{aligned} \tag{16}$$

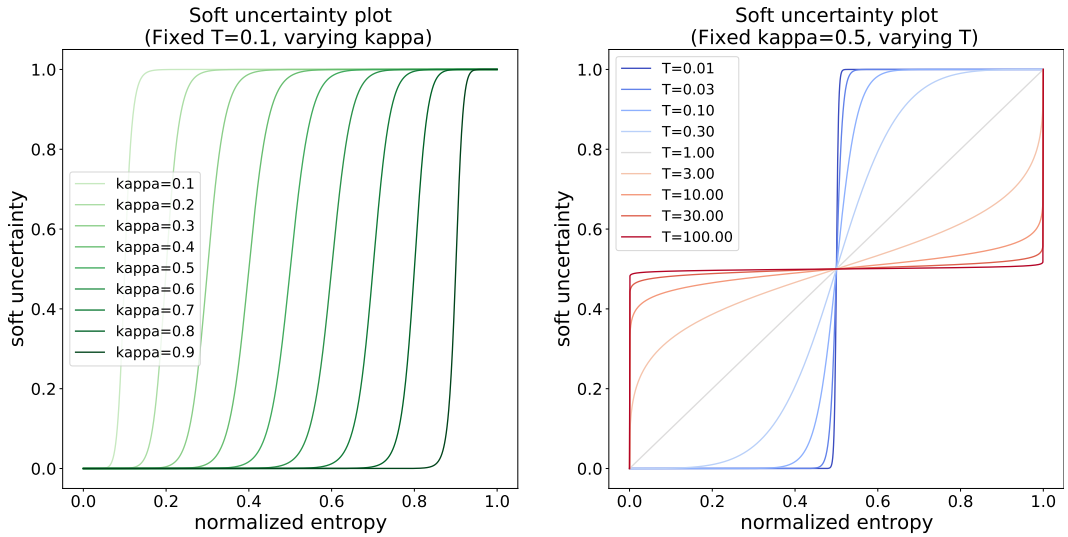

Figure 4: Visualization of the soft uncertainty function $t_{\kappa,T}(h^*)$ which shows that the parameter κ captures the soft-threshold whereas the parameter T captures the sharpness of the thresholding.

5 Results

We compare our soft calibration objectives to recently proposed calibration-incentivizing training objectives such as MMCE and focal loss on the CIFAR-10, CIFAR-100, and ImageNet datasets. We evaluate the full cross product of primary and secondary losses: the options for primary loss are cross entropy/NLL, focal or mean squared error loss; and the options for secondary loss are MMCE, AvUC, SB-ECE or S-AvUC. The results for the mean squared error loss are in the appendix. Our experiments build on the Uncertainty Baselines and Robustness Metrics libraries [Nado et al., 2021, Djolonga et al., 2020].

5.1 Soft Calibration Objectives for End-to-End Training

Our results demonstrate that training losses which include soft calibration objectives obtain state-of-the-art single-model ECE on the test set in exchange for less than 1% reduction in accuracy for all three datasets that we experiment with. In fact, our methods (especially S-AvUC) without post-hoc temperature scaling are better than or as good as other methods with or without post-hoc temperature scaling on all three datasets.

The primary losses we focus on for CIFAR-10/100 are the cross entropy loss and the focal loss. The latter has recently been shown [Lin et al., 2018] to outperform the cross entropy loss. The cross entropy loss outperforms the other two losses on Imagenet, and is thus our sole focus for this dataset.

We use the Wide-Resnet-28-10 architecture [Zagoruyko and Komodakis, 2017] trained for 200 epochs on CIFAR-100 and CIFAR-10. For Imagenet, we use the Resnet-50 [He et al., 2015] architecture training for 90 epochs. All our experiments use the SGD with momentum optimizer with momentum fixed to 0.9 and learning rate fixed to 0.1. The loss function we use in our experiments is PL $+ \beta \cdot$ SL $+ \lambda \cdot$ L2 where PL and SL denote the primary and secondary losses respectively and L2 denotes the weight normalization term with ℓ_2 norm. We tune the β and λ parameters along with the parameters κ and T relevant to the secondary losses SB-ECE$_{lb,p}(M, T, \hat{D}, \boldsymbol{\theta})$ and S-AvUC$(\kappa, T, \hat{D}, \boldsymbol{\theta})$. We tune these hyperparameters sequentially. We fix the learning rate schedule and the number of bins M to keep the search space manageable. The appendix has more details of our hyperparameter search.

