FedLAS: Feature-Modulated Bidirectional Label Smoothing for Neural Network Calibration

Thiru Thillai Nadarasar Bahavan[1], Sachith Seneviratne[1], and Saman Halgamuge[1]

The University of Melbourne, Parkville, Australia
bahavant@student.unimelb.edu.au,
{sachith.seneviratne,saman.halgamuge}@unimelb.edu.au

Abstract. Deep Neural Network (DNN) classifiers suffer from poor calibration when their softmax outputs (predictive confidence) deviate from the empirical likelihoods. This manifests itself as either overconfident incorrect predictions or under-confident correct predictions. Label smoothing (LS) enhances model calibration by introducing entropy regularization during training through redistributing probability mass from the ground-truth label to the remaining classes. LS, including Margin-based LS (MbLS), have restrictive assumptions: they rely on predefined, uniform smoothing rules and only tackle overconfidence. In reality, samples exhibit diverse characteristics, such as difficulty/ambiguity, that interact with the evolving nature of the model being trained. In training, samples may have various degrees of under- or overconfidence. To overcome this, a mechanism that identifies the specific confidence state of each sample and determines the appropriate degree of smoothing in each training step is needed, tailoring the adjustment to the individual sample. We propose FeDLaS: Feature-Modulated Bidirectional Label Smoothing, a plug-and-play algorithm for label smoothing-based losses. In FeDLaS, we introduce a Feature Norm-based Confidence Indicator (NCI) to control smoothing and a Bidirectional Calibration Gating (BCG) module to detect both over and under-confidence. Our algorithm can be integrated with LS and MbLS based losses when applied to standard DNNs, enhancing performance. Extensive experiments on standard and fine-grained high-resolution vision benchmarks show that FeDLaS consistently improves calibration compared to modern baselines, reducing Expected Calibration Error (ECE) and Adaptive ECE while maintaining Top-1 accuracy. Code: github.com/nadarasarbahavan/FEDLAS

Keywords: Network Calibration · Uncertainty Estimation · Label Smoothing

1 Introduction

Reliable confidence estimation is fundamental to decision-making; *e.g.*, in clinical diagnostics, it determines whether the prediction of a DNN classifier can be trusted for treatment or whether the case must be escalated to a human

specialist. A DNN is considered well-calibrated when its predicted confidence estimates align with the true empirical likelihoods. However, a significant issue arises with many over-parameterized DNN-based classifiers: they are often poorly calibrated [10], meaning their predicted probabilities fail to accurately reflect the true likelihood of correctness. For example, if the clinical diagnostics classifier assigns 70% confidence to a set of predictions, ideally 70% of those predictions should be accurate.

In standard DNN training, networks maximize the likelihood of training labels. The use of one-hot encoded labels (Fig. 1(a)) encourages the models to assign near-certain probabilities to the correct class [10], forcing the predictions to closely match the ground truth. This leads to overfitting and causes models to exhibit overconfidence, even when predictions are incorrect [10]. Label smoothing (LS) improves model calibration by introducing entropy regularization during training by redistributing the probability mass from the ground-truth label to the remaining classes (Fig. 1(b)). LS, including Margin-based LS (MbLS) have restrictive assumptions: they rely on predefined, uniform smoothing rules and only tackle overconfidence [22, 28]. In reality, samples exhibit diverse characteristics, such as difficulty/ambiguity, that interact with the evolving nature of the model being trained. In training, samples may have various degrees of under- or overconfidence. To overcome this, *a mechanism that identifies the specific confidence state of each sample and determines the appropriate degree of smoothing in each training step is needed, tailoring the adjustment to the individual sample.*

Park et al. [33] mathematically demonstrated that many advanced network calibration methods are implicitly sample-adaptive variants of label smoothing that rely on softmax output to determine the degree of smoothing [4, 12, 18, 21, 22, 26, 35]. The authors claim that these methods have inherent structural limitations acknowledged by [33], *e.g.*, *Lim1*: these methods do not correct calibration across the complete range of predictive confidence. However, three more critical challenges (*Lim2-Lim4*) remain unaddressed: *Lim2:* Most of these methods remain constrained by a unidirectional focus on overconfidence, overlooking scenarios where samples exhibit under-confidence. Some methods account for under-confidence, but they use a validation dataset [9]. *Lim3:* They do not accommodate the non-stationarity of the model when training. *Lim4:* They use softmax outputs that are naturally prone to be overconfident. Motivated by these four limitations, we address the following two research questions: 1. During training, how can the direction of miscalibration (under-confidence vs. overconfidence) be identified for each sample, without the use of auxiliary data? 2. Beyond the model output, can we leverage the models' internal behavior to address model miscalibration? Our contributions are as follows.

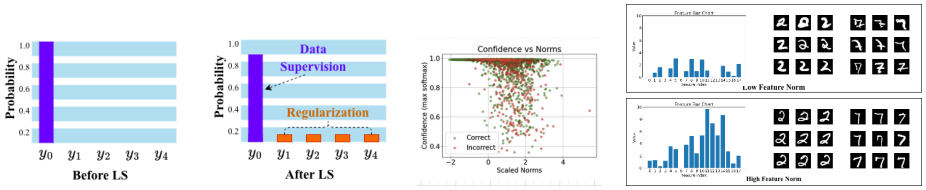
- **Norm-based Confidence Indicator:** To address *Lim3* and *Lim4*, we introduce NCI, a label-free indicator leveraging hidden-layer feature norms. Feature norms are an established class-agnostic proxy of predictive confidence [1, 16, 34, 41], and theoretically the confidence of the network's hidden classifier [34]; we ask *whether they can also correct network miscalibration.*

NCI bypasses softmax overconfidence and normalizes these norms against an Exponentially Moving Average (EMA) reference to handle non-stationarity.
- **Bidirectional Calibration Gating (BCG):** To address *Lim2*, we propose the BCG module, a dynamic mechanism designed to detect and mitigate both overconfidence and under-confidence pointwise throughout the training process, without a validation dataset.
- **Adaptive Smoothing Module (ASM):** By combining NCI and BCG, we develop the ASM, which adjusts label smoothing per-sample across the entire spectrum of predictive confidence, addressing *Lim1*. We demonstrate how to integrate it into Vanilla and Margin-based Label Smoothing.
- **Extensive Empirical Validation:** We conduct comprehensive experiments across a diverse set of benchmarks, including CIFAR-10/100, Tiny-ImageNet, two high-resolution fine-grained classification tasks. Our results consistently demonstrate superior performance in both accuracy and model calibration.

