On Deep Neural Network Calibration by Regularization and its Impact on Refinement

Anonymous authors
Paper under double-blind review

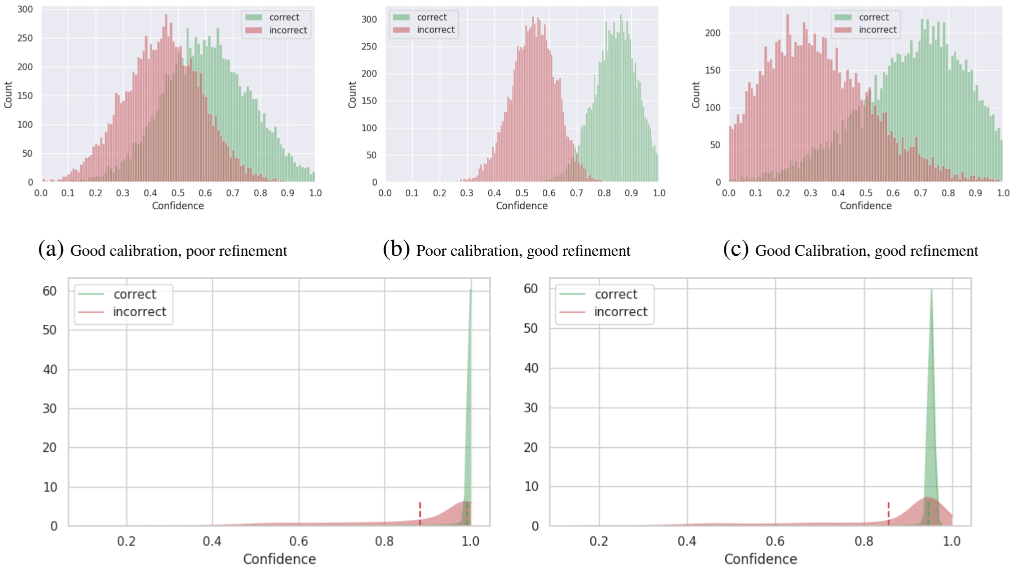
Abstract

Deep neural networks have been shown to be highly miscalibrated. often they tend to be overconfident in their predictions. It poses a significant challenge for safety-critical systems to utilise deep neural networks (DNNs), reliably. Many recently proposed approaches to mitigate this have demonstrated substantial progress in improving DNN calibration. However, they hardly touch upon refinement, which historically has been an essential aspect of calibration. Refinement indicates separability of a network's correct and incorrect predictions. This paper presents a theoretically and empirically supported exposition reviewing refinement of a calibrated model. Firstly, we show the breakdown of expected calibration error (ECE), into predicted confidence and refinement under the assumption of over-confident predictions. Secondly, linking with this result, we highlight that regularisation based calibration only focuses on naively reducing a model's confidence. This logically has a severe downside to a model's refinement as correct and incorrect predictions become tightly coupled. Lastly, connecting refinement with ECE also provides support to existing refinement based approaches which improve calibration but do not explain the reasoning behind it. We support our claims through rigorous empirical evaluations of many state of the art calibration approaches on widely used datasets and neural networks. We find that many calibration approaches with the likes of label smoothing, mixup etc. lower the usefulness of a DNN by degrading its refinement. Even under natural data shift, this calibration-refinement trade-off holds for the majority of calibration methods.

1 Introduction

Guo et al. (2017) showed that many popular deep neural networks are highly miscalibrated. This implies that the model's confidence in its estimate is not reflective of its accuracy. Typically, the output after a softmax layer of a neural network is interpreted as confidence (Hendrycks & Gimpel, 2017; Guo et al., 2017). Many studies have found that DNNs output high confidences for incorrectly classified samples (Guo et al., 2017; Pereyra et al., 2017). For scenarios such as automated driving, medical image analysis etc. where one wishes to avoid failures at all cost, such highly confident incorrect predictions can prove fatal. As a result, calibration is a desired property of the deployed neural networks, which is being actively studied in deep learning research. However, calibration is not the only component that describes a reliable system. Along with calibration we also require the predictions to be refined.

Refinement describes the separability of a binary classification problem (Murphy, 1973; Gneiting et al., 2007). To build trust, it can be interpreted as the degree of confidence separation between correct and incorrect predictions. It serves as an important heuristic for real world deployment as more often than not the predictions are imposed over an operating threshold and the rest are forwarded to fallback mechanism for further evaluation. For example, in estimating if there is an object ahead of a car we might want to rely on the predictions if the estimated confidence lies above a pre-selected (based on validation) value. The idea of using confidence for reliability of predictions is very similar to how calibration is assessed as well. Good refinement indicates an ordinal ranking of predictions which allows better segregation of correct predictions from incorrect ones (Moon et al., 2020). Such a ranking can then allow the user to find an appropriate operating



(a) Good calibration, poor refinement

(b) Poor calibration, good refinement

(c) Good Calibration, good refinement

(d) Density plots of an uncalibrated network(CIFAR-10)

(e) Density plots of a label smoothed network(CIFAR-10)

Figure 1: Hypothetical classification results(a–c) leading to different calibration and refinement scenarios. Figure a) has low calibration error ($ECE = 0.06$) but also a low refinement score ($AUROC = 77.4\%$). In b) we have well refined outputs ($AUROC = 99.46\%$) and poor calibration performance ($ECE = 0.18$). Lastly in figure c), the calibration error ($ECE = 0.08$) and the refinement score ($AUROC = 89\%$) are relatively better. The details of the metrics are provided in section 3.3. (e) shows the effect of applying label smoothing on the output densities vs. an uncalibrated model (d). In (e) though the network is better calibrated ($ECE = 3.93$ vs $ECE = 4.80$), the separation of incorrect predictions with higher confidence than correct predictions has decreased($AUROC$ of 78% vs. 90%).

.

threshold which reduces the chances of encountering incorrect predictions. Moreover, it also plays an important part in describing predictors' effectiveness.

