

# Long-tailed Visual Recognition via Gaussian Clouded Logit Adjustment

Mengke Li<sup>1</sup> Yiu-ming Cheung<sup>1\*</sup> Yang Lu<sup>2</sup>

<sup>1</sup>Department of Computer Science, Hong Kong Baptist University, Hong Kong

<sup>2</sup>Department of Computer Science and Technology, School of Informatics, Xiamen University, China

{csmkli, ymc}@comp.hkbu.edu.hk, luyang@xmu.edu.cn

## Abstract

Long-tailed data is still a big challenge for deep neural networks, even though they have achieved great success on balanced data. We observe that vanilla training on long-tailed data with cross-entropy loss makes the instance-rich head classes severely squeeze the spatial distribution of the tail classes, which leads to difficulty in classifying tail class samples. Furthermore, the original cross-entropy loss can only propagate gradient short-livedly because the gradient in softmax form rapidly approaches zero as the logit difference increases. This phenomenon is called softmax saturation. It is unfavorable for training on balanced data, but can be utilized to adjust the validity of the samples in long-tailed data, thereby solving the distorted embedding space of long-tailed problems. To this end, this paper proposes the Gaussian clouded logit adjustment by Gaussian perturbation of different class logits with varied amplitude. We define the amplitude of perturbation as cloud size and set relatively large cloud sizes to tail classes. The large cloud size can reduce the softmax saturation and thereby making tail class samples more active as well as enlarging the embedding space. To alleviate the bias in a classifier, we therefore propose the class-based effective number sampling strategy with classifier re-training. Extensive experiments on benchmark datasets validate the superior performance of the proposed method. Source code is available at <https://github.com/Keke921/GCLLoss>.

## 1. Introduction

Deep neural networks (DNNs) have been widely utilized in a variety of visual recognition problems [6, 7, 21, 28] by virtue of the large-scale, high-quality, and annotated datasets. DNNs usually require the training dataset to be artificially balanced and have sufficient samples of each class. Unfortunately, from a practical perspective, object frequency usually follows a power law and typically ex-

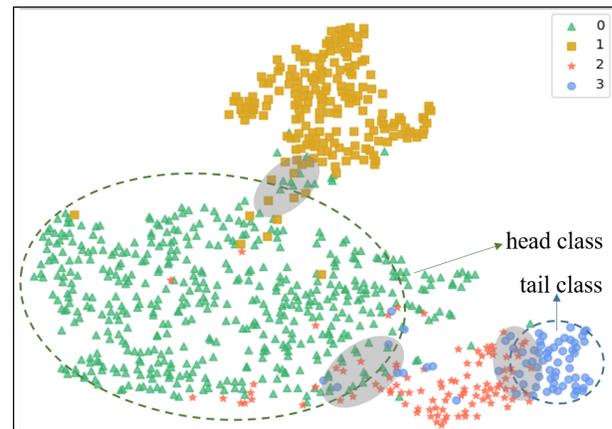


Figure 1. t-SNE visualization of the distorted embedding space. (Color for the best view.) The embeddings are calculated with ResNet-32 on a subset with four classes of CIFAR-10-LT. We randomly select four classes with the training numbers 500, 200, 100, and 50, respectively. The gray areas show the obscure regions between different classes.

hibits a long-tailed distribution. Naive learning on such data is prone to undesirable bias towards the head classes which occupy the majority of the training samples [37]. Since tail classes have few training samples that cannot cover the real distribution in embedding space, their spatial span is severely compressed by head classes. In addition, a vast number of head class samples generate overwhelming discouraging gradients for tail classes. Thus, the learning of a classifier is biased towards the head classes. As a result, directly training on long-tailed data brings two key problems: 1) the distorted embedding space, and 2) the biased classifier.

In the literature, most of the recently proposed approaches focus on addressing the second problem only, *i.e.*, the biased classifier. For example, Menon *et al.* [17] and Hong *et al.* [8] applied post-adjust strategy to the trained model to calibrate the class boundary. Nevertheless, the distorted embedding cannot be adjusted with the post-hoc calibration, which is not conducive to further improving the

\*Yiu-ming Cheung is the Corresponding Author.

model performance. Most recently, the two-stage decoupling methods [2, 10, 31, 35, 40] have been proposed to obtain good embeddings in the first stage and then re-balance the classifier in the second stage. These methods obtain the representation by cross-entropy (CE) loss, which, however, leads to a severely uneven distributed embedding space. We implement a toy experiment to illustrate the distortion of the embedding space as shown in Fig. 1, where t-SNE [25] is utilized to visualize the features of a long-tailed subset from CIFAR-10 dataset. We can observe that the tail class occupies a much small spatial span than the head class. This is because the tail class with fewer samples cannot cover the ground truth distribution. Moreover, Fig. 1 also shows that there are obscure regions (*i.e.*, the grey area) between different classes. Softmax saturation [3] is one of the factors of these obscure regions because it leads to insufficient training. These obscure regions have a severe effect on the tail classes but little on the head classes. Since tail class samples clustered around the class boundary aggravate their spatial squeezing, while the head class samples with enough variety can already cover the true distribution.

