Improving Calibration for Long-Tailed Recognition

Zhisheng Zhong Jiequan Cui Shu Liu Jiaya Jia
Chinese University of Hong Kong SmartMore

Code: https://github.com/Jia-Research-Lab/MiSLAS

Abstract

Deep neural networks may perform poorly when training datasets are heavily class-imbalanced. Recently, two-stage methods decouple representation learning and classifier learning to improve performance. But there is still the vital issue of miscalibration. To address it, we design two methods to improve calibration and performance in such scenarios. Motivated by the fact that predicted probability distributions of classes are highly related to the numbers of class instances, we propose label-aware smoothing to deal with different degrees of over-confidence for classes and improve classifier learning. For dataset bias between these two stages due to different samplers, we further propose shifted batch normalization in the decoupling framework. Our proposed methods set new records on multiple popular long-tailed recognition benchmark datasets, including CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, Places-LT, and iNaturalist 2018.

1. Introduction

With numerous available large-scale and high-quality datasets, such as ImageNet [27], COCO [19], and Places [40], deep convolutional neural networks (CNNs) have made notable breakthrough in various computer vision tasks, such as image recognition [16, 10], object detection [26], and semantic segmentation [6]. These datasets are usually artificially balanced with respect to the number of instances for each object/class. However, in many real-world applications, data may follow unexpected long-tailed distributions, where the numbers of instances for different classes are seriously imbalanced. When training CNNs on these long-tailed datasets, the performance notably degrades. To address this terrible issue, a number of methods were proposed for long-tailed recognition.

Recently, many two-stage approaches have achieved significant improvement comparing with one-stage methods. Deferred re-sampling (DRS, [4]) and deferred re-weighting (DRW, [4]) first train CNNs in a normal way in Stage-1. DRS tunes CNNs on datasets with class-balanced resampling while DRW tunes CNNs by assigning different weights to classes in Stage-2. Zhou *et al*. [39] proposed bilateral branch network (BBN) in one stage to simulate the process of DRS by dynamically combining instance-balanced sampler and the reverse-balanced sampler. Kang *et al*. [15] proposed two-stage decoupling models, classifier re-training (cRT) and learnable weight scaling (LWS), to further boost performance, where decoupling models freeze the backbone and just train the classifier with class-balanced resampling in Stage-2.

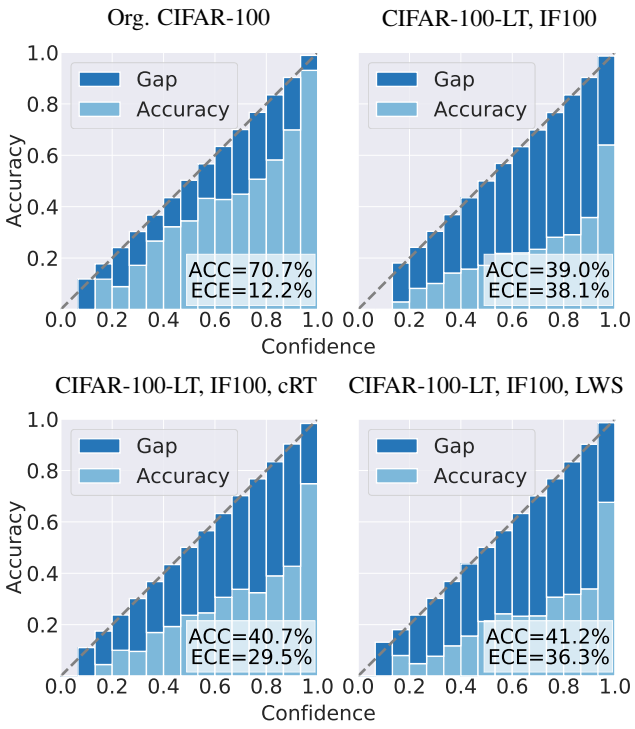

Figure 1: Reliability diagrams of ResNet-32. From top left to bottom right: the plain model trained on the original balanced CIFAR-100 dataset, the plain model, cRT, and LWS trained on CIFAR-100-LT with IF 100.

Confidence calibration [24, 9] is to predict probability by estimating representative of true correctness likelihood. It is important for recognition models in many applications [1,

14]. Expected calibration error (ECE) is widely used in measuring calibration of the network. To compute ECE, all N predictions are first grouped into B interval bins of equal size. ECE is defined as:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|\mathcal{S}_b|}{N} \Big| \mathrm{acc}(\mathcal{S}_b) - \mathrm{conf}(\mathcal{S}_b) \Big| \times 100\%,$$

where $\mathcal{S}_b$ is the set of samples whose prediction scores fall into Bin-b. $\mathrm{acc}(\cdot)$ and $\mathrm{conf}(\cdot)$ are the accuracy and predicted confidence of $\mathcal{S}_b$, respectively.

Our study shows, because of the imbalanced composition ratio of each class, networks trained on long-tailed datasets are more miscalibrated and over-confident. We draw the reliability diagrams with 15 bins in Fig. 1, which compares the plain cross-entropy (CE) model trained on the original CIFAR-100 dataset, the plain CE model, cRT, and LWS trained on CIFAR-100-LT with imbalanced factor (IF) 100. It is noticeable that networks trained on long-tailed datasets usually have higher ECEs. The two-stage models of cRT and LWS suffer from over-confidence as well. Moreover, Figs. 9 and 10 (the first two plots) in Appendix C depict that this phenomenon also commonly exists on other long-tailed datasets, such as CIFAR-10-LT and ImageNet-LT.