To be better calibrated, a predictor can *cheat* by artificially making predictions around the empirical accuracy which is often referred to as predicting the marginal. This implies that for a binary classifier if its accuracy is 50% then making all predictions with confidence of 50% makes it perfectly calibrated but, the prediction thus made are useless. The model learnt is no better than a random coin flip. To emphasize on this example, we provide some more hypothetical settings in figure 1. We can qualitatively observe that it is possible for a network to exhibit varying degree of calibration and refinement in its predictions for the same final accuracy ($\approx 50\%$). In (a), we have a classifier which is well calibrated but poorly refined. As the network makes prediction mostly with a confidence of $40\% - 60\%$ with a matching accuracy, the usefulness of such a predictor is low as you lose a number of correct predictions by operating above 50% confidence. For (b), we see that the predictions are well separated but not well calibrated. We can select an operating threshold for the network to ensure that we don't encounter many false-positives in practice; however, the remaining predictions become uncalibrated. Case (c) shows an ideal scenario where the predictions are well separated and calibrated. The correct predictions are all predicted with very high confidence, and incorrect predictions consist of very low confidence values. We also present a real scenario figures (d, e), wherein the confidence decreased after label smoothing has led to larger degradation of the quality of predictions.

Though commonly studied together in the domains of statistics (Gneiting et al., 2007), meteorological forecast (Murphy & Winkler, 1977), medical analysis (Van Calster et al., 2019); for recent approaches proposed in the deep learning domain, the joint importance has been sidelined for individual improvements. Many of the recently proposed calibration methods employ strictly proper scores such as Brier Score (Brier, 1950) (mean squared error) and negative log-likelihood to measure calibration. Such scores have been known to decompose into calibration, and refinement components (Murphy, 1973). However, a metric which produces a single score reflecting 2 complex attributes can conceal the area in which the improvement is made. Due to this reason, many approaches utilise Expected Calibration Error (ECE) (Niculescu-Mizil & Caruana, 2005; Naeini et al., 2015) or its variants to focus only on the calibration aspect of a forecaster. Motivated from reliability diagrams, it measures the difference between model confidence and accuracy computed over various bins.

Knowing that refinement and calibration play an important part and consequently have been an integral component for describing a trustworthy and reliable predictor, it raises an important question: 'How well do modern calibration approaches fare on refinement?'. The focus of our paper is to investigate this question. Our main contributions are as follows:

- We mathematically highlight the connection between ECE and area under the ROC curve (AUROC) computed for a classification task. AUROC serves as a measure for refinement of predictions in this work. This serves to show that model confidence and confidence refinement are two areas focusing on which we can improve model calibration. This provides theoretical backing to various refinement based methods which improve calibration for which this support didn't exist.
- We also shed light on the link between the calibration approaches (based on regularisation) and the previously derived relationship to highlight the mode of working of such algorithms. We find that these algorithms work only on the confidence aspect of the classification which can in theory lead to predicting the marginal.
- We provide supporting empirical evidence to illustrate improved calibration but at the expense of refinement of many calibration approaches. As overall the confidence is reduced in the final predictions, this leads to poor refinement.
- Lastly, we provide empirical evidence of calibration-refinement trade-off under natural data shift. We find that refinement, in this case, is also degraded w.r.t an uncalibrated baseline.

The structure of the paper is as follows: In Section 2, we first provide formal introduction to the concepts of calibration and refinement. We further show that under a weak assumption the goal of minimising the calibration error falls in line to improve separability between correctly & incorrectly classified samples. Furthermore, we shed light on the working method of many popular calibration approaches. In Section 3, we review the existing approaches proposed for calibration and the employed metrics. Sections 4 and 5 describe the evaluation setting and experiments which empirically verify our theoretical understanding built in Section 2. We discuss the implications of our findings, future work and conclusions in Section 6.

2 Calibration & Refinement

A dataset is composed of tuples of inputs and targets represented as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{L}$, where $x \in \mathbb{R}^d$, $y_i \in \mathcal{Y} = \{1, 2, \dots K\}$ and L are the total number of samples in the dataset. We represent the learnable weights of a network as θ. The output of a network is a probability distribution over K possible outcomes. The predicted category and predicted confidence are respectively

$$\hat{y}_i = argmax_{k \in \mathcal{Y}} P(Y = k | x_i, \theta) \tag{1}$$

$$c_i = max_{k \in \mathcal{Y}} P(Y = k | x_i, \theta), \tag{2}$$

where c_i is referred to as either the winning probability or maximum class probability. We focus on the problem of calibration and refinement for a reduced binary setting. For a multi-class classification problem we form two groups, overall *correctly* classified samples (or positive category) and overall *incorrectly* classified samples (or negative category). We intend to measure calibration and refinement within this reduced setting.

Definition 2.1 (Calibration)**.** A model P_θ is calibrated if $\mathcal{P}(y_i = \hat{y}_i | c_i, \theta) = c_i \; \forall (x_i, y_i) \in \mathcal{D}^t$. $\mathcal{D}^t$ being the test set.

This implies that the accuracy of the model should be reflective of its confidence in the prediction. Deviation from it leads to under-confident (accuracy $>$ predicted confidence) or over-confident (accuracy $<$ predicted confidence) models. A common metric often used to measure calibration in practice is the Expected calibration error (Naeini et al., 2015). It is measured as the difference between the accuracy and predicted confidences computed over several bins. Formally,

$$ECE \triangleq \sum_{m}^{M} \frac{|B_m|}{L} |A_m - C_m|, \tag{3}$$

where average confidence (C) and accuracy (A) is computed after splitting the predictions into predefined M bins sampled uniformly based on the predicted confidence and B_m is the number of total samples falling in bin m.

Definition 2.2 (Refinement)**.** Let S^p and S^n denote correct and incorrect classification of a model on D^t. Predictions are considered refined iff $c_i > c_j \ \forall x_i \in S^p \ , \ \forall x_j \in S^n$.

Refinement enforces a separation between the two sets of prediction. Degroot & Fienberg (1981) provide an alternative definition of refinement for calibrated classifiers. We consider area under the ROC curve (r) (Ling et al., 2003), as an appropriate choice of metric for measuring refinement of a model(Corbière et al., 2019). A common interpretation of r is that it denotes the expectation that a uniformly drawn random positive sample is ranked higher (higher confidence) than a uniformly drawn random negative sample. Hand & Till (2001) calculate r as:

$$r = \frac{R^p - |S^n| \times (|S^n| + 1)/2}{|S^p| \times |S^n|} \tag{4}$$

where, $R^p = \sum_{\forall x \in S^p} rank(x)$ and $rank(x)$ denotes the rank of prediction x in an increasingly sorted list of predictions based on associated confidence. It is straightforward to observe that r for a refined model will always be greater than an unrefined one (switching the rank of an incorrect prediction with the correct one decreases r).