2 Related Work

Post hoc approaches Post-processing methods offer a simple and efficient way to calibrate predictions by transforming the output of a model [10, 40, 45]. Temperature scaling [10], a streamlined variant of Platt scaling [36], uses a single parameter to soften pre-softmax logits. Although effective in-domain, it performs poorly under distributional shift [32]. **Probabilistic and non-probabilistic methods.** Probabilistic methods include Bayesian neural networks [3, 23], stochastic expectation propagation [14], and dropout-based variational inference [8]. Ensemble learning, a popular non-parametric alternative, estimates uncertainty via the empirical variance of predictions and improves calibration by aggregating outputs from diverse models. Common strategies involve varying hyperparameters [43], random seeds and data shuffling [19], Monte Carlo Dropout [8], modeling dataset shift [32], and enforcing model orthogonality [20]. Despite their effectiveness, ensembles are computationally expensive, particularly with large models and datasets.

Explicit and implicit penalties. These methods introduce entropy-based penalties during training to regularize predictions and mitigate overconfidence. ECP [35] and label smoothing [28] explicitly maximize entropy by penalizing overconfident outputs. Focal loss [21] implicitly increases entropy by minimizing the KL divergence between softmax outputs and a uniform distribution [27]. As shown in [27, 28, 38], both act as implicit regularizers that promote uncertainty. On a different track, focal loss based methods have been put forth to handle both over and under-confidence: such as Dual Focal Loss and Adaptive Focal Loss. AdaFocal uses a validation set, which is limiting and DFL applies regularization to limited range of predictive confidence. MbLS relaxes the entropy constraint [22], and ACLS [33] builds on MbLS by adapting it to overconfidence [33]. Overall, loss functions that encourage higher prediction entropy, explicitly or implicitly, has been shown to achieve state-of-the-art calibration performance [27, 28, 33].



(a) One-hot Label **(b)** LS Label **(c)** L_1 vs $\max(p_i)$ **(d)** Low vs High L_1

Fig. 1: (a)-(b) Labels before and after Label Smoothing. (c) For each sample, softmax probability values (X-axis) tends to saturate towards very high values, with little variation. In contrast, L_1 captures more of the variation. (d) The features reflect the activation tendencies of the easy-to-learn and hard-to-learn images. Spurious difficult images have a flatter features, resulting in a low L_1 values. The activation tendency can thus be summarized via its L_1 norm.

Other Methods Tao introduced feature clipping as a method for applying an entropy-based penalty [39]. Earlier works, such as [44], have investigated leveraging features to group similar samples in pursuit of multi-calibration objectives [13]. However none of these explored it for adaptive label smoothing. There are Mixup based methods such as RankMixup, however this involves significant data augmentation [31].

Feature Norms Recently, various variants of feature norms has been shown (both empirically and theoretically) to differentiate via ordering between in-distribution and out-of-distribution samples [1,34,41], low quality and high quality images [16] and data with high and low uncertainty for metric learning [37].

3 Preliminaries & Problem Formulation

Setup. The training dataset $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ consists of N input-label pairs. Each input $\mathbf{x}^{(i)} \in \mathcal{X}$ is associated with a ground truth label $y^{(i)}$ from a label space $\mathcal{Y}$ consisting of $\{1, ..., C\}$ classes. The label $y^{(i)}$ is encoded as a one-hot vector $\mathbf{y}^{(i)}$. Specifically, for a sample i with true class label j, the label is defined as $\mathbf{y}^{(i)} = \{y_k^{(i)}\}_{k=1}^{C}$, where $y_j^{(i)} = 1$ and $y_k^{(i)} = 0$ for all $k \neq j$. We define a neural network classifier $f_\omega(\mathbf{x})$ trained on S that maps an input to a probability distribution $\hat{\mathbf{p}}(\mathbf{x}) \in \Delta^{C-1}$. The model is typically trained by minimizing the *cross-entropy (CE) loss* $\mathcal{L}_{\text{CE}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\text{CE}}^{i}$, where $\mathcal{L}_{\text{CE}}^{i} = -\sum_{k=1}^{C} y_k^{(i)} \log \hat{p}_k^{(i)}$.

Problem statement. From the predicted distribution, the classifier provides a prediction index $i = \arg\max_{k \in \{1,\cdots,C\}} \hat{p_k}^{(i)}$, with an associated **predictive confidence** $\hat{p}_c = \max_{k \in \{1,\cdots,C\}} \hat{p_k}^{(i)}$. A model is **perfectly calibrated** if this confidence $\hat{p}_c$ represents the true likelihood of correctness p. Formally, a model is perfectly calibrated if:

$$P(\hat{y} = y \mid \hat{p}_c = p) = p, \quad \forall p \in [0, 1] \tag{1}$$

The miscalibration manifests itself as a gap between the predicted $\hat{p}_c$ and p, specifically as overconfidence $P(\hat{y} = y \mid \hat{p}_c = p) < p$ or under-confidence $P(\hat{y} = y \mid \hat{p}_c = p) > p$. The primary goal of *calibration methods* is to minimize this disparity. This disparity is measured via Expected Calibration Error (ECE) and Adaptive Expected Calibration Error (AECE) [22, 29, 30].

Motivation for Feature norm as Confidence proxy. Following previous work, we quantify the predictive confidence of the model by examining its internal behavior, using the L_1 norm of the d-dimensional feature vector $\mathbf{z}^{(i)} = F_{backbone}(\mathbf{x}^{(i)})$. Generally, L_p norms are defined as $\|\mathbf{z}^{(i)}\|_p = \left(\sum_{j=1}^{d} |z_j^{(i)}|^p\right)^{1/p}$ for $p \geq 1$.

To provide a theoretical basis for this approach, we draw upon the findings of Park et al. [34]. They demonstrate that for networks utilizing unit-wise rectifier activation functions (*e.g.*, ReLU or GeLU), the L_1 norm of a hidden layer's feature vector acts as the maximum logit of an implicit *hidden classifier* [34]. The logit $s(x)$ is the pre-softmax output of the classifier. For notational consistency with [34], let activations in layer (l) be denoted as $\mathbf{a}^{(l)}$, which corresponds to our extracted feature vector $\mathbf{z}^i$, and let logits be $\psi(x) := s(x)$.

Theorem (Park et al., 2023) [34]. *The final logits of the model are represented by the linear transformation: $\psi(\mathbf{x}) = \mathbf{C}^{(l)}\mathbf{a}^{(l)}$, where $\mathbf{C}^l$ is the derived coefficient matrix and $\mathbf{a}^{(l)}$ represents the activations at layer l. A binarized hidden classifier $\overline{\psi}^l \in \mathbb{R}^C$ can be derived such that $\overline{\psi}^l(\mathbf{x}) := sign(\mathbf{C}^l)\mathbf{a}^{(l)} = \mathbf{B}^l\mathbf{a}^{(l)}$. Under sufficient discriminative training for class j, the feature norm $\|\mathbf{a}^l\|_1$ converges to the hidden confidence $\overline{\psi}_j^l(\mathbf{x})$ (Logit of class j). Specifically, for any class j:*

$$0 \leq \|\mathbf{a}^{(l)}\|_1 - \overline{\psi}_j^l(\mathbf{x}) \leq \|\mathbf{a}^{(l)}\|_\infty \|\text{sign}(\mathbf{a}^{(l)}) - \mathbf{b}_j^l\|_1 \tag{2}$$

