AREA: Adaptive Reweighting via Effective Area for Long-Tailed Classification

Xiaohua Chen[1,2] Yucan Zhou[1*] Dayan Wu[1] Chule Yang[3]
Bo Li[1,2] Qinghua Hu[4] Weiping Wang[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Defense Innovation Institute, Military Academy of Sciences [4]Tianjin University
{chenxiaohua, zhouyucan, wudayan, libo, wangweiping}@iie.ac.cn
yangchule@126.com huqinghua@tju.edu.cn

Abstract

Large-scale data from the real-world usually follow a long-tailed distribution (i.e., a few majority classes occupy plentiful training data, while most minority classes have few samples), making the hyperplanes heavily skewed to the minority classes. Traditionally, reweighting is adopted to make the hyperplanes fairly split the feature space, where the weights are designed according to the number of samples. However, we find that the number of samples in a class can not accurately measure the size of its spanned space, especially for the majority class, where the size of its spanned space is usually larger than the samples' number because of the high diversity. Therefore, weights designed based on the samples' number will still compress the space of minority classes. In this paper, we reconsider reweighting from a totally new perspective of analyzing the spanned space of each class. We argue that, besides statistical numbers, relations between samples are also significant for sufficiently depicting the spanned space. Consequently, we estimate the size of the spanned space for each category, namely effective area, by detailedly analyzing its samples' distribution. By treating samples of a class as identically distributed random variables and analyzing their correlations, a simple and non-parametric formula is derived to estimate the effective area. Then, the weight simply calculated inversely proportional to the effective area of each class is adopted to achieve fairer training. Note that our weights are more flexible as they can be adaptively adjusted along with the optimizing features during training. Experiments on four long-tailed datasets show that the proposed weights outperform the state-of-the-art reweighting methods. Moreover, our method can also achieve better results on statistically balanced CIFAR-10/100. Code is available at https://github.com/xiaohua-chen/AREA.

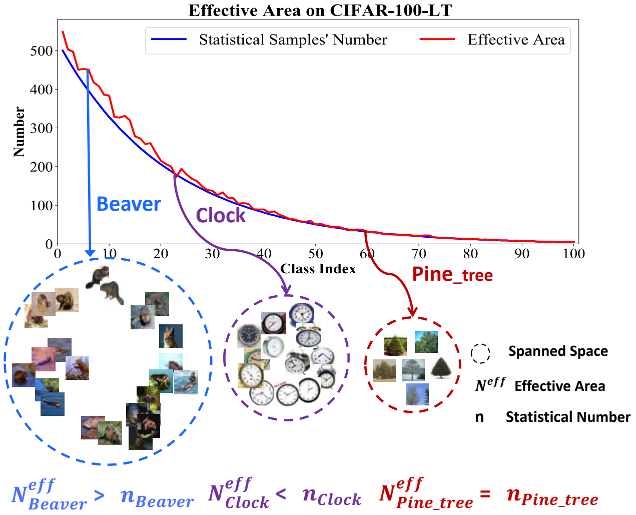

Figure 1. The effective area and statistical number on CIFAR-100-LT under the imbalance ratio of 100. We can see that since samples in "Beaver" are distributed dispersedly with different backgrounds, viewpoints, and poses, their effective area is larger than the samples' number. Whereas, the effective area of "Clock" is smaller than its number of samples, as samples are highly overlapped with limited backgrounds and viewpoints.

1. Introduction

Recently, deep neural networks have achieved great improvements on many tasks [16, 23, 55]. Their dramatic performance significantly depends on high-quality annotated and balanced distributed datasets, such as ImageNet [46] and COCO [29]. However, data are usually long-tailed in the real world, where many minority classes occupy a small number of samples and most samples belong to a few majority classes. Obviously, majority classes will dominate the

*Corresponding author

training, making the hyperplanes heavily skewed to minority classes [43]. Therefore, it is a serious challenge to train a classifier with a severely unbalanced dataset [35, 19, 36, 1].

Rebalancing is an intuitive approach for fair training. To achieve a balanced training set, we can under-sample the majority classes by randomly discarding some samples, or over-sample the minority classes by duplicating their samples [15, 2, 3]. But under-sampling may degrade the performance of majority classes, and over-sampling makes minority classes easily over-fitted [53, 4, 9]. Although data augmentation may intuitively diversify minority classes, the enlarged dataset will tremendously slow down the training procedure [57, 25, 5]. Besides, the reasonability of the augmented data to a target category requires further discussion.