2.1 Connecting ECE and r

Assumption: We assume that $A_m < C_m \forall m$. It implies that the network is over-confident in its prediction throughout. This is partly true in practice as for all deep neural networks the problem of calibration entails over-confident predictions(Thulasidasan et al., 2019). Also, we empirically observed that for networks trained on ImageNet(Deng et al., 2009), CIFAR-100(Krizhevsky, 2009), STL-10(Coates et al., 2011) and CUB-200(Wah et al., 2011) the number of bins for which $A_m <= C_m$ holds true are 80, 95, 94 and 86 respectively for $M = 100$. Recently, a study by Bai et al. (2021) showed that a classifier learnt through well specified logistic regression is destined to be overconfident.

Let, p_m and n_m represent positive and negative category samples in bin m respectively which implies $|S^p| = \sum_m p_m$ and $|S^n| = \sum_m n_m$. We can now describe the accuracy within a bin as $A_m = \frac{p_m}{p_m + n_m}$. Substituting all the above conversions to Equation equation 3, ECE is updated as

$$ECE = \sum_m \frac{(p_m + n_m)}{|S^p| + |S^n|} \left(C_m - \frac{p_m}{p_m + n_m} \right). \tag{5}$$

This can be further expanded to

$$ECE = \underbrace{\sum_m \frac{(p_m + n_m)}{|S^p| + |S^n|} C_m}_{I} - \underbrace{\sum_m \frac{p_m}{|S^p| + |S^n|}}_{II}. \tag{6}$$

I denotes the expected confidence of the predictions, $\mathbb{E}_{C \sim p_\theta(X)}[C]$, of the model, whereas II is the expected model accuracy, $\mathbb{E}[A]$. Equation equation 6 can thus be updated to

$$ECE = \mathbb{E}[C] - \mathbb{E}[A]. \tag{7}$$

For a binary classification task, it has been shown (Hernández-Orallo et al., 2012; Flach & Kull, 2015) that r and $\mathbb{E}[A]$ are linearly related averaged over all possible true-positive rates. They showed that:

$$\mathbb{E}[A] = \frac{P}{|S^p| + |S^n|}(1 - \frac{P}{|S^p| + |S^n|})(2r - 1) + \frac{1}{2}, \tag{8}$$

where r is the area under the ROC curve. Substituting Equation equation 8 for $\mathbb{E}[A]$ in Equation equation 7 and re-arranging the terms gives us the final expression in the form of

$$ECE = \underbrace{\mathbb{E}[C]}_{\alpha} - r \underbrace{\frac{2PN}{(|S^p| + |S^n|)^2}}_{\beta} - \underbrace{\frac{P^2 + N^2}{2(|S^p| + |S^n|)^2}}_{\gamma}. \tag{9}$$

Traditionally, for strictly proper scoring rules such as the Brier score, the decomposition of the metric into calibration and refinement is well known. However, for ECE which is not a strict proper scoring rule, we have shown that the breakdown is into average predicted confidence and refinement under the applied assumption of bins-wide overconfidence.
For a set of predictions, we have the following constraints $P \geq 0$, $N \geq 0$, $|S^p| + |S^n| > 0$, $\beta \geq 0$ and $\gamma > 0$. We can decrease the calibration error by either reducing α and/or increasing r. Moon et al. (2020) have shown that their refinement based approach improves calibration however, they do not provide the reasoning behind such an observation. Their observation can now be supported by the relationship described in Equation equation 9. We also compute calibration of another refinement approach, CFN(Corbière et al., 2019), for which earlier these results were not computed and find that in this case as well the network achieves better calibration after the refinement process (see Section A.3).

2.2 How Regularization Enforces Calibration?

We highlighted the factors which contribute towards lowering of the expected calibration error. In this section, we focus on shedding light on the working route for many regularization based calibration approaches instead. To emphasize, regularization acts as a penalty during the training procedure. Label Smoothing(Müller et al., 2019) provides calibration apart from other benefits. Many existing approaches also have been proven to materialize into label smoothing (LS) such as entropy regularization (ERL) (Pereyra et al., 2017) and focal loss (FL) (Mukhoti et al., 2020). We focus our attention to the label smoothing objective function and decipher the mode of working for this particular algorithm. A training loss consisting of label smoothing can be written as

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{LS}, \tag{10}$$

where CE stands for cross-entropy and LS represents label smoothing contribution. Label smoothing contribution is the KL divergence between uniform distribution (U) and network's output distribution (P_θ). Formally,

$$\mathcal{L}_{LS} = -\mathrm{D}_{KL}(U||P_\theta). \tag{11}$$

$\mathcal{L}_{LS}$ can be expanded as,

$$\mathcal{L}_{LS} = \sum_{i=0}^{i<N} -\underbrace{U(x_i)log(P_\theta(x_i))}_{I} + \underbrace{U(x_i)log(U(x_i)))}_{II}, \tag{12}$$

where x_i is a sample input from a total of N sample points. The value for the uniform distribution is set before hand to a small constant ϵ thus making II a constant term. I is the term which is optimised and for a binary classification problem can be written as

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \epsilon \ logc_i + \epsilon \ log(1 - c_i) \\ \text{s.t.} \quad & 0 \leq c_i \leq 1. \end{aligned} \tag{13}$$

The above expression reaches a minimum value when $c_i = 0.5$. For multi-class classification, the minimum is achieved at $\frac{1}{K}$. This goes on to show that label smoothing works on only reducing the confidence of all of its predictions.
For ERL and FL, the breakdown is similar as they simply rely on slightly different target functions in equation 11. The breakdown is similar when we use their corresponding losses which are:

$$\mathcal{L}_{erl} = -H(P_\theta) \tag{14}$$

$$\mathcal{L}_{focal} = (1 - \gamma)H(P_\theta) \tag{15}$$

where, H is the entropy.
The takeaway is that regularisation added only helps to tone down the winning class confidence and increase the losing confidences. The improvement in calibration is focused more on the α-aspect of Equation equation 9. Intuitively, concentrating predictions at a point will have detrimental effect on a network's refinement as now we have concentrated incorrect and correct predictions.

3 Related Work

3.1 Calibration

This work is focused on calibration of point estimate based deep neural networks. For the Bayesian perspective, we refer the readers to recent works on ensembles(Lakshminarayanan et al., 2017) and cold-posterior(Wenzel et al., 2020). The existing work for calibration of point estimate models can be categorised into the following 3 broad groups based on the commonalities between the approaches.