Another issue is that two-stage decoupling ignores the dataset bias or domain shift [25] in the two stages. In details, two-stage models are first trained on the instanced-balanced dataset $\mathcal{D}_\mathrm{I}$ in Stage-1. Then, models are trained on the class-balanced dataset $\mathcal{D}_\mathrm{C}$ in Stage-2. Obviously, $P_{\mathcal{D}_\mathrm{I}}(\boldsymbol{x}, y) \neq P_{\mathcal{D}_\mathrm{C}}(\boldsymbol{x}, y)$ and distributions of the dataset by different sampling ways are inconsistent. Motivated by transfer learning [17, 33], we focus on the batch normalization [12] layer to deal with the dataset bias problem.

In this work, we propose a **Mi**xup **S**hifted **L**abel-**A**ware **S**moothing model (MiSLAS) to effectively solve above issues. Our key contributions are as follows.

- We discover that models trained on long-tailed datasets are much more miscalibrated and over-confident than those trained on balanced data. Two-stage models suffer from this problem as well.
- We find that mixup can remedy over-confidence and have a positive effect on representation learning but a negative or negligible effect on classifier learning. To further enhance classifier learning and calibration, we propose label-aware smoothing to handle different degrees of over-confidence for classes.
- It is the first attempt to note the dataset bias or domain shift in two-stage resampling methods for long-tailed recognition. To deal with it in the decoupling framework, we propose shift learning on the batch normalization layer, which can greatly improve performance.
- We extensively validate our MiSLAS on multiple long-tailed recognition benchmark datasets – experimental results manifest the effectiveness. Our method yields new state-of-the-art.

2. Related Work

Re-sampling and re-weighting. There are two groups of re-sampling strategies: over-sampling the tail-class images [28, 2, 3] and under-sampling the head-class images [13, 2]. Over-sampling is regularly useful on large datasets and may suffer from heavy over-fitting to tail classes especially on small datasets. For under-sampling, it discards a large portion of data, which inevitably causes degradation of the generalization ability of deep models. Re-weighting [11, 34] is another prominent strategy. It assigns different weights for classes and even instances. The vanilla re-weighting method gives class weights in reverse proportion to the number of samples of classes.

However, with large-scale data, re-weighting makes deep models difficult to optimize during training. Cui *et al*. [7] relieved the problem using the effective numbers to calculate the class weights. Another line of work is to adaptively re-weight each instance. For example, focal loss [18, 22] assigned smaller weights for well-classified samples.

Confidence calibration and regularization. Calibrated confidence is significant for classification models in many applications. Calibration of modern neural networks is first discussed in [9]. The authors discovered that model capacity, normalization, and regularization have strong effect on network calibration. mixup [37] is a regularization technique to train with interpolation of input and labels.

mixup inspires follow-up of manifold mixup [32], CutMix [36], and Remix [5] that have shown significant improvement. Thulasidasan *et al*. [30] found that CNNs trained with mixup are better calibrated. Label smoothing [29] is another regularization technique that encourages the model to be less over-confident. Unlike cross-entropy that computes loss upon the ground truth labels, label smoothing computes loss upon a soft version of labels. It relieves over-fitting and increases calibration and reliability [23].

Two-stage methods. Cao *et al*. [4] proposed deferred re-weighting (DRW) and deferred re-sampling (DRS), working better than conventional one-stage methods. Its stage-2, starting from better features, adjusts the decision boundary and locally tunes features. Recently, Kang *et al*. [15] and Zhou *et al*. [39] concluded that although class re-balance matters for jointly training representation and classifier, instance-balanced sampling gives more general representations.

Based on this observation, Kang *et al*. [15] achieved state-of-the-art results by decomposing representation and classifier learning. It first trains the deep models with

Mark	Stg.-1	Stg.-2	ResNet-50	ResNet-101	ResNet-152
CE	☒		45.7 / 13.7	47.3 / 13.7	48.7 / 14.5
CE	☑		45.5 / 7.98	47.7 / 10.1	48.3 / 10.2
cRT	☒	☒	50.3 / 8.97	51.3 / 9.34	52.7 / 9.05
cRT	☒	☑	50.2 / 3.32	51.3 / 3.38	52.8 / 3.60
cRT	☑	☒	**51.7** / 5.62	**53.1** / 6.86	**54.2** / 6.02
cRT	☑	☑	51.6 / **3.13**	53.0 / **2.93**	54.1 / **3.37**

Mark	Stg.-1	Stg.-2	ResNet-50	ResNet-101	ResNet-152
CE	☒		45.7 / 13.7	47.3 / 13.7	48.7 / 14.5
CE	☑		45.5 / 7.98	47.7 / 10.1	48.3 / 10.2
LWS	☒	☒	51.2 / 4.89	52.3 / 5.10	53.8 / 4.48
LWS	☒	☑	51.0 / 5.01	52.2 / 5.38	53.6 / 5.50
LWS	☑	☒	52.0 / **2.23**	**53.5** / **2.73**	**54.6** / **2.46**
LWS	☑	☑	52.0 / 8.04	53.3 / 6.97	54.4 / 7.74

Table 1: Top-1 accuracy (%) and ECE (%) of the plain cross-entropy (CE) model, and decoupling models of cRT (left) and LWS (right), for ResNet families trained on the ImageNet-LT dataset. We vary the augmentation strategies with (☑), or without (☒) mixup $\alpha = 0.2$, on both of the stages.