Softmax saturation refers to the inopportune early gradients vanishing produced by the softmax [3, 36], which weakens the validity of training samples and impedes model training. However, from another perspective, the seemingly harmful softmax saturation has the ability to balance the valid samples of different classes and thus help calibrate the distortion of embedding space. Specifically, we disturb the logit of different classes with different amplitudes. We name the disturbed logit as Gaussian clouded logit (GCL) and the amplitude of the disturbance as cloud size, because we set the disturbance to a Gaussian distribution. The tail classes have few training samples and thus the training samples of them should be more valid. We therefore disturb the logit of tail classes with large relative cloud sizes to reduce the softmax saturation. In this way, tail class samples can provide more gradients without overfitting and thus indirectly affect their embedding space. In addition, a large cloud size of the tail class logit corresponds to the large cloud size on feature in the direction of the class anchor. Therefore, tail classes can have large margins towards the class boundary, so as to alleviate the severe uneven distribution between the head and tail classes. Conversely, the head classes are set to small cloud sizes, so that they can be automatically filtered out during training. Eventually, as shown in Fig. 2, the tail class samples can be pushed more away from the class boundary so as the distortion of the embedding space can be calibrated.

To address the biased classifier, we re-balance the training data with a class-wise sampling strategy. As training with GCL makes the validity of different classes vary, the so-called “effectiveness” [4] of them are different. Existing class-wise balanced sampling strategies will lead to exces-

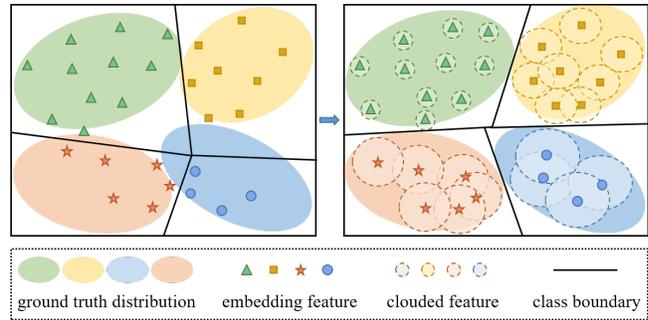


Figure 2. An overview of GCL. (Color for the best view.) The tail class logit is assigned to a larger sample cloud size than the head class, which corresponds to a large relative cloud size of the feature in the direction of the tail class anchor. In this way, the distortion of the embedding space can be calibrated well.

sive training of tail classes for GCL. We thereby propose the class-based effective number (CBEN) sampling strategy, which is based on sample validity and label frequencies to re-balance the classifier. This simple but effective sampling strategy helps mitigate the classifier bias towards the head classes and further boost the performance of GCL.

Extensive experiments on multiple commonly used long-tailed recognition benchmark datasets demonstrate that the proposed GCL surpasses the recently proposed counterparts. In summary, the key contributions of our work are three-fold:

- We propose the GCL adjustment loss function, which utilizes softmax saturation to balance the sample validity of different classes. An evenly distributed embedding can be obtained with the proposed GCL.
- We propose a simple but effective class-based effective number (CBEN) sampling strategy for re-balancing the classifier to avoid repeat training of tail classes. This sampling strategy can further boost the performance of GCL.
- Extensive experiments on popular long-tailed datasets demonstrate that the proposed method outperforms the state-of-the-art counterparts.

## 2. Related Works

Long-tailed classification is one of the long-standing research problems in machine learning. Several kinds of approaches have been proposed to address it. This section briefly introduces the most related three regimes, namely loss modification, logit adjustment, and decoupling representation.

### 2.1. Loss Modification

Modifying the loss function through re-weighting is the most natural method. Sample-wise re-weighting meth-

ods [15, 20] attempt to make the model pay more attention to the difficult samples by introducing fine-grained coefficients in the loss for imbalanced learning. For example, focal loss [15] introduces a tunable focusing parameter, which is negatively correlated with the predicted probability of the target class. This focusing parameter helps the model training focus on hard samples and prevents the numerous easy negatives from overwhelming. Class-wise re-weighting methods [4, 9, 11, 23] assign the standard CE loss with category-specific parameters that are inversely proportional to the class frequencies. For example, Tan *et al.* [23] proposed equalization loss, which utilizes a weight term to randomly ignore the discouraging gradients of head class samples. These methods can alleviate the data imbalance to a certain extent. However, the classification difficulty of a sample is not directly related to its corresponding class size. Further, another side effect of assigning higher weights to difficult samples/tail classes is overly focusing on harmful samples (*e.g.*, noisy data or mislabeled data) [13].

## 2.2. Logit Adjustment

Logit adjustment assigns relatively large margins for tail classes. Most recently, Menon *et al.* [17] have proposed a logit adjustment (LA) method which is consistent with minimizing the balanced error. The logit shifting in LA of different classes is based on label frequencies of training data. Differently, LADE [8] calibrates the logit to the test set using the label distribution of test data, so that the test set can also be imbalanced. Tang *et al.* [24] adopted causal intervention to remove the “bad” SGD momentum and keep the “good” one to avoid the harmful causal effect for tail prediction. DisAlign [35] adjusts the logit by calibrating the model prediction to a reference distribution of classes that favors the balanced prediction. These methods well adjust the model logits through post-hoc shifting but without considering calibrating the embedding space. Another type of approach [1, 2] addresses long-tailed data by leaving large relative margins for tail classes during training. For example, label-distribution-aware margin (LDAM) loss proposed by Cao *et al.* [2] utilizes Rademacher complexity to theoretically prove that the margin should be inversely proportional to a quarter power of label frequencies. The hard margin on target logit helps make the intro-class samples more compact, but does not truly enlarge the tail class span in embedding space.