Compared to rebalancing, reweighting is much more simple and lightweight, which assigns higher costs to the loss of minority classes. In reweighting, weights are important for good performance. Thus, complex re-weighting methods are designed with sophisticated hyper-parameters, held-out validation sets, or expensive training procedures [28, 48, 59]. Usually, weights are devised based on the number of samples. However, the spanned space of a category is essentially far more than the statistical number, where the relation of samples is also significant. Class-Balanced Loss [9] is the preliminary attempt that considers the samples' distribution in reweighting. It estimates the size of a category with the assumption that each sample is either entirely inside or outside the set of previous data. However, this strong assumption is difficult to be satisfied, where instances are naturally partially overlapped. For example, the shape of "Apple" is round, but the color of an instance can be red, green, or yellow, which leads to partial overlap (i.e., sharing the same shape but having different colors).

In contrast to existing work, we consider the size of the spanned space of a category by detailedly analyzing the samples' distribution. We call this size the effective area, which can be larger than, smaller than, or equal to the number of samples. If samples are distributed far away from each other, then the effective area should be larger than the statistical number. Inversely, when samples are highly overlapped, the effective area is smaller. Figure 1 shows the effective areas on CIFAR-100-LT under the imbalance ratio of 100. We can see that since samples in "Beaver" are distributed dispersedly with different backgrounds, viewpoints, and poses, the effective area (452.43) of "Beaver" is larger than the samples' number (415). Whereas, the effective area (173.10) of "Clock" is less than the number of samples (179) as samples are highly overlapped because of the limited backgrounds and viewpoints. While the effective area of the "Pine tree" is equal to the statistical number. Therefore, weights designed by inversing the statistical number will still lead to a biased allocation of the feature space. For example, the weight ratio of "Beaver" to the "Pine tree" in the statistical number-based line is (1:12.96). While it can be (1:14.14) under the effective area. However, estimating the effective area is unexplored.

In this paper, we propose a simple, elegant, and non-parametric formula to calculate the effective area for each category. Thus, our method **A**daptive **R**eweighting based on the **E**ffective **A**rea (**AREA**) can achieve better weights for reweighting. It consists of two stages. In the first stage, we train a basic feature extractor with the standard cross-entropy loss. In the second stage, the trained feature extractor is used to extract features for the training data. Then, for a category, the correlation coefficient between any two samples can be calculated, making up its correlation coefficient matrix $\boldsymbol{R}$. Subsequently, we compute its effective area with the derived formula $1/\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{a}$, where $\boldsymbol{a} = \left(\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N}\right) \in \mathbb{R}^{N\times 1}$, N is its statistical number. Finally, the loss for each category is reweighted by the inversely proportional to its effective area. Notably, the weights in AREA can be adaptively updated during training, since $\boldsymbol{R}$ is calculated with the optimizing features. Extensive experiments demonstrate that the proposed AREA can not only greatly improve the performance on long-tailed datasets, but also promote the accuracy on balanced datasets.

Our key contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose using the effective area instead of the statistical sample number for reweighting.
- To quantify the effective area of a category, we derive a simple and non-parametric formula $1/\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{a}$ according to the correlation between all its samples.
- By directly assigning the inversely proportional effective area as the weight for each category, the simply improved cross-entropy loss can substantially boost the performance of long-tailed classification.
- The proposed AREA outperforms current state-of-the-art reweighting methods on four long-tailed datasets. In addition, it also achieves good improvements on statistically balanced CIFAR-10/100.

2. Related Work

2.1. Resampling and Data Augmentation

A commonly used strategy to deal with the long-tailed problem is rebalancing, including under-sampling the majority categories [20, 2] and over-sampling the minority categories [47, 2, 3]. However, under-sampling may damage feature representation by discarding important majority data [53, 4, 9], while over-sampling may cause over-fitting by duplicating minority samples.

To alleviate over-fitting, data augmentation is introduced to enhance the intra-class diversity for minority categories.

A direct way to synthesize new instances is by combining several samples from a target class [63, 13]. Similarly, mixup [57] creates additional samples with a convex combination of two randomly selected images. While, generative adversarial networks [42, 6] can generate new samples with the generator. MBJ [30] accumulates jitters to enhance the diversity of the minority categories, while CMO [39] uses the rich context of the majority categories. Nevertheless, with the increasing number of augmented samples, the training speed is sharply slowed down, especially for large-scale datasets. Recently, although MetaSAug [25] and RISDA [5] have been proposed to do implicit augmentation, the rationality of the augmented data remains an open issue.

