Uncertainty Weighted Gradients for Model Calibration

Jinxu Lin[1] Linwei Tao[1] Minjing Dong[2] Chang Xu[1]
[1]The University of Sydney, [2]City University of Hong Kong
{jinxu.lin, linwei.tao, c.xu}@sydney.edu.au, minjdong@cityu.edu.hk

Abstract

Model calibration is essential for ensuring that the predictions of deep neural networks accurately reflect true probabilities in real-world classification tasks. However, deep networks often produce over-confident or under-confident predictions, leading to miscalibration. Various methods have been proposed to address this issue by designing effective loss functions for calibration, such as focal loss. In this paper, we analyze its effectiveness and provide a unified loss framework of focal loss and its variants, where we mainly attribute their superiority in model calibration to the loss weighting factor that estimates sample-wise uncertainty. Based on our analysis, existing loss functions fail to achieve optimal calibration performance due to two main issues: including misalignment during optimization and insufficient precision in uncertainty estimation. Specifically, focal loss cannot align sample uncertainty with gradient scaling and the single logit cannot indicate the uncertainty. To address these issues, we reformulate the optimization from the perspective of gradients, which focuses on uncertain samples. Meanwhile, we propose using the Brier Score as the loss weight factor, which provides a more accurate uncertainty estimation via all the logits. Extensive experiments on various models and datasets demonstrate that our method achieves state-of-the-art (SOTA) performance.[1]

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in various domains, including image classification. However, recent studies [5, 30] reveal that DNNs often suffer from mis-calibration in classification task, exhibiting over-confidence or under-confidence in their predictions. For example, a model might output a confidence score of 0.8 for a particular prediction, which does not necessarily correspond to an 80% probability of correctness.

Calibrating DNNs, which aligns the predicted confidence with the true probabilities, is therefore crucial to enhancing their reliability in practical applications, such as autonomous driving, medical imaging, and weather forecasting. To address mis-calibration issues, several methods [26, 28] have been proposed, many of which focus on modifying the loss function during training. A common approach involves adding regularization terms with the Cross-Entropy (CE) loss to improve calibration, as seen in methods like Maximum Mean Calibration Error (MMCE) [15] and Meta Calibration [1]. Additionally, Focal Loss (FL) [17], which adjusts per-sample loss weights based on prediction difficulty, has been shown to improve calibration performance. Mukhoti et al. [18] attribute the effectiveness of FL to its implicit regularization on the entropy of predicted probabilities, which mitigates overconfidence. Similarly, Dual Focal Loss (DFL) [27] incorporates the second-most-probable class in the weighting mechanism to address model under-confidence. Another research interest regarding calibration is the evaluation of calibration performance. Metrics commonly employed include Expected Calibration Error (ECE)[5] and Brier Score[2].

The focal-loss-based methods can be unified under a general loss framework expressed as $u \cdot \text{CE}$, where u is a loss weighting factor. In FL, the factor u_{FL} can be treated as an estimation for sample-wise difficulty . Interestingly, we observe that u_{FL}, originally designed for binary classification, is mathematically equivalent to the Brier Score for binary cases, which can measure the uncertainty. Furthermore, the weighting term u_{DFL} in DFL aligns with the Brier Score in three-class classification scenarios. This observation motivates the hypothesis that weighting the loss with sample calibration metrics, such as the Brier Score, could better identify uncertain samples and facilitate targeted training.

Furthermore, our investigation reveals a limitation of directly applying the weighting factor u to the loss term. Specifically, the vanilla optimization of focal loss would achieve a misalignment with the objective of scaling gradients for harder samples with larger magnitudes. Based on our analysis, we mainly attribute this issue to a differentiable loss weighting factor u, which could disrupt the positive correlation between the CE and its gradient magnitude during backpropagation, impeding the model's ability

[1]Code is available at https://github.com/Jinxu-Lin/BSCE-GRA.

to prioritize higher-uncertain samples effectively.

To address these issues, we first propose applying the weighting factor to scaling the gradient rather than the loss itself. This allows us directly aligns the gradient optimization with sample uncertainty, which ensures that harder samples receive appropriately scaled updates without disrupting the optimization process. Within this framework, we then introduce a generalized form of the Brier Score as a gradient weighting factor, which provides a more accurate estimation of sample uncertainty that considers all categories. This leads to the development of the BSCE-GRA loss function, which adjusts gradient to directly scale optimization based on sample uncertainty by Brier Score.

We conduct extensive experiments to validate the effectiveness of the proposed method, achieving state-of-the-art results across various datasets and model architectures, which demonstrate the effectiveness of our methods. The contributions of this paper are summarized as:

1. We provide a new perspective by unifying some existing loss modification techniques for model calibration under a sample-weighting framework. With this framework, we provide extensive analysis of their limitations.
2. We propose a simple yet effective optimization framework for model calibration with a gradient weighting factor, where we scale the gradients to encourage the model to focus on uncertain samples effectively.
3. We analyze the use of different uncertainty metrics within this framework and introduce BSCE-GRA, a new loss function based on the Brier Score that provides an accurate sample-wise uncertainty estimation.
4. We conduct extensive experiments under different settings to validate our proposed methods. Our uncertainty-weighted framework shows consistent effectiveness across various uncertainty metrics, and the proposed BSCE-GRA loss achieves state-of-the-art results.

2. Related Works

In recent years, numerous techniques have been proposed to address the problem of network miscalibration, which can generally be classified into three categories.

The first category is post-hoc calibration techniques, which adjust model predictions after training by optimizing additional parameters on a held-out validation set. These methods include Platt Scaling [23], which performs a linear transformation on the original prediction logits; Isotonic Regression [34], which uses piecewise functions to transform logits; Bayesian Binning into Quantiles (BBQ)[20], which extends histogram binning with Bayesian model averaging; and Beta Calibration[13], initially designed for binary classification and later generalized to multi-class settings with Dirichlet distributions by Kull et al. [14]. Temperature Scaling [5], one of the most widely used post-hoc calibration methods, tunes the temperature parameter in the SoftMax function to minimize negative log-likelihood on a held-out validation set. In this work, we report calibration performance with post-temperature scaling results.

The second category includes regularization techniques, which are known to effectively calibrate DNNs. Data augmentation methods, such as Mixup [31] and AugMix [7], train DNNs on mixed samples to mitigate overconfident predictions. Model ensemble techniques, which involve independently training multiple DNNs and averaging their predictions, have been shown to enhance both accuracy and predictive uncertainty by aggregating outputs from multiple models [16, 24, 36]. Label smoothing [19], which replaces one-hot labels with soft labels, encourages the model to make less confident predictions, thereby reducing overconfidence. Additionally, weight decay has also been demonstrated to improve confidence calibration [5].

The third category focuses on modifying the training loss to improve calibration. These methods include adding a differentiable auxiliary surrogate loss for Expected Calibration Error (ECE)[1, 10, 11] or replacing the training loss with other loss functions, such as Mean Squared Error (MSE)[9], Focal Loss [18], Inverse Focal Loss [32], and Dual Focal Loss [27]. Among these, Focal Loss [18], which adds a modulation term to the Cross-Entropy loss to focus on hard-to-classify samples, provides a simple and effective way to train calibrated models. Focal Loss and Dual Focal Loss can be categorized as margin-based losses, similar to Hinge Loss [3], Triplet Loss [25], and Margin Ranking Loss [33].

3. Preliminary

3.1. Problem Formulation

Given a K-class dataset $\mathbb{S} = \{\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(N)}\}$, where each training sample $\boldsymbol{z}^{(n)} := (x^{(n)}, y^{(n)})$ is an input-label pair. Let $\mathcal{X}$ and $\mathcal{Y}$ represent the input space and the label space, respectively. The ground truth label $y \in \mathcal{Y}$ is encoded in a one-hot vector format, where $y_i = 1$ if $i \in K$ represents the actual class. We define a classifier f_θ trained on $\mathbb{S}$ that maps an input $\boldsymbol{x} \in \mathcal{X}$ to a probability distribution $\hat{p}(\boldsymbol{x})$. The classifier then provides a prediction $k = \arg\max_{i \in K} \hat{p}_i$, which indicates the index of the predicted label. The predicted label $\hat{y} \in \mathcal{Y}$ is similarly represented as y in the one-hot encoding of prediction k. The confidence $\hat{p}_c$ is defined as the predicted probability associated with the prediction k.

A well-calibrated model ensures that the provided confidence $\hat{p}_c$ accurately reflects the true probability of correct classification. We define the true class-posterior probability vector as $\boldsymbol{\eta}(\boldsymbol{x}) = [\eta_1(\boldsymbol{x}), ..., \eta_K(\boldsymbol{x})]$, where $\eta_k(\boldsymbol{x}) = \mathbb{P}(y = k|\boldsymbol{x})$ is the true probability of class k given input $\boldsymbol{x}$. Formally, a network is considered perfectly calibrated if $\mathbb{P}(\hat{y} = y|\hat{p}_c = p) = p$ for all $p \in [0, 1]$ [5], which can be also written as $\hat{p}(\boldsymbol{x}) = \boldsymbol{\eta}(\boldsymbol{x})$. The uncertainty of a model for a sample $\boldsymbol{x}$, also referred to the calibrated error $c(\boldsymbol{x})$,

can be computed as the difference between $\boldsymbol{\eta}(\boldsymbol{x})$ and $\hat{p}(\boldsymbol{x})$:

$$c(\boldsymbol{x}) = \|\eta(\boldsymbol{x}) - \hat{p}_c(\boldsymbol{x})\|. \tag{1}$$

3.2. Metric for Calibrated Error

In practice, it is hard to access the ground truth probability $\boldsymbol{\eta}(\boldsymbol{x})$ on real-world samples and thus the calibration error cannot be directly computed by Eq. 1, some alternative methods have been proposed to evaluate the uncertainty.