where $\text{sign}(x) = 1$ if $x > 0$ and -1 otherwise; $\mathbf{b}_j^l$ is the binary weight corresponding to class j. Because classifier outputs are class-dependent and normalized relative to other classes, confidence scores are not comparable between different classes and do not provide a globally consistent confidence signal. In contrast, the feature norm reflects the model's overall discriminative strength and activation magnitude. Because this property relies on internal activation patterns rather than specific class semantics, the feature norm serves as a *class-agnostic* indicator of confidence [34]. We empirically validate this relationship by visualizing internal activation tendencies. As illustrated in Fig. 1(d), these characteristic norms act as a concise summary of the activation tendencies within the hidden layers of the network. For example, for the MNIST dataset, an easily identifiable image, Fig. 1(d), shows stronger activations, leading to a high $L1$ norm. In contrast, low-quality images show weaker activations Fig. 1(d), leading to a low $L1$ norm. In addition, in Fig. 1(c), we plot a scatter plot of the normalized L_1 norm (X-axis) and the maximum softmax probability (Y-axis). We make the following observations: (1) Softmax exhibits a saturating tendency, masking fine-grained information; and (2) L_1 has a wider dynamic range, which captures subtle variations. Furthermore, even when L_1 is low, its corresponding softmax confidence

is saturated. In Sec. 4.1, we detail the engineering solution required to adapt this L_1 for sample-specific label smoothing.

4 Method

Label Smoothing (LS) and Margin-based Label Smoothing (MbLS) can be viewed as a cross-entropy-based classification objective augmented with an entropy-penalty regularization term [22]. The trade off between these two components is fixed. We introduce a sample-dependent coefficient $\alpha^{(i)}$ to create a convex objective that adaptively modulates the balance between classification and regularization. This coefficient is calculated from detached features and logits, and is constrained within $[\alpha_{\min}, \alpha_{\max}]$ to prevent the vanishing of the classification signal or the regularization effect during training. **Model Decomposition**. The Neural network classifier can be decomposed as: $f_\omega(\mathbf{x}) = \sigma(F_{\text{head}}\left(F_{\text{backbone}}(\mathbf{x})\right))$ as shown in Fig. 2(a) [15]. Here, the feature extractor $F_{\text{backbone}}(\mathbf{x}^{(i)}) = \mathbf{z}^{(i)} \in \mathbb{R}^d$ maps the input to a d-dimensional feature space. The classification head $F_{\text{head}}(\mathbf{z}^{(i)}) = \mathbf{s}^{(i)} \in \mathbb{R}^C$ maps the features to the class logits $\mathbf{s}^{(i)} = \{s_k^{(i)}\}_{k=1}^C$. These logits are normalized via the *softmax function* $\sigma(\cdot)$ to produce the predicted probability distribution $\hat{\mathbf{p}}^{(i)} = \{\hat{p}_k^{(i)}\}_{k=1}^C$.

Vanilla Label Smoothing. Following the definitions in Sec. 3, Label Smoothing modifies the original one-hot labels $\mathbf{y^{(i)}} \rightarrow \tilde{\mathbf{y}}^{(\mathbf{i})}$, with a smoothing parameter $\alpha \in [0, 1]$ such that each element is modified as $\tilde{y}_k^{(i)} = (1-\alpha) \cdot y_k^{(i)} + \frac{\alpha}{C}$. By applying cross-entropy and rearranging the terms, we get:

$$\mathcal{L}_{\text{LS}}^i = -\sum_{k=1}^{C} \tilde{y}_k^{(i)} \log \hat{p}_k^{(i)} = (1-\alpha) \underbrace{(-\sum_{k=1}^{C} y_k^{(i)} \log \hat{p}_k^{(i)})}_{\text{Classification Loss}} + \alpha \underbrace{(-\sum_{k=1}^{C} \frac{1}{C} \log \hat{p}_k^{(i)})}_{\text{Regularization Loss } E(\hat{\mathbf{p}}^{(i)})}, \tag{3}$$

The final form of the label-smoothed cross-entropy loss, as shown in Eq. (3), offers a clear interpretation: it is a weighted sum of the cross-entropy-based classification loss with the original one-hot labels and a Kullback-Leibler (KL) divergence term $E(\hat{\mathbf{p}}^{(i)})$ with an additive constant [22]. This KL term encourages the predictive distribution to approach the uniform distribution $\mathbf{u} = [\frac{1}{C} ... \frac{1}{C}]$ over all C classes. The smoothing parameter α thus controls the trade-off between fitting the ground truth and distributing the probability mass more broadly, thus regularizing the confidence of the model. To address the limitations of a static parameter, we propose FeDLaS-LS, which extends LS by introducing an adaptive smoothing factor $\alpha^{(i)}$.

$$\mathcal{L}_{FeDLaS-LS}^i = (1-\alpha^{(i)})\mathcal{L}_{\text{CE}}^i + \alpha^{(i)} E(\hat{\mathbf{p}}^{(i)}) \tag{4}$$

Margin based LS (MbLS). In *MbLS* [22], the authors characterize the network calibration through the lens of constrained optimization. Within this framework, regularization is viewed as a constraint enforced on *logit gaps*. These gaps are formulated as follows:

$$\mathbf{d}(\mathbf{s}^{(i)}) = \left[\max_{l \in \{1,\cdots,C\}} (s_l^{(i)}) - s_1^{(i)}, \ldots, \max_{l \in \{1,\cdots,C\}} (s_l^{(i)}) - s_C^{(i)} \right] \tag{5}$$

This perspective reveals that standard calibration methods [21, 28, 35], can be reinterpreted as optimization problems subject to a soft equality constraint $\mathbf{d}(\mathbf{s}^{(i)}) = \mathbf{0}$. By maximizing entropy, these methods implicitly drive all logit gaps toward zero. However, such a strict equality constraint is too restrictive as it constantly pushes the model toward a state of maximum uncertainty [22]. *MbLS* relaxes this constraint by replacing the equality constraint $\mathbf{d}(\mathbf{s}^{(i)}) = \mathbf{0}$ with a margin based set-constraint: $\mathbf{d}(\mathbf{s}^{(i)}) \leq m \cdot \mathbf{1}_C$. As demonstrated in [33], this formulation can also theoretically be interpreted as *conditional label smoothing*; specifically, the logit $s_k^{(i)}$ is regularized only if the logit gap $\max_l(s_l^{(i)}) - s_k^{(i)}$ exceeds the predefined margin m.

$$\mathcal{L}_{\text{MbLS}} = \mathcal{L}_{\text{CE}}^i + \lambda \cdot \sum_{k=1}^{C} \max\left(0, \max_{l \in \{1,\cdots,C\}} (s_l^{(i)}) - s_k^{(i)} - m\right) := \mathcal{L}_{\text{CE}}^i + \lambda \cdot R(\mathbf{s^{(i)}}) \tag{6}$$

However, the regularization strength remains *non-adaptive* [22]. A critical weakness identified in [33] is that the standard additive MbLS penalty is unlikely to regularize the training label of the true class directly. Often, the true logit is the max-logit, therefore, it cancels out: $\max_{l \in \{1,\cdots,C\}}(s_l^{(i)}) - s_k^{(i)} = 0$, which can limit its ability to mitigate overconfidence. Our FeDLaS-MbLS approach employs $\alpha^{(i)}$ as a unified balancing parameter in a convex combination, $\mathcal{L}_{FeDLaS-MbLS}^{(i)} = (1 - \alpha^{(i)})\mathcal{L}_{\text{CE}}^i + \alpha^{(i)} R(\mathbf{s}^{(i)})$, while *MbLS* [22] scales the regularization term in isolation without such an interpolation.

