Improving Calibration for Long-Tailed Recognition (Supplementary Material)

A. Experiment Setup

Following Liu *et al.* [20] and Kang *et al.* [15], we report the commonly used top-1 accuracy over all classes on the balanced test/validation datasets, denoted as *All*. We further report accuracy on three splits of classes: *Head-Many* (more than 100 images), *Medium* (20 to 100 images), and *Tail-Few* (less than 20 images). The detailed setting of hyperparameters and training for all datasets used in our paper are listed in Table 6.

Dataset		Common setting			Stage-1		Stage-2				
		LR	BS	WD	Epochs	LRS	Epochs	LRS	ϵ_1	ϵ_K	$\Delta\boldsymbol{W}$
CIFAR-10-LT	$\beta = 10$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.1	0.0	0.2×
CIFAR-10-LT	$\beta = 50$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.2	0.0	0.2×
CIFAR-10-LT	$\beta = 100$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.3	0.0	0.5×
CIFAR-100-LT	$\beta = 10$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.2	0.0	0.1×
CIFAR-100-LT	$\beta = 50$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.3	0.0	0.1×
CIFAR-100-LT	$\beta = 100$	0.1	128	2×10^{-4}	200	multistep	10	cosine	0.4	0.1	0.2×
ImageNet-LT		0.1	256	5×10^{-4}	180	cosine	10	cosine	0.3	0.0	0.05×
Places-LT		0.1	256	5×10^{-4}	90	cosine	10	cosine	0.4	0.1	0.05×
iNaturalist 2018		0.1	256	1×10^{-4}	200	cosine	30	cosine	0.4	0.0	0.05×

Table 6: Detailed experiment setting on five benchmark datasets. LR: initial learning rate, BS: batch size, WD: weight decay, LRS: learning rate schedule, and $\Delta\boldsymbol{W}$: learning rate ratio of $\Delta\boldsymbol{W}$.

B. Exponential Form of the Related Function $f(\cdot)$

As discussed in Secs. 3.2 and 4.2, the form of the related function $f(\cdot)$ may play an important role for final model performance. We draw the illustration of Eqs. (3.a), (3.b), and (3.c) at the left of Fig. 8. For the CIFAR-100-LT dataset with imbalanced factor 100, $K = 100$, $N_1 = 500$, and $N_{100} = 5$. Based on the ablation study results of ϵ_1 and ϵ_K mentioned in Sec. 4.2, we set $\epsilon_1 = 0.4$ and $\epsilon_{100} = 0.1$ here. After fintuning for 10 epochs in Stage-2, the accuracy of the concave model is the best. We also design an exponential related function, which is written as

$$\epsilon_y = f(N_y) = \epsilon_K + (\epsilon_1 - \epsilon_K)\left(\frac{N_y - N_K}{N_1 - N_K}\right)^p, \qquad y = 1, 2, ..., K, \tag{7}$$

where p is a hyperparameter to control the shape of the related function. For example, we get the concave related function when setting $p < 1$ and convex function otherwise. Illustration of Eq. (7) is given on the right of Fig. 8. Comparing accuracy of all variants, the influence of the related function form is quite limited for the final performance (0.3% increase). Because the concave related function Eq. (3.a) achieves the best performance, we choose it as the default setting of the related function $f(\cdot)$ for other experiments.