ECE. The Expected Calibration Error (ECE) is defined as $\mathbb{E}_{\hat{p}_c}[|\mathbb{P}(\hat{y} = y|\hat{p}_c) - \hat{p}_c|]$. Guo et al. [5] propose an approximation of ECE. Specifically, all samples are divided into M bins $\{B_m\}_{m=1}^{M}$ of equal width based on their confidence, where each bin B_m contains all samples whose confidences fall within the range $\hat{p}_c \in [\frac{m}{M}, \frac{m+1}{M})$. For each bin B_m, the average confidence is computed as $C_m = \frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_c^{(i)}$ and the bin accuracy is computed as $A_m = \frac{1}{|B_m|}\sum_{i\in B_m} \mathbb{1}(\hat{y}_k^{(i)} = y_k^{(i)})$, where $\mathbb{1}$ is the indicator function. The ECE can then be computed as the average L_1 difference between bin accuracy and confidence:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N}|A_m - C_m|, \tag{2}$$

where N denotes the number of samples in each bin. In addition to the ECE in Eq.2, there are several variants used to measure calibration error. For example, AdaECE [21] groups samples into bins B_m with an equal number of smaples such that $|B_m| = |B_n|$ for all bins. Another variant, ClasswiesECE [14] measure the calibration error by considering each of the K classes separately.

Brier Score. Besides ECE computing the expectation of calibrated error among whole datasets, some other techniques have also been used to evaluate the sample-wise uncertainty. Accuracy and calibration are distinct concepts, and one cannot be inferred from the other unequivocally. The Brier Score (BS) [2] unify these two concepts, which is commonly used in calibration literature. It has been shown that this family of metrics can be decomposed into a calibration term and a refinement term. Achieving an optimal score requires both accurate predictions and appropriate confidence levels. For a given sample, the Brier Score is mathematically defined as the Mean Squared Error between the predicted probability distribution $\hat{p}$ and one-hot encoded ground truth label y. We introduce a generalized Brier Score (gBS) form as follows:

$$\text{gBS} = \sum_{i=1}^{K} \|\hat{p}_i(\boldsymbol{x}) - y_i\|_\beta^\gamma \tag{3}$$

where γ and β are the hyperparamters for the exponent and norm order, respectively. When $\gamma = 2$ and $\beta = 2$, this formulation reduces to the original Brier Score.

3.3. Calibrating Method

Temperature Scaling A widely used post-hoc technique for improving classification calibration is temperature scaling. It adjusts the sharpness of the output probability distribution by scaling the logits in the SoftMax function using a temperature parameter, defined as $\hat{p}_i = \frac{\exp(\hat{g}_i/T)}{\sum_{k=1}^{K}\exp(\hat{g}_k/T)}$, where $\hat{g}$ represents the logits before applying the SoftMax function, and T is the temperature that controls the scaling. Calibration performance can be enhanced by tuning T on a held-out validation set.

Focal Loss and Dual Focal Loss Some other previous works have attempted to improve model calibration by modifying the loss function. Focal Loss [17] was initially introduced to address the foreground-background imbalance problem in object detection. It addresses the issue by incorporating a loss weighting factor based on sample complexity, reducing the weight of easy samples and allowing the model to focus on harder-to-classify instances.

Formally, given the predicted probability $\hat{p}(\boldsymbol{x})$ on sample $\boldsymbol{x}$, the focal loss is defined as:

$$\mathcal{L}_{\text{FL}}(\boldsymbol{x}, y) = -\sum_{i=1}^{K} y_i(1-\hat{p}_i)^\gamma \log \hat{p}_i(\boldsymbol{x}), \tag{4}$$

where γ is a pre-defined hyperparameter. Previous studies [18] have demonstrated that optimizing models using Focal Loss results in better calibration compared to Cross-Entropy. This improvement is partly due to the entropy-based regularization effect introduced by Focal Loss, while complexity-based weighting also likely plays a role. However, we noticed that Focal Loss was initially applied to binary classification, where the complexity term $(1-\hat{p}_i)$ is related to the Brier Score, which can also be used for evaluating uncertainty. In multi-class classification, although complexity and uncertainty are not strictly equivalent, the complexity term still captures some class-level uncertainty.

Several variants of Focal Loss exist, including Dual Focal Loss [27], which incorporates the probability from the second most probable class into the scaling factor to address the under-confidence issue caused by Focal Loss:

$$\mathcal{L}_{\text{DFL}}(\boldsymbol{x}, y) = -\sum_{i=1}^{K} y_i(1-\hat{p}_i(\boldsymbol{x})+\hat{p}_j(\boldsymbol{x}))^\gamma \log \hat{p}_i(\boldsymbol{x}), \tag{5}$$

where $\hat{p}_j(x) = \max_i\{\hat{p}_i(x)|\hat{p}_i(x) < \hat{p}_{gt}(x)\}$.

4. Method

4.1. Weighting with Sample-wise Uncertainty

Focal Loss and Dual Focal Loss both support the concept that uncertainty-based weighting in conjunction with Cross Entropy can enhance model calibration. We propose a generalized loss function framework, termed Uncertainty CE

Loss, which incorporates a sample-wise uncertainty metric into Cross Entropy as a weighting factor:

$$\mathcal{L}_{\text{Uncertainty}}(\boldsymbol{x}, y) = -\sum_{i=1}^{K} u(\hat{p}_i(\boldsymbol{x})) \cdot y_i \log \hat{p}_i(\boldsymbol{x}), \quad (6)$$

where $u(\hat{p}_i)$ is the adaptive term that evaluates the uncertainty of sample $\boldsymbol{x}$. The motivation behind this design is to use the predicted calibration error for each sample to scale the loss function, thereby directing the model's optimization more effectively towards samples with higher uncertainty.

Therefore, the weights used in Focal Loss is given by:

$$u_{\text{FL}}(\hat{p}(\boldsymbol{x})) = (1 - \hat{p}_c)^{\gamma}. \quad (7)$$

And the scaling term in Dual Focal Loss is defined as:

$$u_{\text{DFL}}(\hat{p}(\boldsymbol{x})) = (1 - \hat{p}_c(x) + \hat{p}_j(x))^{\gamma}, \quad (8)$$

where $\hat{p}_j(x)$ is the second maximum confidence in prediction. The weight used in Focal Loss can be seen as derived from the Brier Score, focusing on the error of the ground true class. In contrast, the scaling term in Dual Focal Loss incorporates uncertainty information from the second most probable class. However, directly weighting loss function with certain uncertainty metrics, as those used in Focal Loss and Dual Focal Loss, presents several challenges.

Reviewing our objectives, we aim to weight the loss function value by each sample's uncertainty, resulting in greater optimization steps for those samples with higher uncertainty. Our focus actually is not on the loss value itself, but on the optimization. The purpose of uncertainty weighting is to focus the model's attention on samples with higher uncertainty. Thus, the key factor is the gradient: higher weights should lead to larger gradients, resulting in more substantial model updates for those samples. For Cross Entropy, its value is positively correlated with the gradient: a higher CE value yields a larger gradient, leading to more significant updates for optimization. When applying a scalar weight to CE, this relationship remains intact, as it does not disrupt the positive correlation.

However, for differentiable weights, directly applying them to the loss can disrupt the positive correlation between the loss and the gradient, ultimately impairing effective model optimization. To further analyze this, let us consider Focal Loss. For Focal Loss $\mathcal{L}_{\text{FL}}$ and Cross Entropy $\mathcal{L}_{\text{CE}}$, their gradients are given by: $\frac{\partial}{\partial \boldsymbol{w}}\mathcal{L}_{\text{FL}} = g(\hat{p}_i, \gamma)\frac{\partial}{\partial \boldsymbol{w}}\mathcal{L}_{\text{CE}}$, where $g(p, \gamma) = (1-p)^{\gamma} - \gamma p(1-p)^{\gamma-1}\log(p)$, γ is a predefined hyperparameter, and $\boldsymbol{w}$ represents the parameters of the final linear layer. From a gradient perspective, $g(p, \gamma)$ acts as a weight on the CE gradient, which is illustrated in Figure 1. There exists a point $p_0 \in [0, 1]$ such that within the range $[0, p_0]$, $\frac{\partial}{\partial p}g(p, \gamma) > 0$ and within the range $[p_0, 1]$, $\frac{\partial}{\partial p}g(p, \gamma) < 0$. In binary classification, $1 - \hat{p}_i$ is related to

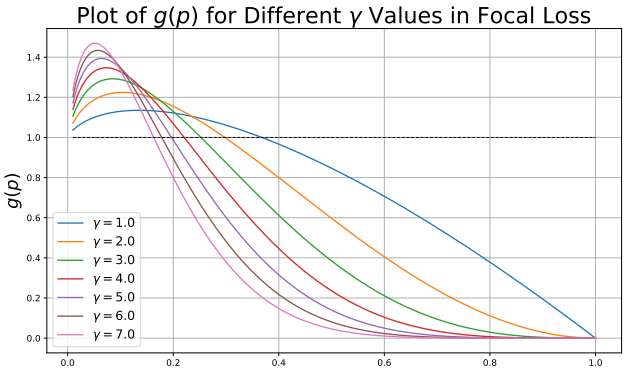


Figure 1. $g(p, \gamma)$ of Focal Loss vs predicted confidence $\hat{p}_c$.

the Brier Score that reveals calibrated error, indicating that the predicted calibrated error is not always positively correlated with the weight on gradient. As uncertainty decreases, the weight on gradient initially increases, before eventually aligning with the level of uncertainty. This causes the model to focus more on samples with moderate uncertainty rather than those with the highest uncertainty, which contradicts our original goal of emphasizing the most uncertain samples. Further, changes in uncertainty are not immediately reflected in the sample weights; instead, multiple training iterations are required for these changes to align properly. Thus, directly applying some kind of uncertainty metrics into Eq. 6 would pose issues to the uncertainty weighting.