4.1 Adaptive Smoothing Module (ASM) $f(.)$.

This plug-in module $f(.)$ modulates the smoothing parameter α per sample, so that for each sample, the smoothing parameter for the corresponding loss is modulated with respect to its features and logits as $\alpha^{(i)} = \alpha \cdot f(\cdot)$ for FeDLaS-LS or $\alpha^{(i)} = \lambda \cdot f(\cdot)$ for FeDLaS-MbLS. α and λ refer to the original hyperparameters of LS and MBLS.

$$\alpha^{(i)} = \alpha f(s^{(i)}, z^{(i)}) = \alpha \cdot 2\phi\left(\beta \cdot \psi\left(sg[s^{(i)}]\right) \cdot \gamma\left(sg[z^{(i)}]\right)\right) \tag{7}$$

where $\phi(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, $sg[\cdot]$ is the gradient detaching function. $\gamma(z^{(i)}) \in \mathbb{R}$ is a Norm-based Confidence Indicator (NCI). $\psi(s^{(i)}) \in \{-1, 1\}$ is the Bidirectional Calibrating Gate (BCG) Module. β is a tunable hyperparameter (selected on the validation set) which controls the sensitivity of the smoothing coefficient to changes in the input. Our design choice of sigmoid is influenced

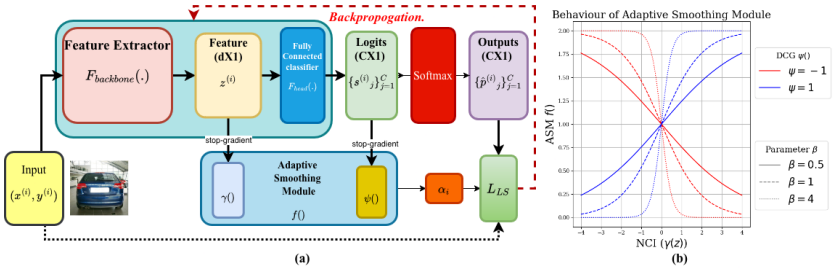


Fig. 2: (a) High-Level Architecture and Data flow of FeDLaS. A standard feature extractor, such as a ResNet($F_{backbone}(\cdot)$), is connected to a fully connected classifier($F_{head}(\cdot)$). Our proposed Adaptive Smoothing Module (ASM) dynamically adjusts the regularization for each sample by leveraging the detached features/logits extracted by the backbone network. (b) The behavior of Adaptive Smoothing Module (ASM) $f(.)$. The intensity of the scaling is controlled by the *Norm-based Confidence Indicator (NCI)* $\gamma(z)$. This estimates prediction confidence from the norm of the intermediate feature representation $z^{(i)}$. The polarity of scaling, if its outputs are monotonically increasing (blue) /decreasing (red) for the NCI signal is determined by *Bidirectional Calibrating Gate (BCG)* $\psi\,(s)$. This compensates for systematic overconfidence or under-confidence for each sample.

by various properties that we wanted to preserve. (1) **Boundedness**: The module scales the base hyperparameter $\alpha^{(i)}$ within the range $(0, 2\alpha)$(reverting to the baseline method when $\beta = 0$). This protects the system from outliers. (2) **Monotonicity**: The sigmoid function is monotonic. In each operating mode, the smoothing factor is scaled proportionally to the NCI (3) **Symmetric**: Essentially, our model has two operating modes: positive and negative. It is determined by the BCG module. Both modes are rotationally symmetric.

Bidirectional Calibration Gating Module $\psi(\cdot)$ (BCG Module). The BCG module acts as a discrete binary selector, $\psi(\mathbf{s}^{(i)}) \in \{-1, 1\}$, determining the smoothing regime for each sample:

$$\psi(\mathbf{s}^{(i)}) = \begin{cases} 1, & \text{if } \textit{Positive Mode} \text{ (Mitigating Overconfidence)} \\ -1, & \text{if } \textit{Negative Mode} \text{ (Counteracting Underconfidence)} \end{cases} \tag{8}$$

In *Positive Mode*, the relationship between the norm-based confidence indicator $\gamma(\cdot)$ and the smoothing parameter $\alpha^{(i)}$ is monotonically increasing ($\frac{d\alpha^{(i)}}{d\gamma^{(i)}} > 0$; Fig. 2 (b) (Blue)), provides an "up-regulation" that prevents pathological overconfidence. Conversely, in *Negative Mode*, the relationship is monotonically decreasing ($\frac{d\alpha^{(i)}}{d\gamma^{(i)}} < 0$; Fig. 2 (b) (Red)), allowing for a "down-regulation" that boosts the model's confidence where it is otherwise lacking.

To determine the optimal regime, we employ a lightweight gating network $F_{gate} : \mathbb{R}^C \to \mathbb{R}^2$ that processes logits detached from the gradient graph sg$[\mathbf{s}^{(i)}]$.

To maintain end-to-end differentiability despite the discrete nature of $\psi(\cdot)$, we utilize a **Straight-Through Estimator (STE)** [2]. The gating logic is formulated as follows:

$$\psi(\mathbf{s}^{(i)}) = \text{sgn}\left(\mathbf{w}_{\text{gate}}^{\top} \cdot \sigma\left(\mathcal{F}_{gate}(\text{sg}[\mathbf{s}^{(i)}])\right)\right), \quad \text{with } \mathbf{w}_{\text{gate}} = [1, -1]^{\top} \tag{9}$$

where $\text{sgn}(x) = \{1 \text{ if } x > 0, -1 \text{ otherwise}\}$, $\mathcal{F}_{gate}$ represents a Multi-Layer Perceptron (MLP), and σ denotes the softmax function. This formulation effectively maps the predicted state to a bipolar directional signal.

