Towards Calibrated Model for Long-Tailed Visual Recognition from Prior Perspective

Anonymous Author(s)
Affiliation
Address
email

Abstract

Real-world data universally confronts a severe class-imbalance problem and exhibits a *long-tailed* distribution, i.e., most labels are associated with limited instances. The naïve models supervised by such datasets would prefer dominant labels, encounter a serious generalization challenge and become poorly calibrated. We propose two novel methods from the *prior* perspective to alleviate this dilemma. First, we deduce a balance-oriented data augmentation named Uniform Mixup (UniMix) to promote *mixup* in long-tailed scenarios, which adopts advanced mixing factor and sampler in favor of the minority. Second, motivated by the Bayesian theory, we figure out the Bayes Bias (Bayias), an inherent bias caused by the inconsistency of *prior*, and compensate it as a modification on standard cross-entropy loss. We further prove that both the proposed methods ensure the classification *calibration* theoretically and empirically. Extensive experiments verify that our strategies contribute to a better-calibrated model and their combination achieves state-of-the-art performance on CIFAR-LT, ImageNet-LT, and iNaturalist 2018.

1 Introduction

Balanced and large-scaled datasets [43, 35] have promoted deep neural networks to achieve remarkable success in many visual tasks [20, 42, 18]. However, real-world data typically exhibits a *long-tailed* (LT) distribution [31, 36, 22, 14], and collecting a minority category (**tail**) sample always leads to more occurrences of common classes (**head**) [47, 23], resulting in most labels associated with limited instances. The paucity of samples may cause insufficient feature learning on the tail classes [56, 10, 30, 36], and such data imbalance will bias the model towards dominant labels [46, 47, 33]. Hence, the generalization of minority categories is an enormous challenge.

The intuitive approaches such as directly over-sampling the tail [7, 3, 39, 44, 4] or under-sampling the head [15, 3, 17] will cause serious robustness problems. *mixup* [53] and its extensions [49, 52, 8] are effective feature improvement methods and contribute to a well-calibrated model in balanced datasets [48, 54], i.e., *the predicted confidence indicates actual accuracy likelihood* [13, 48]. However, *mixup* is inadequately calibrated in an imbalanced LT scenario (Fig.1). In this paper, we raise a conception called ξ-*Aug* to analyze *mixup* and figure out that it tends to generate more head-head pairs, resulting in unsatisfactory generalization of the tail. Therefore, we propose Uniform Mixup (UniMix), which adopts a tail-favored *mixing factor* related to label *prior* and a *inverse sampling strategy* to encourage more head-tail pairs occurrence for better generalization and *calibration*.

Previous works adjust the logits *weight* [28, 10, 46, 51] or *margin* [5, 38] on standard *Softmax* cross-entropy (CE) loss to tackle the bias towards dominant labels. We analyze the inconstancy of label *prior*, which varies in LT train set and balanced test set, and pinpoint an inherent bias named Bayes Bias (Bayias). Based on the Bayesian theory, the *posterior* is proportional to *prior* times *likelihood*. Hence, it's necessary to adjust the *posterior* on train set by compensating different *prior* for each class,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

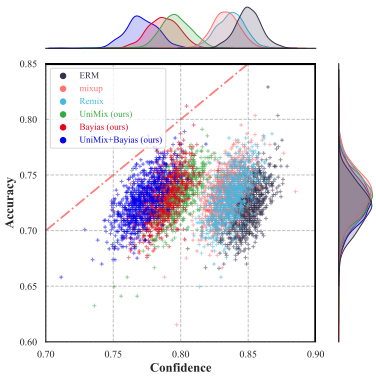


Figure 1: Joint density plots of accuracy vs. confidence to measure the *calibration* of classifiers on CIFAR-100-LT-100 during training. A well-calibrated classifier's density will lay around the red dot line $y = x$, indicating prediction score reflects the actual likelihood of accuracy. *mixup* manages to regularize classifier on balanced datasets. However, both *mixup* and its extensions tend to be overconfident in LT scenarios. Our UniMix reconstructs a more balanced dataset and Bayias-compensated CE erases *prior* bias to ensure better *calibration*. Without loss of accuracy, either of proposed methods trains the same classifier more calibrated and their combination achieves the best. How to measure *calibration* and more visualization results are available in Appendix D.2.

which can serve as a additional *margin* on CE. We further demonstrate that the Bayias-compensated CE ensures classification *calibration* and propose a unified learning manner to combine Bayias with UniMix towards a better-calibrated model (see in Fig.1). Furthermore, we suggest that bad calibrated approaches are counterproductive with each other, which provides a heuristic way to analyze the combined results of different feature improvement and loss modification methods (see in Tab.3).

In summary, our contributions are: 1) We raise the concept of ξ-*Aug* to theoretically explain the reason of *mixup*'s miscalibration in LT scenarios and propose Unimix (Sec.3.1) composed of novel mixing and sampling strategies to construct a more class-balanced virtual dataset. 2) We propose the Bayias (Sec.3.2) to compensate the bias incured by different label *prior*, which can be unified with UniMix by a training manner for better classification *calibration*. 3) We conduct sufficient experiments to demonstrate that our method trains a well-calibrated model and achieves state-of-the-art results on CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018.

2 Analysis of *mixup*

The core of supervised image classification is to find a θ parameterized mapping $\mathcal{F}_\theta : X \in \mathbb{R}^{c\times h\times w} \mapsto Y \in \mathbb{R}^{C\times 1}$ to estimate the empirical Dirac delta distribution $\mathbb{P}_\delta(x, y) = \frac{1}{N}\sum_{i=1}^{N}\delta(x_i, y_i)$ of N instances $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$. The learning progress by minimizing Eq.1 is known as Empirical Risk Minimization (ERM), where $\mathcal{L}(Y = y_i, \mathcal{F}_\theta(X = x_i))$ is x_i's conditional risk.

$$R_\delta(\mathcal{F}_\theta) = \int_{x\in\mathcal{X}} \mathcal{L}\left(Y = y, \mathcal{F}_\theta(X = x)\right) d\mathbb{P}_\delta(x, y) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}\left(Y = y_i, \mathcal{F}_\theta(X = x_i)\right) \quad (1)$$

To overcome the over-fitting caused by insufficient training of N samples, *mixup* utilizes Eq.2 to extend the feature space to its vicinity based on Vicinal Risk Minimization (VRM) [6].

