Feature Clipping for Uncertainty Calibration

Linwei Tao[1], **Minjing Dong**[2], **Chang Xu**[1]

[1]University of Sydney
[2]City University of Hong Kong
linwei.tao@sydney.edu.au, minjdong@cityu.edu.hk, c.xu@sydney.edu.au

Abstract

Deep neural networks (DNNs) have achieved significant success across various tasks, but ensuring reliable uncertainty estimates, known as model calibration, is crucial for their safe and effective deployment. Modern DNNs often suffer from overconfidence, leading to miscalibration. We propose a novel post-hoc calibration method called feature clipping (FC) to address this issue. FC involves clipping feature values to a specified threshold, effectively increasing entropy in high calibration error samples while maintaining the information in low calibration error samples. This process reduces the overconfidence in predictions, improving the overall calibration of the model. Our extensive experiments on datasets such as CIFAR-10, CIFAR-100, and ImageNet, and models including CNNs and transformers, demonstrate that FC consistently enhances calibration performance. Additionally, we provide a theoretical analysis that validates the effectiveness of our method. As the first calibration technique based on feature modification, feature clipping offers a novel approach to improving model calibration, showing significant improvements over both post-hoc and train-time calibration methods and pioneering a new avenue for feature-based model calibration.

Code — https://github.com/Linwei94/AAAI2025-FC.git

Introduction

While deep neural networks achieve significant improvements across various tasks, model calibration—ensuring a model provides reliable uncertainty estimates—is as important as achieving high prediction accuracy. Accurate and reliable uncertainty estimation is vital for many safety-critical downstream tasks, such as autonomous driving (Feng et al. 2019) and medical diagnosis (Chen et al. 2018). However, recent studies (Guo et al. 2017) have found that most modern neural networks struggle to accurately reflect the actual probabilities of their predictions through their confidence scores. Thus, improving model calibration techniques is essential to enhance the reliability of these models.

Efforts to address this issue can be divided into two streams: train-time calibration and post-hoc calibration. The first stream is train-time calibration, which includes training frameworks (Tao et al. 2023a; Liu et al. 2023), data augmentation (Wang et al. 2023; Zhang et al. 2022; Hendrycks et al. 2019), and regularization techniques like label smoothing (Müller et al. 2019; Liu et al. 2022) and entropy regularizers (Pereyra et al. 2017). Training losses such as dual focal loss (Tao et al. 2023b) and focal loss (Mukhoti et al. 2020) are also notable methods. The second stream is post-hoc calibration methods, which are applied to trained models and modify the output probability. Representative works include Isotonic Regression (Zadrozny and Elkan 2002), Histogram Binning (Zadrozny and Elkan 2001), and Temperature Scaling (TS) (Guo et al. 2017). Among these, TS is widely accepted due to its simplicity and good performance. Many subsequent works (Frenkel et al. 2021; Xiong et al. 2023; Yang et al. 2024; Tomani et al. 2022) propose improved versions of TS, often making the temperature factor adaptive according to different criteria.

Guo et al. (2017) identified overconfidence as a major cause of miscalibration in most modern neural networks. Adding a maximum-entropy penalty effectively increases prediction uncertainty, thereby mitigating overconfidence issues. Many calibration methods can be summarized as using entropy regularization in various forms. Pereyra et al. (2017) apply a maximum entropy penalty uniformly to all samples. Similarly, Label Smoothing (Müller et al. 2019) can be transformed into a form of entropy penalty, while Focal Loss (Mukhoti et al. 2020) can be viewed as the upper bound of a form with negative entropy, effectively adding a maximum entropy regularizer. TS often uses a temperature parameter larger than 1, resulting in a smoother probability distribution with higher entropy.

Since the features extracted by neural networks are direct representations of data, a possible way to mimic this maximum-entropy penalty effect is by applying information loss directly to the features, thereby increasing entropy. To explore this idea, we begin by comparing high calibration error samples with low calibration error samples. Accurately obtaining per-sample calibration error is nontrivial, so we choose wrongly predicted samples with high confidence (greater than 0.95) as high calibration error (HCE) samples and correctly predicted samples with high confidence (greater than 0.95) as low calibration error (LCE) samples. We randomly select 100 feature units from the feature of these samples and plot the average unit value in Figure 1.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

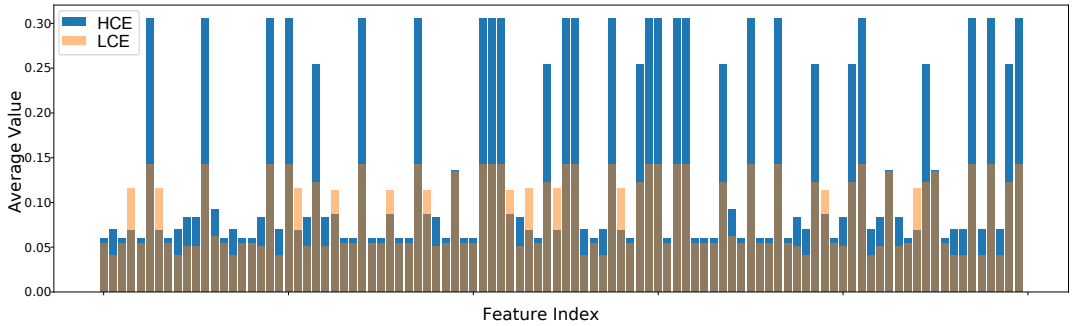

Figure 1: **Average feature value of samples with high or low calibration error.** We randomly select 100 feature units out of 2048 units. The high/low calibration error samples are selected as the wrongly/correctly predicted samples with confidence larger than 0.95. High calibration error samples shows a obvious tendency of higher feature value in around 30% feature. We provide a comparison of full 2048 feature units in Appendix, which shows similar pattern.