Norm-based Confidence Indicator $\gamma(\cdot)$ (NCI) To quantify sample confidence, we define a *Norm-based Confidence Indicator (NCI)*, denoted as $\gamma(\cdot)$. Instead of using the raw feature norms, we normalize it with Exponential Moving Averages. The reasons for this are as follows: (1) Without mean-variance normalization, fluctuations in norm variance cause the modulation signal to frequently enter saturation regions of the sigmoid function in the ASM module, leading to "dead" adaptation. ; (2) By using an exponentially moving baseline, normalization ensures that the smoothing correction for each sample is approximately determined by its relative position within the evolving population of training points, rather than within the current batch. The NCI for a sample i is calculated as:

$$\gamma(z^{(i)}) = \frac{\|sg[z^{(i)}]\|_1 - \mu_z^k}{\sigma_z^k} \tag{10}$$

where $\|\cdot\|_1$ represents the L_1 norm and $sg[\cdot]$ denotes the stop-gradient operation [2]. We use $sg[\cdot]$ to ensure that the normalization statistics serve purely as a measurement tool and do not influence the underlying feature optimization during backpropagation. The mean (μ_z^k) and the standard deviation (σ_z^k) are tracked using EMA:

$$\mu_z^k = \theta\widehat{\mu}_z^k + (1-\theta)\mu_z^{k-1} \tag{11}$$

$$\sigma_z^k = \theta\widehat{\sigma}_z^k + (1-\theta)\sigma_z^{k-1} \tag{12}$$

In these updates, $\widehat{\mu}_z^k$ and $\widehat{\sigma}_z^k$ represent the statistics of the current batch k, while μ and σ represent the global moving statistics. The momentum parameter θ controls the influence of the current batch on the moving averages: a higher $\theta \in (0, 1)$ allows the averages to adapt quickly to recent statistics, while a lower θ provides greater stability by weighting past averages more heavily. Furthermore, $\gamma(\cdot)$ can be instantiated using the maximum softmax probability/maximum logit (This is ablated).

5 Experiments

To thoroughly validate the effectiveness of our proposed method, we conducted extensive experiments in multiple classification regimes. For the standard classification, the results are summarized in Tab. 1. We further evaluate our approach

on high-resolution *fine-grained classification*, a challenging task that requires distinguishing between subordinate categories within a single entry-level class (*e.g.*, specific bird species or aircraft variants). Unlike standard recognition, fine-grained tasks are characterized by subtle differences between classes and high intra-class variance; performance on these high-resolution benchmarks is reported in Tab. 2.

Datasets. For standard classification, we utilize the widely-used CIFAR-10, CIFAR-100 [17], and Tiny-ImageNet [6] benchmarks. For fine-grained visual classification (FGVC), we employ the CUB-200-2011 [42] and FGVC-Aircraft [24].

Architectures. Following [33], for standard image classification (CIFAR/Tiny-ImageNet), we train *ResNet-50* and *ResNet-101* [11] architectures from scratch. For fine-grained tasks, we use the ImageNet-1K pre-trained *ResNet-101* (following [22]) and, beyond prior work, the ImageNet-1K pre-trained *ViT-B/16* [7].

Experimental Setup. All experiments were conducted on a single NVIDIA A100 (40GB) GPU running Linux. To ensure a strictly fair comparison, our training protocol, including the optimizer, scheduler, and data augmentations, perfectly replicate the standardized setup established by ACLS [33]. Detailed training protocols, image pre-processing steps, and augmentation strategies are elaborated in the Appendix.

Metrics. Classification performance is quantified using *Top-1 Accuracy*. Model calibration is assessed primarily via the *Expected Calibration Error (ECE)* and *the Adaptive Expected Calibration Error (AECE)* [22, 29, 30]. Following [33], all calibration metrics are calculated using the $M = 15$ bins. AECE is more robust than ECE due to its adaptive binning strategy [30].

Implementation details of the compared baselines. As our approach is a training-time calibration method based on entropy regularization, we compare it against a broad spectrum of established training-time techniques with implicit/explicit entropy penalties. We include several contemporary methods such as MMCE [18], MDCA [12], CPC [4], and CRL [26].

We also consider the original Focal Loss (FL) [21] and its extensions: Sample Dependent Focal Loss (FLSD) [27], AdaFocal [9], and Dual Focal Loss (DFL) [38]. The hyperparameters for these methods align with those specified in their respective original publications. Furthermore, we evaluated against standard Label Smoothing (LS) with a fixed factor $\alpha = 0.05$ (Exception of $\alpha = 0.1$ for CIFAR100), as well as Margin-Based Label Smoothing (MbLS) [22] and its extension, ACLS [33]. For these, we follow the original configurations: $m = 6, \lambda = 0.1$ for CIFAR-10/100 and $m = 10, \lambda = 0.05$ for Tiny-ImageNet, CUB-2011, and FGVC-Aircraft. Within this comparison, our primary state-of-the-art (SOTA) baselines are *AdaFocal [9], Dual Focal Loss [38], and ACLS [33].* We include Temperature Scaling [10] as a representative post-hoc baseline, but we do not compare against other post-hoc methods because our approach is an in-training objective that actively shapes the model during learning, rather than an adjustment applied to a fixed, trained model.

Implementation details of the proposed method. We integrate our proposed plug-in, hereafter referred to as FeDLaS, into two distinct baseline losses:

Table 1: Results on the Standard Image Recognition/Calibration test splits. We report the Top-1 Accuracy, ECE, AECE. Calibration metrics calculated with 15 bins. Results follow the unified setup of [33]; values are from [33], except rows marked †, which we run as means over three fixed seeds. The best-performing result in each category is indicated in **bold**, second-best is underlined. The final three columns show the average accuracy across datasets/backbones (expanded in Supplemental) and the mean rank of each method for ECE and AECE across all tested scenarios