2.2. Instance-level and Class-level Reweighting

Another typical strategy is reweighting, which aims to make the loss of minority categories more important, including instance-level and class-level reweighting. Instance-level reweighting assigns different weights to each sample. It either requires sophisticated hyper-parameters (e.g., Focal loss[28], IB[41], and FSR[59]), or a held-out validation set (e.g., L2RW [44] and Meta-Weight-Net [48]).

Differently, class-level methods are much easy and lightweight, which assign different weights for each class to make the hyperplane fairly split the feature space. An intuitive method is to reweight each class inversely proportional to its statistical number [18, 52]. However, this strategy performs worse [38, 34] on large-scale long-tailed datasets since the spanned space of a category is never equal to its sample number. Therefore, some researchers try to find complicated relations between weight and number empirically. LDAM [4] introduces a label-distribution-aware marginal loss based on the statistical numbers to extend the decision boundary for tail categories. Difficulty-Net [49] innovatively proposes using the performance of the model to predict class-difficulty scores and then dynamically assigns weights based on the scores instead of statistical numbers.

However, the essential reason for the unsatisfactory performance is that the spanned space of a category is far more than its statistical number, where the samples' distribution is also significant. Class-Balance Loss [9] is proposed to emphasize the effective number of samples based on a strong assumption that each sample is either entirely inside the set of previous data or entirely outside. However, samples from the same category are naturally partially overlapped. Therefore, a more effective way to estimate the size of spanned space for a category by considering its samples' exhaustive distribution is urgently required.

3. The Adaptive Effective Area

Before diving into the mathematical formulations, we first give the definition of effective area.

Definition (**Effective Area**). For a specific category, the effective area is the size of its spanned space.

Given a long-tail distributed dataset $\{x_i, y_i\}_{i=1}^N$, where $y_i \in L = \{l_1, l_2, ..., l_C\}$, C is the number of categories. Our method contains two stages. In stage-I, we train a basic classifier with the standard cross-entropy loss. For a sample x_i, we extract its feature $\boldsymbol{f}_i \in \mathbb{R}^{1\times d}$ with $F(x_i, \Theta_f)$, where Θ_f are parameters of the feature extractor. H is the classifier and Θ_h are its parameters. During stage-II, we use the effective area to design the weights of each class to achieve fair training. In the latent space, we estimate the effective area of a category by considering the distribution of all its samples. Specifically, for a category l_c, by treating its samples as identically distributed random variables, we analyze the variance of its prototype based on sufficient analysis of the correlation between all its samples. Then, with the assumption of rendering the spanned space of category l_c with virtual independent identically distributed samples, we can obtain a compact variance. Finally, by comparing these two variances, a simple and non-parametric formula of the effective area is derived.

3.1. Variance of the Prototype

To measure the effective area, for a category l_c, we should consider the distribution of all its N_c instances. Ideally, we can collect numerous different instance sets to represent l_c, where each set contains N_c instances belonging to l_c. In this case, instance in the i_{th} position is a random variable sampled from the distribution of l_c. Then, its feature $\boldsymbol{f}_i$ is also a random variable. When different N_c instances are collected, its effective area in the feature space will vary, and so is its prototype $\boldsymbol{\mu}_c$. Since the effective area has never been explored, while $\boldsymbol{\mu}_c$ is maturely defined. Inspired by the relation of effective sample size and variance of the prototype in statistics [45], we use $Var(\boldsymbol{\mu}_c)$ as a bridge to quantify the N^{eff}. Consequently, we can estimate the effective area by analyzing the variance of $\boldsymbol{\mu}_c$. Generally, $\boldsymbol{\mu}_c$ can be estimated with $\frac{1}{N_c}\boldsymbol{f}_1 + \frac{1}{N_c}\boldsymbol{f}_2 + \cdots + \frac{1}{N_c}\boldsymbol{f}_{N_c}$. If the N_c samples are independent and identically distributed, then, the variance of their prototype $\boldsymbol{\mu}_c$ is:

$$\begin{aligned} Var\left(\boldsymbol{\mu}_c\right) &= Var\left(\frac{1}{N_c}\boldsymbol{f}_1 + \frac{1}{N_c}\boldsymbol{f}_2 + \cdots + \frac{1}{N_c}\boldsymbol{f}_{N_c}\right) \\ &= \frac{1}{N_c^2}\left(Var\left(\boldsymbol{f}_1\right) + Var\left(\boldsymbol{f}_2\right) + \cdots + Var\left(\boldsymbol{f}_{N_c}\right)\right) \\ &= \frac{1}{N_c^2}\sum_{i=1}^{N_c}\sigma_{\boldsymbol{f}_i}^2. \end{aligned} \tag{1}$$