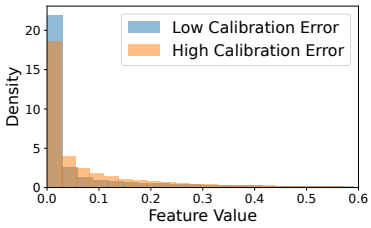

Figure 2: **Histogram of feature values for HCE and LCE samples.** The thicker tail of the HCE distribution indicates a larger variance σ^2 compared to the LCE distribution. These experiments were conducted using ResNet-50 on the CIFAR-10 dataset.

We observe that the feature value of HCE samples is much higher than that of LCE samples in some units. A potential solution is to clip the feature values, making values larger than a threshold c equal to c. This might help reduce the abnormally large feature values, increasing the entropy of HCE samples. For example, in Figure 1, we propose feature clipping at 0.15 to increase the entropy of HCE samples while retaining the information of LCE samples. This removes significant information from HCE samples, making them more uncertain, while maintaining as much information as possible from LCE samples. We also plot the histogram of feature of both HCE samples and LCE samples to examine the feature distribution of both samples as shown in Fig. 2. We observe that HCE samples exhibit a thicker tail, indicating a larger variance compared to LCE samples. These patterns suggest notable differences in features between HCE and LCE samples. Therefore, it is worthwhile to conduct a deeper study on how to calibrate models based on these features.

Motivated by these observation, we propose a simple and effective post-hoc calibration method called *feature clipping* (FC), which clips the feature value to a hyperparameter c, optimized on the validation set to minimize negative log likelihood (NLL), similar to temperature scaling. We also provide a solid theoretical analysis to prove the effectiveness of FC. To the best of our knowledge, we are the first to propose a calibration method based on feature modification. We conduct extensive experiments on a wide range of datasets, including CIFAR-10, CIFAR-100, and ImageNet, and models, including CNNs and transformers. Our method shows consistent improvement. Furthermore, since we are the first to perform calibration on features, our method is orthogonal to previous calibration methods. Extensive experiments demonstrate that FC can enhance calibration performance over both previous post-hoc and train-time calibration methods. Overall, we make the following contributions:

- We propose a simple and effective calibration method called feature clipping, which achieves SOTA calibration performance across multiple models and datasets.
- We provide a solid theoretical analysis to prove the effectiveness of feature clipping by showing feature clipping increases more entropy on HCE samples.
- We are the first to propose calibration based on features, initiating a new avenue for feature-based calibration. Our method serves as a strong baseline for this emerging area.

Related Works

Deep neural networks have long been a focus of calibration research (Guo et al. 2017), with extensive studies examining their calibration properties (Minderer et al. 2021; Wang, Feng, and Zhang 2021; Tao et al. 2023c). Numerous calibration methods have been proposed, generally divided into two categories: train-time calibration and post-hoc calibration.

Train-Time Calibration Train-time calibration aims to improve a model's calibration performance during training. A notable example is focal loss (Mukhoti et al. 2020), with subsequent works such as dual focal loss (Tao et al. 2023b) focusing on both the highest and second-highest probabilities. Adaptive focal loss (Ghosh, Schaaf, and Gormley 2022) modifies hyperparameters for different sample groups based on prior training knowledge. These focal loss-based methods can be transformed into an upper bound of negative entropy, thereby performing an entropy penalty during training. Similarly, label smoothing (Müller et al. 2019) can also

be transformed into a form of entropy penalty.

Post-Hoc Calibration Post-hoc calibration is resource-efficient and can be easily applied to pretrained models without altering their weights, preserving the model's accuracy and robustness. A common technique is temperature scaling (TS), which adjusts the output probability distribution's sharpness via a temperature parameter optimized to minimize negative log likelihood (NLL) on a validation set. TS typically uses larger temperature parameters for CNN models, reducing probability distribution sharpness and acting as an uniform maximum-entropy regularizer. Many subsequent methods aim to improve TS by applying adaptive temperature parameters, treating samples differently for a more effective maximum-entropy regularizer. For example, CTS (Frenkel et al. 2021) adapts temperature based on class labels, while PTS (Tomani et al. 2022) proposes learnable temperature parameters using a neural network. Recent methods like Proximity-based TS (Xiong et al. 2023) and Group Calibration (Yang et al. 2024) adjust temperature based on features, aiming for more precise entropy penalties.

Calibration Using Features Although feature representation is a crucial aspect of deep neural networks and is well-studied in robustness literature (Ilyas et al. 2019), it is underutilized in calibration literature. Pioneering works such as (Xiong et al. 2023; Yang et al. 2024) have explored using features to group similar samples to achieve multi-calibration (Hébert-Johnson et al. 2018). However, they do not perform calibration based on feature modification.