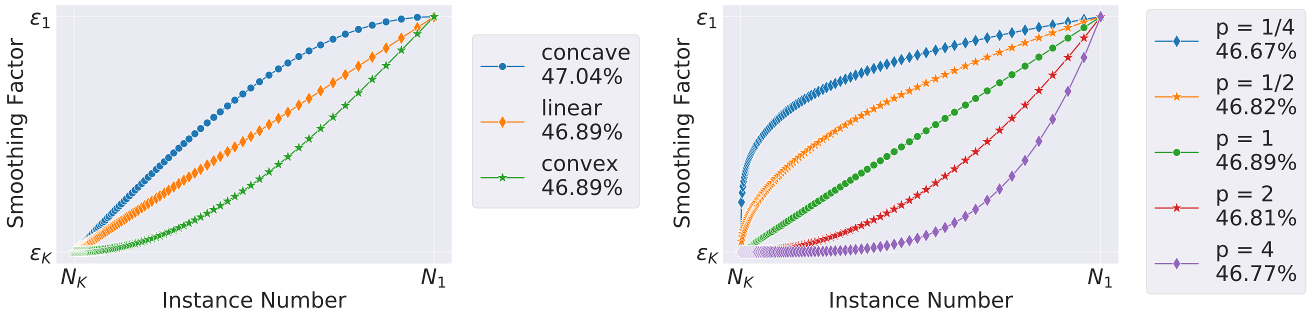

Figure 8: Function illustration and accuracy of Eqs. (3.a), (3.b), and (3.c) (left) and Eq. (7) (right).

C. Calibration Performance

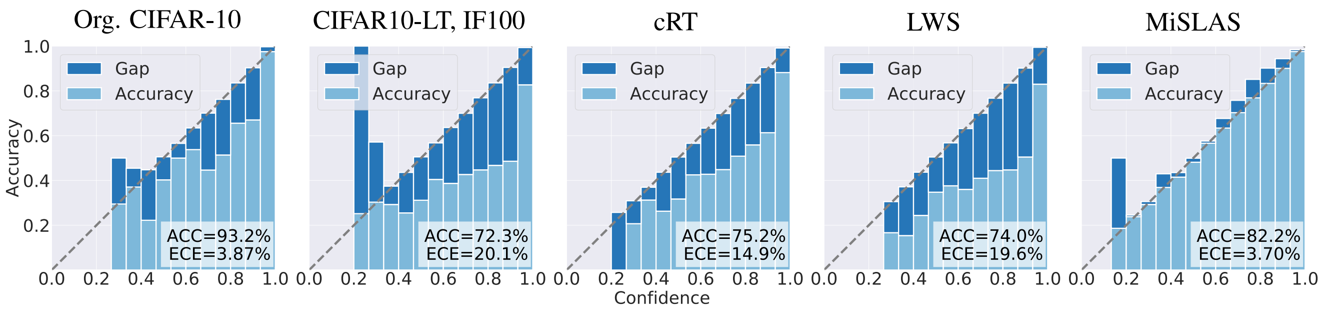

Figure 9: Reliability diagrams on CIFAR10 with 15 bins. From left to right: plain ResNet-32 model trained on the original CIFAR-10 dataset, plain model, cRT, LWS, and MiSLAS trained on long-tailed CIFAR-10 with imbalanced factor 100.

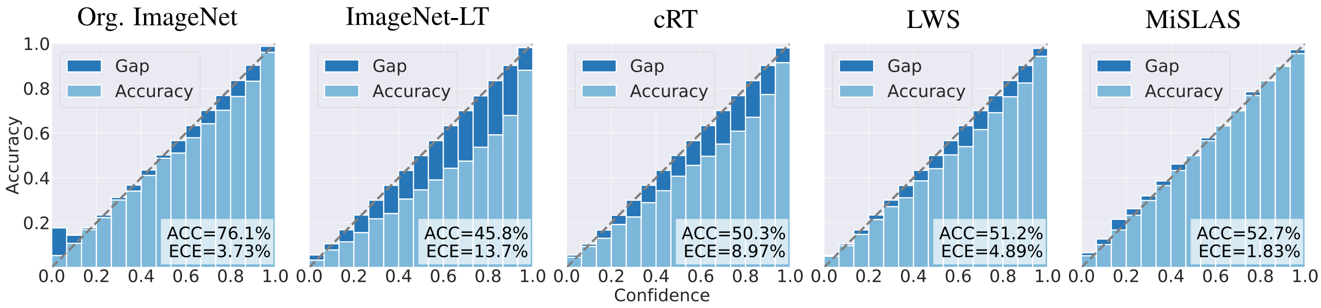

Figure 10: Reliability diagrams on ImageNet with 15 bins. From left to right: plain ResNet-50 model trained on the original ImageNet dataset, plain model, cRT, LWS, and MiSLAS trained on ImageNet-LT.