	Tiny-ImageNet [6]				CIFAR10 [17]				CIFAR100 [17]				Average (Dataset)		
Arch.	ResNet50		ResNet101		ResNet50		ResNet101		ResNet50		ResNet101		Value	Rank	Rank
Metrics	ECE↓	AECE↓	ECE↓	AECE↓	ECE↓	AECE↓	ECE↓	AECE↓	ECE↓	AECE↓	ECE↓	AECE↓	Acc.↑	ECE↓	AECE↓
Training-time Methods															
CE [25]	3.73	3.69	4.97	4.97	5.85	5.84	5.74	5.73	13.59	13.54	12.94	12.94	78.74	14.00	14.00
MMCE [18]	5.15	5.12	4.88	4.88	3.10	3.10	3.61	3.61	12.72	12.71	13.43	13.42	79.37	11.50	11.00
ECP [35]	4.00	3.92	4.68	4.66	3.01	2.99	5.41	5.40	12.29	12.28	13.43	13.42	78.63	11.50	11.00
MDCA [12]	2.77	2.61	6.06	6.06	6.86	6.73	6.88	7.23	12.68	12.65	13.61	13.59	79.40	14.17	14.17
CPC [4]	3.12	3.05	3.90	4.00	3.91	3.91	3.78	3.75	13.29	13.28	13.32	13.28	79.52	12.00	11.50
CRL [26]	1.65	1.52	3.57	3.56	3.14	3.11	3.74	3.73	6.30	6.26	7.29	7.14	79.39	8.50	7.50
FLSD-53 [27]	2.91	2.95	4.91	4.91	3.84	3.60	4.58	4.57	4.23	4.21	5.33	5.26	78.36	8.83	8.67
FL [21]	2.96	3.12	2.55	2.44	3.90	3.86	4.60	4.58	4.81	4.79	5.13	5.14	78.26	8.67	8.83
DFL [38] †	6.50	6.50	4.78	4.77	2.46	2.39	2.12	2.15	4.55	4.38	5.05	5.10	78.89	7.50	6.33
AdaFocal [9] †	5.82	5.80	5.43	5.42	4.9	4.81	3.52	3.29	4.39	4.39	5.31	5.22	78.77	10.00	9.83
LS [28] †	2.56	2.58	2.04	1.96	3.59	4.09	3.07	4.18	5.42	5.72	6.08	8.87	79.69	7.00	9.00
FeDLaS-LS(Ours) †	1.77	1.65	2.56	2.39	1.29	**2.30**	1.56	**1.92**	4.29	4.20	7.48	7.27	79.62	5.50	4.17
MbLS [22] †	1.45	1.47	1.81	1.58	1.21	3.34	1.38	3.25	5.00	5.52	6.10	10.07	79.66	4.33	6.00
ACLS [33] †	1.37	1.36	1.66	1.65	2.32	2.99	2.22	2.88	7.05	7.32	7.09	7.11	**79.78**	5.50	4.83
FeDLaS-MbLS(Ours) †	**1.22**	**1.11**	**1.55**	**1.39**	**1.12**	3.20	**1.27**	3.58	4.42	4.48	4.51	4.42	79.58	**1.83**	**3.83**
Post-hoc Methods															
CE+ TS [10]	1.63	1.52	2.08	2.03	3.68	3.67	3.62	3.62	**2.93**	**2.86**	**3.06**	**2.80**	78.74	5.00	4.83

standard Label Smoothing (LS) and Margin-based Label Smoothing (MbLS) [22]. This results in two configurations: FeDLaS-LS and FeDLaS-MbLS. For the FeDLaS-LS and FeDLaS-MbLS configurations, all original LS and MbLS hyper-parameters are kept unchanged for each dataset to ensure a controlled evaluation. To compute the required feature norms, we utilize the pre-classifier feature vectors for ResNet architectures and the global class token (CLS) for Vision Transformers. Our method introduces a dataset-specific hyperparameter, β, and a batch-size-specific momentum smoothing parameter, θ. The optimal values for these were determined through the validation set. Specifically, we found $\beta = 4.0$ to be effective for the standalone FeDLaS-LS baselines in all datasets, while $\beta = 0.5$ performed optimally for the FeDLaS-MbLS configuration in CIFAR-10 and Tiny-ImageNet, and $\beta = 1$ performed optimally for the FeDLaS-MbLS configuration in CIFAR-100. The momentum parameter θ was set to 0.95 for CIFAR-10 and 0.5 for Tiny-ImageNet and CIFAR-100. For the standard classification benchmarks, the hyperparameters were tuned directly in the validation sets. However, since fine-grained datasets typically lack a formal validation split and have a smaller training dataset size, we followed the protocol established in [22] and adopted the optimal hyper-parameters identified for Tiny-ImageNet for all experiments involving CUB-2011 and FGVC-Aircraft.

Table 2: Results on the Fine-Grained High-Resolution Classification test splits [42]. We report the Top-1 Accuracy, ECE, and AECE. Calibration metrics are calculated with 15 bins. The best-performing result in each category is indicated in **bold**, second-best is underlined. Following the same evaluation protocol in [22], all relevant hyper-parameters were selected on a held-out proxy dataset (Tiny-ImageNet) and kept fixed across all benchmarks. All methods use a single fixed seed due to compute cost.

Arch.	FGVC-Aircraft [24]						CUB-2011-Birds [42]						Average		
Metrics	ResNet-101			ViT-B/16 @384			ResNet-101			ViT-B/16 @384			Value	Rank	Rank
	Acc.↑	ECE↓	AECE↓	Acc.↑	ECE↓	AECE↓	Acc.↑	ECE↓	AECE↓	Acc.↑	ECE↓	AECE↓	Acc.↑	ECE↓	AECE↓
CE [25]	87.42	4.79	4.72	68.01	25.62	25.61	70.83	12.12	12.12	65.62	26.90	26.89	72.97	8.75	8.75
MMCE [18]	88.44	3.88	3.87	67.23	25.35	25.34	72.55	8.82	8.71	76.18	17.27	17.26	76.10	7.75	7.75
DFL [38]	86.79	2.80	2.39	64.41	24.67	24.67	71.34	**2.75**	**2.74**	74.93	15.97	15.91	74.36	4.75	4.25
AdaFocal [9]	88.02	9.31	9.08	67.71	17.76	17.75	71.24	7.63	7.63	74.64	11.90	11.90	75.40	5.50	5.50
ACLS [33]	87.93	3.51	3.67	73.02	22.40	22.40	**72.93**	7.11	7.06	**83.58**	11.82	11.81	**79.36**	5.00	5.00
LS [28]	88.39	3.01	2.61	69.21	20.53	20.44	71.72	3.51	3.65	82.74	5.36	6.43	78.01	3.00	3.00
FeDLaS-LS(Ours)	88.50	**1.53**	**1.44**	73.17	**12.38**	**12.11**	72.67	3.62	3.39	82.33	**2.85**	**2.74**	79.16	**1.50**	**1.25**
MbLS [22]	88.35	2.47	2.80	**73.29**	21.89	21.86	72.45	5.62	5.38	82.49	12.40	12.42	79.14	4.50	5.00
FeDLaS-MbLS(Ours)	**88.71**	2.16	2.55	72.00	23.99	23.98	72.73	4.41	4.20	83.32	12.03	12.03	79.19	4.25	4.50

6 Results, Discussion and Ablation Studies

FeDLaS is an Effective Plug-in for LS and MbLS. To analyze the results, we calculate the mean rank for each of the metrics (AECE, ECE) across all dataset-backbone combinations for each method for both standard classification datasets (Tab. 1) and fine-grained high-resolution classification datasets Tab. 2. These ranks, displayed in the final columns of Tab. 1 and Tab. 2, provide a holistic view of performance beyond individual data set wins. In standard benchmarks, FeDLaS-MbLS has the best (lowest) rank among all compared methods for both calibration metrics. In particular, FeDLaS-MbLS shows a significant ranking boost over its base method, MbLS, suggesting that our proposed modifications effectively bridge the calibration gap found in margin-based losses. Similarly, FeDLaS-LS consistently outperforms label smoothing (LS), maintaining a lower mean rank. The advantages of our approach are even more pronounced in the fine-grained setting (Tab. 2). The winner across settings depends entirely on whether a specific task is fundamentally better suited for a LS or MbLS. Based on the results in Tab. 1,Tab. 2, as a rule of thumb, we recommend FeDLaS-MbLS for standard classification and FeDLaS-LS for fine-grained classification.