Methodology

Problem Formulation In a classification task, let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the label space. The classifier f maps an input to a probability distribution $\hat{p}_{[1,2,\ldots,K]} \in [0,1]^K$ over K classes. The confidence of a prediction is defined as the largest probability, $\max(\hat{p}_i)$. For simplicity, we use $\hat{p}$ to represent confidence in the following discussion.

A network is perfectly calibrated if the predicted confidence $\hat{p}$ accurately represents the true probability of the classification being correct. Formally, a perfectly calibrated network satisfies $\mathbb{P}(\hat{y} = y|\hat{p} = p) = p$ for all $p \in [0,1]$ (Guo et al. 2017), where $\hat{y}$ is the predicted label and y is the ground truth label. Given the confidence score and the probability of correctness, the *Expected Calibration Error* (ECE) is defined as $\mathbb{E}_{\hat{p}}[|\mathbb{P}(\hat{y} = y|\hat{p}) - \hat{p}|]$. In practice, since the calibration error cannot be exactly derived from finite samples, an approximation of ECE is introduced (Guo et al. 2017). Specifically, samples are grouped into M bins $\{B_m\}_{m=1}^M$ based on their confidence scores, where B_m contains samples with confidence scores $\hat{p}_i \in \left[\frac{m-1}{M}, \frac{m}{M}\right)$. For each bin B_m, the average confidence is computed as $C_m = \frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_i$ and the bin accuracy as $A_m = \frac{1}{|B_m|}\sum_{i\in B_m} \mathbb{1}(\hat{y}_i = y_i)$, where $\mathbb{1}$ is the indicator function. The ECE is then approximated as the expected absolute difference between bin accuracy and average confidence:

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

where N is the total number of samples. Besides this estimated ECE, there are variants like Adaptive ECE (Krishnan and Tickoo 2020), which groups samples into bins with equal sample sizes, and Classwise ECE (Kull et al. 2019), which computes ECE over K classes.

Feature Clipping We propose feature clipping (FC), a simple and effective post-hoc calibration method designed to reduce overconfidence problem in deep neural networks. The key idea is to clip the feature values to a specified threshold, thereby increasing entropy in HCE samples while preserving the information in LCE samples. This approach helps mitigate overconfidence issues in HCE samples and improves overall model calibration. Given feature values x, we apply feature clipping as follows:

$$\tilde{x} = \max(\min(x, c), -c) \tag{1}$$

where c is a positive hyperparameter optimized on a validation set to minimize negative log likelihood (NLL).

Theoretical Evidence

In this section, we present theoretical evidence to explain the effectiveness of feature clipping by analyzing the information loss in features of HCE and LCE samples. Our aim is to demonstrate that after feature clipping, HCE samples, characterized by larger variance, experience greater information loss compared to LCE samples, which have smaller variance. Consequently, we perform the entropy penalty differently to HCE and LCE samples and make HCE samples more uncertain.

Entropy of original feature We consider the case where the output are all postive values after ReLU activation function. Consider the feature vector $\mathbf{x} := \{x_1, \ldots, x_n\}$ extracted from a sample, which is normally the output of penultimate layer of a neural network. Suppose the feature value X follows a rectified normal distribution (Socci, Lee, and Seung 1997), which is a mixture distribution with both discrete variables and continuous variables. To calculate the entropy for this mixture distribution[1] (Politis 1991), first, we treat the continuous variables as the truncated normal distribution (Burkardt 2014).

For a standard truncated normal distribution, suppose X has a normal distribution with mean $\mu = 0$ and variance σ^2 and lies within the interval (a, b). The probability density function (PDF) of truncated normal distribution is given by:

$$f(x;\mu,\sigma,a,b) = \frac{1}{\sigma}\frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)} \tag{2}$$

and by $f = 0$ otherwise. Here, $\phi(\xi) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\xi^2\right)$ is the probability density function of the standard normal distribution and $\Phi(\cdot)$ is its cumulative distribution function $\Phi(x) = \frac{1}{2}\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$ and $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$ is

[1]For a mixture distribution that includes both discrete and continuous variables, we treat the entropy of the discrete and continuous parts individually. This means the overall entropy is a weighted combination of the entropy of these two parts.

the error function. By definition, if $b = \infty$, then $\Phi\left(\frac{b-\mu}{\sigma}\right) = 1$. The entropy of truncated normal distribution is given by:

$$H_c(x;\mu,\sigma,a,b) = -\int_a^b f(x)\log f(x)\,dx = \log(\sqrt{2\pi e\sigma Z}) + \frac{\alpha\phi(\alpha)-\beta\phi(\beta)}{2Z} \quad (3)$$

where $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$ and $Z = \Phi(\beta) - \Phi(\alpha)$.

Thus, the PDF of the continuous variables of the mixture distribution is $f(x;0,\sigma,0,+\infty)$ and the corresponding differential entropy is $H_c(x;0,\sigma,0,+\infty)$. The probability mass function (PMF) for discrete variables in the mixture distribution is given by:

$$p(x) = \begin{cases} 100\% & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

and the corresponding Shannon entropy is $H_d(x) = 0$.