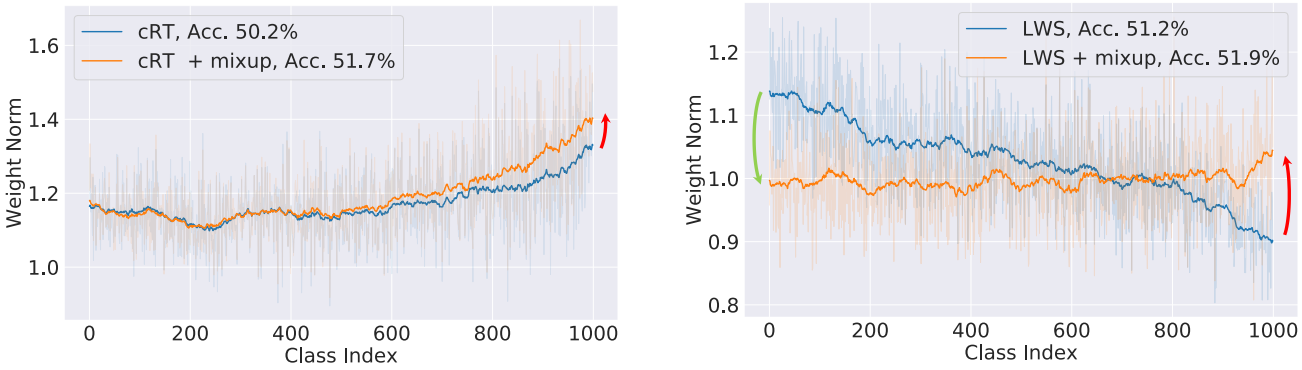

Figure 2: Classifier weight norms for the ImageNet-LT validation set where classes are sorted by descending values of N_j, where N_j denotes the number of training sample for Class-j. Left: weight norms of cRT with or without mixup. Right: weight norms of LWS with or without mixup. Light shade: true norm. Dark lines: smooth version. *Best viewed on screen.*

instance-balanced sampling, and then fine-tunes the classifier with class-balanced sampling with parameters of representation learning fixed. Similarly, Zhou *et al*. [39] integrated mixup training into the proposed cumulative learning strategy. It bridges the representation learning and classifier re-balancing. The cumulative learning strategy requires dual samplers of instance-balanced and reversed instance-balanced sampler.

3. Main Approach

3.1. Study of mixup Strategy

For the two-stage learning framework, Kang *et al*. [15] and Zhou *et al*. [39] found that instance-balanced sampling gives the most general representation among all for long-tailed recognition. Besides, Thulasidasan *et al*. [30] show that networks trained with mixup are better calibrated. Based on these findings, when using instance-balanced sampling, we explore the effect of mixup in the two-stage decoupling framework for higher representation generalization and over-confidence reduction.

We train a plain cross-entropy model, and two two-stage models of cRT and LWS, on ImageNet-LT for 180 epochs in Stage-1 and finetune them for 10 epochs in Stage-2, respectively. We vary the training setup (with/without mixup $\alpha = 0.2$) for both stages. Top-1 accuracy of these variants is listed in Table 1. It reveals the following. (i) When applying mixup, improvement of CE can be ignored. But the performance is greatly enhanced for both cRT and LWS. (ii) Applying additional mixup in Stage-2 yields no obvious improvement or even damages performance. The reason is that mixup encourages representation learning and is yet with adverse or negligible effect on classifier learning.

Besides, we draw the final classifier weight norms of these variants in Fig. 2. We show the L_2 norms of the weight vectors for all classes, as well as the training data distribution sorted in a descending manner concerning the number of instances. We observe that when applying mixup (in orange), the weight norms of the tail classes tends to be large and the weight norms of the head classes decrease. It means mixup may be more friendly to tail classes.

We also list ECEs of the above models in Table 1. When adding mixup in just Stage-1, both cRT and LWS models can consistently obtain better top-1 accuracy and lower ECEs for different backbones (Row-4 and Row-6). Due to the unsatisfied top-1 accuracy enhancement and unstable ECE decline of mixup for classifier learning (by adding mixup in Stage-2), we propose a label-aware smoothing to further improve both calibration and classifier learning.

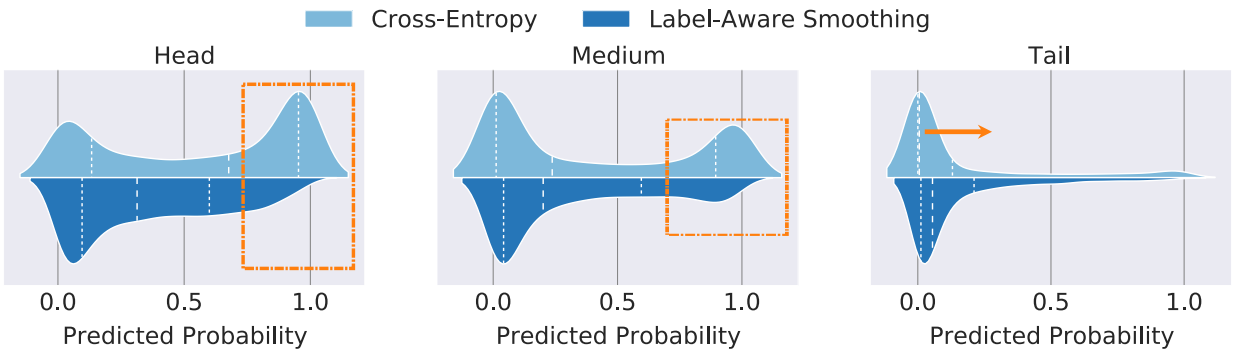

Figure 3: Violin plot of predicted probability distributions for different parts of the classes, head (100+ images per class), medium (20-100 images per class), and tail (less than 20 images per class) on CIFAR-100-LT with IF 100. The upper half part in light blue denotes "LWS + cross-entropy". The bottom half part in deep blue represents "LWS + label-aware smoothing".