Regularisation based approaches apply a calibrating penalty to the supervised learning objective. Pereyra et al. (2017) added negative entropy of the predictions to encourage the model to predict less 'peaky' estimates. Subsequently, many approaches have been proposed along this direction which adds noise to the labels (Müller et al., 2019), optimise a proxy for the calibration error metric (Kumar et al., 2018), and replace the cross-entropy objective with focal loss (Mukhoti et al., 2020). Peterson et al. (2019) utilised human inferred soft-targets to improve robustness. This approach can be understood as being along the lines of label smoothing.

Post-hoc approaches re-scale the confidence scores of an uncalibrated neural network to make it calibrated. The scaling hyper-parameters are chosen on a held-out validation set. Some of the recently proposed approaches are temperature scaling (Guo et al., 2017), scaling and binning calibration (Kumar et al., 2019), Dirichlet calibration (Kull et al., 2019), and beta calibration (Kull et al., 2017). These approaches find motivation from classical methods such as Platt scaling (Platt, 1999), binning (Zadrozny & Elkan, 2001), and isotonic regression (Zadrozny & Elkan, 2002).

In the last group, we list the remaining approaches. Mixup (Zhang et al., 2018; Thulasidasan et al., 2019) and AugMix (Hendrycks et al., 2020) combine data augmentation and regularization. Pretraining (Hendrycks et al., 2019a) and self-supervised learning (Hendrycks et al., 2019b) have also been highlighted to be beneficial in this regard.

3.2 Refinement

By refining prediction, methods seek to find a good ordinal ranking of predictions. This may or may not result in a calibrated model as it has not been studied for this problem extensively. Moon et al. (2020) incorporated 'Correctness Ranking Loss' to allow a DNN to learn appropriate ordinal rankings for classified samples. They also observed that their approach helped in calibrating the network; however, do not discuss the reasoning behind this observation. As a replacement for confidence estimate, Jiang et al. (2018) introduced 'TrustScore', which provides a better ordinal ranking of predictions than the output of the network. They utilised the ratio between the distance from the sample to the nearest class different from the predicted class and the distance to the predicted class as the trust score. ConfidNet (Corbière et al., 2019) incorporates the learning of this trust score as an additional branch in the network. In the post-hoc stage, ConfidNet branch of the classifier is trained to predict a confidence score which mimics the reliability of the network on its prediction. Meta-cal(Ma & Blaschko, 2021), is a recent attempt to ensure that calibration ensures usability of the classifier though post-hoc ranking on an existing calibrated network.

3.3 Metrics

For the scores utilised to assess calibration, the most commonly used are Brier score, negative log-likelihood (NLL), Expected Calibration Error (ECE) and Overconfidence Error (OE).

Brier score (Brier, 1950) and NLL are strictly proper scoring rules (Gneiting & Raftery, 2007; Dawid & Musio, 2014). It has been shown that strictly proper scoring rules decompose into calibration and refinement components (Murphy, 1973; Blattenberger & Lad, 1985). The presence of the refinement component describes the utility of the calibration approach. However, the implicit combination of the two can conceal the area of improvement.

ECE and OE (Degroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005; Naeini et al., 2015) are proper scoring rules (**not** strict) and are adapted from reliability diagrams for judging the calibration of the models. They are not strict as the optimum value of 0 can be achieved with more than one set of predictions. These metrics also suffer from high sensitivity to the bin hyper-parameter (Nixon et al., 2020). Finding a good calibration metric is an active area of research (Geifman et al., 2019; Nixon et al., 2020).

		VGG-16					ResNet-50					DenseNet-121				
	Method	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	AUPR (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	AUPR (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	AUPR (↑)
CIFAR-10	Baseline	$93.74_{0.08}$	$10.92_{0.20}$	$4.80_{0.12}$	$90.90_{0.64}$	$99.17_{0.12}$	$95.65_{0.02}$	$7.12_{0.12}$	$2.69_{0.07}$	$93.80_{0.04}$	$99.66_{0.01}$	$95.55_{0.03}$	$7.43_{0.13}$	$2.83_{0.12}$	$92.12_{0.18}$	$99.49_{0.03}$
	LS	$94.11_{0.04}$	$10.47_{0.09}$	$3.86_{0.36}$	$\mathbf{75.96}_{4.26}$	$\mathbf{96.78}_{0.99}$	$\mathbf{95.44}_{0.14}$	$\mathbf{8.00}_{0.19}$	$\mathbf{3.82}_{0.17}$	$\mathbf{73.46}_{1.41}$	$\mathbf{96.83}_{0.32}$	$\mathbf{95.39}_{0.00}$	$\mathbf{8.51}_{0.04}$	$2.80_{0.08}$	$\mathbf{73.56}_{1.71}$	$\mathbf{96.95}_{0.31}$
	ERL	$93.86_{0.05}$	$10.53_{0.07}$	$4.32_{0.04}$	$\mathbf{88.28}_{1.37}$	$\mathbf{98.77}_{0.25}$	$95.66_{0.18}$	$7.01_{0.18}$	$2.37_{0.02}$	$\mathbf{93.64}_{0.49}$	$99.66_{0.03}$	$\mathbf{95.47}_{0.01}$	$7.19_{0.01}$	$2.18_{0.04}$	$92.83_{0.18}$	$99.55_{0.03}$
	MX	$93.99_{0.05}$	$9.79_{0.18}$	$3.75_{1.01}$	$\mathbf{84.38}_{1.99}$	$\mathbf{97.87}_{0.44}$	$96.06_{0.12}$	$6.35_{0.15}$	$2.67_{0.27}$	$\mathbf{90.10}_{1.36}$	$\mathbf{99.40}_{0.10}$	$95.81_{0.03}$	$6.82_{0.01}$	$\mathbf{3.67}_{0.36}$	$\mathbf{88.98}_{0.02}$	$\mathbf{99.25}_{0.08}$
	FL	$\mathbf{93.63}_{0.12}$	$10.88_{0.16}$	$3.48_{0.06}$	$\mathbf{84.50}_{0.50}$	$\mathbf{98.00}_{0.11}$	$\mathbf{95.28}_{0.08}$	$\mathbf{7.50}_{0.04}$	$1.73_{0.06}$	$\mathbf{92.40}_{0.59}$	$\mathbf{99.53}_{0.05}$	$\mathbf{95.05}_{0.05}$	$\mathbf{7.69}_{0.12}$	$1.71_{0.06}$	$\mathbf{91.72}_{0.15}$	$\mathbf{99.41}_{0.03}$
CIFAR-100	Baseline	$72.46_{0.14}$	$43.26_{0.94}$	$16.29_{1.39}$	$84.97_{0.45}$	$92.23_{0.61}$	$78.31_{0.18}$	$34.05_{0.38}$	$12.17_{0.06}$	$85.69_{0.12}$	$94.59_{0.28}$	$79.57_{0.11}$	$30.68_{0.28}$	$8.47_{0.07}$	$87.07_{0.17}$	$96.09_{0.09}$
	LS	$73.79_{0.88}$	$39.45_{1.04}$	$9.17_{0.59}$	$\mathbf{82.57}_{0.45}$	$\mathbf{90.14}_{1.14}$	$79.02_{0.24}$	$32.07_{0.20}$	$6.21_{0.13}$	$\mathbf{81.96}_{0.64}$	$\mathbf{91.81}_{0.37}$	$\mathbf{78.74}_{0.11}$	$\mathbf{33.01}_{0.33}$	$8.22_{0.02}$	$\mathbf{81.36}_{0.50}$	$\mathbf{91.85}_{0.43}$
	ERL	$72.51_{0.17}$	$42.09_{0.72}$	$13.96_{1.43}$	$\mathbf{84.21}_{0.55}$	$91.37_{1.01}$	$78.63_{0.35}$	$32.56_{0.59}$	$9.80_{0.12}$	$\mathbf{85.41}_{0.59}$	$\mathbf{94.49}_{0.34}$	$79.09_{0.23}$	$30.50_{0.33}$	$6.10_{0.11}$	$\mathbf{86.78}_{0.12}$	$\mathbf{95.81}_{0.09}$
	MX	$73.88_{0.78}$	$38.18_{0.64}$	$7.68_{1.17}$	$\mathbf{83.63}_{0.88}$	$\mathbf{90.99}_{1.10}$	$79.62_{0.14}$	$30.27_{0.53}$	$6.02_{1.57}$	$85.78_{0.24}$	$95.00_{0.22}$	$80.06_{0.12}$	$28.93_{0.03}$	$3.55_{0.09}$	$\mathbf{85.80}_{0.32}$	$\mathbf{95.26}_{0.16}$
	FL	$72.93_{0.40}$	$39.68_{0.49}$	$9.05_{0.48}$	$\mathbf{83.12}_{0.37}$	$\mathbf{90.67}_{0.26}$	$78.68_{0.31}$	$30.61_{0.33}$	$3.98_{0.48}$	$85.91_{0.24}$	$95.17_{0.17}$	$79.00_{0.01}$	$29.91_{0.03}$	$3.06_{0.16}$	$\mathbf{86.59}_{0.01}$	$\mathbf{95.96}_{0.04}$