## 2.3. Decoupling Representation

Many recent works have focused on improving the long-tailed visual recognition performance by decoupling the representation and classifier. Most recently, LDAM-DRW [2] has been proposed, which learns features in the first stage and adopts the deferred re-weighting (DRW) to fine-tune the decision boundary in the second stage. It

significantly improves the long-tailed prediction accuracy, but the theoretical explanation of DRW is not clear. After that, Kang *et al.* [10] precisely pointed out that the learning process of representation and classifier can be decoupled into two separate stages. The representation learning is conducted on the original long-tailed data in the first stage and the classifier learning is performed on class-balanced re-sampling data in the second stage. A lot of works [31, 32, 35, 39] have further refined this strategy. For example, Zhang *et al.* [35] proposed an adaptive calibration function to calibrate the predicted logits of different classes into a balanced class prior in the second stage. Zhong *et al.* [39] proposed label distribution-based soft label to deal with different degrees of over-confidence for classes and can improve the classifier learning in the second stage. Another alternative direction is proposed by Zhou *et al.* [40], which splits the network structure into two branches that focus on learning the representation of head and tail classes, respectively. This method incorporates feature mixup [27] into a cumulative learning strategy and also achieves the state-of-the-art results. Following [40], Wang *et al.* [30] introduced contrastive learning into this bilateral-branch network to further improve the long-tailed classification performance.

## 3. Proposed Approach: GCL

The key idea of our proposed GCL is to utilize the softmax saturation to automatically balance the valid samples of head and tail classes. The theoretical motivation and the formulation of the loss function of the proposed approach are presented as follows.

### 3.1. Motivation

Fig. 1 shows that the obscure region among different classes, especially the tail class, is large. One important factor of this obscure region is the softmax saturation in CE loss [3]. Suppose  $\{x, y\} \in \mathcal{T}$  represents a sample  $\{x, y\}$  from the training set  $\mathcal{T}$  with the total  $N$  samples in  $C$  classes, and  $y \in \{1, \dots, C\}$  is the ground truth label. The softmax loss function for the input image  $x$  can be written as:

$$\mathcal{L}(x) = -\log p_y, \text{ with } p_y = \frac{e^{z_y}}{\sum_{j=1}^C e^{z_j}}, \quad (1)$$

where  $z_j$  represents the predicted logit of class  $j$ . We use the subscript  $y$  ( $j \neq y$ ) to represent the target class. That is,  $z_y$  indicates the target logit and  $z_j$  ( $j \neq y$ ) is the non-target logit.

In backward propagation, the gradients on  $z_j$  is calculated by:

$$\frac{\partial \mathcal{L}}{\partial z_j} = \begin{cases} p_j - 1, & j = y \\ p_j, & j \neq y. \end{cases} \quad (2)$$

Without loss of generality, we use the binary classification as an example. Supposing  $x$  is from class 1, the gradients on  $z_1$  is then calculated by:

$$\frac{\partial \mathcal{L}}{\partial z_1} = -\frac{1}{1 + e^{z_1 - z_2}}. \quad (3)$$

Eq. (3) indicates that the gradient of the target class rapidly approaches zero with the increase of the logit difference. Softmax can only slightly separate various classes, and lacks the power to evenly distribute each class in the embedded space. Therefore, there are many overlapping areas among the classes. In particular, under the circumstances of long-tailed classification, the tail class features are not sufficient to cover the real distribution in embedding space. The early gradient vanish caused by softmax saturation exacerbates the squeezing of their embedding space. A straightforward approach is to introduce hard margin [2, 5, 36]. However, the hard margin will cause the samples to shrink towards the class anchor and easy to overfit tail classes, which cannot evenly distribute the embedding space well. Fortunately, softmax saturation can help filter out the head class samples and make the tail class samples fully participate in training. In this way, the tail classes can be pushed away from the head classes and indirectly enlarge their embedding space.

### 3.2. Embedding Space Calibration

Suppose the features of different class samples satisfy Gaussian distribution. We can obtain a disturbed feature  $\mathbf{f}^{cld}$  of the input by Gaussian sampling, which is represented as:

$$\mathbf{f}^{cld} \triangleq \mathbf{f} + \delta \mathbf{E}, \quad (4)$$

where  $\mathbf{f} \in \mathbb{R}^D$  is the feature obtained from the embedding layer with the dimension of  $D$ .  $\mathbf{E} \sim \mathcal{N}(\mathbf{u}, \Sigma)$  is the disturbance sampled from Gaussian distribution, and the mean vector and covariance matrix are represented by  $\mathbf{u} \in \mathbb{R}^D$  and  $\Sigma \in \mathbb{R}^{D \times D}$ , respectively.  $\delta > 0$  is a parameter that is used to adjust the amplitude of disturbance. In addition,  $\delta$  should be a small number because a large disturbance will mislead the model. This disturbed feature is the input of the classifier. We use  $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_C\} \in \mathbb{R}^{D \times C}$  to represent the weight matrix of the classifier, where  $\mathbf{w}_j$  represents the anchor vector of class  $j$  in the classifier. Then, the corresponding disturbed logit  $z_j^{cld}$  of class  $j$  is calculated by:

$$\begin{aligned} z_j^{cld} &= \mathbf{w}_j^T \mathbf{f}^{cld} + \mathbf{b}_j \\ &= \mathbf{w}_j^T \mathbf{f} + \mathbf{b}_j + \mathbf{w}_j^T (\delta \mathbf{E}) \\ &= z_j + \delta (\mathbf{w}_j^T \mathbf{E}). \end{aligned} \quad (5)$$