Assume the input of ReLU layer follows Gaussian distribution with mean at 0, we can derive that feature x with probability $q = 0.5$ to be discrete variables and $1-q$ to be continuous variables. According to the entropy calculation of mixture distribution (Politis 1991), the entropy of original feature x is given by:

$$\begin{aligned} H(x) =& -q\log q - (1-q)\log(1-q) \\ &+ qH_d(x) + (1-q)H_c(x;0,\sigma,0,+\infty) \\ =& -\log(\frac{1}{2}) - \frac{1}{2}H_c(x;0,\sigma,0,+\infty) \\ =& -\log(\frac{1}{2}) - \frac{1}{2}\log(\sqrt{\pi e\sigma}) \end{aligned} \quad (5)$$

Entropy of clipped feature Similarly, the clipped feature $\tilde{x}$ follows mixture distribution with discrete variables and continuous variables, where the PDF of the continuous variables is $f(\tilde{x};0,\sigma,0,c)$ and the corresponding differential entropy is $H_c(x;0,\sigma,0,c)$ The PMF for discrete variables is given by:

$$p(\tilde{x}) = \begin{cases} \frac{\Phi(0)}{\Phi(0)+(1-\Phi\left(\frac{c}{\sigma}\right))} & \text{if } \tilde{x} = 0 \\ \frac{1-\Phi\left(\frac{c}{\sigma}\right)}{\Phi(0)+(1-\Phi\left(\frac{c}{\sigma}\right))} & \text{if } \tilde{x} = c \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

and the corresponding entropy is

$$\begin{aligned} H_d(\tilde{x}) =& -\sum p(\tilde{x})\log(p(\tilde{x})) \\ =& -\frac{0.5}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\log\left(\frac{0.5}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\right) \\ &-\frac{1-\Phi\left(\frac{c}{\sigma}\right)}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\log\left(\frac{1-\Phi\left(\frac{c}{\sigma}\right)}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\right) \end{aligned} \quad (7)$$

Since $\tilde{x}$ with probability $\tilde{q} = \Phi(0) + (1-\Phi\left(\frac{c}{\sigma}\right))$ to be discrete variables and $1-\tilde{q}$ to be continuous variables, similarly, the entropy of clipped feature $\tilde{x}$ can be derived as following form according to (Politis 1991),

$$\begin{aligned} H(\tilde{x}) =& -\tilde{q}\log\tilde{q} - (1-\tilde{q})\log(1-\tilde{q}) \\ &+ \tilde{q}H_d(\tilde{x}) + (1-\tilde{q})H_c(\tilde{x};0,\sigma,0,c) \\ =& -\left(1.5-\Phi\left(\frac{c}{\sigma}\right)\right)\log\left(1.5-\Phi\left(\frac{c}{\sigma}\right)\right) \\ &-\left(\Phi\left(\frac{c}{\sigma}\right)-0.5\right)\log\left(\Phi\left(\frac{c}{\sigma}\right)-0.5\right) \\ &-0.5\log\left(\frac{0.5}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\right) \\ &-\left(1-\Phi\left(\frac{c}{\sigma}\right)\right)\log\left(\frac{1-\Phi\left(\frac{c}{\sigma}\right)}{1.5-\Phi\left(\frac{c}{\sigma}\right)}\right) \\ &+(\Phi\left(\frac{c}{\sigma}\right)-0.5)H_c(\tilde{x};0,\sigma,0,c) \end{aligned} \quad (8)$$

	$H_{\text{sm}}(X)$	$H_{\text{sm}}(\tilde{X})$	ΔH_{sm}
HCE	0.0824	0.5723	**0.4908**
LCE	0.0032	0.1525	0.1493

Table 1: Entropy values calculated on Softmax probability before and after clipping, and their differences for HCE and LCE samples. FC makes HCE samples more uncertain. The experiment is conducted on ResNet-50 on CIFAR-10.

Entropy Difference Then, the Shannon entropy difference between features before and after clipping is given by

$$\begin{aligned} \Delta H =& -\left(\Phi\left(\frac{c}{\sigma}\right)-0.5\right)\log\left(\Phi\left(\frac{c}{\sigma}\right)-0.5\right) \\ &+0.5\log(0.5) \\ &-\left(1-\Phi\left(\frac{c}{\sigma}\right)\right)\log\left(1-\Phi\left(\frac{c}{\sigma}\right)\right) \\ &+(\Phi\left(\frac{c}{\sigma}\right)-0.5)H_c(\tilde{x};0,\sigma,0,c) \\ &+\frac{1}{2}\log(\sqrt{\pi e\sigma}) \end{aligned} \quad (9)$$

and ΔH is determined by the clipping threshold c and σ. We adopt the empirical result that $\sigma_{HCE} > \sigma_{LCE}$, as discussed in previous section. Thus, we can derive the theorem,

Theorem 1. *High calibration error samples suffer larger entropy difference compared to low calibration error samples after feature clipping.*

$$\Delta H_{LCE} < \Delta H_{HCE}$$

The detailed proof of Theorem 1 is given in Appendix. To verify our conclusion, we further calculate the entropy difference at Softmax layer, which is consistent with our observation. Specifically, we numerically calculate the entropy based on Softmax probability before and after feature clipping. As shown in Table 1, both entropy of HCE samples $H_{\text{sm}}^{\text{HCE}}(X)$ and entropy of LCE samples $H_{\text{sm}}^{\text{LCE}}(X)$ are