3.2. Label-aware Smoothing

In this subsection, we analyze and deal with the two issues of over-confidence and limited improvement by classifier learning. Suppose weight of the classifier is $\boldsymbol{W} \in \mathbb{R}^{M\times K}$, where M is the number of features and K is the number of classes. The cross-entropy encourages the whole network to be over-confident on the head classes. The cross-entropy loss after the softmax activation is $l(y, \boldsymbol{p}) = -\log(\boldsymbol{p}_y) = -\boldsymbol{w}_y^\top \boldsymbol{x} + \log(\sum \exp(\boldsymbol{w}_i^\top \boldsymbol{x}))$, where $y \in \{1, 2, ..., K\}$ is the label. $\boldsymbol{x} \in \mathbb{R}^M$ is the feature vector send to classifier and $\boldsymbol{w}_i$ is the i-th column vector of $\boldsymbol{W}$. The optimal solution is $\boldsymbol{w}_y^{*\top}\boldsymbol{x} = \inf$, while other $\boldsymbol{w}_i^\top \boldsymbol{x}$, $i \neq y$ are small enough.

Because the head classes contain much more training examples, the network makes the weight norm $\|\boldsymbol{w}\|$ of the head classes larger to approach the optimal solution. It results in predicted probabilities mainly near 1.0 (see Fig. 3, the upper half in light blue). Another fact is that distributions of predicted probability are related to instance numbers. Unlike balanced recognition, applying different strategies for these classes is necessary for solving the long-tailed problem.

Here, we propose label-aware smoothing to solve the over-confidence in cross-entropy and varying distributions of predicted probability issues. It is expressed as

$$l(\boldsymbol{q}, \boldsymbol{p}) = -\sum_{i=1}^{K} \boldsymbol{q}_i \log \boldsymbol{p}_i, \qquad \boldsymbol{q}_i = \begin{cases} 1-\epsilon_y = 1 - f(N_y), & i = y, \\ \frac{\epsilon_y}{K-1} = \frac{f(N_y)}{K-1}, & \text{otherwise}, \end{cases} \tag{1}$$

where ϵ_y is a small label smoothing factor for Class-y, relating to its class number N_y. Now the optimal solution becomes (proof presented in Appendix E)

$$\boldsymbol{w}_i^{*\top}\boldsymbol{x} = \begin{cases} \log\left(\frac{(K-1)(1-\epsilon_y)}{\epsilon_y}\right) + c, & i = y, \\ c, & \text{otherwise}, \end{cases} \tag{2}$$

where c is an arbitrary real number. Compared with the optimal solution in cross-entropy, the label-aware smoothing encourages a finite output, more general and remedying overfit. We suppose the labels of the long-tailed dataset are assigned in a descending order concerning the number of instances, *i.e.*, $N_1 \geq N_2 \geq ... \geq N_K$. Because the head classes contain more diverse examples, the predicted probabilities are more promising than those of tail classes. Thus, we require the classes with larger instance numbers to be penalized with stronger label smoothing factors – that is, the related function $f(N_y)$ should be negatively correlated to N_y. We define three types of related function $f(N_y)$ as

- Concave form:

$$f(N_y) = \epsilon_K + (\epsilon_1 - \epsilon_K)\sin\left[\frac{\pi(N_y - N_K)}{2(N_1 - N_K)}\right]; \tag{3.a}$$

- Linear form:

$$f(N_y) = \epsilon_K + (\epsilon_1 - \epsilon_K)\frac{N_y - N_K}{N_1 - N_K}; \tag{3.b}$$

- Convex form:

$$f(N_y) = \epsilon_1 + (\epsilon_1 - \epsilon_K)\sin\left[\frac{3\pi}{2} + \frac{\pi(N_y - N_K)}{2(N_1 - N_K)}\right], \tag{3.c}$$

where ϵ_1 and ϵ_K are two hyperparameters. Illustration of these functions is shown in Fig. 6. If we set $\epsilon_1 \geq \epsilon_K$, $\epsilon_1 \geq \epsilon_2 \geq ... \geq \epsilon_K$ is obtained. For large instance number N_y for Class-y, label-aware smoothing allocates a strong smoothing factor. It lowers the fitting probability to relieve over-confidence because the head and medium classes are more likely to be over-confident than the tail classes (see Fig. 3).

As the form of label-aware smoothing is more complicated than cross-entropy, we propose a generalized classifier learning framework to fit it. Here we give a quick review about cRT and LWS. cRT learns a classifier weight, which contains KM learnable parameters, while LWS is restricted to learning the weight scaling vector $\boldsymbol{s} \in \mathbb{R}^K$ with only K learnable parameters.

In contrast, cRT has more learnable parameters and more powerful representation ability. LWS tends to obtain better validation losses and performance on large-scale datasets (refer to the experiment part in [15]). So LWS has a better generalization property. To combine the advantages of cRT and LWS, we design the classifier framework in Stage-2 as

$$\boldsymbol{z} = \mathrm{diag}(\boldsymbol{s})\left(r\boldsymbol{W} + \Delta\boldsymbol{W}\right)^{\top}\boldsymbol{x}. \tag{4}$$

In Eq. (4), we fix the original classifier weight $\boldsymbol{W}$ in Stage-2. If we make the learnable scaling vector $\boldsymbol{s}$ fixed, set $\boldsymbol{s} = \boldsymbol{1}$ and retention factor $r = 0$, and just learn the new classifier weight $\Delta\boldsymbol{W} \in \mathbb{R}^{M\times K}$, Eq. (4) degrades to cRT.

Because LWS fixes the original classifier weights $\boldsymbol{W}$ and only learns the scaling $\boldsymbol{s}$, Eq. (4) degrades to LWS if we set $r = 1$ and $\Delta\boldsymbol{W} = \boldsymbol{0}$. In most cases, LWS achieves better results on large-scale datasets. Thus, we let $\boldsymbol{s}$ learnable and set $r = 1$. We also make $\Delta\boldsymbol{W}$ learnable to improve the representation ability and optimize $\Delta\boldsymbol{W}$ by a different learning rate. $\Delta\boldsymbol{W}$ can be viewed as shift transformation on $\boldsymbol{W}$. It changes the direction of weight vector $\boldsymbol{w}$ in $\boldsymbol{W}$, which LWS does not similarly achieve.