$$\widetilde{x} = \xi \cdot x_i + (1-\xi)\cdot x_j \qquad \widetilde{y} = \xi \cdot y_i + (1-\xi)\cdot y_j \quad (2)$$

where $\xi \sim Beta(\alpha, \alpha), \alpha \in [0, 1]$, the sample pair $(x_i, y_i), (x_j, y_j)$ is drawn from training dataset $\mathcal{D}_{train}$ randomly. Hence, Eq.2 converts $\mathbb{P}_\delta(X, Y)$ into empirical *vicinal distribution* $\mathbb{P}_\nu(\widetilde{x}, \widetilde{y}) = \frac{1}{N}\sum_{i=1}^{N}\nu(\widetilde{x}, \widetilde{y}|x_i, y_i)$, where $\nu(\cdot)$ describes the manner of finding virtual pairs $(\widetilde{x}, \widetilde{y})$ in the vicinity of arbitrary sample (x_i, y_i). Then, we construct a new dataset $\mathcal{D}_\nu := \{(\widetilde{x}_k, \widetilde{y}_k)\}_{k=1}^{M}$ via Eq.2 and minimize the empirical vicinal risk by Vicinal Risk Minimization (VRM):

$$R_\nu(\mathcal{F}_\theta) = \int_{\widetilde{x}\in\widetilde{\mathcal{X}}} \mathcal{L}\left(\widetilde{Y} = \widetilde{y}, \mathcal{F}_\theta(\widetilde{X} = \widetilde{x})\right) d\mathbb{P}_\nu(\widetilde{x}, \widetilde{y}) = \frac{1}{M}\sum\nolimits_{i=1}^{M} \mathcal{L}\left(\widetilde{Y} = \widetilde{y}_i, \mathcal{F}_\theta(\widetilde{X} = \widetilde{x}_i)\right) \quad (3)$$

mixup is proven to be effective on balanced dataset due to its improvement of *calibration* [48, 13], but it is unsatisfactory in LT scenarios (see in Tab.3). In Fig.1, *mixup* fails to train a calibrated model, which surpasses baseline (ERM) a little in accuracy and seldom contributes to *calibration* (far from $y = x$). To analyze the insufficiency of *mixup*, the definition of ξ-*Aug* is raised.

Definition 1 ξ-*Aug*. *The virtual sample $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ generated by Eq.2 with mix factor ξ is defined as a ξ-Aug sample, which is a robust sample of class y_i (class y_j) iff $\xi \geq 0.5 (\xi < 0.5)$ that contributes to class y_i (class y_j) in model's feature learning.*

In LT scenarios, we reasonably assume the instance number n of each class is *exponential* with parameter λ [10] if indices are descending sorted by n_{y_i} where $y_i \in [1, C]$ and C is the total class number. Generally, the imbalance factor is defined as $\rho = n_{y_1}/n_{y_C}$ to measure how skewed the LT dataset is. It is easy to draw $\lambda = \ln \rho/(C-1)$. Hence, we can describe the LT dataset as Eq.4:

$$\mathbb{P}(Y = y_i) = \frac{\iint_{x_i \in \mathcal{X}, y_j \in \mathcal{Y}} \mathbb{1}(X = x_i, Y = y_i) dx_i dy_j}{\iint_{x_i \in \mathcal{X}, y_j \in \mathcal{Y}} \mathbb{1}(X = x_i, Y = y_j) dx_i dy_j} = \frac{\lambda}{e^{-\lambda} - e^{-\lambda C}} e^{-\lambda y_i}, y_i \in [1, C] \tag{4}$$

Then, we derive the following corollary to illustrate the limitation of naïve *mixup* strategy.

Corollary 1 *When $\boldsymbol{\xi} \sim \boldsymbol{Beta(\alpha, \alpha)}, \alpha \in [0, 1]$, the newly mixed dataset $\mathcal{D}_\nu$ composed of ξ-Aug samples $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ follows the same long-tailed distribution as the origin dataset $\mathcal{D}_{train}$, where (x_i, y_i) and (x_j, y_j) are **randomly** sampled from $\mathcal{D}_{train}$. (See detail derivation in Appendix A.2.)*

$$\begin{aligned} \mathbb{P}_{mixup}(Y^* = y_i) &= \mathbb{P}^2(Y = y_i) + \mathbb{P}(Y = y_i) \iint_{y_i \neq y_j} Beta(\alpha, \alpha) \mathbb{P}(Y = y_j) d\xi dy_j \\ &= \frac{\lambda}{e^{-\lambda} - e^{-\lambda C}} e^{-\lambda y_i}, y_i \in [1, C] \end{aligned} \tag{5}$$

In *mixup*, the probability of any $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ belongs to class y_i or class y_j is strictly determined by ξ and $\mathbb{E}(\xi) \equiv 0.5$. Furthermore, both (x_i, y_i) and (x_j, y_j) are randomly sampled and concentrated on the head instead of tail, resulting in that the head classes get more ξ-*Aug* samples than the tail ones.

3 Methodology

3.1 UniMix: balance-oriented feature improvement

mixup and its extensions tend to generate head-majority pseudo data, which leads to the deficiency on the tail feature learning and results in a bad-calibrated model. To obtain a more balanced dataset $\mathcal{D}_\nu$, we propose the UniMix Factor ξ^*_{ij} related to the *prior* probability of each category and a novel UniMix Sampler to obtain sample pairs. Our motivation is to generate comparable ξ-*Aug* samples of each class for better generalization and *calibration*.

UniMix Factor. Specifically, the *prior* in imbalanced train set and balanced test set of class y_i is defined as $\mathbb{P}_{train}(Y = y_i) \triangleq \pi_{y_i}$, and $\mathbb{P}_{test}(Y = y_i) \equiv 1/C$, respectively. We design the UniMix Factor $\xi^*_{i,j}$ for each virtual sample $\widetilde{x}_{i,j}$ instead of a fixed ξ in *mixup*. Consider adjusting ξ with the class *prior* probability π_{y_i}, π_{y_j}. It is intuitive that a proper factor $\xi_{ij} = \pi_{y_j}/(\pi_{y_i} + \pi_{y_j})$ ensures $\widetilde{x}_{i,j}$ to be a ξ-*Aug* sample of class y_j if $\pi_{y_i} \geq \pi_{y_j}$, i.e., class y_i occupies more instances than class y_j.