4.2. Sample-wise Uncertainty on Gradients

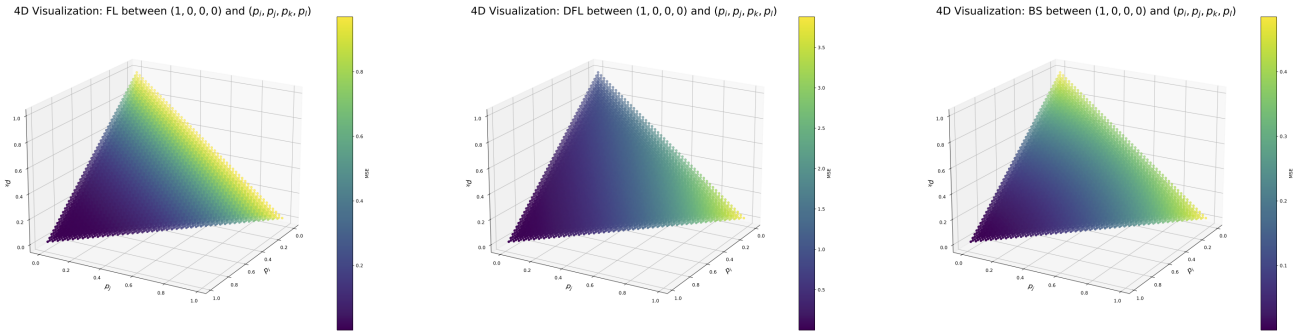
The discussion in Sec. 4.1 motivates us to ensure the weight on gradients for a sample to align with its uncertainty. This could be intuitively solved by directly applying the model uncertainty on samples as a weight of the gradient. We define the modified gradient as:

$$\frac{\partial}{\partial \theta}\mathcal{L}_{\text{Uncertainty-GRA}}(\boldsymbol{x}, y) = u(\hat{p}(\boldsymbol{x}))\frac{\partial}{\partial \theta}\mathcal{L}_{\text{CE}}(\boldsymbol{x}, y). \quad (9)$$

Taking a SGD optimizer as an example, the model update can be expressed as: $\theta_{t+1} = \theta + \alpha \cdot \frac{\partial}{\partial \theta}\mathcal{L}_{\text{Uncertainty-GRA}}(\boldsymbol{x}, y)$, where θ is the model parameters and α is the learning rate. When the predicted uncertainty on sample $\boldsymbol{x}$ is higher, the optimization step results in a larger update, encouraging the model to focus more on uncertain samples. A general form of the loss function for our uncertainty-weighted gradient framework, termed Uncertainty-GRA CE Loss, is defined as:

$$\mathcal{L}_{\text{Uncertainty-GRA}}(\boldsymbol{x}, y) = -\int \sum_{i=1}^{K} u(\hat{p}(\boldsymbol{x})) \cdot \frac{y_i}{\hat{p}_i(\boldsymbol{x})} \mathrm{d}\hat{p}(\boldsymbol{x}), \quad (10)$$

In practice, to compute the $\mathcal{L}_{\text{Uncertainty-GRA}}$, we can detach the gradient of $u(\hat{p}(\boldsymbol{x}))$ and multiply it with the Cross Entropy, instead of calculating the gradient integration.



(a) An illustration of the u_{FL}. (b) An illustration of the u_{DFL}. (c) An illustration of the u_{BS}.

Figure 2. An illustration of value of gradient weight function on a 4 class classification. It is obvious that u_{FL} varies only along the p_i axis and u_{DFL} changes along the p_i and p_j axes. u_{BS} responds to changes across all axes, providing a more complete uncertainty evaluation.

4.3. Sample-wise Uncertainty Metrics

With Eq. 10, calibration can be achieved by employing the sample's ground-truth uncertainty as the scaling term. However, accessing the ground-truth uncertainty for real-world image samples is impractical. Therefore, alternative metrics are required to estimate the sample-wise uncertainty $u(\boldsymbol{x})$.

One approach to evaluate sample-wise uncertainty is based on computing the generalized Brier Score (gBS):

$$u_{\text{gBS}}(\hat{p}(\boldsymbol{x})) = \sum_{i=1}^{K} \|\hat{p}_i(\boldsymbol{x}) - y_i\|_\beta^\gamma. \tag{11}$$

We provide a brief discussion on the effectiveness of the generalized Brier Score (gBS) as a measure of calibration error. When using the Brier Score (BS) to evaluate calibration error, with $\beta = 2$ and $\gamma = 2$, the difference between the expected and predicted calibration error is given by:

$$\begin{aligned} c(\boldsymbol{x}) - u_{\text{BS}}(\hat{p}(\boldsymbol{x})) &= \mathbb{E}_{y\sim\eta(x)}[\|\hat{p}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_2^2 - \|\hat{p}(\boldsymbol{x}) - y\|_2^2] \\ &= \sum_{k=1}^{K} \boldsymbol{\eta}_k(\boldsymbol{x})(\boldsymbol{\eta}_k(\boldsymbol{x}) - 1). \end{aligned} \tag{12}$$

Thus, the term $\sum_{k=1}^{K} \boldsymbol{\eta}_k(\boldsymbol{x})(\boldsymbol{\eta}_k(\boldsymbol{x}) - 1)$ depends solely on $\boldsymbol{\eta}(\boldsymbol{x})$ which is fixed for a given sample $\boldsymbol{x}$. Consequently, the error remains constant for a specific sample $\boldsymbol{x}$. We provide a detailed computation of Eq. 12 in Appendix.

Several variants of $u_{\text{gBS}}(\hat{p})$ have demonstrated their effectiveness in previous studies. When $\beta = 1$ and only the actual class is considered, the generalized Brier Score can be interpreted as the scaling term in Focal Loss in Eq. 7. When $\beta = 1$ and both the actual class and the maximum predicted class (excluding the actual class) are considered, the generalized Brier Score can be interpreted as the scaling term in Dual Focal Loss in Eq. 8.

However, these two metrics only consider several classes in the predicted probability . In the case of Focal Loss, which was originally designed for binary classification, considering one-dimensional uncertainty is sufficient due to the constraint that the sum of probabilities must equal 1. When extended to multi-class classification, these metrics are not accurate enough to evaluate uncertainty.

We visualize the gradient weights u_{FL}, u_{DFL} and u_{BS} in Figure 2 for a 4-class scenario. The three axes represent three dimensions of the predicted probability, while the fourth is implied by the probability sum constraint. From the figure, it is evident that u_{FL} and u_{DFL} are sensitive only to changes in one or two dimensions. However, different points in the coordinate system have different uncertainties. They fail to adequately capture uncertainty change across all dimensions. But the u_{BS} can accurately respond to changes along any coordinate axis.

To further evaluate whether these metrics accurately measure the ground truth uncertainty of samples, we conducted experiments on a toy dataset. The toy dataset consists of 5 two-dimensional Gaussian distributions, representing 5 groups of data: $\mathcal{N}(\mu_i, \Sigma), i \in \{0, ..., 5\}$. The mean vectors μ_i were randomly sampled from the range $[-10, 10]$ and $\Sigma = \boldsymbol{I}$ was used as the shared covariance matrix for all groups. We generated 10,000 data points from each group to form the training dataset, which was used to train a two layer CNN model for 5 epochs. An additional 1,000 samples from each distribution were used to create the test dataset. Therefore, the ground truth probability $\eta(\boldsymbol{x})$ can be computed using the probability density function (PDF) of each distribution:

$$\eta(\boldsymbol{x}) = \frac{p^n(\boldsymbol{x})}{\sum_i^N p^i(\boldsymbol{x})}, \tag{13}$$

where $N = 5$ in this case, $p^n(\boldsymbol{x})$ and $p^i(\boldsymbol{x})$ are the PDF of corresponds class for $\boldsymbol{x}$ and remaining classes, respectively. The ground truth uncertainty of each sample is calculated by Eq. 1, representing the model's ground truth uncertainty for each sample, along with the sample-wise uncertainty metrics u_{FL}, u_{DFL} and u_{gBS}. To validate the accuracy

Dataset	Model	CE		BL		MMCE		FLSD		DFL		BSCE		BSCE-GRA	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T
CIFAR10	ResNet50	4.36	1.32	4.29	1.50	4.48	1.41	1.26	1.15	1.00	1.00	0.88	0.88	**0.74**	**0.74**
	ResNet110	4.7	1.56	4.48	1.66	4.80	1.29	1.81	1.17	1.01	1.01	0.99	0.99	**0.87**	**0.87**
	WideResNet	3.35	**0.94**	2.86	1.10	3.65	1.28	1.84	1.04	3.32	1.16	1.7	0.95	**1.46**	1.12
	DenseNet	4.64	1.46	3.96	1.46	4.81	1.67	1.37	1.17	0.87	**0.77**	1.01	1.01	**0.87**	0.87
CIFAR100	ResNet50	18.05	3.05	7.87	4.27	15.87	3.32	5.53	2.57	2.54	2.56	1.90	1.90	**1.59**	**1.59**
	ResNet110	18.84	4.63	16.77	4.30	18.65	3.93	6.88	3.71	3.47	3.47	2.75	2.75	**2.53**	**2.53**
	WideResNet	14.81	3.27	7.74	4.46	14.58	2.99	2.70	2.71	5.45	2.52	2.63	**2.42**	**2.46**	2.46
	DenseNet	19.1	3.43	8.13	2.99	17.56	2.87	3.38	1.30	4.68	1.83	1.63	1.62	**1.62**	**1.61**
TinyImageNet	ResNet50	14.94	5.16	7.81	1.47	14.58	2.99	**2.18**	2.18	6.71	2.28	4.0	1.76	4.57	**1.47**

Table 1. **Comparison of Calibration Methods Using ECE Across Various Datasets and Models.** ECE values are reported using 15 bins, with the best-performing method for each dataset-model combination highlighted in bold. Results are averaged over three runs with different random seeds.