Balance of Accuracy and Calibration. A perennial challenge in model calibration is the potential degradation of Top-1 accuracy [10]. To evaluate this trade-off, we calculate the mean Top-1 accuracy across all data set-backbone combinations for standard (Tab. 1) and fine-grained high-resolution datasets (Tab. 2) (Extended results are available in the Supplementary Material). Standard Classification: As shown in Tab. 1, the addition of FeDLaS does not compromise predictive performance. Both variants FeDLaS-LS (79.62% vs. 79.69%) and FeDLaS-MbLS (79.58% vs. 79.66%) maintain high accuracies competitive to their baselines (LS, MbLS). Notably, our methods outperforms most existing

Table 3: OOD detection performance (AUROC %) Comparison across CIFAR benchmarks for standard and augmented objectives. Our FeDLaS plug-in increases the OOD discriminative power of the base objective, closing the gap with Cross-entropy. Additional baselines are provided for reference. Uses trained models from Tab. 1.

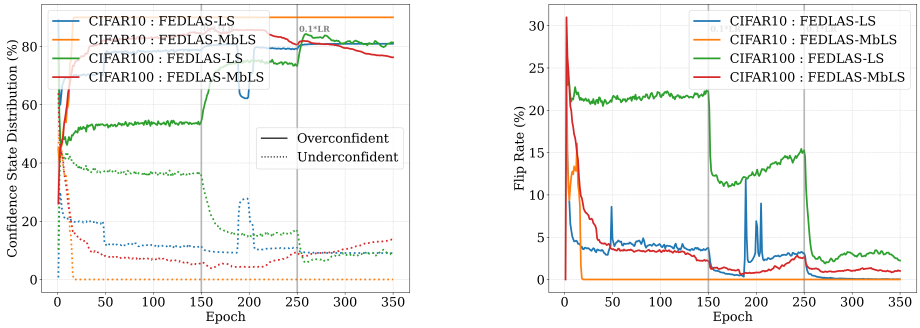
Datasets			Baselines					LS-based		MbLS-based	
Training	**Testing**	**Metric**	CE	TS	DFL	AdaFocal	ACLS	LS	**FeDLaS-LS**	MbLS	**FeDLaS-MbLS**
C10	SVHN	AUROC↑	91.37	92.20	93.87	96.37	89.14	74.23	**85.04**	89.90	**91.40**
	C100	AUROC↑	87.82	88.00	86.43	87.30	84.25	75.20	**77.20**	77.20	**84.15**
C100	SVHN	AUROC↑	79.47	79.77	81.85	83.57	75.09	72.92	**84.33**	73.88	**78.60**
	C10	AUROC↑	74.82	74.84	77.60	76.87	71.76	72.33	**73.71**	73.19	**73.72**

calibration techniques; for instance, FeDLaS variants (79.58%–79.62%) consistently exceed the accuracy range of Focal Loss-based methods [9, 21, 38], which typically fall between 78.26% and 78.89%. This trend is even more pronounced in the fine-grained setting (Tab. 2). Our methods (ranging from 79.16% to 79.19%) significantly outperform Focal Loss variants (74.36%–75.40%), while showing a slight improvement over the base LS (78.01%) and MbLS (79.14%) models. These results demonstrate that FeDLaS not only enhances calibration but also serves as a robust objective for maintaining, or even slightly improving, Top-1 accuracy in complex classification tasks.

FeDLaS Improves OOD Detection. Tab. 3 compares Out-of-Distribution (OOD) [5] detection performance (AUROC %) across standard CIFAR benchmarks. We observe that both static LS and MbLS consistently degrade OOD discriminative power compared to the Cross-Entropy (CE) baseline across all training-testing configurations, indicating a clear trade-off between calibration and OOD separation. In particular, the integration of our FeDLaS plug-in consistently narrows this performance gap.

BCG is Stable We track each sample's state by its index across training epochs. Using the BCG module's output, we record whether a sample is in an overconfident or under-confident state. In Fig. 3a, we plot the proportion of samples in each state across the entire dataset per epoch. The proportions remain steady, with shifts occurring mainly when the learning-rate scheduler triggers (epochs 150 and 250), indicating stability. In Fig. 3b, following the same methodology, we introduce the Flip Rate (FR), which measures how frequently a sample switches state between consecutive epochs on average. The FR decreases steadily, indicating that the gating stabilizes and converges.

BCG Rectifies Both Underconfidence and Overconfidence We decompose ECE into overconfident ECE (O-ECE) and underconfident ECE (U-ECE) and report it in Tab. 4. We compare our FeDLaS variants against their base losses (LS, MbLS) and the Cross-Entropy (CE) baseline on standard benchmarks. The results reveal a clear trade-off. CE is strongly overconfident, so while



(a) Confidence-state distribution **(b)** Flip Rate (FR)

Fig. 3: Stability of the BCG module. (a) Proportion of samples classified as overconfident vs. under-confident across training epochs. The proportions stay steady, shifting only at learning-rate scheduler steps (epochs 150, 250). (b) Flip Rate (FR), the average per-epoch rate at which samples switch state. The steady decline shows the gating converges and stabilizes over training.

Table 4: Decomposed calibration error (ECE %) Comparison across standard benchmarks on the decomposed ECE (overconfident (O-ECE) and underconfident (U-ECE)). Our FeDLaS plug-in reduces calibration error in *both* directions, whereas the base objectives lower O-ECE only by inflating U-ECE.

Dataset	Metric	CE	LS	**FeDLaS-LS**	MbLS	**FeDLaS-MbLS**
C10	O-ECE↓	5.85	1.16	**0.89**	1.09	**1.11**
	U-ECE↓	0.00	2.45	**0.40**	0.12	**0.02**
C100	O-ECE↓	13.59	2.73	**3.70**	2.62	**2.27**
	U-ECE↓	0.00	2.69	**0.59**	2.38	**2.15**
Tiny	O-ECE↓	3.67	0.19	**1.30**	0.64	**0.46**
	U-ECE↓	0.05	2.37	**0.48**	0.81	**0.76**

its U-ECE is near zero (it rarely makes underconfident predictions), its O-ECE is large. The base losses LS and MbLS reduce O-ECE but over-correct, shifting many predictions into underconfidence and substantially raising U-ECE. Our FeDLaS variants reduce O-ECE while keeping U-ECE low, achieving the best balance across all three benchmarks. This supports our central claim: bidirectional gating corrects miscalibration in *both* directions rather than trading one failure mode for the other.

7 Conclusions, Limitations and Future Work

FeDLaS is an adaptive label smoothing framework that improves calibration by dynamically adjusting smoothing. By addressing both under- and over-confidence, it consistently outperforms state-of-the-art methods. Furthermore, the introduction of a class-agnostic confidence indicator enables future work in unsupervised, self-supervised, semi-supervised and noisy-label calibration. Limitations-wise,

the NCI relies on the L1 feature norm as a confidence proxy, which inherits the assumptions of the underlying theory.