However, $\xi_{i,j}$ is uniquely determined by π_{y_i}, π_{y_j}. To improve the robustness and generalization, original $Beta(\alpha, \alpha)$ is adjusted to obtain UniMix Factor $\xi^*_{i,j}$. Notice that ξ is close to 0 or 1 and symmetric at 0.5, we hence transform it to maximize the probability of $\xi_{ij} = \pi_{y_j}/(\pi_{y_i} + \pi_{y_j})$ and its vicinity. Specifically, if note $\xi \sim Beta(\alpha, \alpha)$ as $f(\xi; \alpha, \alpha)$, we define $\xi^*_{i,j} \sim \mathscr{U}(\pi_{y_i}, \pi_{y_j}, \alpha, \alpha)$ as:

$$\xi^*_{i,j} \sim \mathscr{U}(\pi_{y_i}, \pi_{y_j}, \alpha, \alpha) = \begin{cases} f(\xi^*_{i,j} - \dfrac{\pi_{y_j}}{\pi_{y_i} + \pi_{y_j}} + 1; \alpha, \alpha), & \xi^*_{i,j} \in [0, \dfrac{\pi_{y_j}}{\pi_{y_i} + \pi_{y_j}}); \\ f(\xi^*_{i,j} - \dfrac{\pi_{y_j}}{\pi_{y_i} + \pi_{y_j}}; \alpha, \alpha), & \xi^*_{i,j} \in [\dfrac{\pi_{y_j}}{\pi_{y_i} + \pi_{y_j}}, 1] \end{cases} \tag{6}$$

Rethink Eq.2 with $\xi^*_{i,j}$ described as Eq.6:

$$\widetilde{x}_{i,j} = \xi^*_{i,j} \cdot x_i + (1 - \xi^*_{i,j}) \cdot x_j \qquad \widetilde{y}_{i,j} = \xi^*_{i,j} \cdot y_i + (1 - \xi^*_{i,j}) \cdot y_j \tag{7}$$

We have the following corollary to show how ξ^*_{ij} ameliorates the imbalance of $\mathcal{D}_{train}$:

Corollary 2 *When $\boldsymbol{\xi^*_{i,j} \sim \mathscr{U}(\pi_{y_i}, \pi_{y_j}, \alpha, \alpha)}, \alpha \in [0, 1]$, the newly mixed dataset $\mathcal{D}_\nu$ composed of ξ-Aug samples $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ follows a middle-majority distribution (see Fig.2), where (x_i, y_i) and (x_j, y_j) are both **randomly** sampled from $\mathcal{D}_{train}$. (See detail derivation in Appendix A.3.)*

$$\begin{aligned} \mathbb{P}^*_{mixup}(Y^* = y_i) &= \mathbb{P}(Y = y_i) \int_{y_j < y_i} \mathbb{1}\left(\int \xi^*_{ij} \mathscr{U}(\pi_i, \pi_j, \alpha, \alpha) d\xi^*_{ij} \geq 0.5 \right) \mathbb{P}(Y = y_j) dy_j \\ &= \frac{\lambda}{(e^{-\lambda} - e^{-C\lambda})^2} \left(e^{-\lambda(y_i+1)} - e^{-2\lambda y_i} \right), y_i \in [1, C] \end{aligned} \tag{8}$$

UniMix Sampler. UniMix Factor facilitates ξ-*Aug* samples more balance-distributed over all classes. However, most samples are still ξ-*Aug* for the head or middle (see Fig.2(green)). Actually, the constraint that pair x_i, x_j drawn from the head and tail respectively is preferred, which dominantly generates ξ-*Aug* samples for tail classes with $\xi^*_{i,j}$. To this end, we consider sample x_j from $\mathcal{D}_{train}$ with probability inverse to the label *prior*:

$$\mathbb{P}_{inv}(Y = y_i) = \frac{\mathbb{P}^{\tau}(Y = y_i)}{\int_{y_j \in \mathcal{Y}} \mathbb{P}^{\tau}(Y = y_j) dy_j} \tag{9}$$

When $\tau = 1$, UniMix Sampler is equivalent to a random sampler. $\tau < 1$ indicates that x_j has higher probability drawn from tail class. Note that x_i is still randomly sampled from $\mathcal{D}_{train}$, i.e., it's most likely drawn from the majority class. The virtual sample $\widetilde{x}_{i,j}$ obtained in this manner is mainly a ξ-*Aug* sample of the tail composite with x_i from the head. Hence Corollary3 is derived:

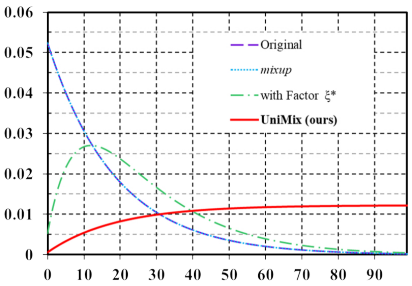


Figure 2: Visualization of ξ-*Aug* samples distribution ($C = 100, \rho = 200$) in Corollary1,2,3. x-axis: class induces. y-axis: probability of each class. *mixup* (blue) exhibits the same LT distribution as origin (purple). ξ^* (green) alleviates such situation and the full pipeline ($\tau = -1$) (red) constructs a more uniform distributed dataset. See more results in Appendix A.5.

Corollary 3 *When* $\boldsymbol{\xi^*_{i,j}} \sim \mathscr{U}(\boldsymbol{\pi_{y_i}}, \boldsymbol{\pi_{y_j}}, \boldsymbol{\alpha}, \boldsymbol{\alpha}), \alpha \in [0, 1]$, *the newly mixed dataset* $\mathcal{D}_{\nu}$ *composed of* ξ-*Aug samples* $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ *follows a tail-majority distribution (see Fig.2), where* (x_i, y_i) *is* ***randomly*** *and* (x_j, y_j) *is* ***inversely*** *sampled from* $\mathcal{D}_{train}$, *respectively. (See detail derivation in Appendix A.4.)*

$$\begin{aligned}\mathbb{P}_{UniMix}(Y^* = y_i) &= \mathbb{P}(Y = y_i) \int_{y_j < y_i} \mathbb{1}\left(\int \xi^*_{ij} \mathscr{U}(\pi_i, \pi_j, \alpha, \alpha) d\xi^*_{ij} \geq 0.5\right) \mathbb{P}_{inv}(Y = y_j) dy_j \\ &= \frac{\lambda}{(e^{-\lambda} - e^{-C\lambda})(e^{-C\tau\lambda} - e^{-\tau\lambda})} \left(e^{-\lambda y_i(\tau+1)} - e^{-\lambda(\tau + y_i)}\right), y_i \in [1, C]\end{aligned} \tag{10}$$