In Table 1, we report the runs with the best ECE values that also came within 1% accuracy of the primary loss run with the highest accuracy. We do not list the originally proposed AvUC [Krishnan and Tickoo, 2020] in these tables since adding it as secondary loss did not help in all three settings (the paper only reported results with SVI). We report all results as an average over 10 runs and we

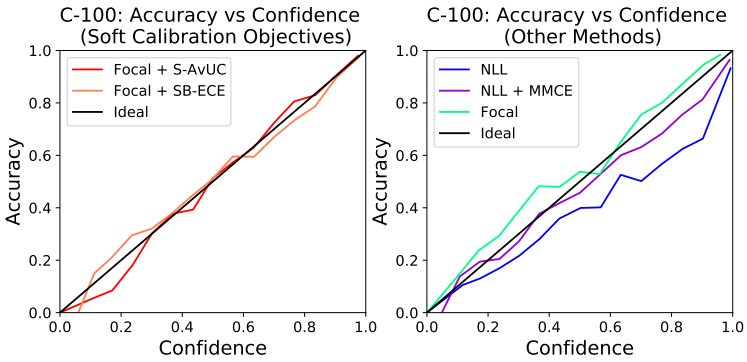

Figure 5: Accuracy vs Confidence plots for various methods on CIFAR-100. NLL is significantly overconfident and NLL + MMCE is somewhat overconfident. While Focal loss is underconfident, augmenting it with Soft Calibration Objectives fixes this issue, resulting in curves closest to the ideal.

use the ℓ_2 norm and equal-mass binning when computing ECE. See the Appendix for the full cross product between all primary and secondary losses.

Table 1: We report average accuracy (with standard error across 10 trials), ECE, and ECE obtained after post-hoc temperature scaling (TS) for models trained with different objectives on the CIFAR-10, CIFAR-100, and ImageNet datasets. ECE is computed with the ℓ_2 norm and equal-mass binning. We find soft calibration objectives (SB-ECE, S-AvUC) result in better or equivalent ECEs compared to previous methods with or without TS. We also find that TS does not always improve ECE. The best ECE value is highlighted for each dataset. The best ECE with TS value is highlighted if it improves over the best value from the ECE column.

(a) CIFAR-100

Loss Fn.	Accuracy	ECE	ECE with TS
NLL	78.7±0.122	9.10±0.139	5.36±0.091
NLL + MMCE	74.9±0.392	3.87±0.123	4.56±0.156
Focal	78.3±0.086	4.66±0.130	6.47±0.140
Focal + SB-ECE	78.6±0.062	2.30±0.105	5.16±0.108
Focal + S-AvUC	78.1±0.084	**1.57**±0.122	4.15±0.090

(b) CIFAR-10

Loss Fn.	Accuracy	ECE	ECE with TS
NLL	95.5±0.040	5.59±0.119	1.95±0.127
NLL + MMCE	95.0±0.031	1.55(0.053)	**1.09**± 0.098
Focal	95.0± 0.085	5.45±0.079	2.69±0.190
Focal + SB-ECE	95.1± 0.056	**1.19**± 0.088	2.08± 0.143
Focal + S-AvUC	94.4± 0.145	1.58± 0.146	1.34± 0.172

(c) ImageNet

Loss Fn.	Accuracy	ECE	ECE with TS
NLL	75.8±0.036	3.81±0.043	2.17±0.045
NLL + MMCE	75.2±0.048	3.12±0.025	2.18±0.029
NLL + SB-ECE	74.7 ±0.028	3.11 ±0.039	**1.92**±0.024
NLL + S-AvUC	75.6 ±0.053	**2.26** ±0.055	2.02 ±0.041

5.2 Soft Calibration Objectives for Post-Hoc Calibration

Standard temperature scaling (TS) uses a cross entropy objective to optimize the temperature. However, using our differentiable soft binning calibration objective, we can optimize the temperature using a loss function designed to directly minimize calibration error. In Figure 2, we compare temperature scaling with a soft binning calibration objective (TS-SB-ECE) to standard temperature scaling with a cross entropy objective (TS) on out-of-distribution shifts of increasing magnitude on both CIFAR-10 and CIFAR-100. Here, we apply both post-hoc calibration techniques to a model trained with standard cross entropy loss, and the distribution shifts come from either the CIFAR-10-C or CIFAR-100-C datasets. We find that TS-SB-ECE outperforms TS under shift in all cases, and the performance increase is similar across shifts of varying magnitude. We also ran additional comparisons between TS-SB-ECE and TS for various combinations of primary and secondary loss, and found that TS-SB-ECE often outperforms TS on these objectives (see Appendix).