Dataset	Model	Original Feature		TS (Guo et al. 2017)		ETS (Zhang et al. 2020)		PTS (Tomani et al. 2022)		CTS (Frenkel et al. 2021)		GC (Yang et al. 2024)	
		base	+ours(c)	base	+ours	base	+ours	base	+ours	base	+ours	base	+ours
CIFAR-10	ResNet-50	4.34	1.10(0.23) ▼	1.39	1.22 ▼	1.37	1.22 ▼	1.36	1.25 ▼	1.46	1.25 ▼	0.97	**0.49** ▼
	ResNet-110	4.41	0.96(0.23) ▼	0.98	0.94 ▼	0.98	0.94 ▼	0.95	**0.90** ▼	1.13	**0.90** ▼	1.24	1.78 ▲
	DenseNet-121	4.51	**1.05**(0.45) ▼	1.41	1.11 ▼	1.40	1.12 ▼	1.38	1.14 ▼	1.44	1.14 ▼	1.27	2.51 ▲
CIFAR-100	ResNet-50	17.52	3.98(0.60) ▼	5.72	4.26 ▼	5.68	4.29 ▼	5.64	4.37 ▼	6.03	4.37 ▼	3.43	**1.70** ▼
	ResNet-110	19.06	4.40(0.61) ▼	5.12	4.81 ▼	5.10	4.81 ▼	5.05	4.98 ▼	5.43	4.98 ▼	**2.71**	3.45 ▲
	DenseNet-121	20.99	3.28(1.40) ▼	5.15	3.92 ▼	5.09	3.95 ▼	5.06	4.04 ▼	4.87	4.04 ▼	2.84	**1.75** ▼
ImageNet	ResNet-50	3.69	1.74(2.06) ▼	2.08	1.64 ▼	2.08	1.65 ▼	2.11	1.63 ▼	3.05	1.63 ▼	1.30	**1.00** ▼
	DenseNet-121	6.66	3.08(3.45) ▼	1.65	1.19 ▼	1.65	1.20 ▼	1.61	1.20 ▼	2.21	1.20 ▼	2.67	**0.63** ▼
	Wide-Resnet-50	5.52	2.52(3.05) ▼	3.01	2.21 ▼	3.01	2.20 ▼	3.00	2.18 ▼	4.31	2.18 ▼	3.01	**0.88** ▼
	MobileNet-V2	2.72	1.36(1.73) ▼	1.92	1.41 ▼	1.92	1.41 ▼	1.93	1.44 ▼	2.34	1.44 ▼	1.81	**0.50** ▼

Table 2: **ECE↓ before and after after feature clipping.** ECE is measured as a percentage, with lower values indicating better calibration. ECE is evaluated for different post hoc calibration methods, both before (base) and after (+ours) feature clipping. The results are calculated with number of bins set as 15. The optimal c is determined on the validation set, included in brackets.

close to zero before clipping. However, after feature clipping, the entropy of HCE samples $H_{\text{sm}}^{\text{HCE}}(\tilde{X})$ become much larger than entropy of LCE samples $H_{\text{sm}}^{\text{LCE}}(\tilde{X})$.

$$\Delta H_{\text{sm}}^{\text{LCE}} \ll \Delta H_{\text{sm}}^{\text{HCE}}.$$

In other words, feature clipping successfully differentiated the handling of HCE and LCE samples, increase more entropy in HCE samples compared to LCE samples and make HCE samples more uncertain.

Experiments

Experiment Setup

Models and Datasets We evaluate our methods on various deep neural networks (DNNs), including ResNet (He et al. 2016), Wide-ResNet (Zagoruyko and Komodakis 2016), DenseNet (Huang et al. 2017), MobileNet (Howard et al. 2017), and ViT (Dosovitskiy et al. 2020), using the CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009), and ImageNet-1K (Deng et al. 2009) datasets to assess the effectiveness of feature clipping. Pre-trained weights for post hoc calibration evaluation are provided by PyTorch.torchvision. Pre-trained weights trained by other train-time calibration methods are provided by Mukhoti et al. (2020).

Metrics We use the Expected Calibration Error (ECE) and accuracy as our primary metrics for evaluation. Additionally, we incorporate Adaptive ECE, a variant of ECE, which groups samples into bins of equal sizes to provide a balanced evaluation of calibration performance. For both ECE and Adaptive ECE, we use bin size at 15. We also measure the influence of calibration methods on prediction accuracy.

Comparison methods We compare our methods with several popular and state-of-the-art (SOTA) approaches. For post-hoc methods, we evaluate the widely used temperature scaling (TS) and other subsequent methods such as ETS (Zhang et al. 2020), PTS (Tomani et al. 2022), CTS (Frenkel et al. 2021), and a recently proposed SOTA calibration method called Group Calibration (Yang et al. 2024). For all TS-based methods, we determine the temperature by tuning the hyperparameter on the validation set to minimize the Negative Log Likelihood (NLL). To maintain consistency with TS, we also determine the optimal clipping threshold c on the validation set by minimizing the NLL. For training-time calibration methods, we include training with Brier loss (Brier 1950), label smoothing (Müller et al. 2019) with a smoothing factor of 0.05, FLSD-53 (Mukhoti et al. 2020) using the same γ scheduling scheme as in (Mukhoti et al. 2020), and Dual Focal Loss (Tao et al. 2023b). Detailed settings are following the settings in (Mukhoti et al. 2020).

Calibration Performance

To evaluate the performance, we assess feature clipping on both post-hoc methods and train-time calibration. We also find that post-hoc calibration methods hardly improve calibration performance on ViT and provide an empirical analysis to support this finding.