With the proposed UniMix Factor and UniMix Sampler, we get the complete UniMix manner, which constructs a uniform ξ-*Aug* samples distribution for VRM and greatly facilitates model's *calibration* (See Fig.2 (red) & 1). We construct $\mathcal{D}_{\nu} := \{(\widetilde{x}_k, \widetilde{y}_k)\}_{k=1}^{M}$ where $\{\widetilde{x}_k, \widetilde{y}_k\}$ is $(\widetilde{x}_{i,j}, \widetilde{y}_{i,j})$ generated by (x_i, y_i) and (x_j, y_j). We conduct training via Eq.3 and the loss via VRM is available as:

$$\mathcal{L}(\widetilde{y}_k, \mathcal{F}_{\theta}(\widetilde{x}_k)) = \xi^*_{i,j} \mathcal{L}(y_i, \mathcal{F}_{\theta}(\widetilde{x}_{i,j})) + (1 - \xi^*_{i,j}) \mathcal{L}(y_j, \mathcal{F}_{\theta}(\widetilde{x}_{i,j})) \tag{11}$$

3.2 Bayias: an inherent bias in LT

The bias between LT set and balanced set is ineluctable and numerous studies [10, 47, 50] have demonstrated its existence. To eliminate the systematic bias that classifier tends to predict the head, we reconsider the parameters training process. Generally, a classifier can be modeled as:

$$\hat{y} = \arg\max_{y_i \in \mathcal{Y}} \frac{e^{\sum_{d_i \in D} [(W^T)^{(d_i)}_{y_i} \mathcal{F}(x;\theta)^{(d_i)}] + b_{y_i}}}{\sum_{y_j \in \mathcal{Y}} e^{\sum_{d_i \in D} [(W^T)^{(d_i)}_{y_j} \mathcal{F}(x;\theta)^{(d_i)}] + b_{y_j}}} \triangleq \arg\max_{y_i \in \mathcal{Y}} \frac{e^{\psi(x;\theta,W,b)_{y_i}}}{\sum_{y_j \in \mathcal{Y}} e^{\psi(x;\theta,W,b)_{y_j}}} \tag{12}$$

where $\hat{y}$ indicts the predicted label, and $\mathcal{F}(x;\theta) \in \mathbb{R}^{D \times 1}$ is the D-dimension feature extracted by the backbone with parameter θ. $W \in \mathbb{R}^{D \times C}$ represents the parameter matrix of the classifier.

Nevertheless, previous works [10, 47] have demonstrated that simply make no exception of bias is not suitable for imbalance learning. In LT scenarios, the instances number in each class of the train set varies greatly, which means the corresponding *prior* probability $\mathbb{P}_{train}(Y = y)$ is highly skewed whereas the distribution on the test set $\mathbb{P}_{test}(Y = y)$ is uniform.

According to Bayesian theory, *posterior* is proportional to *prior* times *likelihood*. The supervised training process of $\psi(x;\theta,W,b)$ in Eq.12 can regard as the estimation of *likelihood*, which is equivalent to get *posterior* for inference in balanced dataset. Considering the difference of *prior* during training and testing, we have the following theorem (See detail derivation in Appendix B.1):

Theorem 3.1 *For classification, let* $\psi(x;\theta,W,b)$ *be a hypothesis class of neural networks of input* $X = x$, *the classification with Softmax should contain the influence of prior, i.e., the predicted label*

during training should be:

$$\hat{y} = \arg\max_{y_i \in \mathcal{Y}} \frac{e^{\psi(x;\theta,W,b)_{y_i} + \mathbf{log}\,(\boldsymbol{\pi_{y_i}}) + \mathbf{log}\,(\boldsymbol{C})}}{\sum_{y_j \in \mathcal{Y}} e^{\psi(x;\theta,W,b)_{y_j} + \mathbf{log}\,(\boldsymbol{\pi_{y_j}}) + \mathbf{log}\,(\boldsymbol{C})}} \tag{13}$$

In balanced datasets, all classes share the same *prior*. Hence, the supervised model $\psi(x;\theta,W,b)$ could use the estimated *likelihood* $\mathbb{P}(X = x|Y = y)$ of train set to correctly obtain *posterior* $\mathbb{P}(Y = y|X = x)$ in test set. However, in LT datasets where $\mathbb{P}_{train}(Y = y_i) = \pi_{y_i}$ and $\mathbb{P}_{test}(Y = y_i) \equiv 1/C$, *prior* cannot be regard as a constant over all classes any more. Due to the difference on *prior*, the learned parameters $\theta, W, b \triangleq \Theta$ will yield class-level bias, i.e., the optimization direction is no longer as described in Eq.12. Thus, the bias incurred by *prior* should compensate at first. To correctness the bias for inferring, the offset term that model in LT dataset to compensate is:

$$\mathscr{B}_y = log(\pi_y) + log(C) \tag{14}$$

Furthermore, the proposed Bayias $\mathscr{B}_y$ enables predicted probability reflecting the actual correctness likelihood, expressed as Theorem 3.2. (See detail derivation in Appendix B.2.)

Theorem 3.2 *$\mathscr{B}_y$-compensated cross-entropy loss in Eq.15 ensures classification calibration.*

$$\mathcal{L}_{\mathscr{B}}(y_i, \psi(x;\Theta)) = \log\left[1 + \sum_{y_k \neq y_i} e^{(\mathscr{B}_{y_k} - \mathscr{B}_{y_i})} \cdot e^{\psi(x;\Theta)_{y_k} - \psi(x;\Theta)_{y_i}}\right] \tag{15}$$

Here, the optimization direction during training will convert to $\psi(X;\theta,W,b) + \mathscr{B}_y$. In particular, if the train set is balanced, $\mathbb{P}_{train}(Y = y) \triangleq \pi_y \equiv 1/C$, then $\mathscr{B}_y = log(1/C) + log(C) \equiv 0$, which means the Eq.12 is a balanced case of Eq.13. We further raise that $\mathscr{B}_y$ is critical to the classification *calibration* in Theorem 3.2. The pairwise loss in Eq.15 will guide model to avoid over-fitting the tail or under-fitting the head with better generalization, which contributes to a better calibrated model.