Table 1: Joint evaluation for calibration and refinement. We highlight the values which are on an average worse than the baseline in bold. The arrows indicate that higher (↑) and lower (↓) values are better respectively.

Method	STL-10				CUB-200				ImageNet			
	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)
Baseline	$84.09_{0.27}$	$25.55_{0.34}$	$9.63_{0.02}$	$87.16_{0.18}$	$81.48_{0.31}$	$27.54_{0.54}$	$6.70_{0.28}$	$87.00_{0.51}$	75.83	33.62	3.37	86.78
LS	$84.10_{0.11}$	$23.85_{0.19}$	$6.44_{0.21}$	$\mathbf{85.93}_{0.09}$	$\mathbf{81.22}_{0.10}$	$27.69_{0.04}$	$4.38_{0.05}$	$\mathbf{86.14}_{0.14}$	76.01	**34.21**	**5.86**	**85.14**
ERL	$\mathbf{83.54}_{0.25}$	$\mathbf{25.79}_{1.07}$	$8.70_{1.80}$	$\mathbf{87.09}_{0.13}$	$\mathbf{81.14}_{0.08}$	$27.58_{0.08}$	$3.97_{0.23}$	$\mathbf{86.26}_{0.23}$	75.85	33.59	2.02	**86.41**
MX	$84.91_{0.07}$	$22.50_{0.21}$	$3.66_{0.55}$	$\mathbf{86.19}_{0.37}$	$82.67_{0.08}$	$26.52_{0.10}$	$\mathbf{6.96}_{0.01}$	$\mathbf{85.22}_{0.09}$	76.72	33.50	1.78	**86.45**
FL	$\mathbf{83.15}_{0.25}$	$\mathbf{25.64}_{0.60}$	$6.93_{0.61}$	$\mathbf{85.87}_{0.15}$	$\mathbf{80.63}_{0.30}$	$28.34_{0.62}$	$4.17_{1.16}$	$\mathbf{85.61}_{0.28}$	**74.41**	**36.77**	**8.21**	**85.17**

Table 2: Joint evaluation for calibration and refinement for STL-10(VGG-16), CUB-200(ResNet-50) and ImageNet(ResNet-50).

4 Implementation Details

To empirically verify our findings we employ the following calibration approaches in our study. • Label Smoothing (**LS**) • Entropy Regularization (**ERL**) • Mixup (**MX**) • Focal Loss (**FL**). We compare these approaches to a cross-entropy trained model referred to as **baseline**.

For the datasets we rely on CIFAR-10/100 (Krizhevsky, 2009), STL-10(Coates et al., 2011), CUB-200(Wah et al., 2011) and ImageNet (Deng et al., 2009) which have been used extensively in recent calibration studies. The neural network architectures chosen are Resnet-50 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2015) and DenseNet-121 (Huang et al., 2017) for CIFARs as to reflect on architecture wide occurrence of the calibration-refinement trade-off. Resnet-50 for (Pre-trained) CUB-200 and ImageNet. VGG-16(with batch norm) for STL-10.

Alongside accuracy we report ECE and Brier score for calibration errors whereas, AUROC and AUPR for refinement. All values provided are $\times 100$. We report mean and deviation(as subscript) over 3 trials where applicable. Training details are provided in the supplemental (see Section A.4).