As the range of  $z_j^{cld}$  is enlarged with random Gaussian disturbances, we call it Gaussian clouded logit, and  $\delta (\mathbf{w}_j^T \mathbf{E})$  is

the clouded term. Please note that the clouded term has the different degrees of influence on the final predicted results based on different predicted logits. It has a relatively small impact on  $z_j^{cld}$  when the original logit  $z_j$  is large. On the contrary, it will play a key role for  $z_j^{cld}$  when  $z_j$  is small. As a result, we need to normalize the effect caused by different predicted logits and maintain the consistency of the influence of the clouded term. Inspired by [5, 28, 29], we normalize the clouded logits based on cosine distance. In this way, the norm of the feature and the class anchor can be normalized to the fixed numbers. We use  $s_1$  and  $s_2$  to represent these two numbers. The normalized clouded logit is named *clouded cosine logit*, which is calculated by:

$$\begin{aligned} \tilde{z}_j^{cld} &= \frac{s_1 \mathbf{w}_j^T \cdot s_2 \mathbf{f}^{cld}}{\|\mathbf{w}_j^T\| \|\mathbf{f}^{cld}\|} \\ &= s \cdot \left( \frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta \mathbf{E}\|} + \delta \frac{\mathbf{w}_j^T \mathbf{E}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta \mathbf{E}\|} \right), \end{aligned} \quad (6)$$

where  $s = s_1 \cdot s_2$  is a constant. In the first term of Eq. (6),  $\|\mathbf{f} + \delta \mathbf{E}\| \approx \|\mathbf{f}\|$  because  $\delta$  is a small number. In the second term, the norm of feature is normalized to  $s_1$ . Thus,  $\tilde{z}_j^{cld}$  can be simplified as:

$$\tilde{z}_j^{cld} \approx s \cdot \left( \frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f}\|} + \frac{\delta}{s_1} I_j \mathbf{E} \right), \quad (7)$$

where  $I_j$  is the identity vector that has the same direction as  $\mathbf{w}_j^T$ . In order to simplify the calculation, we make the clouded cosine logit still satisfy the Gaussian distribution. Thus, we introduce a constant  $\sigma$  and set the covariance matrix  $\Sigma = \sigma \mathbf{I}$ , where  $\mathbf{I} \in \mathbb{R}^{D \times D}$  is the identity matrix. Then,  $I_j \mathbf{E}$  is the projection of the noise sampled by Gaussian in the direction of the anchor vector of class  $j$ . We denote its magnitude by  $\varepsilon_j$ . Therefore,  $\tilde{z}_j^{cld}$  can be calculated by:

$$\begin{aligned} \tilde{z}_j^{cld} &= s \cdot (\tilde{z}_j + \frac{\delta}{s_1} \varepsilon_j) \\ &\Leftrightarrow s \cdot (\tilde{z}_j + \delta_j \varepsilon), \end{aligned} \quad (8)$$

where  $\tilde{z}_j = \cos \theta_j$  is the cosine distance, and  $\theta_j$  is the angle between  $\mathbf{f}$  and  $\mathbf{w}_j$ .  $\delta_j$  is the logit cloud size that depends on different classes.

To achieve the two goals mentioned in Sec. 3.1, i.e., 1) encourage tail class samples to participate more in training; 2) enlarge the embedding space for the tail classes, the size of logit cloud should be negatively correlated with the number of training samples. For the most frequent class, the diversity of training samples is sufficient and we set its logit cloud size to zero, while utilizing larger cloud sizes for tail classes. The merits of this large relative cloud size of tail classes are three-fold: 1) reduce the softmax saturation and thereby increase the training degree of tail classes;

2) different values are sampled randomly from the Gaussian cloud so as to avoid overfitting; 3) enlarge the margin of class boundary for tail classes and can calibrate the distortion of the embedding space. We therefore empirically set the cloud size for class  $j$  as:

$$\delta_j = \log n_{max} - \log n_j, \quad (9)$$

where  $n_{max}$  is the sample numbers of the most frequent class. We experimentally verify the effectiveness of this cloud size adjustment strategy in Sec. 4.5.2.

The Gaussian clouded logit difference  $\Delta_{y-j}$  between the target and non-target classes is:

$$\begin{aligned} \Delta_{y-j} &= z_y^{cld} - z_j^{cld} \\ &= z_y - z_j + \varepsilon(\delta_y - \delta_j). \end{aligned} \quad (10)$$

If  $\varepsilon > 0$ ,  $\Delta_{y-j}$  for tail classes will be increased. However, our goal is to reduce the logit difference to alleviate the softmax saturation for tail classes. In addition, a reduced logit corresponds to the feature that is relatively far from the class anchor. If the relatively distant feature can be predicted correctly, the closer one will be able to assign the right label. Therefore, we require  $\varepsilon$  to be negative. Subsequently, the clouded cosine logit can be written in the following form:

$$\tilde{z}_j^{cld} = s \cdot (\tilde{z}_j - \delta_j \|\varepsilon\|). \quad (11)$$

Taking the clouded cosine logit into the original softmax, we can obtain the loss function of GCL:

$$\mathcal{L}_{GCL} = -\frac{1}{N} \sum_i \log \frac{e^{\tilde{z}_{y_i}^{cld}}}{\sum_j e^{\tilde{z}_j^{cld}}}. \quad (12)$$

### 3.3. Classifier Re-balance

The gradients derived in Eq. (2) demonstrate that the sample of the target class  $y$  punishes the classifier weights  $\mathbf{w}_j$  of non-target class  $j, j \neq y$  w.r.t.  $p_j$ . The head classes have enormously greater training instances than tail classes. Therefore, the classifier weights of tail classes receive much more penalty than positive signals during training. Consequently, the classifier will bias towards the head classes, and the predicted logits of the tail classes will be seriously suppressed, resulting in low classification accuracy of the tail classes. A straightforward approach is to use the re-sampled data to re-train the classifier. We apply the classifier re-training (cRT), which was adopted by Kang *et al.* [10] and Wang *et al.* [31]. As the GCL loss enables different class samples to participate in training to different degrees, the effectiveness of different class samples is varied. Class-balanced sampling will lead to repeat training for tail classes. Drawing on the effective number proposed by Cui *et al.* [4], we propose the class-based effective number (CBEN) sampling to avoid excessive training of tail classes.