3.3 Towards calibrated model with UniMix and Bayias

It's intuitive to integrate feature improvement methods with loss modification ones for better performance. However, we find such combinations fail in most cases and are counterproductive with each other, i.e., the combined methods reach unsatisfactory performance gains. We suspect that these methods take contradictory trade-offs and thus result in overconfidence and bad *calibration*. Fortunately, the proposed UniMix and Bayias are both proven to ensure *calibration*. To achieve a better-calibrated model for superior performance gains, we introduce Alg.1 to tackle the previous dilemma and integrate our two proposed approaches to deal with poor generalization of tail classes. Specially, inspired by previous work [21], we overcome the coverage difficulty in *mixup* [53] by removing UniMix in the last several epochs and thus maintain the same epoch as baselines. Note that Bayias-compensated CE is only adopted in the training process as discussed in Sec.3.2.

Algorithm 1 Integrated training manner towards calibrated model.

Input: $\mathcal{D}_{train}$, Batch Size $\mathcal{N}$, Stop Steps T_1, T_2, Random Sampler $\mathcal{R}$, UniMix Sampler $\mathcal{R}^*$
Output: Optimized Θ^*, i.e., feature extractor parameters θ^*, classifier parameters W^*, b^*

```
Initialize the parameters Θ^(0) randomly and calculate B_y via Eq.14
for t = 0 to T_1 do
    Sample a mini-batch B = {x_i, y_i}_{i=1}^N ← R(D_train, N)
    Sample a mini-batch B* = {x*_j, y*_j}_{j=1}^N ← R*(D_train, N)
    Calculate UniMix factor ξ* via Eq.6
    Construct VRM dataset B_ν = {x̃_k, ỹ_k}_{k=1}^N via Eq.7
    Calculate L_{B_ν} = E[ξ*_ij L_B(y_i, ψ(x̃; Θ^(t))) + (1 − ξ*_ij) L_B(y*_j, ψ(x̃; Θ^(t)))] via Eq.11,15
    Update Θ^(t+1) ← Θ^(t) − α∇_{Θ^(t)} L_{B_ν}
end for
for t = T_1 to T_2 do
    Sample a mini-batch B = {x_i, y_i}_{i=1}^N ← R(D_train, N)
    Calculate L_B = E[L_B(y_i, ψ(x_i; Θ))] via Eq.15
    Update Θ^(t+1) ← Θ^(t) − α∇_{Θ^(t)} L_B
end for
```

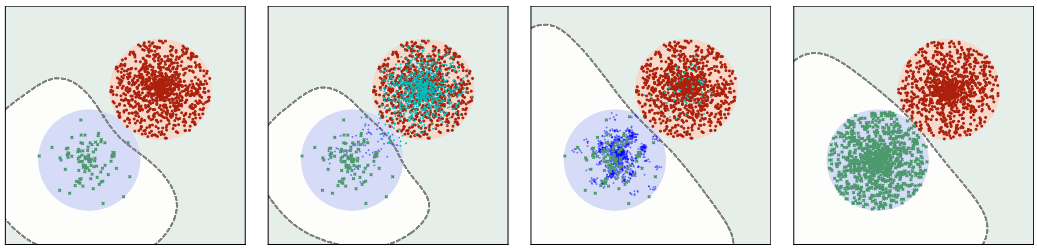
4 Experiment

4.1 Results on synthetic dataset

We make an ideal binary classification using Support Vector Machine (SVM) [11] to show the distinguish effectiveness of UniMix. Suppose there are samples from two disjoint circles respectively:

$$\begin{aligned} z^+ &= \{(x,y)|(x-x_0)^2+(y-y_0)^2 \leq r^2\} \\ z^- &= \{(x,y)|(x+x_0)^2+(y+y_0)^2 \leq r^2\} \end{aligned} \tag{16}$$

To this end, we randomly sample m discrete point pairs from z^+ to compose positive samples $z_p^+ = \{(x_1^+, y_1^+), \cdots, (x_m^+, y_m^+)\}$, and m negative samples $z_n^- = \{(x_1^-, y_1^+), \cdots, (x_m^-, y_m^-)\}$ from z^- correspondingly, thus to generate a balanced dataset $\mathcal{D}_{bal} = \{z_p^+, z_n^-\}$ with $\mathbb{P}\left((x,y) \in z_p^+\right) = \mathbb{P}\left((x,y) \in z_n^-\right) = 0.5$. For imbalance data, we sample $n (n \ll m)$ negative data from z^- to generate $z_{n'}^- = \{(x_1'^-, y_1'^-), \cdots, (x_n'^-, y_n'^-)\}$, so as to compose the imbalance dataset $\mathcal{D}_{imbal} = \{z_p^+, z_{n'}^-\}$, with $\mathbb{P}\left((x,y) \in z_p^+\right) \gg \mathbb{P}\left((x,y) \in z_{n'}^-\right)$. We train the SVM model on the two synthetic datasets, and visualize the classification boundary of each dataset in Fig.3.



(a) imbalanced scenario (b) with *mixup* (c) with UniMix (d) balanced scenario

Figure 3: SVM decision boundary on the synthetic balanced dataset (Fig.3(d)) and imbalanced dataset (Fig.3(a),3(b),3(c)). The theoretical classification boundary of the synthetic dataset is y=$-x$. "$\circ$" represents generated pseudo data, where blue and green represent belong to z^- and z^+, respectively.

The SVM reaches an approximate ideal boundary on balanced datasets (Fig.3(d)) but severely deviates from the $y = -x$ in the imbalanced dataset (Fig.3(a)). As proven in Sec.2, *mixup* (Fig.3(b)) is incapable of shifting imbalance distribution, resulting in no better result than the original one (Fig.3(a)). After adopting the proposed UniMix, the classification boundary in Fig.3(c) shows much better results than the original imbalanced dataset, which gets closed to the ideal boundary.