Compare with Post-Hoc Calibration Methods We compare the post-hoc calibration performance across multiple datasets and models, as shown in Table 2. FC consistently improves over the original features. With similar computational overhead and simplicity, FC outperforms TS in most cases, as seen when comparing columns 2 and 3. When combined with other post-hoc calibration methods, FC achieves state-of-the-art results. While Group Calibration also shows competitive results, it requires training an additional neural network based on features, resulting in higher computational overhead. Additionally, feature clipping is not compatible with Group Calibration in some cases, likely because Group Calibration separates groups based on features, while FC clips features, reducing information and making them less separable. Notably, in several instances, FC alone achieves the best performance, highlighting the potential of feature-based calibration. The simplicity of FC as a baseline method suggests significant opportunities for enhancement and optimization in future work. This demonstrates that even straightforward approaches like FC can yield substantial improvements, paving the way for more sophisticated feature-

	Vanilla	TS	ETS	PTS	CTS
w/o FC	5.24	5.73	5.73	5.73	6.07
w/ FC	**5.04**▼	5.59▼	5.60▼	5.60▼	5.60▼

Table 3: **ECE Calibration performance on Vision Transformer.** Feature Clipping provides little but consistent improvement on Vision Transformer. Experiments are conducted on ViT-L-16 on ImageNet.

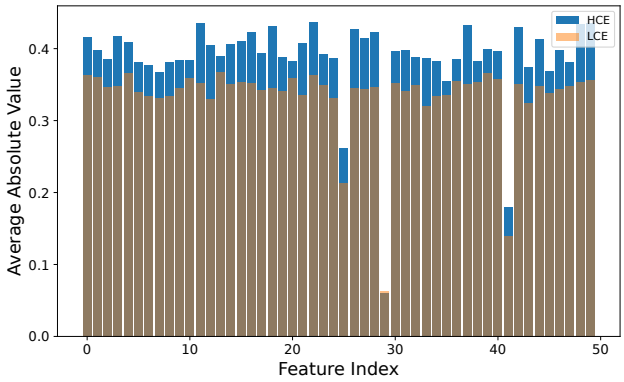

Figure 3: **Average absolute feature value of samples with high or low calibration error on Vision Transformer.** We randomly select 50 feature units out of 2048 units. The high-/low calibration error samples are selected as the wrongly/-correctly predicted samples with confidence larger than 0.8.

based calibration techniques. We also evaluate feature clipping using Adaptive ECE, a balanced version of ECE, with the results presented in the Appendix. FC demonstrates competitive results in this evaluation as well.

	ResNet-50		ViT-L-16	
Confidence	Correct	Wrong	Correct	Wrong
> 0.80	5921	471	6431	455
> 0.90	5221	271	3429	92
> 0.95	4526	161	44	0
> 0.99	3173	53	0	0

Table 4: **Number of high confidence samples in ImageNet test set.** The total number of samples is 50,000.

Performance on Vision Transformer Although Vision Transformers do not end with a ReLU layer, the difference between HCE samples and LCE samples still exists, indicating that feature clipping can significantly influence HCE samples. As shown in Figure 3, we take the mean of the absolute value of features for better visualization. The HCE samples show higher average feature values than LCE samples. However, the improvement is not as pronounced compared to CNN models. We show the ECE performance of ViT-L-16 in Table 3. To investigate the reason, we count the number of overconfident samples, as shown in Table 4. The number of samples with confidence larger than 0.8 is similar for both CNN and ViT. However, for samples with confidence greater than 0.95, CNN has significantly more samples than ViT. When examining samples with confidence greater than 0.99, CNN still has many samples, while ViT has none within this confidence range. This indicates that transformers face far fewer overconfidence issues compared to CNN models. Theoretically, clipping feature values results in a loss of information, increasing entropy and mitigating overconfidence problems. Since transformers exhibit fewer overconfidence problems compared to CNNs, our method has less impact on transformer-based models compared to CNNs. However, the difference in feature values among samples still exists, indicating significant potential for future improvements in transformer models.

Compare with Train-time Calibration Methods We also compare the effectiveness of feature clipping when applied on top of various train-time calibration methods. Feature clipping consistently demonstrates improvement across all these train-time calibration methods and different models, as shown in Table 5. On simpler datasets like CIFAR-10, models trained with "maximum-entropy penalty" methods such as focal loss and label smoothing adequately address the overconfidence issue. These methods effectively mitigate the overconfidence problem, leaving little room for additional improvement through feature clipping. However, when applied to more complex datasets like CIFAR-100, these training losses may not entirely resolve the overconfidence problem, providing an opportunity for feature clipping to further alleviate this issue and enhance calibration. Feature clipping's ability to improve calibration in such scenarios underscores its potential as a valuable addition to existing training-time calibration methods.

Ablation Study

Feature clipping is a straightforward method that causes samples to lose information. We are interested in understanding how this loss of information affects various aspects of model performance. Therefore, we study its influence on accuracy, how hyperparameter c affect performance, and its performance when applied to different layers.