3.3. Shift Learning on Batch Normalization

In the two-stage training framework, models are first trained with instance-balanced sampling in Stage-1 and then trained with class-balanced sampling in Stage-2. Since the framework involves two samplers, or two datasets – instance-balanced dataset $\mathcal{D}_{\mathrm{I}}$ and class-balanced dataset $\mathcal{D}_{\mathrm{C}}$ – we regard this two-stage training framework as a variant of transfer learning. If we view the two-stage decoupling training framework from the transfer learning perspective, fixing the backbone part and just tuning the classifier in Stage-2 are clearly unreasonable, especially for the batch normalization (BN) layers.

Specifically, we suppose the input to network is $\boldsymbol{x}_i$, the input feature of some BN layer is $g(\boldsymbol{x}_i)$, and the mini-batch size is m. The mean and running variance of Channel-j for these two stages are

$$\begin{aligned} &\boldsymbol{x}_i \sim P_{\mathcal{D}_{\mathrm{I}}}(\boldsymbol{x}, y), \quad \boldsymbol{\mu}_{\mathrm{I}}^{(j)} = \frac{1}{m}\sum_{i=1}^{m} g(\boldsymbol{x}_i)^{(j)}, \\ &\boldsymbol{\sigma}_{\mathrm{I}}^{2\,(j)} = \frac{1}{m}\sum_{i=1}^{m}\left[g(\boldsymbol{x}_i)^{(j)} - \boldsymbol{\mu}_{\mathrm{I}}^{(j)}\right]^2, \end{aligned} \tag{5}$$

$$\begin{aligned} &\boldsymbol{x}_i \sim P_{\mathcal{D}_{\mathrm{C}}}(\boldsymbol{x}, y), \quad \boldsymbol{\mu}_{\mathrm{C}}^{(j)} = \frac{1}{m}\sum_{i=1}^{m} g(\boldsymbol{x}_i)^{(j)}, \\ &\boldsymbol{\sigma}_{\mathrm{C}}^{2\,(j)} = \frac{1}{m}\sum_{i=1}^{m}\left[g(\boldsymbol{x}_i)^{(j)} - \boldsymbol{\mu}_{\mathrm{C}}^{(j)}\right]^2. \end{aligned} \tag{6}$$

Due to different sampling strategies, the composition ratios of head, medium, and tail classes are also different, which lead to $P_{\mathcal{D}_{\mathrm{I}}}(\boldsymbol{x}, y) \neq P_{\mathcal{D}_{\mathrm{C}}}(\boldsymbol{x}, y)$. By Eqs. (5) and (6), there exist biases in $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ under two sampling strategies, *i.e.*, $\boldsymbol{\mu}_{\mathrm{I}} \neq \boldsymbol{\mu}_{\mathrm{C}}$ and $\boldsymbol{\sigma}_{\mathrm{I}}^2 \neq \boldsymbol{\sigma}_{\mathrm{C}}^2$. Thus, it is infeasible for the decoupling framework that BN shares mean and variance across datasets with two sampling strategies. Motivated by AdaBN [17] and TransNorm [33], we update the running mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ and yet fix the learnable linear transformation parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for better normalization in Stage-2.

4. Experiments

4.1. Datasets and Setup

Our experimental setup including the implementation details and evaluation protocol mainly follows [4] for CIFAR-10-LT and CIFAR-100-LT, and [15] for ImageNet-LT, Places-LT, and iNuturalist 2018. Please see Appendix A for more details of training and hyperparameter setting.

4.1.1 Datasets Explanation

CIFAR-10 and CIFAR-100. CIFAR-10 and CIFAR-100 both have 60,000 images, 50,000 for training and 10,000 for validation with 10 categories and 100 categories. For fair comparison, we use the long-tailed versions of CIFAR datasets with the same setting as those used in [4]. It is by controlling the degrees of data imbalance with an imbalanced factor $\beta = \frac{N_{\max}}{N_{\min}}$, where $N_{\max}$ and $N_{\min}$ are the numbers of training samples for the most and the least frequent classes. Following Cao *et al*. [4] and Zhou *et al*. [39], we conduct experiments with IF 100, 50, and 10.

ImageNet-LT and Places-LT. ImageNet-LT and Places-LT were proposed by Liu *et al*. [20]. ImageNet-LT is a long-tailed version of the large-scale object classification dataset ImageNet [27] by sampling a subset following the Pareto distribution with power value $\alpha = 6$. It contains 115.8K images from 1,000 categories, with class cardinality ranging from 5 to 1,280. Places-LT is a long-tailed version of the large-scale scene classification dataset Places [40]. It consists of 184.5K images from 365 categories with class cardinality ranging from 5 to 4,980.

iNaturalist 2018. iNaturalist 2018 [31] is a classification dataset, which is on a large scale and suffers from extremely imbalanced label distribution. It is composed of 437.5K images from 8,142 categories. In addition, on iNaturalist 2018 dataset, we also face the fine-grained problem.