4.2 Results on CIFAR-LT

The imbalanced datasets CIFAR-10-LT and CIFAR-100-LT are constructed via suitably discarding training samples following previous works [56, 10, 38, 5]. The instance numbers exponentially decay per class in train dataset and keep balanced during inference. We extensively adopt $\rho \in \{10, 50, 100, 200\}$ for comprehensive comparisons. See implementation details in Appendix C.1.

Comparison methods. We evaluate the proposed method against various representative and effective approaches extensively, summarized into the following groups: **a) Baseline.** We conduct plain training with CE loss called ERM as baseline. **b) Feature improvement methods** modify the input feature to cope with LT datasets. *mixup* [53] convexly combines images and labels to build virtual data for VRM. Manifold mixup [49] performs the linear combination in latent states. Remix [8] conducts the same combination on images and adopts tail-favored rules on labels. M2m [30] converts majority images to minority ones by adding noise perturbation, which need an additional pre-trained classifier. BBN [56] utilizes features from two branches in a cumulative learning manner. **c) Loss modification methods** either adjust the logits *weight* or *margin* before the *Softmax* operation. Specifically, focal loss [34], CB [10] and CDT [51] re-weight the logits with elaborate strategies, while LDAM [5] and Logit Adjustment [38] add the logits *margin* to shift decision boundary away from tail classes. **d) Other methods.** We also compare the proposed method with other two-stage approaches (e.g. DRW [5]) for comprehensive comparisons.

Results. We present results of CIFAR-10-LT and CIFAR-100-LT in Tab.1. Our proposed method achieves state-of-the-art results against others on each ρ, with performance gains improved as ρ gets

Table 1: Top-1 validation accuracy(%) of ResNet-32 on CIFAR-10/100-LT. E2E: end to end training. Underscore: the best performance in each group. †: our reproduced results. ‡: reported results in [56]. ⋆: reported results in [51]. Our *calibration* ensured method achieves the best performance.

Dataset	E2E	CIFAR-10-LT				CIFAR-100-LT			
ρ (easy $\rightarrow$ hard)	-	10	50	100	200	10	50	100	200
ERM†	✓	86.39	74.94	70.36	66.21	55.7	44.02	38.32	34.56
mixup‡ [53]	✓	87.10	77.82	73.06	67.73	58.02	44.99	39.54	34.97
Manifold Mixup‡ [49]	✓	87.03	77.95	72.96	-	56.55	43.09	38.25	-
Remix [8]	✓	88.15	79.20	75.36	67.08	59.36	46.21	41.94	36.99
M2m [30]	✗	87.90	-	78.30	-	58.20	-	42.90	-
BBN‡ [56]	✗	88.32	82.18	79.82	-	59.12	47.02	42.56	-
Focal⋆ [34]	✓	86.55	76.71†	70.43	65.85	55.78	44.32†	38.41	35.62
Urtasun et al [41]	✓	82.12	76.45	72.23	66.25	52.12	43.17	38.90	33.00
CB-Focal [10]	✓	87.10	79.22	74.57	68.15	57.99	45.21	39.60	36.23
τ-norm⋆ [27]	✓	87.80	82.78†	75.10	70.30	59.10	48.23†	43.60	39.30
LDAM† [5]	✓	86.96	79.84	74.47	69.50	56.91	46.16	41.76	37.73
LDAM+DRW† [5]	✗	88.16	81.27	77.03	74.74	58.71	47.97	42.04	38.45
CDT⋆ [51]	✓	89.40	81.97†	79.40	74.70	58.90	45.15†	44.30	40.50
Logit Adjustment [38]	✓	89.26†	83.38†	79.91	75.13†	59.87†	49.76†	43.89	40.87†
Ours	✓	**89.66**	**84.32**	**82.75**	**78.48**	**61.25**	**51.11**	**45.45**	**42.07**

increased (See Appendix D.1). Specifically, our method overcomes the ignorance in tail classes effectively with better *calibration*, which integrates advantages of two group approaches and thus surpass most two-stage methods (i.e., BBN, M2m, LDAM+DRW). However, not all combinations can get ideal performance gains as expected. More details will be discussed in Sec.4.4.

4.3 Results on large-scale datasets

We further verify the proposed method's effectiveness quantitatively on large-scale imbalanced datasets, i.e. ImageNet-LT and iNaturalist 2018. ImageNet-LT is the LT version of ImageNet [43] by sampling a subset following *Pareto* distribution, which contains about $115K$ images from $1,000$ classes. The number of images per class varies from 5 to $1,280$ exponentially, i.e., $\rho = 256$. In our experiment, we utilize the balanced validation set constructed by Cui *et al.* [10] for fair comparisons. The iNaturalist species classification dataset [22] is a large-scale real-world dataset which suffers from extremely label LT distribution and fine-grained problems [22, 56]. It is composed of $435,713$ images over $8,142$ classes with $\rho = 500$. The official splits of train and validation images [5, 56, 27] are adopted for fair comparisons. See implementation details in Appendix C.2.

Table 2: Top-1 validation accuracy(%) of ResNet-10/50 on ImageNet-LT and ResNet-50 on iNaturalist 2018. E2E: end to end training. †: our reproduced results. ‡: results reported in origin paper.

Dataset	ImageNet-LT					iNaturalist 2018	
Method	E2E	ResNet-10	Δ	ResNet-50	Δ	ResNet-50	Δ
CE†	✓	35.88	-	38.88	-	60.88	-
CB-CE† [10]	✓	37.06	+1.18	40.85	+1.97	63.50	+2.62
LDAM [5]	✓	36.05†	+0.17	41.86†	+2.98	64.58‡	+3.70
OLTR‡ [36]	✗	35.60	-0.28	40.36	+1.48	63.90	+3.02
LDAM+DRW [5]	✗	38.22†	+2.34	45.75†	+6.87	68.00‡	+7.12
BBN‡ [56]	✗	-		-		66.29	+5.41
c-RT [27]	✗	41.80‡	+5.92	47.54†	+8.66	67.60†	+6.72
Ours	✓	**42.90**	**+7.02**	**48.41**	**+9.53**	**69.15**	**+8.27**

Results. Tab.2 illustrates the results on large-scale datasets. Ours is consistently effective and outperforms existing mainstream methods, achieving distinguish improvement compared with previous SOTA c-RT [27] in the compared backbones. Especially, our method outperforms the baseline on ImageNet-LT and iNaturalist 2018 by **9.53%** and **8.27%** with ResNet-50, respectively. As can be noticed in Tab.2, the proposed method also surpasses the well-known two-stage methods [27, 5, 56], achieving superior accuracy with less computation load in a concise training manner.