---

### Algorithm 1: Gaussian clouded logit

---

**Input:** Training dataset  $\mathcal{T}$ ;  
**Output:** Predicted labels;

- 1 Initialize the model parameters  $\omega$  of the CNN network  $\phi((x, y); \omega)$  randomly;
- 2 **for**  $iter = 1$  to  $I_0$  **do**
- 3     Sample a batch samples  $\mathcal{B}$  from the original long-tailed data  $\mathcal{T}$  with batch size  $b$ ;
- 4     Obtain the logit cloud size:  
 $\delta_j \leftarrow \log n_{max} - \log n_j$ ;
- 5     Calculate the loss by Eq. (12):  
 $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x, y) \in \mathcal{B}} L_{GCL}(x, y)$ ;
- 6     Update model parameters:  
 $\omega = \omega - \alpha \nabla_{\omega} \mathcal{L}((x, y); \omega)$ .
- 7 **end**
- 8 **for**  $iter = I_0 + 1$  to  $I_0 + I_1$  **do**
- 9     Calculate sampling rate:  
 $\beta_j \leftarrow b \times \frac{\delta_j - \delta_{max}}{\delta_{max} - \delta_{min}} + a$ ;  $\rho_j \leftarrow \frac{1 - \beta_j}{1 - \beta_j^{n_j}}$ ;  
 $\rho_j \leftarrow \frac{\rho_j}{\sum_i \rho_i}$ ;
- 10     Sample a batch samples  $\mathcal{B}'$  with the sampling probability  $\rho_j$  and the batch size  $b$ ;
- 11     Calculate the loss by Eq. (12):  
 $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x, y) \in \mathcal{B}'} L_{GCL}(x, y)$ ;
- 12     Update classifier parameters  $\omega_{cls}$  (representation parameters are frozen):  
 $\omega_{cls} = \omega_{cls} - \alpha \nabla_{\omega_{cls}} \mathcal{L}((x, y); \omega_{cls})$ .
- 13 **end**

---

The sampling probability  $\rho_j$  of a sample from class  $j$  is calculated by:

$$\rho_j = \frac{1 - \beta_j}{1 - \beta_j^{n_j}}. \quad (13)$$

Since the sum of the sampling probability for all data needs to be 1, we normalize  $\rho_j$  by  $\rho_j \leftarrow \frac{\rho_j}{\sum_i \rho_i}$ .  $\beta_j$  reflects the validity of different class samples. The class samples with large cloud size participate more in training. Therefore,  $\beta_j$  is positively correlated with cloud size  $\delta_j$ . We set  $\beta_j$  as:

$$\beta_j = b \times \frac{\delta_j - \delta_{min}}{\delta_{max} - \delta_{min}} + a, \quad (14)$$

so that  $\beta_j$  can be in the region  $[a, a + b]$ , where  $a$  and  $b$  are the range hyper-parameters.

The overall training procedure of the proposed method is summarized in Algorithm 1.

## 4. Experiments

### 4.1. Datasets

We use five benchmarks: long-tailed CIFAR datasets that include CIFAR-10-LT and CIFAR-100-LT, long-tailed

ImageNet-2012 (ImageNet-LT), iNaturalist 2018 [26] and long-tailed Places-2 (Places-LT). The original version of CIFAR-10/100 [14], ImageNet-2012 [22] and Places-2 [41] are all balanced datasets. We follow Cao *et al.* [2] and Cui *et al.* [4] to create long-tailed versions of CIFAR-10/100 and use the long-tailed versions of ImageNet-2012 and Places-2 produced by Liu *et al.* [16].

**CIFAR-10/100-LT.** The original CIFAR-10 and CIFAR-100 consist of 10 and 100 classes, respectively. They both have 60,000 color images of size  $32 \times 32$ . 50,000 of them are used for training and the remaining images are for validation. Following [2, 4], we down-sampling training samples per class with the exponential function  $n_i = n_{o_i} \times \mu^i$ , where  $i$  is the class index (0-indexed),  $n_{o_i}$  is the number of training samples in original CIFAR and  $\mu \in (0, 1)$ . The validation sets are kept unchanged. The imbalance ratio  $\gamma$  is defined as the ratio of the sample size of the most and the least frequent classes, *i.e.*  $\gamma = \max(n_i) / \min(n_i)$ ,  $i = 0, 1, \dots, C - 1$ .  $\gamma$  is set at its common values, *i.e.*  $\gamma = 50, 100$  and  $200$ , in our experiments.

**ImageNet-LT and Places-LT.** The balanced versions of ImageNet-2012 and Places-2 are large-scale real-world datasets for classification and localization. We follow Liu *et al.*'s work [16] to construct the long-tailed version of these two datasets by truncating a subset with the Pareto distribution with the power value  $\alpha = 6$  from the balanced versions. The original balanced validation sets remain unchanged. Overall, ImageNet-LT has 115.8K training images from 1,000 categories with  $\gamma = 1,280/5$ . Places-LT contains 62.5K training images from 365 categories with  $\gamma = 4,980/5$ .