5 Experiments & Results

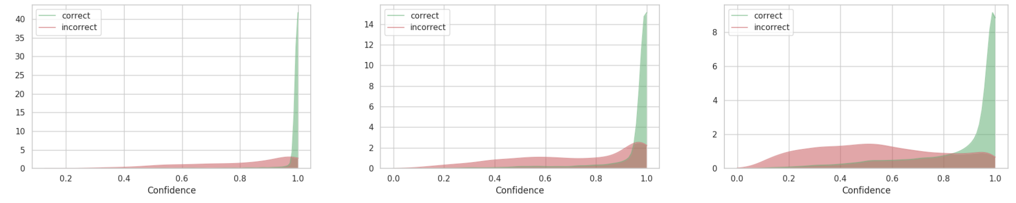
5.1 Calibration & Refinement

Tables 1 and 2 show the joint calibration and refinement on various datasets. Unsurprisingly, calibration approaches attain lower calibration errors for most of scenarios. Also, in many cases brier score is also better than the baseline which hides the shortcoming.

In table 1 we can observe that in terms of refinement, the baseline performs superior to calibrated models. Focusing on AUPR and AUROC, these metrics capture slightly different aspects of the quality of predictions. AUPR is typically a preferred metric when there is an imbalance due to the negative category. But, as the overall accuracy of networks considered is > 50 we believe that is not the case. Additionally, AUPR prioritises positive class samples but not their ordering which forms the definition of refinement. Keeping this in mind, we believe AUROC is a stronger indicator of refinement with AUPR serving a similar but softened purpose.

ERL provides the least improvement in terms of calibration and achieves slightly worse AUROC w.r.t the baseline at times. Out of all the approaches assessed, LS consistently acquires the lowest refinement performance. MX and FL provide moderate to low decay of refinement.

For other datasets in table 2 similar observation of weakening refinement can be drawn. Another point to notice is the varying degree of calibration and refinement across datasets. This can be attributed to over-parameterized training. Mukhoti et al. (2020) argued that over-fitting to training leads to miscalibration. We suspect since the network's overfit to varying degree on different datasets. This results in varied improvement in calibration and hence the impact on refinement also varies. For example, on ImageNet we achieve a baseline training accuracy of 77% as opposed to the



(a) CIFAR-10

(b) CIFAR-100

(c) ImageNet

Figure 2: Density plots for correct & incorrect classification confidences for baseline Resnet-50 models.

Method	CIFAR-10.1				CIFAR10.2				ImageNet-v2			
	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)	Acc	Brier (↓)	ECE (↓)	AUROC (↑)
Baseline	$85.90_{0.64}$	$24.50_{1.23}$	$11.28_{0.71}$	$86.23_{0.91}$	$80.59_{0.25}$	$34.25_{0.73}$	$16.32_{0.60}$	$83.39_{1.72}$	63.17	49.58	8.12	84.97
LS	$86.43_{0.20}$	$23.60_{0.07}$	$6.84_{0.09}$	$\mathbf{77.44}_{0.91}$	$81.15_{0.52}$	$32.37_{0.08}$	$11.26_{0.19}$	$\mathbf{74.05}_{2.85}$	63.52	48.93	3.57	**83.70**
ERL	$86.15_{0.35}$	$23.70_{0.58}$	$10.16_{0.27}$	$\mathbf{84.32}_{1.19}$	$80.62_{0.68}$	$34.23_{1.26}$	$15.60_{0.59}$	$\mathbf{81.85}_{0.19}$	63.17	48.80	4.54	85.27
MX	$86.67_{0.06}$	$20.89_{0.71}$	$6.87_{0.60}$	$\mathbf{84.76}_{1.24}$	$81.77_{0.23}$	$29.35_{0.75}$	$9.77_{1.79}$	$\mathbf{80.25}_{1.67}$	64.23	47.85	5.45	**84.82**
FL	$\mathbf{85.23}_{0.41}$	$\mathbf{25.13}_{0.69}$	$10.51_{0.29}$	$\mathbf{80.55}_{0.89}$	$81.05_{0.19}$	$33.05_{0.36}$	$14.61_{0.29}$	$\mathbf{77.74}_{1.18}$	**61.82**	**51.15**	4.80	**83.10**

Table 3: Joint evaluation for calibration and refinement under natural data shift. For CIFARs we use the VGG-16 models trained on CIFAR-10 and for ImageNet-v2 we employ the Resnet-50s trained on ImageNet.

CIFARs' training accuracy $> 99\%$. Figure 1 we also notice that the density plots for ImageNet are vastly different from CIFARs as the concentration of misclassified samples in the baseline are well separated from the corrects ones.

5.2 Impact on Refinement Under Data Shift

Previously, the test set consisted of samples originating from the same distribution as that of training. In this experiment, we aim to assess the deterioration under natural distribution shift of the datasets. Natural shift implies a subtle change in scene composition, object types, lighting conditions, and many others (Taori et al., 2020). It is logical to assume that a DNN is bound to confront such images in the real world.

Examples of naturally shifted datasets are CIFAR-10.1 (Recht et al., 2018), CIFAR-10.2 (Lu et al., 2020) and ImageNet-v2 (Recht et al., 2019). These datasets are collected following the identical process to that of the original reference dataset. Such datasets have been utilised to measure the lack of generalisation and robustness of many classification models (Taori et al., 2020; Ovadia et al., 2019). This is the first attempt at evaluating calibration-refinement under natural data-shift to the best of our knowledge. Assessment of calibration under synthetic shift has been reported by Ovadia et al. (2019). However, we believe natural data-shift is a scenario which a deployed DNN is more likely to face and hence requires equal if not more attention. By evaluating calibration-refinement trade-off we will also be able to highlight the severity and extent of the problem induced by many calibration approaches.

5.2.1 Results

Table 3 shows the performance of models trained on original datasets and tested on shifted variants. For CIFAR-10.x we use the VGG-16 model trained on CIFAR-10 and for ImageNet-v2 we employee the ResNet-50 trained on ImageNet.

We spot that the trend of worsening refinement continues for models under data shift as well. Similar to what we have already seen for LS, it also provides the lowest refinement performance under natural shift. A surprising observation to note is the poor performance of MX. MX as shown by Thulasidasan et al. (2019) performs well on out-of-distribution detection. However, when the data shift is not severe it appears that mixup provides no added benefit in terms of refinement.

We also observe that calibration approaches provide better calibration than the baseline under the natural shift. This observation has not yet been highlighted in existing studies which focus on ood performance or some form of generalisation metric (relative accuracy) to investigate robustness of a model. For synthetic shifts, Ovadia et al. (2019) made a similar observation and noted that calibration approaches to a certain extent improve calibration on corrupted images w.r.t the baseline.