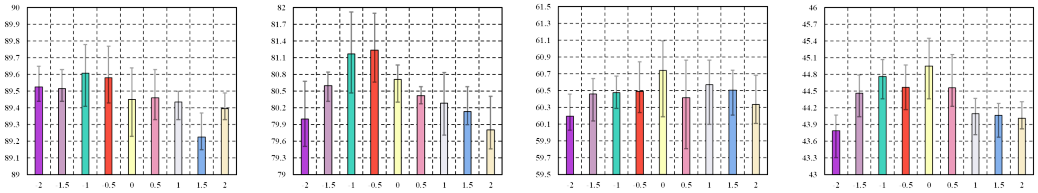
4.4 Further Analysis

Effectiveness of UniMix and Bayias. We conduct extensive ablation studies in Tab.3 to demonstrate the effectiveness of the proposed UnixMix and Bayias, with detailed analysis in various combinations of feature-wise and loss-wise methods on CIFAR-10-LT and CIFAR-100-LT. Indeed, both UniMix and Bayias turn out to be effective in LT scenarios. Further observation shows that with *calibration* ensured, the proposed method gets significant performance gains and achieve state-of-the-art results. Noteworthy, LDAM [5] makes classifiers miscalibrated (see Appendix D.2), which leads to unsatisfactory improvement when combined with *mixup* manners.

Table 3: Ablation study between feature-wise and loss-wise methods. LDAM is counterproductive to *mixup* and its extensions. Bayias-compensated CE ensures *calibration* and shows excellent performance gains especially combined with UniMix.

Dataset		CIFAR-10-LT				CIFAR-100-LT			
ρ (easy $\rightarrow$ hard)		100		200		100		200	
Mix	Loss	Top1 Acc	Δ	Top1 Acc	Δ	Top1 Acc	Δ	Top1 Acc	Δ
None	CE	70.36	-	66.21	-	38.32	-	34.56	-
Mixup	CE	73.06	+2.70	67.73	+1.52	39.54	+1.22	34.97	+0.41
Remix	CE	75.36	+5.00	67.08	+0.87	41.94	+3.62	36.99	+2.43
UniMix	CE	76.47	+6.11	68.42	+2.21	41.46	+3.14	37.63	+3.07
None	LDAM	74.47	-	69.50	-	41.76	-	37.73	-
Mixup	LDAM	73.96	-0.15	67.89	-1.61	40.22	-1.54	37.52	-0.21
Remix	LDAM	74.33	-0.14	69.66	+0.16	40.59	-1.17	37.66	-0.07
UniMix	LDAM	75.35	+0.88	70.77	+1.27	41.67	-0.09	37.83	+0.01
None	Bayias	78.70	-	74.21	-	43.52	-	38.83	-
Mixup	Bayias	81.75	+3.05	76.69	+2.48	44.56	+1.04	41.19	+2.36
Remix	Bayias	81.55	+2.85	75.81	+1.60	45.01	+1.49	41.44	+2.61
UniMix	Bayias	**82.75**	**+4.05**	**78.48**	**+4.27**	**45.45**	**+1.93**	**42.07**	**+3.24**

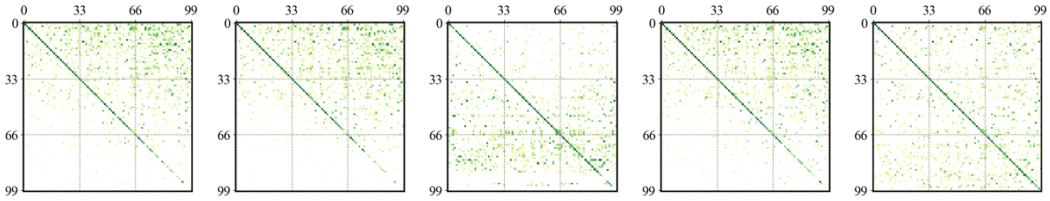
Evaluating different UniMix Sampler. Corollary 1,2,3 demonstrate distinguish influence of UniMix. However, the ξ-*sample* can not be completely equivalent with orginal ones. Hence, an appropriate τ in Eq.9 is also worth further searching. Fig.4 illustrates the accuracy with different τ on CIFAR-10-LT and CIFAR-100-LT setting $\rho = 10$ and 100. For CIFAR-10-LT (Fig.4(a),4(b)), $\tau = -1$ is possibly ideal, which forces more head-tail instead of head-head pairs get generated to compensate tail classes. In the more challenging CIFAR-100-LT, $\tau = 0$ achieves the best result. We suspect that unlike simple datasets (e.g., CIFAR-10-LT), where overconfidence occurs in head classes, all classes need to get enhanced in complicated LT scenarios. Hence, the augmentation is effective and necessary both on head and tail. $\tau = 0$ allows both head and tail get improved simultaneously.



(a) CIFAR-10-LT-10. (b) CIFAR-10-LT-100. (c) CIFAR-100-LT-10. (d) CIFAR-100-LT-100.

Figure 4: Comparison of top-1 validation accuracy(%) of ResNet-32 on CIFAR-LT when varying τ in Eq.9 for UniMix. The histogram indicates average results in repeated experiments.

Do minorities really get improved? To observe the amelioration on tail classes, Fig.5 visualizes log-confusion matrices on CIFAR-100-LT-100. In Fig.5(e), our method exhibits satisfactory generalization on the tail. Vanilla ERM model (Fig.5(a)) is a trivial predictor which simplifies tail instances as head labels to minimize the error rate. Feature improvement [8] and loss modification [5, 51] methods do alleviate LT problem to some extent. The misclassification cases (i.e., non-diagonal elements) in Fig.5(b),5(c),5(d) become smaller and more balanced distributed compared with ERM. However, the error cases are still mainly in the upper or lower triangular, indicating the existence of inherent bias between the head and tail. Our method (Fig.5(e)) significantly alleviates such dilemma. The non-diagonal elements are more uniformly distributed throughout the matrix rather than in



(a) ERM (b) LDAM+DRW [5] (c) CDT [51] (d) Remix [8] (e) Ours

Figure 5: The log-confusion matrix on CIFAR-100-LT-100 validation dataset. The x-axis and y-axis indicate the ground truth and predicted labels, respectively. Deeper color indicates larger values.

the corners, showing superiority to erase the bias in LT scenarios. Our method enables effective feature improvement for data-scarce classes and alleviates the *prior* bias, suggesting our success in regularizing tail remarkably.