4.1.2 Implementation Details

For all experiments, we use the SGD optimizer with momentum 0.9 to optimize networks. For CIFAR-LT, we mainly follow Cao *et al*. [4]. We train all MiSLAS models with the ResNet-32 backbone on one GPU and use the multistep

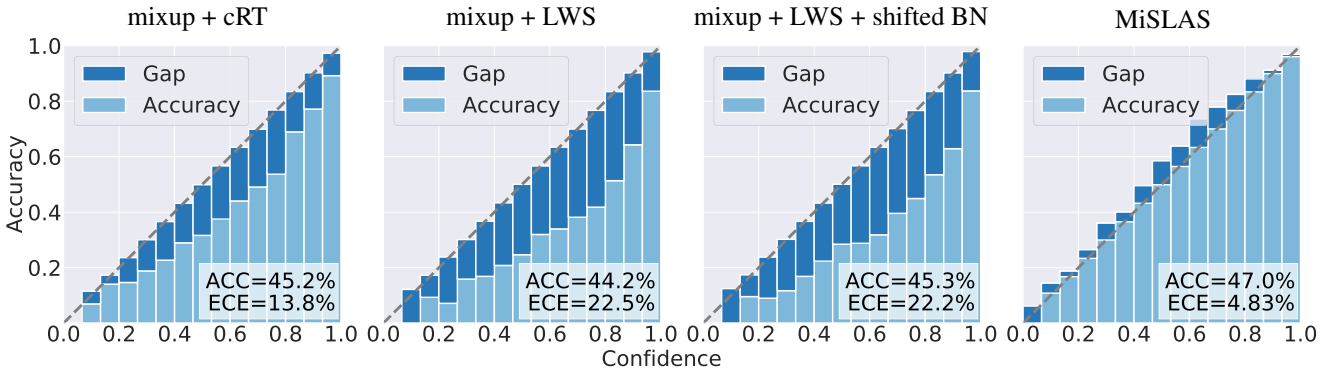

Figure 4: Reliability diagrams of ResNet-32 trained on CIFAR-100-LT with IF 100. From left to right: cRT with mixup, LWS with mixup, LWS with mixup and shifted BN, and MiSLAS (*complying with Fig. 1*).

learning rate schedule, which decreases the learning rate by 0.1 at the 160^{th} and 180^{th} epochs in Stage-1. For ImageNet-LT, Places-LT, and iNaturalist 2018, we mainly follow Kang *et al*. [15] and use the cosine learning rate schedule [21] to train all MiSLAS models with the ResNet-10, 50, 101, and 152 backbones on four GPUs.

4.2. Ablation Study

Calibration performance. Here we show the reliability diagrams with 15 bins of our methods on CIFAR-100-LT with IF 100 in Fig. 4. Comparing with Fig. 1, both mixup and label-aware smoothing can not only largely enhance the network calibration (even lower ECEs than those on balanced datasets) but also greatly improve the performance for long-tailed recognition. The similar trends can also be noticed on CIFAR-10-LT, ImageNet-LT, and Places-LT (see Table 1 and figures in Appendix C for detail), which proves the power of the proposed method on calibration.

All experimental results show that the training networks on imbalanced datasets lead to severe over-confidence. Since the conventional mixup and label-smoothing both contain the operation of softening the ground truth labels, it may suggest that training with hard labels is likely to be another contributing factor leading to network over-confidence.

Comparing re-weighting with label-aware smoothing. Here we compare the proposed label-aware smoothing (LAS) with the re-weighting methods. The main difference is on label transformation. In particular, label-aware smoothing changes the hard label to the soft version based on label distribution (see the otherwise case of Eq. (1): $\boldsymbol{q}_i = \frac{f(N_y)}{K-1}$, $i \neq y$). While re-weighting methods do not contain such critical transformation and just set the values to zero by $\boldsymbol{q}_i = 0, i \neq y$.

Further, due to the transformation of labels, the optimal solution of $\boldsymbol{w}_i^{*\top}\boldsymbol{x}$ in LAS becomes Eq. (2). In contrast, the optimal solution of re-weighting is the same as that of cross-entropy $\boldsymbol{w}_i^{*\top}\boldsymbol{x} = \inf$, which cannot properly change the predicted distribution and leads to over-confidence. Based on our experimental results in Table 2, using the re-weighting method in Stage-2 degrades performance and calibration compared with the case of LAS.

Method	100	50	10
CB-CE [7]	44.3 / 20.2	50.5 / 19.1	62.5 / 13.9
LAS	**47.0 / 4.83**	**52.3 / 2.25**	**63.2 / 1.73**

Table 2: Comparison in terms of test accuracy (%) / ECE (%) of label-aware smoothing (LAS) with re-weighting, class-balanced cross-entropy (CB-CE, [7]) in Stage-2. Both models are based on ResNet-32 and trained on CIFAR-100-LT with IF 100, 50, and 10.

How ϵ_1 and ϵ_K affect label-aware smoothing? In our label-aware smoothing, there are two hyperparameters in Eqs. (3.a), (3.b), and (3.c). They are ϵ_1 and ϵ_K, which control penalty of classes. In a recognition system, if the predicted probability of Class-y is larger than 0.5, the classifier would classify the input to Class-y. Thus, to make it reasonable, we limit $0 \leq \epsilon_K \leq \epsilon_1 \leq 0.5$.

Here we conduct experiments by varying ϵ_1 and ϵ_K both from 0.0 to 0.5 on CIFAR-10-LT with IF 100. We plot the performance matrix upon ϵ_1 and ϵ_K in Fig. 5 for all possible variants. It shows that the classification accuracy is further improved by 3.3% comparing with conventional cross-entropy ($\epsilon_1 = 0$ and $\epsilon_K = 0$, green square) when we pick $\epsilon_1 = 0.3$, and $\epsilon_K = 0.0$ (orange square) for label-aware smoothing. Consistent improvement 0.9% is yielded on CIFAR-100-LT with IF 100 when picking $\epsilon_1 = 0.4$ and $\epsilon_K = 0.1$ for label-aware smoothing.