5.3 Training for Calibration Under Distribution Shift

In previous sections we have shown that training for calibration outperforms the popular cross entropy loss coupled with post-hoc TS on the in-distribution test set. We find that methods which train for calibration (not always our proposed methods) also outperform the cross entropy loss with TS under dataset shift. Prior work has shown that temperature scaling can perform poorly under distribution

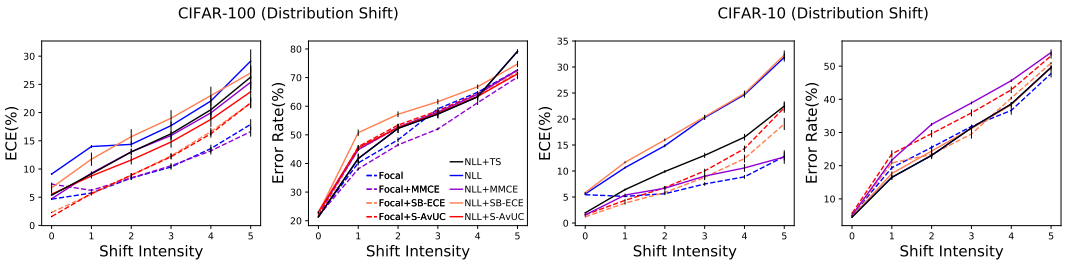

Figure 6: For the CIFAR-10/100 datasets, methods which train for calibration outperform the popular methods - NLL and NLL + TS - under distribution shift. The Focal and Focal + MMCE methods result in the lowest ECEs under shift for good error rates. NLL + MMCE does well in terms of ECE under shift but has high error rates. We note that these methods start off with worse ECEs than our SB-ECE and S-AvUC methods on the test set but end up with better ECE under increasing levels of skew. The OOD datasets that we have used here are CIFAR-10/100-C with skew levels 1-5.

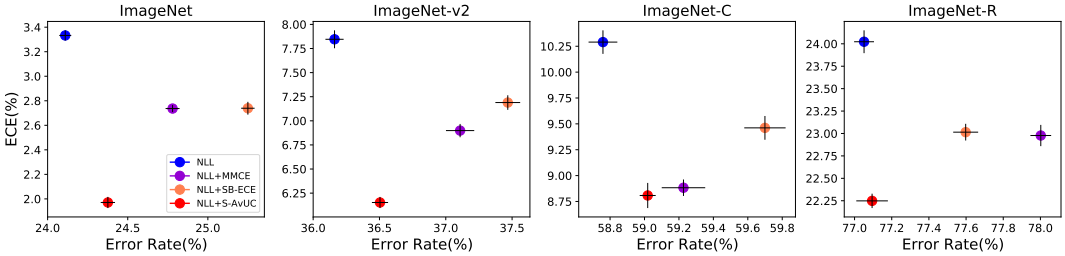

Figure 7: A comparison of methods for ImageNet (primary loss: cross entropy) shows that the S-AvUC secondary loss yields lowest ECEs under dataset shift. Error bars are ± 1 SEM over 10 runs.

shift [Ovadia et al., 2019], and our experiments reproduce this issue. Moreover, we show that training for calibration makes progress towards fixing this problem. However, different methods perform best under distribution shift on different datasets. Whereas S-AvUC does well on ImageNet OOD (see figure 7), Focal loss does better than our methods on CIFAR-10-C and CIFAR-100-C (see figure 6). We cannot prescribe one method in particular under distribution shift given these results but we have shown a crucial benefit of using methods to train for calibration as a whole.