Does Feature Clipping Affect Accuracy? Although post-hoc methods do not change the model weights and can maintain prediction performance by keeping the original features, we are still interested in how the optimal clipping value affects accuracy. In Table 6, we compare the prediction accuracy of different train-time calibration methods with our feature clipping method. The baseline column indicates the model trained with cross-entropy loss using the original features, while the FC column shows the results of applying our feature clipping on the baseline. All models are trained with the same training recipe, which is included in the Appendix. We observe that feature clipping does not significantly affect accuracy. Despite reducing the information contained in the feature representation, FC sometimes even improves accuracy. On the other hand, some train-time methods, such as Brier loss, can negatively impact accuracy in most cases. This suggests that while these methods aim to improve calibration, they may inadvertently reduce the model's ability

Dataset	Model	Cross Entropy		Brier Loss (Brier 1950)		LS-0.05 (Müller et al. 2019)		FLSD-53 (Mukhoti et al. 2020)		Dual Focal Loss (Tao et al. 2023b)	
		base	+ours	base	+ours	base	+ours	base	+ours	base	+ours
CIFAR-10	ResNet-50	4.34	1.10(0.23) ▼	1.80	1.49(0.98) ▼	2.97	2.97(1.18) ▼	1.55	1.50(0.75) ▼	0.46	0.45(0.80) ▼
	ResNet-110	4.41	0.96(0.23) ▼	2.57	2.34(1.15) ▼	2.09	2.09(1.19) ▼	1.88	1.28(0.51) ▼	0.98	0.98(0.55) ▼
	DenseNet-121	4.51	1.05(0.45) ▼	1.52	1.52(2.69) ▼	1.87	1.87(2.05) ▼	1.23	1.20(1.76) ▼	0.57	0.57(1.96) ▼
	Wide-Resnet-26	3.24	1.35(0.28) ▼	1.24	1.24(2.08) ▼	4.25	4.25(1.74) ▼	1.58	1.58(2.20) ▼	0.81	0.81(2.12) ▼
CIFAR-100	ResNet-50	17.52	3.98(0.60) ▼	6.57	3.96(2.11) ▼	7.82	7.82(3.67) ▼	4.49	3.83(2.17) ▼	1.08	1.01(2.23) ▼
	ResNet-110	19.06	4.40(0.61) ▼	7.87	4.05(1.83) ▼	11.04	6.83(1.03) ▼	8.55	5.12(1.50) ▼	2.90	2.53(1.51) ▼
	DenseNet-121	20.99	3.28(1.40) ▼	5.22	3.50(4.11) ▼	12.87	3.06(1.89) ▼	3.70	2.96(4.02) ▼	1.81	1.53(3.52) ▼
	Wide-Resnet-26	15.34	4.38(0.98) ▼	4.34	3.11(2.24) ▼	4.88	4.88(3.10) ▼	3.02	1.90(2.47) ▼	1.79	1.18(2.30) ▼

Table 5: **ECE↓ before and after after feature clipping.** ECE is measured as a percentage, with lower values indicating better calibration. ECE is evaluated for different train-time calibration methods, both before (base) and after (+ours) feature clipping. The results are calculated with number of bins set as 15. The optimal c is determined on the validation set, included in brackets.

to generalize, thereby lowering prediction accuracy. The detailed comparison of accuracy across different methods and datasets illustrates that our feature clipping method maintains competitive performance.

Dataset	Model	Base	Brier	LS	Focal	FC
CIFAR-10	ResNet-50	95.05	95.0	94.71	95.02	94.93
	ResNet-110	95.11	94.52	94.48	94.58	95.01
	DenseNet-121	95.0	94.89	94.91	94.54	95.13
CIFAR-100	ResNet-50	76.7	76.61	76.57	76.78	76.74
	ResNet-110	77.27	74.9	76.57	77.49	77.06
	DenseNet-121	75.48	76.25	75.95	77.33	75.52

Table 6: **Accuracy↑ for different train-time methods and feature clipping.** Feature clipping does not impact prediction accuracy performance.

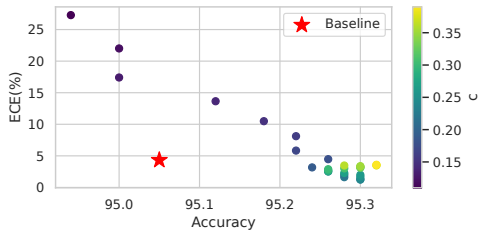

Figure 4: **Feature clipping at different value.** Points to the right bottom cornor indicate better performance. The experiment is conducted on ResNet-50 on CIFAR-10.

How does clip threshold affect performance? To test how the clipping threshold influences performance, we clip the features of a ResNet-50 trained with cross-entropy loss on CIFAR-10 using different clipping thresholds. We plot the resulting performance in terms of ECE and accuracy, as shown in Figure 4. The red star indicates the performance of the original features. Generally, within a certain range (between 0.15 and 0.35 in this case), feature clipping does not significantly affect model accuracy. However, feature clipping can substantially influence calibration performance. For instance, a clipping value of 0.15 (point at the top left corner) achieves similar accuracy to the baseline but results in much worse calibration performance, with an ECE exceeding 25%. With the optimal clipping value, the model can achieve an ECE as low as 1.10, as shown in Table 2. We believe the reason feature clipping has a larger influence on calibration is that excessive clipping may significantly increase entropy, affecting correct predictions with high confidence. As a result, the model faces underconfidence, leading to a large ECE.