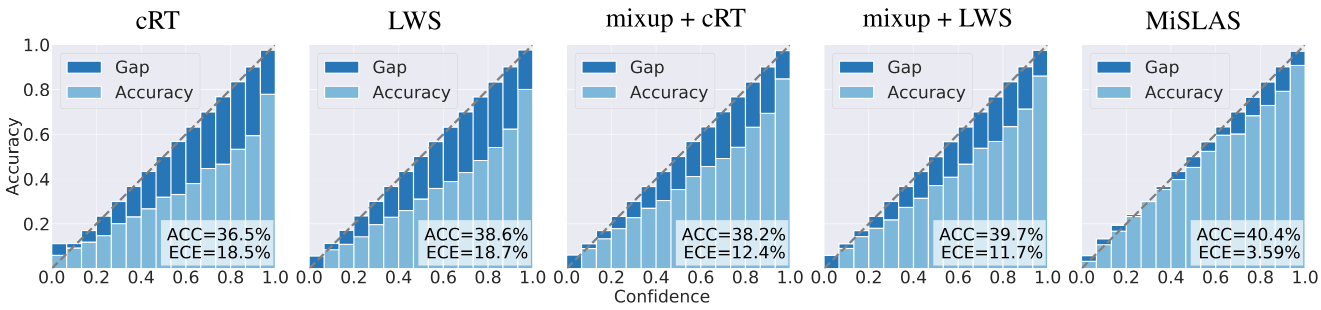

Figure 11: Reliability diagrams of ResNet-152 trained on Places-LT with 15 bins. From left to right: cRT, LWS, cRT with mixup, LWS with mixup, and MiSLAS.

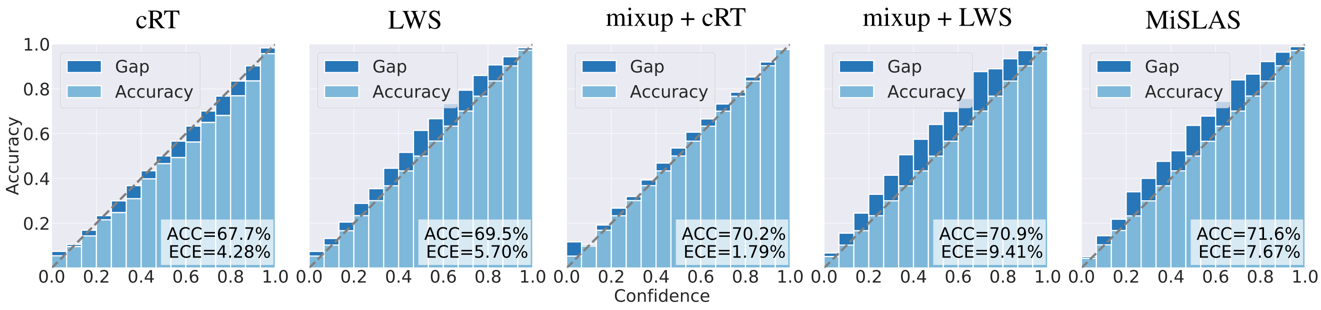

Figure 12: Reliability diagrams of ResNet-50 trained on iNaturalist 2018 with 15 bins. From left to right: cRT, LWS (under-confidence), cRT with mixup, LWS with mixup (under-confidence), and MiSLAS (under-confidence).