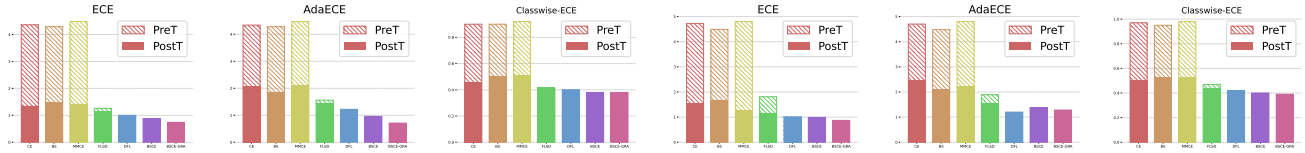


Figure 3. Comparison of different ECE metrics. The first three plots show the uncertainty for CIFAR-10 using ResNet-50, while the remaining plots represent ResNet-110 on CIFAR-10.

of these metrics, we used Pearson correlations between the calibrated error and the uncertainty metrics to verify their positive correlation among 5 runs. We use grid-search to find the optimal hyper-parameter for each methods. Among the methods, gBS achieves the highest Pearson correlation coefficient of 0.664, indicating the strongest linear relationship between the predicted values and the true targets. DFL follows with a correlation of 0.638, while FL has the lowest correlation at 0.550. This suggests that gBS provides the most accurate predicted uncertainty alignment with ground-truth uncertainty, followed by DFL and FL.

Based on the results, we conclude that although ground-truth uncertainty is inaccessible for real-world datasets, these alternative metrics provide a reliable means of estimating uncertainty, making them suitable for use during training. Besides, the metric u_{gBS} has the best performance compared to the uncertainty metric used in Focal Loss and Dual Focal Loss. We incorporate u_{gBS} into the gradient-weighted framework discussed in Section 4.2 and introduce a new loss function called BSCE-GRA. This loss function uses the generalized Brier Score as an adaptive uncertainty metric to weight the Cross Entropy gradients, defined as:

$$\mathcal{L}_{\text{BSCE-GRA}}(\boldsymbol{x}, y) = -\int \sum_{i=1}^{K} u_{\text{gBS}}(\hat{p}(\boldsymbol{x})) \cdot \frac{y_i}{\hat{p}_i(\boldsymbol{x})} \mathrm{d}\hat{p}(\boldsymbol{x}). \quad (14)$$

To further validate the effectiveness of proposed method, we provide a comprehensive theoretical evidence that optimizing with BSCE-GRA, the K-class predicted probability q would equal the actual class-posterior probability η, thereby preventing over/under-confidence when convergence. Due to page limitation, the proof is provided in Appendix.

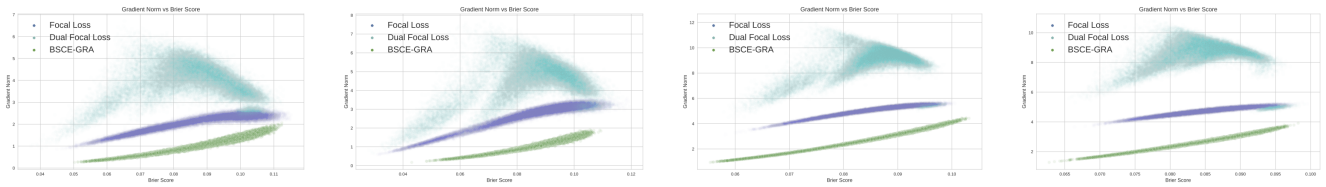
Besides, the Uncertainty-GRA Loss makes it possible to use non-differentiable uncertainty metrics for model calibration. However, it requires the weights to capture sample-wise uncertainty, whereas some existing measurements like ECE can only compute the uncertainty for a group of samples, making them unsuitable for the proposed framework.

5. Experiment

We evaluate our methods on multiple deep neural networks (DNNs), including ResNet50, ResNet110 [6], WideResNet [35], and DenseNet [8]. Our experiments are conducted on CIFAR-10, CIFAR-100 [12], and Tiny-ImageNet [4] to assess calibration performance. Further details about the datasets can be found in the appendix.

Baselines. We compare our methods, BSCE, BSCE-GRA, and ECE-CE, with multiple existing approaches, including training with Cross Entropy (CE), Brier Loss (BL)[2], MMCE Loss[15], Focal Loss with Adaptive Exponent (FLSD)[18], and Dual Focal Loss[27]. For Focal Loss, we employ the FLSD-53 strategy [18] to adaptively adjust the gamma value sample-wise, setting $\gamma_{\text{FL}} = 5$ for $\hat{p}_c \in [0, 0.2)$ and $\gamma_{\text{FL}} = 3$ for $\hat{p}_c \in [0.2, 1)$. For Dual Focal Loss, the gamma value is set to 5, as reported in the original work.

Training Setup. Our training setup follows prior works [18, 27]. We train CIFAR-10 and CIFAR-100 for 350



(a) Gradient norm distribution at epoch 50

(b) Gradient norm distribution at epoch 150

(c) Gradient norm distribution at epoch 250

(d) Gradient norm distribution at epoch 350

Figure 4. Evolution of gradient norm distributions across different training epochs for various loss functions. The scatter plots show the relationship between gradient norm and Brier Score for different loss functions (Focal Loss, Dual Focal Loss, BSCE-GRA).

Metrics	FLSD		FLSD-GRA		DFL		DFL-GRA		BSCE		BSCE-GRA	
	pre T	post T	pre T	post T	pre T	post T	pre T	post T	pre T	post T	pre T	post T
Acc	**95.04%**	**95.04%**	94.72%	94.72%	94.63%	94.63%	94.76%	94.76%	95.03%	95.03%	94.69%	94.69%
ECE	1.26	1.15	0.88	0.88	1.00	1.00	0.93	0.84	0.88	0.88	**0.74**	**0.74**
Ada ECE	1.56	1.45	1.19	1.19	1.22	1.22	0.77	0.81	0.96	0.96	**0.71**	**0.71**

Table 2. Comparison of weighting different uncertainty metrics on gradient or loss function, including u_{FL} and u_{FL}. The results validate the effectiveness of the gradient-weighting strategy among different uncertainty metrics.

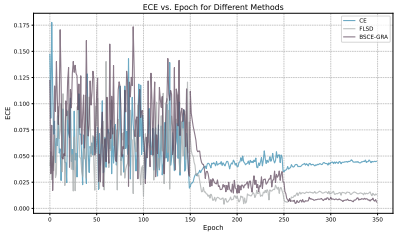
epochs, using 5,000 images from the training set for validation. The learning rate is initially set to 0.1 for the first 150 epochs, then reduced to 0.01 for the next 100 epochs, and further reduced to 0.001 for the remaining epochs. For Tiny-ImageNet, we train for 100 epochs, with the learning rate set to 0.1 for the first 40 epochs, 0.01 for the next 20 epochs, and 0.001 for the remaining epochs. All experiments are conducted using SGD with a weight decay of 5×10^{-4} and a momentum of 0.9. The training and testing batch sizes for all datasets are set to 128. We re-run all baseline methods using three different random seeds (1, 42, and 71), and report the average results. All experiments are performed on a single Nvidia 4090 GPU. For temperature scaling, the temperature parameter T is optimized through a grid search with $T \in [0, 0.1, 0.2, ..., 10]$ on the validation set, selecting the value that yields the best post-temperature-scaling Expected Calibration Error (ECE). The same optimized temperature parameter is applied to other metrics, such as AdaECE. Further details on the datasets and additional experimental setup are provided in the appendix.

5.1. Calibration Performance

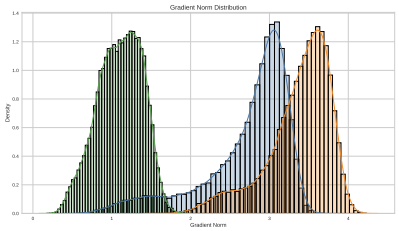
We report the average ECE before and after temperature scaling among three random seeds, along with the corresponding optimal temperatures, in Table 1. BSCE-GRA achieves state-of-the-art ECE performance in most cases, particularly in the pre-temperature-scaling results. Notably, the fact that most of the optimal temperatures for BSCE-GRA are found to be 1 indicates that BSCE-GRA trains an inherently calibrated model, capable of achieving strong calibration performance without the need for additional temperature scaling. This is a crucial advantage for developing accurate and reliable models that are efficient and require minimal post-processing. The results on CIFAR-10 generally show better calibration performance compared to datasets with more labels (e.g., CIFAR-100 and Tiny-ImageNet) across multiple models. Regarding network architecture, ResNet-50 demonstrates the best calibration performance among the four DNNs tested (ResNet-50, ResNet-110, Wide-ResNet-26-10, and DenseNet-121) on both CIFAR-10 and CIFAR-100 datasets.