Conclusion

In conclusion, our proposed feature clipping method demonstrates substantial improvements in model calibration across various datasets and models. FC effectively reduces overconfidence in predictions, enhancing calibration performance while maintaining accuracy. Despite its simplicity, FC achieves state-of-the-art calibration performance and provides a solid foundation for future research on feature-based calibration. However, there are several limitations and opportunities for improvement. The performance on transformer models, for instance, can be further improved. Future work should focus on developing more sophisticated methods, such as starting with a better threshold and conducting faster hyperparameter tuning. Additionally, exploring ways to find optimal clipping values using feature statistics and employing smoothed or adaptive thresholds instead of fixed ones are promising directions. These enhancements will potentially lead to even better calibration. Furthermore, investigating the impact of feature clipping on different neural network architectures and understanding its effects on various types of data can provide deeper insights. Our method serves as a strong baseline for feature-based calibration, and we believe that future developments can build upon this foundation to achieve even greater calibration improvements.

Acknowledgments

This work was supported in part by the Australian Research Council under Projects DP240101848 and FT230100549.

References

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3.

Burkardt, J. 2014. The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 1(35): 58.

Chen, W.; Sahiner, B.; Samuelson, F.; Pezeshk, A.; and Petrick, N. 2018. Calibration of medical diagnostic classifier scores to the probability of disease. *Statistical methods in medical research*, 27(5): 1394–1409.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Feng, D.; Rosenbaum, L.; Glaeser, C.; Timm, F.; and Dietmayer, K. 2019. Can we trust you? on calibration of a probabilistic object detector for autonomous driving. *arXiv preprint arXiv:1909.12358*.

Frenkel, L.; Goldberger, J.; Goldberger, J.; and Goldberger, J. 2021. Network calibration by class-based temperature scaling. In *2021 29th European Signal Processing Conference (EUSIPCO)*, 1486–1490. IEEE.

Ghosh, A.; Schaaf, T.; and Gormley, M. 2022. Adafocal: Calibration-aware adaptive focal loss. *Advances in Neural Information Processing Systems*, 35: 1583–1595.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hébert-Johnson, U.; Kim, M.; Reingold, O.; and Rothblum, G. 2018. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, 1939–1948. PMLR.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Krishnan, R.; and Tickoo, O. 2020. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33: 18237–18248.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *arXiv preprint arXiv:2010.11929*.

Kull, M.; Perello Nieto, M.; Kängsepp, M.; Silva Filho, T.; Song, H.; and Flach, P. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32.

Liu, B.; Ben Ayed, I.; Galdran, A.; and Dolz, J. 2022. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 80–88.

Liu, B.; Rony, J.; Galdran, A.; Dolz, J.; and Ben Ayed, I. 2023. Class adaptive network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16070–16079.

Minderer, M.; Djolonga, J.; Romijnders, R.; Hubis, F.; Zhai, X.; Houlsby, N.; Tran, D.; and Lucic, M. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34: 15682–15694.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P.; and Dokania, P. 2020. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33: 15288–15299.

Müller, R.; Kornblith, S.; Hinton, G. E.; and Hinton, G. E. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, Ł.; and Hinton, G. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Politis, D. N. 1991. On the Entropy of a Mixture Distribution. Technical Report 91-67, Purdue University.

Socci, N.; Lee, D.; and Seung, H. S. 1997. The rectified Gaussian distribution. *Advances in neural information processing systems*, 10.

Tao, L.; Dong, M.; Liu, D.; Sun, C.; and Xu, C. 2023a. Calibrating a deep neural network with its predecessors. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 4271–4279.

Tao, L.; Dong, M.; Xu, C.; and Xu, C. 2023b. Dual focal loss for calibration. In *International Conference on Machine Learning*, 33833–33849. PMLR.

Tao, L.; Zhu, Y.; Guo, H.; Dong, M.; and Xu, C. 2023c. A benchmark study on calibration. *arXiv preprint arXiv:2308.11838*.

Tomani, C.; Cremers, D.; Buettner, F.; and Sun, Y. 2022. Parameterized temperature scaling for boosting the expressive power in post-hoc uncertainty calibration. In *European Conference on Computer Vision*, 555–569. Springer.

Wang, D.-B.; Feng, L.; and Zhang, M.-L. 2021. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34: 11809–11820.

Wang, D.-B.; Li, L.; Zhao, P.; Heng, P.-A.; and Zhang, M.-L. 2023. On the pitfall of mixup for uncertainty calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7609–7618.

Xiong, M.; Deng, A.; Koh, P. W. W.; Wu, J.; Li, S.; Xu, J.; and Hooi, B. 2023. Proximity-informed calibration for deep neural networks. *Advances in Neural Information Processing Systems*, 36: 68511–68538.

Yang, J.-Q.; Zhan, D.-C.; Gan, L.; and Sun, Y. 2024. Beyond probability partitions: Calibrating neural networks with semantic aware grouping. *Advances in Neural Information Processing Systems*, 36.

Zadrozny, B.; and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, 609–616.

Zadrozny, B.; and Elkan, C. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 694–699.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, J.; Kailkhura, B.; Han, T. Y.-J.; and Sun, Y. 2020. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, 11117–11128. PMLR.

Zhang, L.; Deng, Z.; Kawaguchi, K.; and Zou, J. 2022. When and how mixup improves calibration. In *International Conference on Machine Learning*, 26135–26160. PMLR.