**iNaturalist 2018.** The 2018 version of iNaturalist is a real-world fine-grained dataset for classification and detection, which exhibits extremely imbalanced distribution. It contains 437.5K training images and 24.4K validation images from 8,142 categories. We follow the official splits of training and validation sets in the experiments.

## 4.2. Experimental Setting

The pre-setting parameters in the first stage were the Gaussian distribution parameters  $(\mu, \sigma^2)$  and the region  $[a, b]$  of sample validity  $\beta_j$ . We know that  $\tilde{z}_i \in [-1, 1]$ , thus the maximum feature cloud size cannot exceed 1. Since Gaussian distribution has a probability of about 99.7% falling in  $[\mu - 3\sigma, \mu + 3\sigma]$ , we set  $\mu = 0$  and  $\sigma = \frac{1}{3}$ . We further clamped the  $\varepsilon$  to  $[-1, 1]$  to prevent its amplitude from exceeding 1. We set  $\beta_j \in [0.999, 0.9999]$ , *i.e.*  $a = 0.999$  and  $b = 0.0009$ . Moreover, we normalized  $\delta_i$ ,  $i = \{1, 2, \dots, C\}$  by  $\delta_i \triangleq \delta_i / \delta_{max}$  to ensure that the maximum value of  $\delta_i$  did not exceed 1. Similar with Zhong *et al.* [39], the mixup [33] strategy was also adopted in our experiments.

We utilized PyTorch [19] to implement all the back-

bones. SGD optimizer with momentum of 0.9 and the multi-step learning rate schedule were adopted. All the models were trained from scratch except ResNet-152 that was pre-trained on the original balanced version of ImageNet-2012. For the first stage, we selected ResNet-32 as the backbone network and followed the setting in Cao *et al.* [2] for CIFAR-10/100-LT. For the large-scale dataset, namely ImageNet-LT, iNaturalist 2018, and Places-LT, we mainly followed Kang *et al.* [10] except the learning rate schedule. For the second stage, *i.e.*, re-balancing the classifier, we followed Kang *et al.* [10] for all datasets.

## 4.3. Competing Methods

To verify the effectiveness of the proposed method, we have conducted extensive experiments to compare with the previous methods, including the following two groups:

**Baseline Methods.** We implemented vanilla training with cross-entropy (CE) loss as one of our baseline methods. Many visual recognition works [12, 18, 34, 38] have shown the efficacy of mixup, CE loss cooperated with mixup was therefore also compared.

**State-of-the-art Methods.** The recently proposed representation learning method, namely OLTR [16] and logit adjustment method, namely De-confound-TDE inference [24] were compared. We also compared with the two-stage methods including LDAM-DRW [2] and MisLAS [39], which both achieve satisfactory classification accuracy on the aforementioned long-tailed datasets. For CIFAR-10/100-LT datasets, we made comparison with BBN [40] and contrastive learning [30]. For the large-scale datasets, we compared with the most recently proposed two-stage methods, including decoupling [10], logit adjustment [17] and DisAlign [35]. For a fair comparison, we additionally conducted the comparison experiment with the two-stage strategy which added classifier re-training (cRT) [10] to CE loss + mixup on all datasets.

## 4.4. Comparison Results

Comparative studies have been conducted to show the efficacy of the proposed GCL. The results are presented in Tab. 1 and Tab. 2. We use top-1 accuracy on test sets as the performance metric. For the results from those papers that have yet to release the code or relevant hyper-parameters, we directly quote their results from the original papers.

### 4.4.1 Experimental Results on CIFAR-10/100-LT

The results on CIFAR-10/100-LT datasets are summarized in Tab. 1. We can observe that our proposed GCL outperforms the previous methods by notable margins with all imbalanced ratios. Especially for the largest one, *i.e.*,  $\gamma = 200$ , the proposed approach has obvious improvement. We get 79.03% and 44.88% in top-1 classification accuracy

Table 1. Comparison results on CIFAR-10/100-LT in terms of top-1 accuracy (%), where the best and the second-best results are shown in **underline bold** and **bold**, respectively. \*indicates that the results are quoted from the corresponding references. The other results are obtained by re-implementing with the official codes.

Dataset	CIFAR-10-LT			CIFAR-100-LT		
Backbone Net	ResNet-32					
Imbalance ratio	200	100	50	200	100	50
CE loss	65.68	70.70	74.81	34.84	38.43	43.9
CE loss + mixup [33] (2018)	65.84	72.96	79.48	35.84	40.01	45.16
LDAM-DRW [2] (2019)	73.52	77.03	81.03	38.91	42.04	47.62
De-confound-TDE * [24] (2020)	-	80.60	83.60	-	44.15	50.31
CE loss + mixup + cRT [10] (2020)	73.06	79.15	84.21	41.73	45.12	50.86
BBN [40] (2020)	73.47	79.82	81.18	37.21	42.56	47.02
Contrastive learning * [30] (2021)	-	81.40	<b>85.36</b>	-	46.72	51.87
MisLAS [39] (2021)	<b>77.31</b>	<b>82.06</b>	85.16	<b>42.33</b>	<b>47.50</b>	<b>52.62</b>
GCL	<b>79.03</b>	<b>82.68</b>	<b>85.46</b>	<b>44.88</b>	<b>48.71</b>	<b>53.55</b>

Table 2. Comparison results on ImageNet-LT, iNaturalist 2018 and Places-LT in terms of top-1 accuracy (%), where the best and the second-best results are shown in **underline bold** and **bold**, respectively. \*indicates that the results are quoted from the corresponding references. The other results are obtained by re-implementing with the official codes.