D. More Results on ImageNet-LT, iNaturalist 2018, and Places-LT

Backbone	**Method**	Many	Medium	Few	All
ResNet-50	cRT	62.5	47.4	29.5	50.3
	LWS	61.8	48.6	33.5	51.2
	cRT+mixup	**63.9**	49.1	30.2	51.7
	LWS+mixup	62.9	49.8	31.6	52.0
	MiSLAS	61.7	**51.3**	**35.8**	**52.7**
ResNet-101	cRT	63.8	48.5	30.0	51.4
	LWS	63.1	49.9	33.8	52.3
	cRT+mixup	**65.2**	50.6	31.6	53.1
	LWS+mixup	64.5	51.2	34.1	53.5
	MiSLAS	64.3	**52.1**	**35.8**	**54.1**
ResNet-152	cRT	64.9	50.4	30.6	52.7
	LWS	64.1	51.8	35.5	53.8
	cRT+mixup	**66.5**	51.6	32.8	54.2
	LWS+mixup	66.1	52.2	34.5	54.6
	MiSLAS	65.4	**53.2**	**37.1**	**55.2**

Table 7: Comprehensive accuracy results on ImageNet-LT with different backbone networks (ResNet-50, ResNet-101 & ResNet-152) and training 180 epochs.

Backbone	**Method**	Many	Medium	Few	All
ResNet-50	cRT	73.2	68.8	66.1	68.2
	τ-normalized	71.1	68.9	69.3	69.3
	LWS	71.0	69.8	68.8	69.5
	cRT+mixup	**74.2**	71.1	68.2	70.2
	LWS+mixup	72.8	71.6	69.8	70.9
	MiSLAS	73.2	**72.4**	**70.4**	**71.6**

Table 8: Comprehensive accuracy results on iNaturalist 2018 with ResNet-50 and training 200 epochs.

Backbone	**Method**	Many	Medium	Few	All
ResNet-152	Lifted Loss	41.1	35.4	24.0	35.2
	Focal Loss	41.1	34.8	22.4	34.6
	Range Loss	41.1	35.4	23.2	35.1
	FSLwF	43.9	29.9	29.5	34.9
	OLTR	**44.7**	37.0	25.3	35.9
	OLTR+LFME	39.3	39.6	24.2	36.2
	cRT	42.0	37.6	24.9	36.7
	τ-normalized	37.8	40.7	31.8	37.9
	LWS	40.6	39.1	28.6	37.6
	cRT+mixup	44.1	38.5	27.1	38.1
	LWS+mixup	41.7	41.3	33.1	39.7
	MiSLAS	39.6	**43.3**	**36.1**	**40.4**

Table 9: Detailed accuracy results on Places-LT, starting from an ImageNet pre-trained ResNet-152.

E. Proof of Eq. (2), the Optimal Solution of LAS

In this section, we prove the optimal solutions of cross-entropy, the re-weighting method, and LAS. Furthermore, the comparison among above three methods will also be discussed.

The general loss function form of these three methods for K classes can be written as

$$l = -\sum_{i=1}^{K} \boldsymbol{q}_i \log \boldsymbol{p}_i, \qquad \boldsymbol{p}_i = \mathrm{softmax}(\boldsymbol{w}_i^\top \boldsymbol{x}), \qquad s.t., \quad \sum_{i}^{K} \boldsymbol{p}_i = 1, \tag{8}$$

where $\boldsymbol{p}$, $\boldsymbol{w}$, and $\boldsymbol{x}$ are the predicted probability, the weight parameter of the last fully-connected layer, and the input of the last fully-connected layer, respectively. When the target label $\boldsymbol{q}$ is defined as

$$\boldsymbol{q}_i = \begin{cases} 1, & i = y, \\ 0, & i \neq y, \end{cases}$$

where y is the original ground truth label. Eq. (8) becomes the commonly used cross-entropy loss function. Similarly, when the target label $\boldsymbol{q}$ is defined as

$$\boldsymbol{q}_i = \begin{cases} w_i, & i = y, \ \text{and} \ w_i > 0, \\ 0, & i \neq y, \end{cases}$$