How $f(\cdot)$ affects label-aware smoothing? As discussed in Sec. 3.2, the related function $f(\cdot)$ may play a significant role for the final model performance. We draw illustration of Eqs. (3.a), (3.b), and (3.c) in Fig. 6. For CIFAR-100-LT with IF 100, we set $K = 100$, $N_1 = 500$, and $N_{100} = 5$. Based on the ablation study results of ϵ_1 and ϵ_K mentioned

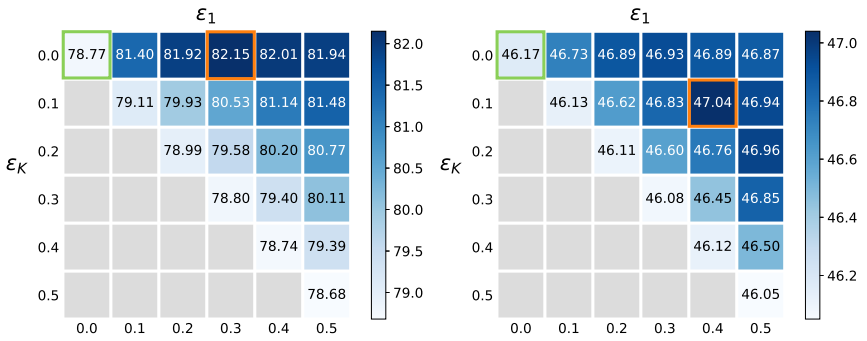

Figure 5: Ablation study of two hyperparameters ϵ_1 and ϵ_K in label-aware smoothing. Heat map visualization on CIFAR-10-LT with IF 100 (left) and on CIFAR-100-LT with IF 100 (right).

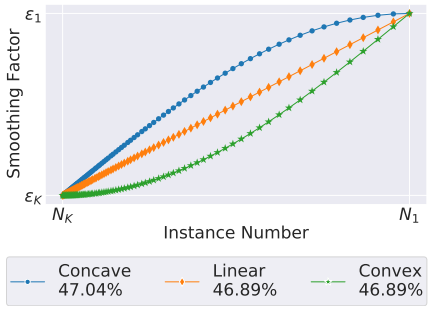

Figure 6: Function illustration and the test performance of Eqs. (3.a), (3.b), and (3.c). Concave form achieves the best result.

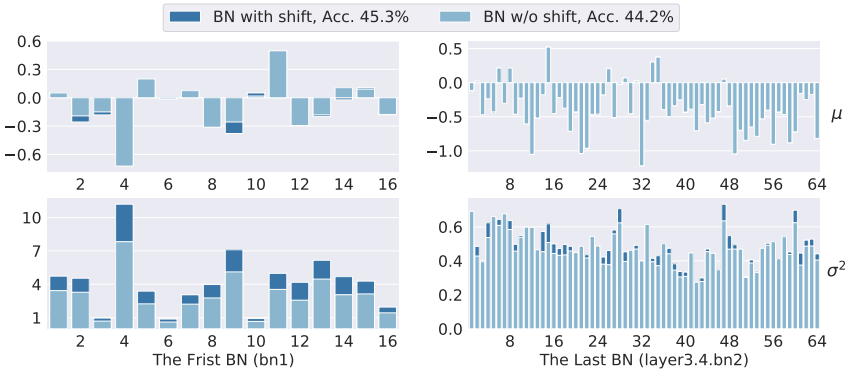

Figure 7: Visualization of the changes in the running mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. The ResNet-32 based model is trained on CIFAR-100-LT with IF 100. Left: $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ in the first BN of ResNet-32, which contains 16 channels. Right: $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ in the last BN of ResNet-32, which contains 64 channels.

Module			CIFAR-100-LT		
MU	SL	LAS	100	50	10
☒	☒	☒	41.2	46.0	58.5
☑	☒	☒	44.2	50.6	62.2
☑	☑	☒	45.3	51.4	62.8
☑	☑	☑	**47.0**	**52.3**	**63.2**
☒	☒	☒	36.3	34.2	27.5
☑	☒	☒	22.5	18.4	14.3
☑	☑	☒	22.2	19.2	13.7
☑	☑	☑	**4.83**	**2.25**	**1.73**

Table 3: Ablation study for all proposed modules on CIFAR-100-LT. Top: accuracy (%). Bottom: ECE (%). MU: applying mixup only in Stage-1. SL: shift learning on BN. LAS: label-aware smoothing.

above, we set $\epsilon_1 = 0.4$ and $\epsilon_{100} = 0.1$ here. After tuning for 10 epochs in Stage-2, accuracy of the concave model is the best. We also exploit other forms, *e.g.*, exponential form of $f(\cdot)$, in Appendix B. The gain of changing form is quite limited compared with varying ϵ_1 and ϵ_K.

How label-aware smoothing affects prediction distribution? To visualize the change in predicted probability distributions, we train two LWS models, one with cross-entropy and the other with label-aware smoothing on CIFAR-100-LT with IF 100. The cross-entropy-based distributions of the head, medium, and tail classes are shown in the upper part of Fig. 3 in light blue. The label-aware smoothing-based distributions are in the bottom half in deep blue. We observe that the over-confidence of head and medium classes is much reduced, and the whole distribution of the tail classes slightly moves right when using label-aware smoothing. These empirical results are consistent with our analysis in Sec. 3.2.

Further analysis of shift learning. In this part, we conduct experiments to show the effectiveness and suitability of shift learning on BN. We train the LWS model on CIFAR-100-LT with IF 100. After 10-epoch finetuning in Stage-2, the model trained with BN shifting achieves accuracy 45.3%, 1.1% higher than that without BN shifting. We also visualize the change in BN. As shown in Fig. 7, there exist biases in $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ between datasets using different sampling strategies.

Due to different composition ratios of the head, medium and tail classes, the statistic mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ vary. We also notice intriguing phenomena in Fig. 7: (i) the change in variance $\boldsymbol{\sigma}^2$ is larger than that on mean $\boldsymbol{\mu}$. (ii) Change of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ in the deep BN layers is much smaller than that in the shallow BN layers.