Different Metrics. The methods are further evaluated using several widely-accepted metrics to assess calibration performance across models, including Adaptive ECE and Classwise-ECE. Adaptive ECE measures the expected calibration error while accounting for the distribution of the data, whereas Classwise-ECE is a variant that evaluates calibration error for each class individually. Figure 3 presents the results of multiple methods using ResNet-50 and ResNet-110 on the CIFAR-10 dataset. The figure demonstrates that BSCE-GRA is the only method that achieves both inherently calibrated models and state-of-the-art performance across various metrics. Additional results are presented in the appendix, providing further evidence of the effectiveness of our method in model calibration.

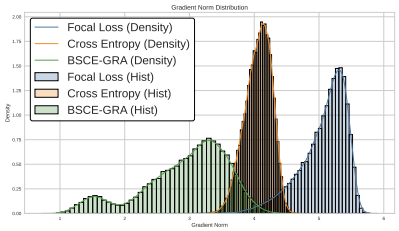
Calibration over Training. Figure 5a presents the ECE on the test set for models trained with Focal Loss and Cross-Entropy loss over the entire training period on CIFAR-10 using ResNet-50. To improve visualization, the ECE values are smoothed using an exponential moving average. The figure suggests that, after the initial warm-up epochs, where predicted probabilities are unstable, the ECE of models trained with BSCE-GRA and Focal Loss consistently remains lower compared to models trained with



(a) ECE over epochs using CE, Focal Loss and BSCE-GRA.



(b) Gradient Density at epoch 150 using CE, Focal Loss and BSCE-GRA.



(c) Gradient Density at epoch 250 using CE, Focal Loss and BSCE-GRA.

Figure 5. Figure 5a presents the evolution of ECE throughout the training process, demonstrating that our method rapidly converges to the best result by epoch 250. The subsequent figures depict the gradient magnitudes of various methods between epochs 150 and 250.

Cross-Entropy. It also indicates that, during training with a moderate learning rate from epochs 150 to 200, Focal Loss tends to produce better-calibrated models than BSCE-GRA. This may be due to the fact that, during these epochs, the model makes more predictions with mid-range confidence levels, and as shown in Figure 1, Focal Loss directs the model's attention towards these moderately uncertain samples. In contrast, BSCE-GRA imposes stronger regularization on the gradient, resulting in smaller optimization steps compared to Focal Loss. To further validate this hypothesis, we present the gradient density of the last linear layer across the entire training set at epochs 150 and 250 in Figure 5b and Fig. 5c, which shows that BSCE-GRA results in smaller gradient magnitudes during these epochs. After 250 epochs, when the learning rate undergoes its second reduction, the model trained with BSCE-GRA soon achieves better calibration performance compared to FL and CE.

5.2. Gradient Value among Uncertainty

We extend our analysis to examine the relationship between gradient magnitude and uncertainty. Specifically, we compute the gradient norms of the last linear layer for all samples in the training set at epochs 50, 150, 250, and 350. To quantify uncertainty, we calculate the Brier Score for each training sample. We present the results for Focal Loss (FL), Dual Focal Loss (DFL), and BSCE-GRA. Figure 4 illustrates the relationship between gradient magnitude and Brier Score. The figure clearly shows that the gradients produced by BSCE-GRA are the most sensitive to changes in the Brier Score, as indicated by the narrow distribution compared to other loss functions. This aligns with our goal: we aim for BSCE-GRA to exhibit sufficient sensitivity to uncertainty, allowing the model to adjust its gradient values based on changes in uncertainty, thereby focusing more on highly uncertain samples. Notably, the gradient distribution for Dual Focal Loss is relatively more dispersed concerning the Brier Score, with a wider range of possible values compared to other loss functions. This could be attributed to the partial derivatives involving other classes, as Dual Focal Loss takes the second most probable class into account in its loss calculation. In contrast, Focal Loss and BSCE-GRA only involve gradients along a single dimension.

5.3. Weighting FL and DFL on Gradients

We evaluate the performance of directly applying the uncertainty terms u_{FL} and u_{DFL} as weights on the gradients. Experiments are conducted using the default settings discussed above, on CIFAR-10 with ResNet-50. The results are presented in Table 2. It is evident that applying weights on gradients results in performance improvement for both u_{FL} and u_{DFL}, further validating its effectiveness.

We conduct additional experiments to comprehensively validate our proposed method, BSCE-GRA, under various settings. Due to page limitations, the extended experiments are provided in the Appendix.

6. Conclusion

In this paper, we proposed a novel approach to model calibration by directly weighting gradients based on uncertainty. We analyzed the strengths and limitations of Focal Loss and Dual Focal Loss from the perspective of sample weighting and introduced a framework that scales gradient magnitudes based on model uncertainty to focus more on uncertain samples. Additionally, we introduced BSCE-GRA, a loss function incorporating uncertainty metrics to enhance model calibration. Extensive experiments on various datasets and network architectures demonstrated significant improvements in calibration performance, achieving state-of-the-art results. Our findings emphasize the value of integrating uncertainty-aware mechanisms directly into the optimization process, providing a reliable framework for training calibrated deep neural networks suitable for real-world applications requiring trustworthy predictions. Future work may explore further optimization strategies for uncertainty-weighted approaches and their impact on tasks like active learning and robustness to adversarial attacks.

7. Acknowledgment

This work was supported in part by the Start-up Grant (No. 9610680) of the City University of Hong Kong, Young Scientist Fund (No. 62406265) of NSFC, and the Australian Research Council under Projects DP240101848 and FT230100549.

References

[1] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Meta-calibration: Learning of model calibration using differentiable expected calibration error. *arXiv preprint arXiv:2106.09613*, 2021. 1, 2

[2] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 1, 3, 6

[3] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 1

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 1, 2, 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[7] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. 2

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[9] Like Hui and Mikhail Belkin. {EVALUATION} {of} {neural} {architectures} {trained} {with} {square} {loss} {vs} {cross}-{entropy} {in} {classification} {tasks}. In *International Conference on Learning Representations*, 2021. 2

[10] Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. *Advances in Neural Information Processing Systems*, 34:29768–29779, 2021. 2

[11] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33: 18237–18248, 2020. 2

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6, 1

[13] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial intelligence and statistics*, pages 623–631. PMLR, 2017. 2

[14] 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. *Advances in neural information processing systems*, 32, 2019. 2, 3

[15] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR, 2018. 1, 6

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 2

[17] T Lin. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1, 3

[18] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020. 1, 2, 3, 6

[19] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 2

[20] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, 2015. 2

[21] Khanh Nguyen and Brendan O'Connor. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, 2015. 3

[22] 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*, pages 9617–9626, 2019. 6

[23] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 2

[24] Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in neural information processing systems*, 34:20063–20075, 2021. 2

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2

[26] Linwei Tao, Minjing Dong, Daochang Liu, Changming Sun, and Chang Xu. Calibrating a deep neural network with its predecessors. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4271–4279, 2023. 1

[27] Linwei Tao, Minjing Dong, and Chang Xu. Dual focal loss for calibration. In *International Conference on Machine Learning*, pages 33833–33849. PMLR, 2023. 1, 2, 3, 6

[28] Linwei Tao, Minjing Dong, and Chang Xu. Feature clipping for uncertainty calibration. *arXiv preprint arXiv:2410.19796*, 2024. 1

[29] Linwei Tao, Haolan Guo, Minjing Dong, and Chang Xu. Consistency calibration: Improving uncertainty calibration via consistency among perturbed neighbors. *arXiv preprint arXiv:2410.12295*, 2024. 6

[30] Linwei Tao, Younan Zhu, Haolan Guo, Minjing Dong, and Chang Xu. A benchmark study on calibration. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[31] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in neural information processing systems*, 32, 2019. 2

[32] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021. 2

[33] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer, 2011. 2

[34] 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*, pages 694–699, 2002. 2

[35] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 6

[36] Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR, 2020. 2

Uncertainty Weighted Gradients for Model Calibration

Supplementary Material

8. Proof of Equation 12

For the MSE term in Eq. 12, we will have:

$$||\hat{p} - y||^2 = \sum_{k=1}^{K}(p_k^2 - 2p_k y_k + y_k), \tag{15}$$

as the y is a one-hot vector. Besides, the one-hot labels y are sampled from Bernoulli Distributions: $y_k \sim \text{Bernoulli}(\eta_k(x))$, and the expectation of MSE can be computed as:

$$\mathbb{E}_{y\sim\eta(x)}[||\hat{p} - y||^2] = \sum_{k=1}^{K}(p_k^2 - 2p_k\mathbb{E}[y_k] + \mathbb{E}[y_k]). \tag{16}$$

Since $\mathbb{E}[y_k] = \eta_k$, we will have:

$$\begin{aligned} c(\boldsymbol{x}) - u_{\text{BS}}(\hat{p}(\boldsymbol{x})) &= \mathbb{E}_{y\sim\eta(x)}[\|\hat{p}(\boldsymbol{x}) - \boldsymbol{\eta}(\boldsymbol{x})\|_2^2 - \|\hat{p}(\boldsymbol{x}) - y\|_2^2] \\ &= \sum_{k=1}^{K}\boldsymbol{\eta}_k(\boldsymbol{x})(\boldsymbol{\eta}_k(\boldsymbol{x}) - 1). \end{aligned} \tag{17}$$

9. Theoretical Evidence for the Effectiveness of BSCE-GRA

Here, we prove that under strict convergence, the K-class predicted probability q equals the actual class-posterior probability η, thereby preventing over/under-confidence. For BSCE-GRA, we introduce the Lagrangian equation of BSCE-GRA as

$$L = [\sum_{i=1}^{K}(q_i - \eta_i)^2](-\sum_{i=1}^{K}\eta_i \log q_i) + \mu(\sum_{i=1}^{K} q_i - 1) \tag{18}$$

under the constraint $\sum_{i=1}^{K} q_i = 1$, where $[\cdot]$ denotes detaching the gradient. Since the MSE term can be considered as a constant C, considering the derivatives w.r.t q_i^* as 0, we have $\mu = C\eta_i/q_i$ for any i. Therefore, for any class i, there exists a constant k s.t. $q_i = k\eta_i$. Considering the constraint $\sum_{i=1}^{K} q_i = 1$, we find that $k = 1$ and thus $q_i = \eta_i$, which implies that this is an optimal minimum solution. When $q = \eta$, both MSE and CE equal 0. When $q \neq \eta$, BSCE-GRA> 0. Thus, the BSCE-GRA achieves a minimum when $q = \eta$.