Eq. (8) becomes the re-weighting loss function. Moreover, when the target label $\boldsymbol{q}$ is

$$\boldsymbol{q}_i = \begin{cases} 1 - \epsilon_y = 1 - f(N_y), & i = y, \\ \frac{\epsilon_y}{K-1} = \frac{f(N_y)}{K-1}, & i \neq y, \end{cases} \tag{9}$$

Eq. (8) becomes the proposed LAS method. To get the optimal solution of Eq. (8), we define its Lagrange multiplier form as

$$L = l + \lambda \left(\sum_{i}^{K} \boldsymbol{p}_i - 1 \right) = -\sum_{i=1}^{K} \boldsymbol{q}_i \log \boldsymbol{p}_i + \lambda \left(\sum_{i}^{K} \boldsymbol{p}_i - 1 \right), \tag{10}$$

where λ is the Lagrange multiplier. The first order conditions of Eq. (10) *w.r.t.* λ and $\boldsymbol{p}$ can be written as

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \sum_{i=1}^{K} \boldsymbol{p}_i - 1 = 0, \\ \frac{\partial L}{\partial \boldsymbol{p}_i} &= -\frac{\boldsymbol{q}_i}{\boldsymbol{p}_i} + \lambda = 0. \end{aligned} \tag{11}$$

According to Eq. (11), we get $\boldsymbol{p}_i = \frac{\boldsymbol{q}_i}{\sum_{j=1}^{K} \boldsymbol{q}_j}$. Then, in the case of cross-entropy and re-weighting loss function, we get $\boldsymbol{p}_i = 1, i = y$ and $\boldsymbol{p}_i = 0, i \neq y$. Noting that

$$\boldsymbol{p}_i = \mathrm{softmax}(\boldsymbol{w}_i^\top \boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_i^\top \boldsymbol{x})}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^\top \boldsymbol{x})},$$

the optimal solutions of $\boldsymbol{w}_i^\top \boldsymbol{x}$ for both cross-entropy and re-weighting loss functions are the same, that is, ${\boldsymbol{w}_i^*}^\top \boldsymbol{x} = \inf$. This means that both cross-entropy and re-weighting loss functions make the weight vector of the right class $\boldsymbol{w}_i, i = y$ large enough while the others $\boldsymbol{w}_j, j \neq y$ sufficiently small. As a result, they cannot change the predicted distribution and relieve over-confidence effectively. In contrast, in our LAS, according to Eqs. (9) and (11), we get

$$\boldsymbol{p}_i = \frac{\exp(\boldsymbol{w}_i^\top \boldsymbol{x})}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^\top \boldsymbol{x})} = \frac{\boldsymbol{q}_i}{\sum_{j=1}^{K} \boldsymbol{q}_j} = \begin{cases} 1 - \epsilon_y, & i = y, \\ \frac{\epsilon_y}{K-1}, & i \neq y, \end{cases} \implies {\boldsymbol{w}_i^*}^\top \boldsymbol{x} = \begin{cases} \log \left[\frac{(K-1)(1-\epsilon_y)}{\epsilon_y} \right] + c, & i = y, \\ c, & i \neq y, \end{cases} \tag{12}$$

where $c \in \mathbb{R}$ can be an arbitrary real number. Overall, comparing with the infinite optimal solution in cross-entropy and re-weighting method, LAS encourages a finite output, which leads to a more general result, properly refines the predicted distributions of the head, medium, and tailed classes, and remedies over-confidence effectively.