Summary. Overall, Table 3 shows the ablation investigation on the effects of mixup (adding mixup in Stage-1, MU), shift learning on batch normalization (SL), and label-aware smoothing (LAS). We note each proposed module can not only improves accuracy (top of Table 3), but also greatly relieves over-confidence (bottom of Table 3) on CIFAR-100-LT for all commonly-used imbalanced factors, *i.e.*, 100, 50, and 10. They firmly manifest the effectiveness.

Method	CIFAR-10-LT			CIFAR-100-LT		
	100	50	10	100	50	10
CE	70.4	74.8	86.4	38.4	43.9	55.8
mixup [37]	73.1	77.8	87.1	39.6	45.0	58.2
LDAM+DRW [4]	77.1	81.1	88.4	42.1	46.7	58.8
BBN(include mixup) [39]	79.9	82.2	88.4	42.6	47.1	59.2
Remix+DRW(300 epochs) [5]	79.8	-	89.1	46.8	-	61.3
cRT+mixup	79.1 / 10.6	84.2 / 6.89	89.8 / 3.92	45.1 / 13.8	50.9 / 10.8	62.1 / 6.83
LWS+mixup	76.3 / 15.6	82.6 / 11.0	89.6 / 5.41	44.2 / 22.5	50.7 / 19.2	62.3 / 13.4
MiSLAS	**82.1 / 3.70**	**85.7 / 2.17**	**90.0 / 1.20**	**47.0 / 4.83**	**52.3 / 2.25**	**63.2 / 1.73**

Table 4: Top-1 accuracy (%) / ECE (%) for ResNet-32 based models trained on CIFAR-10-LT and CIFAR-100-LT.

Method	ResNet-50
CE	44.6
CE+DRW [4]	48.5
Focal+DRW [18]	47.9
LDAM+DRW [4]	48.8
CRT+mixup	51.7 / 5.62
LWS+mixup	52.0 / 2.23
MiSLAS	**52.7 / 1.83**

(a) ImageNet-LT

Method	ResNet-50
CB-Focal [7]	61.1
LDAM+DRW [4]	68.0
BBN(include mixup) [39]	69.6
Remix+DRW [5]	70.5
cRT+mixup	70.2 / **1.79**
LWS+mixup(under-conf.)	70.9 / 9.41
MiSLAS(under-conf.)	**71.6** / 7.67

(b) iNaturalist 2018

Method	ResNet-152
Range Loss [38]	35.1
FSLwF [8]	34.9
OLTR [20]	35.9
OLTR+LFME [35]	36.2
cRT+mixup	38.3 / 12.4
LWS+mixup	39.7 / 11.7
MiSLAS	**40.4 / 3.59**

(c) Places-LT

Table 5: Top-1 accuracy (%) / ECE (%) on ImageNet-LT (left), iNaturalist 2018 (center) and Places-LT (right).

4.3. Comparison with State-of-the-arts

To verify the effectivity, we compare the proposed method against previous one-stage methods of Range Loss [38], LDAM Loss [4], FSLwF [8], and OLTR [20], and against previous two-stage methods, including DRS-like, DRW-like [4], LFME [35], cRT, and LWS [15]. For fair comparison, we add mixup on the LWS and cRT models. Remix [5] is a recently proposed augmentation method for long-tail recognition. Because BBN [39] has double samplers and is trained in a mixup-like manner, we directly compare our method with it.

Experimental results on CIFAR-LT. We conduct extensive experiments on CIFAR-10-LT and CIFAR-100-LT with IF 100, 50, and 10, using the same setting as previous work [4, 39]. The results are summarized in Table 4. Compared with previous methods, our MiSLAS outperforms all previous methods by consistently large margins both in top-1 accuracy and ECE. Moreover, the superiority holds for all imbalanced factors, *i.e.*, 100, 50, and 10, on both CIFAR-10-LT and CIFAR-100-LT.

Experimental results on large-scale datasets. We further verify the effectiveness of our method on three large-scale imbalanced datasets, *i.e.*, ImageNet-LT, iNaturalist 2018, and Places-LT. Table 5 lists experimental results on ImageNet-LT (left), iNaturalist 2018 (center), and Places-LT (right). Notably, our MiSLAS outperforms other approaches and sets a new state-of-the-art with better accuracy and confidence calibration on almost all three large-scale long-tailed benchmark datasets. More results about the split class accuracy and different backbones on these three datasets are listed in Appendix D.

5. Conclusion

In this paper, we have discovered that models trained on long-tailed datasets are more miscalibrated and over-confident than those trained on balanced datasets. We accordingly propose two solutions of using mixup and designing label-aware smoothing to handle different degrees of over-confidence for classes. We note the dataset bias (or domain shift) in two-stage resampling methods for long-tailed recognition. To reduce dataset bias in the decoupling framework, we propose shift learning on the batch normalization layer, which further improves the performance. Extensive quantitative and qualitative experiments on various benchmarks show that our MiSLAS achieves decent performance for both top-1 recognition accuracy and confidence calibration, and makes a new state-of-the-art.

References

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[2] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[3] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881, 2019.

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pages 1567–1578, 2019.

[5] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *ECCVW*, 2020.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019.

[8] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[13] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[14] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, 2012.

[15] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[17] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[20] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019.

[21] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *ICLR*, 2017.

[22] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, pages 15288–15299, 2020.

[23] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *NeurIPS*, pages 4694–4703, 2019.

[24] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, pages 625–632, 2005.

[25] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[28] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, pages 467–482, 2016.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[30] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, pages 13888–13899, 2019.

[31] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018.

[32] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019.

[33] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *NeurIPS*, pages 1953–1963, 2019.

[34] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, pages 7029–7039, 2017.

[35] Liuyu Xiang and Guiguang Ding. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *ECCV*, 2020.

[36] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019.

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

[38] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, pages 5409–5418, 2017.

[39] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, pages 9719–9728, 2020.

[40] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2017.