6 Conclusions

We proposed soft calibration objectives with the motivation to train directly for calibation. These objectives—SB-ECE and S-AvUC—are softened versions of the usual ECE measure and the recently-proposed AvUC loss respectively. They are easy-to-implement augmentations for the popular cross entropy loss. We performed a thorough comparison of similar methods to train for calibration. Experiments show that methods based on soft calibration objectives can be used to obtain the best ECE among such methods in exchange for less that 1% drop in accuracy. We note that a model being better calibrated overall does not necessarily mean that it is better calibrated for every group and hence the fairness of our methods as well as related methods must be studied. We also showed that post-hoc temperature scaling works better when tuned for the SB-ECE objective. From the point of view of model developers, all our methods are easy transitions from the popular setup of neural networks trained with the cross entropy loss. Finally we demonstrated that the uncertainty estimates of methods which train for calibration generalize better under dataset shift as compared to post-hoc calibration, which is a fundamental motivation for transitioning to training for calibration.

References

Mari-Liis Allikivi and Meelis Kull. Non-parametric Bayesian isotonic calibration: Fighting overconfidence in binary classification. In *ECML/PKDD*, 2019.

Mariusz Bojarski, D. Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, L. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, X. Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *ArXiv*, abs/1604.07316, 2016.

R. Caruana, Yin Lou, J. Gehrke, Paul Koch, M. Sturm, and Noémie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

Terrance Devries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *ArXiv*, abs/1802.04865, 2018.

Josip Djolonga, Minderer Matthias, Zack Nado, Jeremy Nixon, Rob Romijnders, Dustin Tran, and Mario Lucic. Robustness Metrics, 2020. URL https://github.com/google-research/robustness_metrics.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pages 2782–2792. PMLR, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ArXiv*, abs/1706.04599, 2017.

Kartik Gupta, Amir M. Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, C. Sminchisescu, and R. Hartley. Calibration of neural networks using splines. *ArXiv*, abs/2006.12800, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks, 2021.

Xiaoqian Jiang, M. Osl, J. Kim, and L. Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association : JAMIA*, 19:263 – 274, 2012.

Simon Kocbek, Primoz Kocbek, Leona Cilar, and Gregor Stiglic. Local interpretability of calibrated prediction models: A case of type 2 diabetes mellitus screening test. *arXiv preprint arXiv:2006.13815*, 2020.

R. Krishnan and O. Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *ArXiv*, abs/2012.07923, 2020.

Volodymyr Kuleshov and S. Ermon. Estimating uncertainty online against an adversary. In *AAAI*, 2017.

Meelis Kull, Telmo de Menezes e Silva Filho, and Peter A. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *AISTATS*, 2017.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with Dirichlet calibration. In *NeurIPS*, 2019.

A. Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, 2018.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *NeurIPS*, 2019.

Balaji Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

M. E. J. Masson and G. R. Loftus. Using confidence intervals for graphically based data interpretation. *Canadian Journal of Experimental Psychology*, 57:203–220, 2003. URL http://web.uvic.ca/psyc/masson/ML03.pdf.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, S. Golodetz, P. Torr, and P. Dokania. Calibrating deep neural networks using focal loss. *ArXiv*, abs/2002.09437, 2020.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help?, 2020.

Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning, 2021. URL https://github.com/google/uncertainty-baselines.

M. Naeini and G. Cooper. Binary classifier calibration using an ensemble of near isotonic regression models. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 360–369, 2016.

M. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 2015.

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

Yaniv Ovadia, E. Fertig, J. Ren, Zachary Nado, D. Sculley, S. Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 1999.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration, 2017.

R. Roelofs, N. Cain, Jonathon Shlens, and M. Mozer. Mitigating bias in calibration error estimation. *ArXiv*, abs/2012.08668, 2020.

Zhihui Shao, Jianyi Yang, and Shaolei Ren. Calibrating deep neural network classifiers on out-of-distribution datasets. *ArXiv*, abs/2006.08914, 2020.

Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

Jonathan Wenger, H. Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *AISTATS*, 2020.

B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *ICML*, 2001.

B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The abstract is consistent with the three contributions. This is also consistent with the experiments, figures and conclusions.
   (b) Did you describe the limitations of your work? [Yes] In figure 6 and sections 1 and 5.3, we mention that our methods don't always work well under dataset shift.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discussed the need to inspect the fairness of our and related methods in section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code will be open-sourced.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 5.1 has many details and the appendix will have the rest.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See figures 1, 6 and 7.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Will include details in appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]