We consider the extreme case for further evidence. When $\eta_i = 1$, $L(q) \supset (q_i - 1)^2(-\log q_i)$. The loss becomes 0 when $q_i = \eta_i = 1$. When $\eta_i = 0$, $L(q) \supset (q_i - 0)^2(-0 \cdot \log q_i) = 0$. For all classes, $\sum_{i=1}^{K} q_i = 1$. Therefore, $q_i = 0$ is the optimal solution for the class where $\eta_i = 0$. The optimal solution $q = \eta$ ensures the mitigation of over/under-confidence.

10. Dataset Desciption

We evaluate the performance of our proposed method, BSCE-GRA on multiple datasets to assess its calibration capabilities and robustness. The datasets include CIFAR-10/100 [12] and Tiny-ImageNet [4]. Below, we provide specific details for each dataset used:

CIFAR-10 CIFAR-10 consists of 60,000 32×32 color images divided into 10 classes, with 6,000 images per class (50,000 training and 10,000 test images). The classes include airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. This dataset is widely used in image classification tasks due to its simplicity and balanced class distribution. For our evaluation, we use 5,000 images from the training set for validation, ensuring a balanced split between training and validation data.

CIFAR-100. CIFAR-100 follows a similar structure to CIFAR-10 but with 100 classes, each containing 600 images (500 training and 100 test images per class). The classes in CIFAR-100 are more fine-grained compared to CIFAR-10, making it a more challenging dataset for image classification. Each class belongs to one of 20 superclasses, adding an additional layer of complexity to the classification task. This dataset allows us to evaluate the performance of our methods on a more complex and diverse set of visual categories.

Tiny ImageNet. Tiny ImageNet is a subset of the larger ImageNet dataset, consisting of 100,000 images across 200 classes, with each image resized to 64×64 pixels. Each class contains 500 training images, 50 validation images, and 50 test images. Tiny ImageNet is commonly used for benchmarking image classification models, providing a challenging task due to the increased number of classes compared to CIFAR-10/100 and the reduced image resolution compared to the original ImageNet dataset. The diversity and scale of Tiny ImageNet make it suitable for evaluating the robustness and scalability of our proposed methods.

11. Comparison Methods

To assess the effectiveness of our proposed algorithm, we compare it against several established methods. Details of these comparison methods are provided below:

Brier Loss [2]. Brier Loss calculates the squared error between the softmax logits and the one-hot encoded labels. It serves as a measure of both model calibration and accuracy.

MMCE Loss [15]. Maximum Mean Calibration Error (MMCE) is a kernel-based auxiliary loss used alongside Negative Log-Likelihood (NLL) to enhance calibration per-

Dataset	Model	CE	BL	MMCE	FLSD	DFL	BSCE	BSCE-GRA
CIFAR10	ResNet50	95.08	94.34	95.04	95.04	94.63	95.03	94.69
	ResNet110	94.84	94.41	94.91	94.76	94.79	94.88	94.72
	WideResNet	96.03	95.88	95.74	95.75	95.82	95.78	95.77
	DenseNet	94.95	94.35	94.73	94.92	94.58	94.76	94.84
CIFAR100	ResNet50	77.22	72.47	77.49	77.69	76.70	77.12	76.84
	ResNet110	77.44	74.42	77.42	77.77	77.27	77.30	77.16
	WideResNet	79.51	78.72	79.14	80.44	80.35	79.96	80.28
	DenseNet	76.76	73.32	76.07	77.29	77.02	76.82	76.96
TinyImageNet	ResNet50	49.88	27.66	48.81	51.98	51.04	50.06	50.21

Table 3. **Comparison of Calibration Methods Using Accuracy Across Various Datasets and Models.**

Dataset	Model	CE		BL		MMCE		FLSD		DFL		BSCE		BSCE-GRA	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T
CIFAR10	ResNet50	4.34	2.09	4.28	1.87	4.47	2.11	1.56	1.45	1.22	1.22	0.96	0.96	**0.71**	**0.71**
	ResNet110	4.70	2.46	4.48	2.11	4.80	2.24	1.89	1.56	**1.20**	**1.20**	1.38	1.38	1.28	1.28
	WideResNet	3.35	1.87	2.86	1.82	3.62	1.98	1.92	1.57	3.12	**1.43**	**1.72**	1.53	1.76	1.60
	DenseNet	4.61	2.43	3.96	1.67	4.81	2.38	1.44	1.52	**0.85**	**0.96**	0.99	0.99	1.09	1.09
CIFAR100	ResNet50	18.04	3.84	7.86	4.27	15.85	3.28	5.50	2.76	2.68	2.85	2.22	2.22	**1.82**	**1.82**
	ResNet110	18.84	5.90	16.77	4.41	18.65	4.69	6.85	3.71	3.90	3.90	2.71	2.71	**2.43**	**2.43**
	WideResNet	14.79	3.43	7.55	4.52	14.57	3.22	2.67	2.64	5.50	2.58	2.64	**2.37**	**2.48**	2.48
	DenseNet	19.09	3.93	8.05	3.09	17.55	2.85	3.29	**1.50**	4.69	1.76	1.62	1.62	**1.52**	1.52
TinyImageNet	ResNet50	14.93	5.15	6.80	1.38	13.50	4.92	**1.90**	1.90	6.71	2.20	4.00	1.70	4.56	**1.34**

Table 4. **Comparison of Calibration Methods Using AdaECE Across Various Datasets and Models.** AdaECE values are reported using adaptive binning, with the best-performing method for each dataset-model combination highlighted in bold. Results are averaged over three runs with different random seeds.

formance. It leverages a Reproducing Kernel Hilbert Space (RKHS) to evaluate and reduce miscalibration during training.

Focal Loss [18]. FLSD-53 is a simplified version of the sample-dependent gamma (γ) approach in Focal Loss. Mukhoti et al. [18] introduced a scheduling mechanism for gamma, replacing the original fixed value. Specifically, they set $\gamma_{\text{focal}} = 5$ for $\hat{p}_c \in [0, 0.2)$ and $\gamma_{\text{focal}} = 3$ for $\hat{p}_c \in [0.2, 1]$.

Dual Focal Loss [27]. Dual Focal Loss (DFL) extends Focal Loss by incorporating the second highest predicted probability into the uncertainty metric. This helps mitigate model underconfidence and improve calibration. In our experiments, we set $\gamma_{\text{DualFocal}} = 5$ as suggested by their reported findings.

12. Performance on Different Metrics

We report the accuracy of each method in different settings in Table 3. Although BSCE-GRA has the best calibration performance according to the ECE results in Table 1, it shows a competitive performance in accuracy compared with other methods. Adaptive-ECE is a calibration performance measure designed to address the bias inherent in the equal-width binning scheme used by ECE. It adapts the bin size based on the number of samples, ensuring an even distribution of samples across bins. The formula for Adaptive-ECE is as follows:

$$\text{Adaptive-ECE} = \sum_{i=1}^{B} \frac{|B_i|}{N} |I_i - C_i| \text{ s.t.} \forall i, j \cdot |B_i| = |B_j| \tag{19}$$

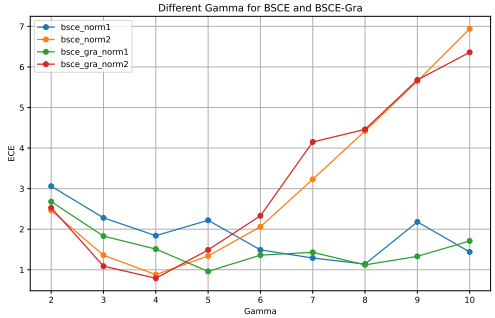
Table 4 shows that BSCE-GRA has the most optimal case compared to other methods, especially in the CIFAR100. Classwise-ECE is an alternative measure of calibration performance that overcomes the limitation of ECE, which only evaluates the calibration of the predicted class. It can be formulated as:

$$\text{Classwise-ECE} = \frac{1}{K} \sum_{i=1}^{B} \sum_{j=1}^{K} \frac{|B_{i,j}|}{N} |I_{i,j} - C_{i,j}| \tag{20}$$

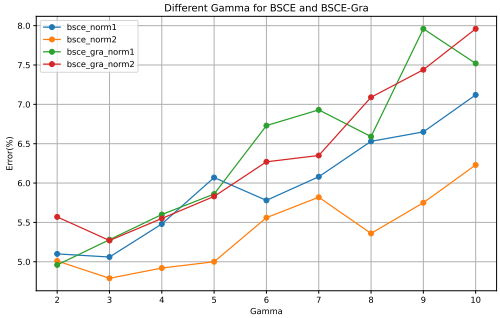
where $B_{i,j}$ denotes the set of samples with the j^{th} class label in the i^{th} bin, $I_{i,j}$ and $C_{i,j}$ represents the accuracy and confi-