5 Related work and discussion

Why need *calibration*? To quantify the predictive uncertainty, *calibration* [2] is put forward to describe the relevance between predictive score and actual correctness likelihood. A well-calibrated model is more reliable with better interpretability, which probabilities indicate optimal expected costs in Bayesian decision scenarios [37]. Guo *et al.* [13] firstly provide metric to measure the *calibration* of CNN and figure out well-performed models are always in lack of *calibration*, indicating that CNN is sensitive to be overconfidence and lacks robustness. Thulasidasan *et al.* [48] point out that the effectiveness of *mixup* in balanced datasets originates from superior *calibration* modification. Menon *et al.* [38] further show how to ensure optimal classification *calibration* for a pair-wise loss.

Feature-wise methods. Intuitively, under-sampling the head [15, 3, 17] or over-sampling the tail [7, 3, 39, 44, 4] can improve the inconsistent performance of imbalanced datasets but tend to either weaken the head or over-fitting the tail. Hence, many effective works generate additional samples [9, 55, 30] to compensate the tail classes. BBN [56] uses two branches to extract features from head and tail simultaneously, while c-RT [27] trains feature representation learning and classification stage separately. *mixup* [53] and its variants [49, 52, 8] are effective and easy-implement feature-wise methods that convexly combine input and label pairs to generate virtual samples. However, naïve *mixup* manners are deficient in LT scenarios as we discussed in Sec.2. In contrast, our UniMix tackles such a dilemma by constructing class balance-oriented virtual data as describe in Sec.3.1 and shows satisfactory *calibration* as Fig.1 exhibits.

Loss modification. Numerous experimental and theoretical studies [40, 10, 47, 50] have demonstrated the existence of inherent *bias* between LT train set and balanced test set in supervised learning. Previous works [24, 25, 28, 10, 34] make networks prefer learning tail samples by additional class-related weight on CE loss. Some works further correct CE according to the gradient generated by different samples [46, 32] or from the perspective of Gaussian distribution and Bayesian estimation [16, 29]. Meta-learning approaches [12, 1, 45, 41, 26] optimize the weights of each class in CE as learnable parameters and achieve remarkable success. Cao *et al.* [5] theoretically provides the ideal optimal *margin* for CE from the perspective of *VC* generalization bound. Compared with Logit Adjustment [38] motivated by balance error rate, our Bayias-compensated CE eliminates *bias* incured by *prior* and is consistent with balanced datasets, which ensures classification *calibration* as well.

6 Conclusion

We systematically analyze the limitations of mainstream feature improvement methods, i.e., *mixup* and its extensions in the label-imbalanced situation, and propose the UniMix to construct a more class-balanced virtual dataset that significantly improves classification *calibration*. We further pinpoint an inherent bias induced by the inconstancy of label distribution *prior* between long-tailed train set and balanced test set. We prove that the standard cross-entropy loss with the proposed Bayias compensated can ensure classification *calibration*. The combination of UniMix and Bayias achieves state-of-the-art performance and contributes to a better-calibrated model (Fig.1). Further study in Tab.3 shows that the bad *calibration* methods are counterproductive with each other. However, more in-depth analysis and theoretical guarantees are still required, which we leave for our future work.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3981–3989, 2016.

[2] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[3] 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.

[4] Jonathon Byrd and Zachary Chase Lipton. What is the effect of importance weighting in deep learning? In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881. PMLR, 2019.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1565–1576, 2019.

[6] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 416–422. MIT Press, 2000.

[7] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[8] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12540 of *Lecture Notes in Computer Science*, pages 95–110. Springer, 2020.

[9] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pages 694–710. Springer, 2020.

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277. Computer Vision Foundation / IEEE, 2019.

[11] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines: Theory and applications. In *Machine Learning and Its Applications, Advanced Lectures*, volume 2049 of *Lecture Notes in Computer Science*, pages 249–257. Springer, 2001.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[14] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019.

[15] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer, 2005.

[16] Munawar Hayat, Salman H. Khan, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Gaussian affinity for max-margin class imbalanced learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6468–6478. IEEE, 2019.

[17] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[20] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 558–567. Computer Vision Foundation / IEEE, 2019.

[21] Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Data augmentation revisited: Rethinking the distribution gap between clean and augmented data. *CoRR*, abs/1909.09148, 2019.

[22] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8769–8778. IEEE Computer Society, 2018.

[23] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *CoRR*, abs/1709.01450, 2017.

[24] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5375–5384. IEEE Computer Society, 2016.

[25] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(11):2781–2794, 2020.

[26] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7607–7616. IEEE, 2020.

[27] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[28] Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Ahmed Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Networks Learn. Syst.*, 29(8):3573–3587, 2018.

[29] Salman H. Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 103–112. Computer Vision Foundation / IEEE, 2019.

[30] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13893–13902. IEEE, 2020.

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017.

[32] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 8577–8584. AAAI Press, 2019.

[33] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020.

[34] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society, 2017.

[35] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[36] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2537–2546. Computer Vision Foundation / IEEE, 2019.

[37] Juan Maroñas, Roberto Paredes, and Daniel Ramos. Calibration of deep probabilistic models with decoupled bayesian neural networks. *Neurocomputing*, 407:194–205, 2020.

[38] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

[39] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1695–1704. IEEE, 2019.

[40] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[41] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4331–4340. PMLR, 2018.

[42] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[44] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, pages 467–482. Springer, 2016.

[45] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1917–1928, 2019.

[46] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11659–11668. IEEE, 2020.

[47] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[48] 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 *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13888–13899, 2019.

[49] 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 *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 2019.

[50] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[51] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *CoRR*, abs/2001.01385, 2020.

[52] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6022–6031. IEEE, 2019.

[53] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[54] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021.

[55] Yongshun Zhang, Xiu-Shen Wei, Boyan Zhou, and Jianxin Wu. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In *AAAI*, 2021.

[56] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9716–9725. IEEE, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sec.1
   (b) Did you describe the limitations of your work? [Yes] See Sec.6
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sec.3.1 & 3.2
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A & B
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix C
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec.4.4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]