Acknowledgment

NB acknowledges Melbourne Graduate Research Scholarship. We would like to thank Chathura Jayasankha, Asela Hevapathige, and Zhaoyuan Qiu for providing valuable feedback. This research was supported by The University of Melbourne's Research Computing Services.

References

1. Bahavan, T.T.N., Seneviratne, S., Halgamuge, S.: Sphor: A representation learning perspective on open-set recognition for identifying unknown classes in deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings. pp. 6901–6910 (June 2026)
2. Bengio, Y., Léonard, N., Courville, A.C.: Estimating or propagating gradients through stochastic neurons for conditional computation. ArXiv **abs/1308.3432** (2013), https://api.semanticscholar.org/CorpusID:18406556
3. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on Machine Learning (ICML). p. 1613–1622. ICML'15, JMLR.org (2015)
4. Cheng, J., Vasconcelos, N.: Calibrating deep neural networks by pairwise constraints. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13699–13708 (2022). https://doi.org/10.1109/CVPR52688.2022.01334
5. De Bernardi, G., Narteni, S., Cambiaso, E., Mongelli, M.: Rule-based out-of-distribution detection. IEEE Transactions on Artificial Intelligence **5**(6), 2627–2637 (2024). https://doi.org/10.1109/TAI.2023.3323923
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009). https://doi.org/10.1109/CVPR.2009.5206848
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
8. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. p. 1050–1059. ICML'16, JMLR.org (2016)
9. Ghosh, A., Schaaf, T., Gormley, M.: Adafocal: calibration-aware adaptive focal loss. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. p. 1321–1330. ICML'17, JMLR.org (2017)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
12. Hebbalaguppe, R., Prakash, J., Madan, N., Arora, C.: A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16081–16090 (June 2022)
13. Hébert-Johnson, Ú., Kim, M.P., Reingold, O., Rothblum, G.N.: Multicalibration: Calibration for the (computationally-identifiable) masses. In: International Conference on Machine Learning (2018), https://api.semanticscholar.org/CorpusID:51880858
14. Hernández-Lobato, J.M., Adams, R.P.: Probabilistic backpropagation for scalable learning of bayesian neural networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. p. 1861–1869. ICML'15, JMLR.org (2015)
15. Jordahn, M., Olmos, P.M.: Decoupling feature extraction and classification layers for calibrated neural networks. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, JMLR.org (2024)
16. Kim, M., Jain, A.K., Liu, X.: Adaface: Quality adaptive margin for face recognition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18729–18738 (2022). https://doi.org/10.1109/CVPR52688.2022.01819
17. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
18. Kumar, A., Sarawagi, S., Jain, U.: Trainable calibration measures for neural networks from kernel mean embeddings. In: Dy, J.G., Krause, A. (eds.) ICML. Proceedings of Machine Learning Research, vol. 80, pp. 2810–2819. PMLR (2018), http://dblp.uni-trier.de/db/conf/icml/icml2018.html#KumarSJ18
19. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6405–6416. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
20. Larrazabal, A.J., Martínez, C., Dolz, J., Ferrante, E.: Orthogonal ensemble networks for biomedical image segmentation. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III. p. 594–603. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-87199-4_56, https://doi.org/10.1007/978-3-030-87199-4_56
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 318–327 (2020). https://doi.org/10.1109/TPAMI.2018.2858826
22. Liu, B., Ayed, I.B., Galdran, A., Dolz, J.: The Devil is in the Margin: Margin-based Label Smoothing for Network Calibration . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 80–88. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2022). https://doi.org/10.1109/CVPR52688.2022.00018, https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00018
23. Louizos, C., Welling, M.: Structured and efficient variational deep learning with matrix gaussian posteriors. In: Proceedings of the 33rd International Conference on Machine Learning (ICML). p. 1708–1716. JMLR.org (2016)

24. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013), https://arxiv.org/abs/1306.5151
25. Mao, A., Mohri, M., Zhong, Y.: Cross-entropy loss functions: theoretical analysis and applications. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
26. Moon, J., Kim, J., Shin, Y., Hwang, S.: Confidence-aware learning for deep neural networks. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)
27. Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P.H.S., Dokania, P.K.: Calibrating deep neural networks using focal loss. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
28. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA (2019)
29. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. p. 2901–2907. AAAI'15, AAAI Press (2015)
30. Nixon, J., Dusenberry, M.W., Zhang, L., Jerfel, G., Tran, D.: Measuring calibration in deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)
31. Noh, J., Park, H., Lee, J., Ham, B.: Rankmixup: Ranking-based mixup training for network calibration. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1358–1368 (2023). https://doi.org/10.1109/ICCV51070.2023.00131
32. Ovadia, Y., Fertig, E., Ren, J.J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J.V., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Neural Information Processing Systems (2019), https://api.semanticscholar.org/CorpusID:174803437
33. Park, H., Noh, J., Oh, Y., Baek, D., Ham, B.: Acls: Adaptive and conditional label smoothing for network calibration. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3913–3922 (2023). https://doi.org/10.1109/ICCV51070.2023.00364
34. Park, J., Chai, J.C.L., Yoon, J., Teoh, A.B.J.: Understanding the Feature Norm for Out-of-Distribution Detection. In: Proceedings - 2023 IEEE/CVF International Conference on Computer Vision, ICCV 2023. pp. 1557–1567. Institute of Electrical and Electronics Engineers Inc. (2023)
35. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.E.: Regularizing neural networks by penalizing confident output distributions. ArXiv **abs/1701.06548** (2017), https://api.semanticscholar.org/CorpusID:9545399
36. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers. pp. 61–74. MIT Press (1999)
37. Scott, T.R., Gallagher, A.C., Mozer, M.C.: von Mises–Fisher Loss: An Exploration of Embedding Geometries for Supervised Learning . In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10592–10602. IEEE Computer Society, Los Alamitos, CA, USA (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.01044, https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.01044

38. Tao, L., Dong, M., Xu, C.: Dual focal loss for calibration. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
39. Tao, L., Dong, M., Xu, C.: Feature clipping for uncertainty calibration. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 20841–20849. AAAI Press (2025). https://doi.org/10.1609/aaai.v39i19.34297, https://ojs.aaai.org/index.php/AAAI/article/view/34297
40. Tomani, C., Gruber, S., Erdem, M.E., Cremers, D., Buettner, F.: Post-hoc uncertainty calibration for domain drift scenarios. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10119–10127 (2021). https://doi.org/10.1109/CVPR46437.2021.00999
41. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Open-set recognition: a good closed-set classifier is all you need? In: International Conference on Learning Representations (2022)
42. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology (2011)
43. Wenzel, F., Snoek, J., Tran, D., Jenatton, R.: Hyperparameter ensembles for robustness and uncertainty quantification. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
44. Yang, J.Q., Zhan, D.C., Gan, L.: Beyond probability partitions: calibrating neural networks with semantic aware grouping. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
45. Zhang, J., Kailkhura, B., Han, T.Y.J.: Mix-n-match: ensemble and compositional methods for uncertainty calibration in deep learning. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)