Dataset	Model	CE		BL		MMCE		FLSD		DFL		BSCE		BSCE-GRA	
		Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T	Pre T	Post T
CIFAR10	ResNet50	0.90	0.46	0.90	0.50	0.92	0.51	0.41	0.42	0.40	0.40	0.38	0.38	**0.38**	**0.38**
	ResNet110	0.97	0.50	0.95	0.53	0.98	0.53	0.47	0.44	0.42	0.42	0.40	0.40	**0.39**	**0.39**
	WideResNet	0.71	0.38	0.63	0.40	0.76	0.41	0.44	0.35	0.83	0.37	0.44	0.35	**0.38**	**0.35**
	DenseNet	0.96	0.52	0.85	0.49	0.99	0.53	0.41	0.38	0.42	0.38	0.40	0.40	**0.37**	**0.37**
CIFAR100	ResNet50	0.40	0.21	0.24	0.24	0.36	0.21	0.21	0.21	0.21	0.20	0.21	0.21	**0.20**	**0.20**
	ResNet110	0.41	0.22	0.38	0.23	0.41	0.21	0.22	0.22	0.21	0.21	0.21	0.21	**0.21**	**0.21**
	WideResNet	0.33	0.21	0.22	0.22	0.32	0.21	**0.18**	0.19	0.23	0.19	0.20	0.19	0.19	**0.19**
	DenseNet	0.42	0.23	0.25	0.24	0.39	0.23	**0.19**	0.20	0.24	0.20	0.21	0.21	0.20	**0.20**
TinyImageNet	ResNet50	0.22	0.17	0.17	**0.14**	0.21	0.17	**0.16**	0.16	0.17	0.16	0.16	0.16	0.17	0.16

Table 5. **Comparison of Calibration Methods Using Classwise ECE Across Various Datasets and Models.** Classwise ECE values are reported for each dataset-model combination, with the best-performing method highlighted in bold. Results are averaged over three runs with different random seeds.



(a) Different gamma and norm performs different bsce and bsce-gra ECE



(b) Different gamma and norm performs different bsce and bsce-gra Error

dence of samples in $B_{i,j}$. Table 5 indicates that all methods perform similarly in terms of Classwise-ECE, yet BSCE-GRA consistently achieves the best results across most settings.

13. Reliability Diagram Variants Across Different Settings

We track the number of test samples classified correctly or incorrectly throughout the training process at epochs 50, 150, 250, and 350, as shown in Figure 7 and Figure 8. The confidence represents the probability assigned to the ground-truth class, and we report the frequency of both correct and incorrect predictions.

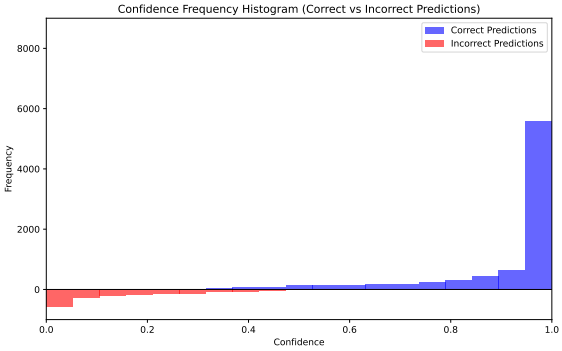
These figures provide insight into how different loss functions influence model predictions. Notably, Cross Entropy tends to produce predictions with high confidence from early on in training. By the final epoch, Cross Entropy frequently assigns near 100% confidence to predictions, even when they are incorrect. In contrast, other loss functions impose constraints that limit overconfident predictions.

Exp	3	4	5	6	7	8
ECE	2.78	1.13	2.83	5.13	6.96	9.28

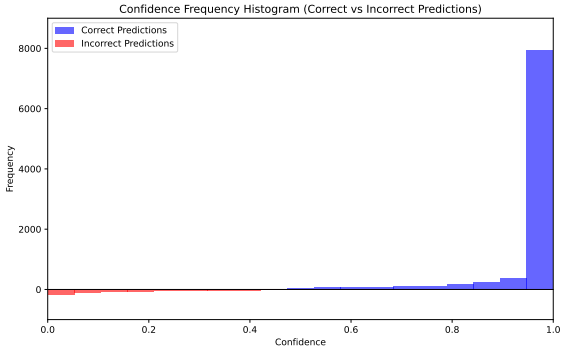
Table 6. Exponent Comparison on CIFAR100 with ResNet50

14. Hyperparameter Selection of BSCE-GRA

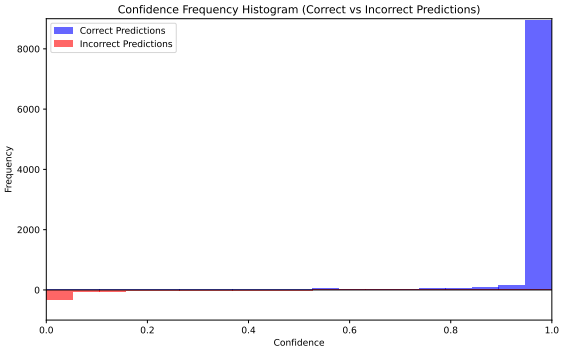
We determine the optimal hyperparameters for BSCE-GRA, including γ and β, using cross-validation, a standard approach as mentioned by [18]: "Finding an appropriate γ is normally done using cross-validation. Traditionally, γ is fixed for all samples in the dataset." We observe that the FLSD-53 strategy is employed in [18] to better control gradient magnitudes by achieving a more favorable trade-off in the function$u(\hat{p}(\boldsymbol{x}))$ in Eq. 9, as discussed in the same work. The primary reason we use fixed hyperparameters in our method, rather than an adaptive γ strategy, is that BSCE-GRA inherently achieves gradient magnitude control and favorable trade-offs during optimization. We acknowledge that tuning γ can improve calibration performance with Focal Loss. However, our proposed DFL also fulfills the requirements described in [18] by incorporating additional logits into the calculation. Moreover, we provide further empirical results for the selection of γ and β using ResNet50 on CIFAR-10, including ECE and predic-