Dataset	ImageNet-LT	iNaturalist 2018	Places-LT
Backbone Net	ResNet-50	ResNet-50	ResNet-152
CE loss	44.51	63.80	27.13
CE loss + mixup [33] (2018)	45.66	65.77	29.51
LDAM-DRW [2] * (2019)	48.80	68.00	-
OLTR * [16] (2019)	-	-	35.9
Decoupling [10] (2020)	47.70	69.49	37.62
CE loss + mixup + cRT [10] (2020)	51.68	70.16	38.51
Logit adjustment * [17](2021)	51.11	66.36	-
DisAlign * [35] (2021)	<b>52.91</b>	70.06	39.30
MisLAS [39] (2021)	52.11	<b>71.57</b>	<b>40.15</b>
GCL	<b>54.88</b>	<b>72.01</b>	<b>40.64</b>

for CIFAR-10-LT and CIFAR-100-LT with  $\gamma = 200$ , which surpasses the second best method, *i.e.*, MisLAS by a significant margin of 1.72% and 2.55%, respectively.

#### 4.4.2 Experimental Results on Large-scale Latasets

The results on three large-scale long-tailed datasets, *i.e.*, ImageNet-LT, iNaturalist 2018, and Place-LT, are reported in Tab. 2. Our approach is superior to prior art on all datasets. On ImageNet-LT, our method achieves 54.88% top-1 accuracy, outperforming DisAlign by a large margin at 1.97% and MisLAS at 2.77%, respectively. On iNaturalist 2018, the proposed approach achieves 72.01% top-1 accuracy, which outperforms the second-best method by 0.44%. On Place-LT, our method achieves 40.64% top-1 classification accuracy, with a performance gain at 0.49% over MisLAS. Although the performance gain compared with MisLAS on iNaturalist 2018 and Place-LT is not as high as other datasets, our method does not require hyper-parameters searching for different datasets, and thus it is relatively easy to implement.

#### 4.5. Model Validation and Analysis

We conduct a series of ablation studies to further analyze the proposed method.

##### 4.5.1 The Role of Gaussian Clouded Logit

In order to obtain additional insight, we utilized t-SNE projection of the embedding for visualization. Since the loss functions of baseline and MisLAS are both CE loss and MisLAS performed the second-best in most cases we have tried so far, we visualized CE loss embedding for comparison. The embeddings were calculated from the samples in CIFAR-10-LT with  $\gamma = 100$ . Fig. 3 shows the visualization results on the training and test set. From the result of the training set (Fig. 3a), we can see that the embeddings obtained via GCL of different classes are more scattered. Therefore, the GCL embedding of each class is much easier to separate. The results of the test set shown in Fig. 3b justify the efficacy of our proposed approach. The obscure region of CE loss embedding is larger than that of GCL em-

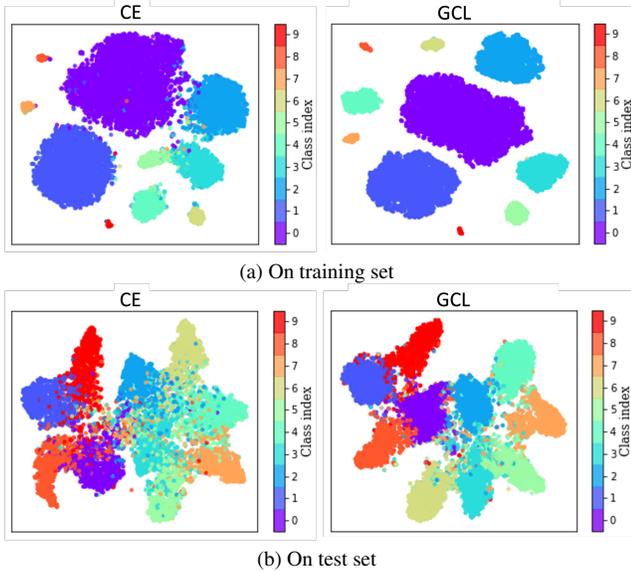


Figure 3. Visualization of the embedding via t-SNE from CIFAR-10-LT with  $\gamma = 100$ , where backbone network is ResNet-32. (Color for the best view.)

Table 3. Ablation experiment of different cloud size adjustment strategies (AS) on CIFAR-10-LT with  $\gamma = 100$ .

AS	Expression	Acc.(%)
cos.	$\cos(n_j/n_{max} \cdot \pi/2)$	79.21
pow. diff. (e:1/3)	$n_{max}^{1/3} - n_j^{1/3}$	80.80
pow. diff. (e:1/4)	$n_{max}^{1/4} - n_j^{1/4}$	82.31
log. diff.	$\log n_{max} - \log n_j$	<b>82.68</b>

bedding. Good embedding helps improve the model performance. We only re-fine the classifier with the simple cRT without any other complicated technologies, but the classification accuracy can be improved a lot.

#### 4.5.2 Cloud Size Adjustment Strategy

We explored several different cloud size adjustment strategies (AS), which included cosine form (cos.), power difference (pow. diff.) with different exponents (e:1/3 and e:1/4), and logarithmic difference (log. diff.). For a fair comparison, the sampler and re-training strategy were selected as CBEN and cRT, respectively. Tab. 3 shows the results. We choose the log. diff. strategy according to Tab. 3.