6 Discussion & Conclusion

In this paper we have brought forth a downside of many calibration approaches. We believe refinement is an important aspect which communicates the usefulness of safety-critical DNNs. Discussed theoretically and empirically, we have shed light on the current state of calibration-refinement trade-off.

Many regularization based calibration approaches disregard the role of refinement, leading to severe loss in the utility of DNNs thus trained. We successfully presented the case of declining refinement for a wide variety of approaches tested on many different datasets. The derived relationship in equation 9 showed how improving refinement can help better calibrate the model. This provides justification for calibration observed for refinement approach of Moon et al. (2020). In the appendix (A.3), we show that calibration is induced by the refinement technique proposed by Corbière et al. (2019). In the future, we aim to focus on finding balanced calibration methods which preserve if not improve refinement of predictions.

The benefits of label smoothing have been highlighted by Müller et al. (2019). We were able to shed light on a severe limitation of the approach, which practitioners were currently unaware of. Similar to LS, other easy to apply calibration methods are also damaging in practice. A similar trend is observed for a NLP classification task reported in appendix A.1.

We observed that the degree of refinement degradation varies from one dataset to another. Mukhoti et al. (2020) discussed the causes for miscalibration and accredited it to the over-fitting on the training data (under cross-entropy loss). We found that the training accuracy achieved by the baseline is 99.99%, 99.4% and 77.9% for CIFAR-10, CIFAR-100 and ImageNet respectively. This signals towards a comparably lower over-fitting of baseline trained on ImageNet and subsequently, a lower impact on calibration leading to a lower refinement degradation.

We also noted the extension of calibration to naturally shifted data. Akin to the observations made by (Ovadia et al., 2019) on their evaluation on synthetically shifted datasets, we observed that existing solutions provide calibration on naturally shifted datasets as well. However, this calibration comes at a cost and as a result refinement aspect of the models is comparably poorer than their uncalibrated counterparts. An important point to note was the failure of Mixup under datashift. Thulasidasan et al. (2019) has demonstrated Mixup's ability to distinguish ood samples however, we believe that natural shift is a weaker notion of data shift than ood evaluation and MX fails to provide any benefit in this regard. We also noted the varying impact of this degradation across datasets. We suspect that the lack of evident over-fitting on ImageNet is the root cause behind the visibly lower calibration-refinement impact on it.

Apart from relying on ECE and Brier score, incorporating metrics like AUROC, AUPR etc. helps in further distinguishing useful calibration approaches. Utilizing such measures can help researchers to make an intelligent and well-formed decision regarding the suitability of an approach for their application. Additionally, many evaluation protocols have also been proposed which extend the problem of calibration to a multi-class setting (Widmann et al., 2019). A natural extension will be to study refinement conjointly with calibration in a similar manner.

To conclude, we have demonstrated a theoretically motivated study of calibration and refinement of many recently proposed calibration approaches. Though these methods improve calibration, they negatively impact refinement when compared to a heavily miscalibrated baseline.

References

Yu Bai, Song Mei, Huan Wang, and Caiming Xiong. Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification. In *ICML*, 2021.

Gail Blattenberger and Frank Lad. Separating the brier score into calibration and refinement components: A graphical exposition. *The American Statistician*, 1985.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 1950.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *NeurIPS*. 2019.

Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. 2014.

M. Degroot and S. Fienberg. Assessing probability assessors: Calibration and refinement. 1981.

Morris H. Degroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 1983.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Peter A. Flach and Meelis Kull. Precision-recall-gain curves: Pr analysis done right. In *NeurIPS*, 2015.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *ICLR*, 2019.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

David Hand and Robert Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 2001. doi: 10.1023/A:1010920819831.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. 2021. URL `https://arxiv.org/abs/2104.05704`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *ICML*, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019b.

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *ICLR*, 2020.

José Hernández-Orallo, Peter Flach, and Cèsar Ferri. A unified view of performance metrics: Translating threshold choice into expected classification loss. In *JMLR*, 2012.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. In *NeurIPS*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Meelis Kull, Telmo M. Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 2017.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, 2019.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, 2019.

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

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

Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*, 1995.

Y. Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

Charles Ling, Jin Huang, and Harry Zhang. Auc: a statistically consistent and more discriminating measure than accuracy. *IJCAI*, 2003.

S. Lu, B. Nott, A. Olson, A. Todeschini, H. Vahabi, Y. Carmon, and L. Schmidt. Harder or different?a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2020.

Xingchen Ma and Matthew B. Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In *ICML*, 2021.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *ICML*, 2020.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. 2020.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *NeurIPS*, 2019.

Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology (1962-1982)*, 1973.

Allan H. Murphy and Robert L. Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1977.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning, 2020.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR, Workshop Track Proceedings*, 2017.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, 1999.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018. `https://arxiv.org/abs/1806.00451`.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, pp. 5389–5400, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Aditya Singh and Alessandro Bay. [Re] Improved Calibration and Predictive Uncertainty for Deep Neural Networks. *ReScience C*, 2020. URL `https://zenodo.org/record/3818605/files/article.pdf`.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL `https://arxiv.org/abs/2007.00644`.

S. Thulasidasan, Gopinath Chennupati, J. Bilmes, Tanmoy Bhattacharya, and S. Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, 2019.

Ben Van Calster, David J. McLernon, Maarten van Smeden, Laure Wynants, Ewout W. Steyerberg, Patrick Bossuyt, Gary S. Collins, Petra Macaskill, Karel G. M. Moons, Ben Van Calster, Maarten van Smeden, and Andrew J. Vickers. Calibration: the achilles heel of predictive analytics. *BMC Medicine*, 2019.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *ICML*, 2020.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *NeurIPS*, 2019.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

A Appendix

A.1 Natural Language Task

Dataset	Meth.	Acc	Brier (↓)	ECE (↓)	AUROC (↑)
	Baseline	73.31	36.60	17.92	83.95
20News	LS	73.96	36.37	4.79	82.71
	FL	70.74	39.59	8.67	83.46

Table 4: Calibration and refinement on 20NewsGroup.

Classification performance on NewsGroup-20 dataset(Lang, 1995) for baseline, label smoothing and focal loss. The pre-trained models were obtained from the official repository of focal loss calibration.
We observe that unsurprisingly calibration approaches reduce ECE over the baseline. However, baseline achieves higher refinement performance than the other two calibration approaches.