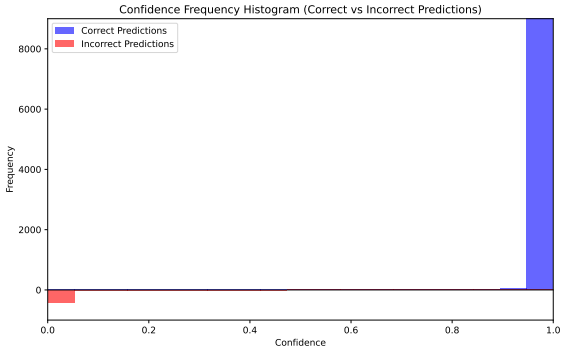
(a) CE at epoch 50



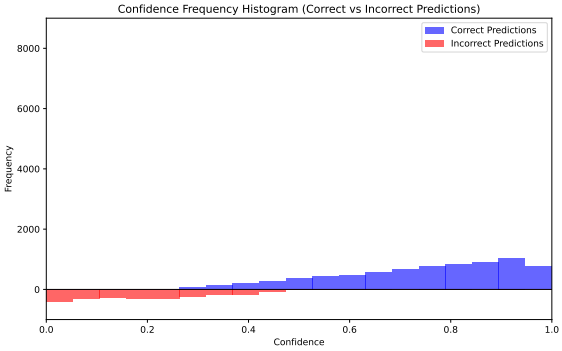
(b) CE at epoch 150



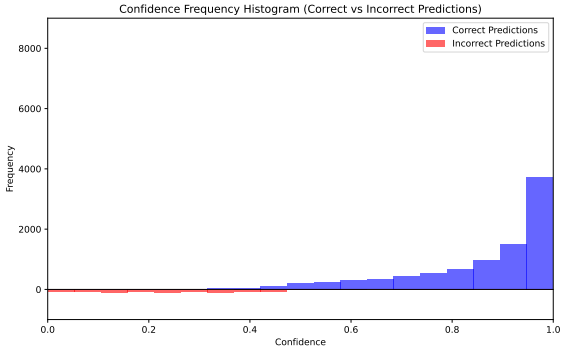
(c) CE at epoch 250



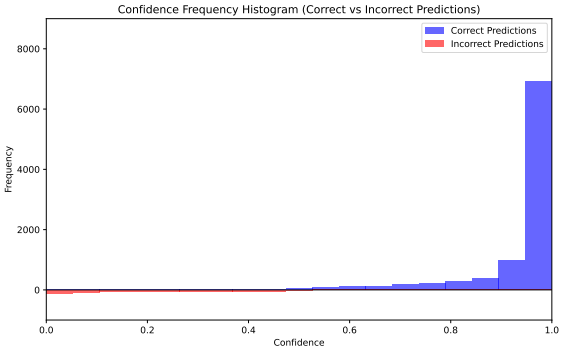
(d) CE at epoch 350



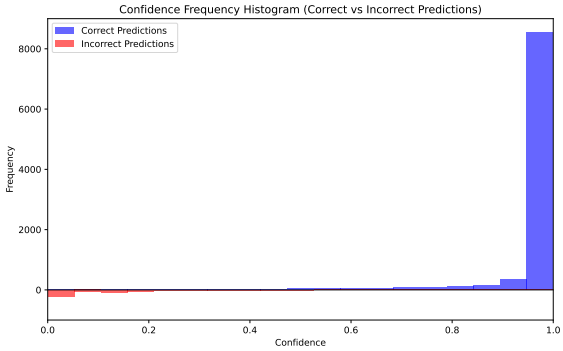
(e) FL at epoch 50



(f) FL at epoch 150

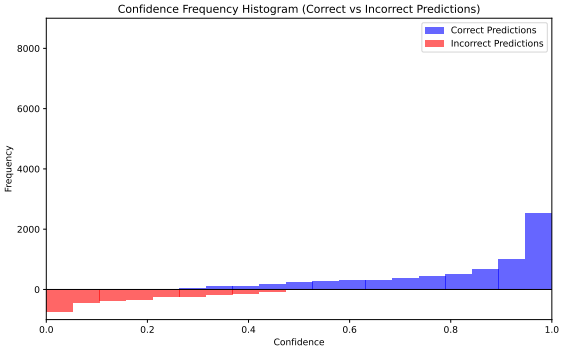


(g) FL at epoch 250

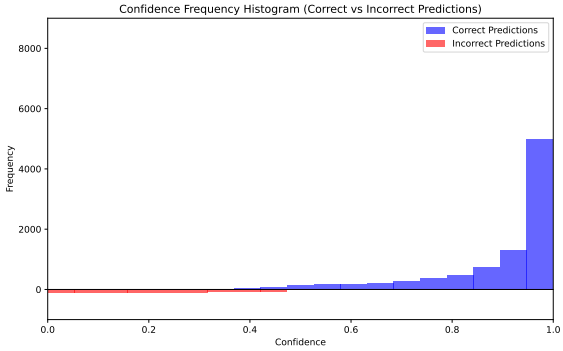


(h) FL at epoch 350

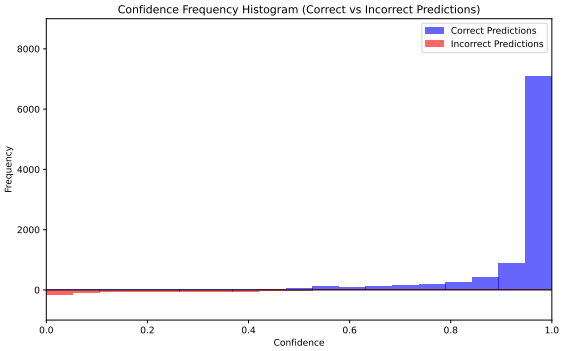
Figure 7. Correct and Wrong Predictions with Cross Entropy and Focal Loss among different epochs.



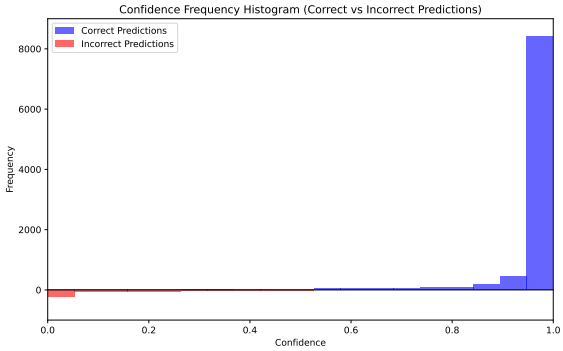
(a) DFL at epoch 50



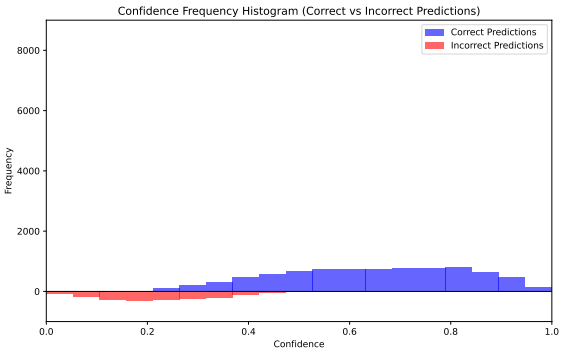
(b) DFL at epoch 150



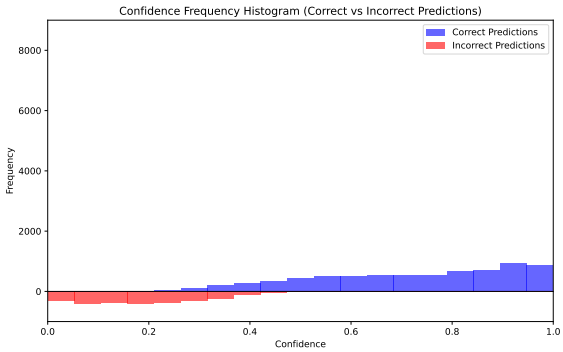
(c) DFL at epoch 250



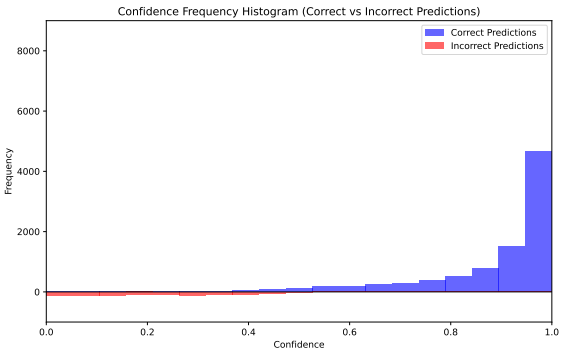
(d) DFL at epoch 350



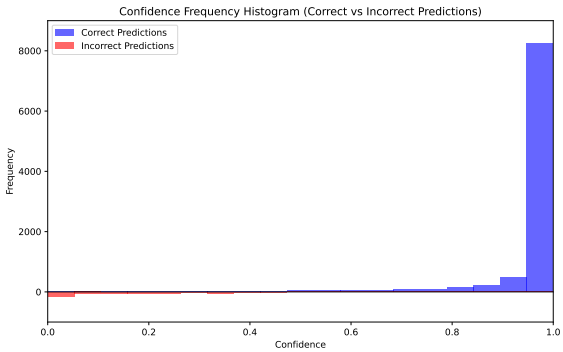
(e) BSCE at epoch 50



(f) BSCE at epoch 150



(g) BSCE at epoch 250



(h) BSCE at epoch 350

Figure 8. Evolution of confidence distributions under different training epochs for various loss functions. The histograms and density curves illustrate the confidence distributions of different loss functions (FL, DFL, BSCE, BSCE-GRA) during training. This comparison reveals how different loss functions shape the model's confidence throughout the training process.

Loss	CE	FL	BSCE-GRA
ECE	2.07	1.16	0.93

Table 7. ECE performance on ViT among CE, FL and BSCE-GRA

Loss	CE	FL	BSCE-GRA
ECE	3.69 (2.26)	3.28 (2.51)	2.63 (1.61)

Table 8. ECE performance on ImageNet with ResNet50

tion error, as shown in Figure 6a and Figure 6b. We also conduct the exponent hyperparameter comparison on CIFAR100 with ResNet50, the results are provided in Table 6.

15. Computation Efficiency

We further conduct experiments to validate the computation efficiency of BSCE-GRA. Although BSCE-GRA introduces an additional MSE calculation compared to CE, but does not affect backpropagation and has no significant impact on training time. We train a ResNet50 on CIFAR10 with default experiment setting. The running time of CE and BSCE-GRA are 128 and 139 mins, separately.

16. Effectiveness on More Model Structure

To further validate the effectiveness of the proposed method across a wider range of model architectures, we fine-tune a ViT model pretrained on IN-1K for 50 epochs on CIFAR-10 using different loss functions, including CE, FL, and BSCE-GRA. The backbone model is obtained from Hugging Face [2], and we follow their fine-tuning guide throughout the process. The results are reported in Table 7.

16.1. Sample-wise Calibration Metric

The proposed framework leverages sample-wise uncertainty as a gradient weight to enhance calibration. However, most existing calibration metrics, such as Expected Calibration Error (ECE), rely on binning strategies, making them unsuitable for directly measuring sample-wise calibration. A potential solution for evaluating sample-wise calibration is to measure the difference between ground truth and predicted probabilities. Although obtaining accurate ground truth probabilities is challenging, datasets like CIFAR-10H [22] approximate them through human annotations. Predicted probabilities from models may still exhibit bias, but they can be calibrated using methods like "consistency" [29] or temperature scaling. Tao et al. [29] perturb the model feature for a sample with a noise several times and consider the expectation of predicted probability as a local consistency of the sample. If the sample is less certain, the "consistency" will have a high variance.

To further validate the effectiveness of proposed framework, we utilize "1-consistency" [29] as an uncertainty weight to evaluate the proposed framework. We conduct experiments on CIFAR10 using the ResNet50 architecture, achieving an ECE of 0.98 and an accuracy of 94.7%.

[2] https://huggingface.co/google/vit-base-patch16-224.

17. Effectiveness on Large Scale Dataset

To thoroughly evaluate the effectiveness of the proposed method, we fine-tune a ResNet-50 model, pretrained on IN-1K and provided by PyTorch, for 90 epochs on the IN-1K dataset. The fine-tuning process utilizes three different loss functions: Cross Entropy (CE), Focal Loss (FL), and our proposed BSCE-GRA. For optimization, we employ the Adam optimizer, which effectively balances convergence speed and stability. Additionally, we utilize a Cosine Learning Rate Scheduler to adapt the learning rate throughout training, promoting efficient convergence. The detailed fine-tuning process follows standard practices to ensure fair comparison among different loss functions. The performance results, including accuracy and calibration metrics, are summarized in Table 8. These results provide insight into the robustness and adaptability of the proposed method across diverse training settings.