F. More Results about the Effect of mixup on cRT and LWS

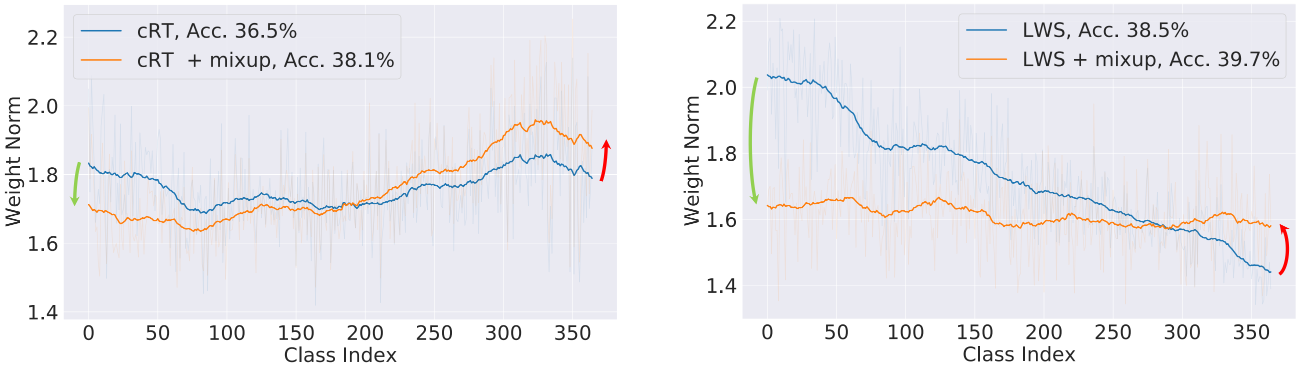

Figure 13: Classifier weight norms for the Places-LT evaluation set (365 classes in total) when classes are sorted by descending values of N_j, where N_j denotes the number of training sample for Class-j. Left: weight norms of cRT with/without mixup. Right: weight norms of LWS with/without mixup. Light shade: true norm. Dark lines: smooth version. *Best viewed on screen.*

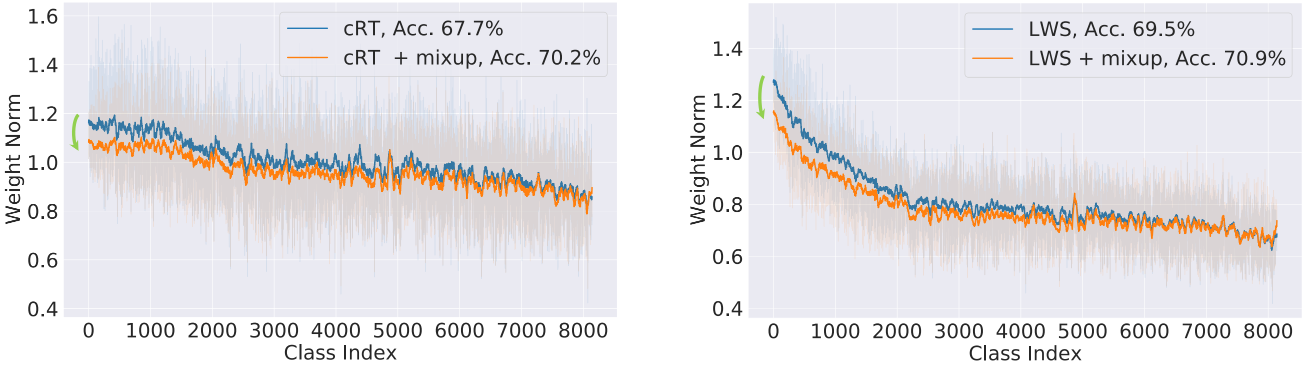

Figure 14: Classifier weight norms for the iNaturalist 2018 validation set (8,142 classes in total) when classes are sorted by descending values of N_j, where N_j denotes the number of training sample for Class-j. Left: weight norms of cRT with or without mixup. Right: weight norms of LWS with or without mixup. Light shade: true norm. Dark lines: smooth version. *Best viewed on screen.*

As mentioned in Sec. 3.1 and Fig. 2, we observe that when applying mixup (orange line), the weight norms of the tail classes tend to be larger and the weight norms of the head classes are decreased, which means mixup may be more friendly to the tail classes. Here, we show more evidences that mixup reduces dominance of the head classes. In Figs. 13 and 14, norm of these variants are trained on Places-LT and iNaturalist 2018, respectively. The results are similar and consistent with those trained on ImageNet-LT.