A.2 Calibration and Refinement for Transformer Based Networks

We utilize CCT and CVT networks as proposed by Hassani et al. (2021) in their recent work. These networks don;t require excess pre-training data to obtain comparable accuracy to popular feed-forward convolution only architectures. As the underlying architecture is significantly different from the baselines considered from our work, we still try to compare calibration and refinement of these models with a comparable baseline (in-terms of accuracy).

A.2.1 Results

The results don't indicate that transformers produce calibrated outputs. However, we did observe that for majority of the bins while computing ECE, the accuracy > confidence. This indicates towards the problem of under-confidence.

	CIFAR-10			CIFAR-100		
	Accuracy(↑)	ECE(↓)	AUROC(↑)	Accuracy(↑)	ECE(↓)	AUROC(↑)
R-50(Baseline)	95.65	2.69	93.8	77.2	12.7	85.69
CCT6_3	95.29	7.88	88.83	77.31	5.69	84.53
VGG-16(Baseline)	93.74	4.8	90.9	-	-	-
CVT6	92.58	6.76	88.39	-	-	-
VGG-16(Baseline)	–	–	–	72.46	16.29	84.97
CVT7	–	–	–	73.01	4.23	85.94

A.3 Calibration by Refinement

In this section we present the results of the refinement approach of Corbière et al. (2019). ConfidNet (CFN) learns as a post-processing step a point-estimate for new predictions. The pre-trained classification branch drives the classification of an input sample, and for estimating the confidence for the prediction, the estimate from the confidence branch is employed.
The authors highlight the refinement advantage over baseline and TrustScore Jiang et al. (2018) by employing AUPR, AUROC, etc. We utilize the official source code and train VGG-16 Simonyan & Zisserman (2015) with batch normalization. We retain 10% of training data to validate CFN training parameters and report the calibration and refinement results on the official test split for CIFARs Krizhevsky (2009). The results are reported over 3 independent runs of the experiment.

A.3.1 Result

Results in Table 5 show the CFN performance in comparison to an uncalibrated and unrefined baseline. Not only does CFN provide better refinement, it is also able to reduce the calibration errors over the datasets. This provides further support to our understanding of calibrating a model by improving refinement.

	CIFAR-100		CIFAR-10	
	ECE(↓)	AUROC(↑)	ECE(↓)	AUROC(↑)
Baseline	19.12 ± 0.13	85.18 ± 0.21	5.38 ± 0.15	92.5 ± 0.01
CFN	$\mathbf{13.95 \pm 2.7}$	$\mathbf{86.0 \pm 0.18}$	$\mathbf{4.1 \pm 0.2}$	$\mathbf{92.55 \pm 0.1}$

Table 5: Calibration and refinement results aggregated over 3 runs. Values in bold font indicates the best value w.r.t the corresponding metric.

A.4 Implementation Details

For CIFARs, we train the models for 300 epochs with a starting learning rate of 0.1 decayed by a factor of 5 (baseline, ERL, Mixup) or 10 (LS, FL) at 150 and 225 epochs. For calibration approaches many of the respective hyper-parameters are borrowed from the original work. For TS we use the temperature of 1.5. For MX, we use $\alpha = 0.2$ based on the results provided by (Thulasidasan et al., 2019; Singh & Bay, 2020). For LS, we use $\epsilon = 0.05$ following the work of Müller et al. (2019) and Mukhoti et al. (2020). We employ the fixed gamma variant for FL with $\gamma = 3.0$. The strength of the entropy regularizer in ERL is set to 0.1 based on the experiments of Thulasidasan et al. (2019).

For ImageNet, the total number of epochs is 100 with learning rate decay by 10 at milestones $30, 60, 90$. This is the standard approach for training Resnet-50 on ImageNet. For the method specific hyper-parameters we rely on existing experiments and their logical extensions. For LS, we use $\epsilon = 0.1$ as utilized by Müller et al. (2019) and Thulasidasan et al. (2019). For FL, we rely on using $\gamma = 3.0$ as the authors utilized it for experiments on the Tiny-ImageNet (Le & Yang, 2015) dataset. For ERL, we use the strength to be 0.1 based on the experiments of Thulasidasan et al. (2019). We found that for TS the temperature 1.1 provides reasonably well calibration. For MX, we employ $\alpha = 0.2$.

We report ECE and Brier score as calibration errors whereas, AUROC for refinement. All values provided are ×100. We report mean and std. deviation over 3 trials where applicable. We report the accuracies in the supplementary document as we found them to be highly similar across different methods.

We utilize publicly available datasets and code implementations for majority of our experiments. We use PyTorch Paszke et al. (2019) as the deep learning framework. Github links for the approaches investigated are provided below:

1. Mixup Calibration (MX): `https://github.com/paganpasta/OnMixup`
2. Focal Loss Calibration (FL): `https://github.com/torrvision/focal_calibration`
3. ConfidNet (CFN): `https://github.com/valeoai/ConfidNet`

The remaining approaches can be easily implemented. We provide short python scripts describing their implementation below:

Listing 1: Entropy Regularization(ERL)

```
from torch.nn import functional as F
def erlloss(logits, targets, eps=0.1, **kwargs):
    h_c = F.cross_entropy(logits, targets, reduction='sum')
    h_p = torch.sum(torch.sum(-F.softmax(logits,dim=1) * F.log_softmax(logits,dim=1),1))
    return h_c - eps*h_p
```

Listing 2: Label Smoothing(LS)

```
import torch.nn.functional as F
import torch.nn as nn
def linear_combination(x, y, epsilon):
    return epsilon * x + (1 - epsilon) * y
def reduce_loss(loss, reduction='sum'):
    return loss.mean() if reduction == 'mean' else loss.sum() if reduction == 'sum' else loss
class LabelSmoothingLoss(nn.Module):
    def __init__(self, epsilon = 0.1, reduction='sum'):
        super().__init__()
        self.epsilon = epsilon
        self.reduction = reduction
    def forward(self, preds, target):
        n = preds.size()[-1]
        log_preds = F.log_softmax(preds, dim=-1)
        loss = reduce_loss(-log_preds.sum(dim=-1), self.reduction)
        nll = F.nll_loss(log_preds, target, reduction=self.reduction)
        return linear_combination(loss / n, nll, self.epsilon)
```

Lastly, temperature scaling (TS) requires dividing the output **logits** by the chosen temperature.

We plan to release the pre-trained models to assist future research for all the methods after the review period.