#### 4.5.3 Classifier Re-balance Strategies

We compared different strategies of data re-sampling and the classifier re-training to better analyze our proposed method. The re-sampling strategy (sam.) included: instance balance (IB) [10], class balance (CB) [10], class

Table 4. Ablation experiment of different re-sampling strategy on CIFAR-10-LT with  $\gamma = 100$ . Table 5. Ablation experiment of different re-training strategies on CIFAR-10-LT with  $\gamma = 100$ .

Sam.	RT	Acc.(%)	Sam.	RT	Acc.(%)
IB	cRT	80.41	-	w/o RT	80.52
CB	cRT	82.43	CBEN	LWS	82.25
EN	cRT	82.47	CBEN	$\tau$ -nor.	82.16
CBEN	cRT	<b>82.68</b>	CBEN	cRT	<b>82.68</b>

balance with effective number (EN) [4], and our proposed class-based effective number (CBEN). For a fair comparison, the re-training strategies for all samplers were cRT. Tab. 4 shows the effectiveness of CBEN. For the selection of classifier re-training strategy, we first trained the backbone without any classifier re-training technology. Then, we fixed the representation and re-balance the classifier with learnable weight scaling (LWS) [10],  $\tau$ -normalization ( $\tau$ -nor.) [10], and cRT, respectively. Tab. 5 presents the top-1 accuracy of CIFAR-10-LT with  $\gamma = 100$ . We can observe that, even without any classifier re-training technique, our approach can still beat most state-of-the-arts including two-stage methods. For example, our GCL without classifier re-training suppresses BBN by 0.7%. Further, cRT performs the best among the classifier re-training strategies, which improves the top-1 accuracy by 1.64%. From Tab. 4 and Tab. 5, we can observe that IB+cRT degrades model performance, which indicates that training the classifier with IB may lead to classifier overfitting.

## 5. Conclusion

In this paper, we have found that softmax saturation reduces sample validity, which has different effects on head and tail classes. This implies that, from another perspective, softmax saturation can be utilized to automatically adjust the training sample validity of different classes. Subsequently, we have proposed the GCL. The tail class logits are set to relatively large cloud sizes to encourage more tail class samples to participate in training as well as leave large margins, which help obtain evenly distributed embedding space. The effectiveness of different classes is varied via GCL. Then, the simple but effective CBEN sampling strategy incorporated with cRT for classifier balancing has been proposed, which can further boost the model performance. Extensive experiments on various benchmark datasets have demonstrated that the proposed GCL has superior performance compared to the existing state-of-the-art methods.

**Acknowledgment** This work was supported in part by NSFC/RGC JRS Grant: N\_HKBU214/21, ORP of Zhejiang Lab: 2021KB0AB03, GRF Grant: 12201321, NSFC Grants: 62002302 and 61672444, NSF of Fujian Province: 2020J01005, HKBU Grants: RC-FNRA-IG/18-19/SCI/03.

## References

- [1] Dong Cao, Xiangyu Zhu, Xingyu Huang, Jianzhu Guo, and Zhen Lei. Domain balancing: Face recognition on long-tailed domains. In *CVPR*, 2020. 3
- [2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arachiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pages 1567–1578, 2019. 2, 3, 4, 6, 7
- [3] Binghui Chen, Weihong Deng, and Junping Du. Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, 2017. 2, 3
- [4] 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. 2, 3, 5, 6, 8
- [5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 4
- [6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1
- [8] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *CVPR*, pages 6626–6636, 2021. 1, 3
- [9] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016. 3
- [10] 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. 2, 3, 5, 6, 7, 8
- [11] Salman H. Khan, Munawar Hayat, Mohammed Bannamoun, Ferdous Ahmed Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE TNNLS*, 29(8):3573–3587, 2018. 3
- [12] Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with super-modular diversity. In *ICLR*, 2021. 6
- [13] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, volume 70, pages 1885–1894, 2017. 3
- [14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009. 6
- [15] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE TPAMI*, 42(2):318–327, 2020. 3
- [16] 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. 6, 7
- [17] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021. 1, 3, 6, 7
- [18] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020. 6
- [19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 6
- [20] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, volume 80, pages 4331–4340, 2018. 3
- [21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2016. 1
- [22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6
- [23] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, pages 11659–11668, 2020. 3
- [24] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 3, 6, 7
- [25] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 2
- [26] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018. 6
- [27] 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. 3
- [28] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface:  $L_2$  hypersphere embedding for face verification. In *ACM MM*, pages 1041–1049, 2017. 1, 4
- [29] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018. 4
- [30] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *CVPR*, pages 943–952, 2021. 3, 6, 7
- [31] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, volume 12359, pages 728–744, 2020. 2, 3, 5
- [32] Xin Wang, Thomas E. Huang, Joseph Gonzalez, Darrell Trevor, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, volume 119, pages 9919–9928, 2020. 3
- [33] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 6, 7

- [34] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *ICLR*, 2021. 6
- [35] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, pages 2361–2370, 2021. 2, 3, 6, 7
- [36] Wanping Zhang, Yongru Chen, Wenming Yang, Guijin Wang, Jing-Hao Xue, and Qingmin Liao. Class-variant margin normalized softmax loss for deep face recognition. *IEEE TNNLS*, 32(10):4742–4747, 2021. 2, 4
- [37] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021. 1
- [38] 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*, pages 3447–3455, 2021. 6
- [39] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, pages 16489–16498, 2021. 3, 6, 7
- [40] 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. 2, 3, 6, 7
- [41] 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. 6