Since each $\boldsymbol{f}_i$ is a random variable following the distribution of category l_c, each $\sigma_{\boldsymbol{f}_i}$ is equal to σ_c, where $\sigma_c^2 = E\left[\left(\boldsymbol{f}_i - \boldsymbol{\mu}_c\right)\left(\boldsymbol{f}_i - \boldsymbol{\mu}_c\right)^{\mathrm{T}}\right]$ denotes the variance of category

l_c. Therefore, we can rewrite Eq.(1) as:

$$Var\left(\boldsymbol{\mu}_c\right) = \frac{1}{N_c^2} \cdot \left(N_c \sigma_c^2\right) = \frac{\sigma_c^2}{N_c}. \tag{2}$$

However, samples from the same class in the real world are essentially correlated to each other (i.e., non-independent and identically distributed). So the variance $\hat{Var}\left(\boldsymbol{\mu}_c\right)$ is:

$$\begin{aligned} \hat{Var}\left(\boldsymbol{\mu}_c\right) &= \hat{Var}\left(\frac{1}{N_c}\boldsymbol{f}_1 + \frac{1}{N_c}\boldsymbol{f}_2 + \cdots + \frac{1}{N_c}\boldsymbol{f}_{N_c}\right) \\ &= \frac{1}{N_c^2}\sum_{i}^{N_c} \sigma_{\boldsymbol{f}_i}^2 + \frac{1}{N_c^2}\sum_{i,j=1,i\neq j}^{N_c} Cov(\boldsymbol{f}_i, \boldsymbol{f}_j) \\ &= \frac{1}{N_c^2}\left(N_c\sigma_c^2 + \sum_{i,j=1,i\neq j}^{N_c} Cov(\boldsymbol{f}_i, \boldsymbol{f}_j)\right) \\ &= \boldsymbol{a}_c^{\mathrm{T}} \cdot \boldsymbol{M}_c \cdot \boldsymbol{a}_c, \end{aligned} \tag{3}$$

where $\boldsymbol{a}_c = \left(\frac{1}{N_c}, \frac{1}{N_c}, ..., \frac{1}{N_c}\right) \in \mathbb{R}^{N_c \times 1}$, and $\boldsymbol{M}_c$ is the covariance matrix of all the samples in class l_c. The covariance between two random variables can be obtained by:

$$Cov\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) \quad = E\left[\left(\boldsymbol{f}_i - \boldsymbol{\mu}_c\right)\left(\boldsymbol{f}_j - \boldsymbol{\mu}_c\right)^{\mathrm{T}}\right]. \tag{4}$$

Next, we will review the Pearson correlation coefficient to reflect the linear correlation between two random variables $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, which is as follows:

$$\rho_{\boldsymbol{f}_i, \boldsymbol{f}_j} = \frac{Cov\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right)}{\sigma_{\boldsymbol{f}_i}\sigma_{\boldsymbol{f}_j}} = \frac{Cov\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right)}{\sigma_c^2}. \tag{5}$$

With Eq.(5), we can rewrite $\boldsymbol{M}_c^{ij}$ as:

$$\boldsymbol{M}_c^{ij} = Cov(\boldsymbol{f}_i, \boldsymbol{f}_j) = \rho_{\boldsymbol{f}_i, \boldsymbol{f}_j} \times \sigma_c^2. \tag{6}$$

We use $\boldsymbol{R}_c$ to denote the the correlation matrix of $\{\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_{N_c}\}$. Since the covariance matrix $\boldsymbol{M}_c$ is σ_c^2 times of $\boldsymbol{R}_c$, we can rewrite Eq.(3) as follows:

$$\hat{Var}\left(\boldsymbol{\mu}_c\right) = \sigma_c^2 \times \boldsymbol{a}_c^{\mathrm{T}}\boldsymbol{R}_c\boldsymbol{a}_c. \tag{7}$$

3.2. Effective Area Estimation

Inspired by Eq.(2), we want to obtain a compact form for $\hat{Var}\left(\boldsymbol{\mu}_c\right)$. Following the assumption that each sample has the unit volume of 1 [9], and suppose the effective area is N_c^{eff}, then, we can use N_c^{eff} virtual tangent instances to render the feature space spanned by N_c real correlational samples. Figure 2 (a) and (b) show an intuitive geometric interpretation. In this case, according to Eq.(2), the variance calculated with these virtual independent instances is:

$$\hat{Var}\left(\boldsymbol{\mu}_c\right) = \frac{\sigma_c^2}{N_c^{eff}}. \tag{8}$$

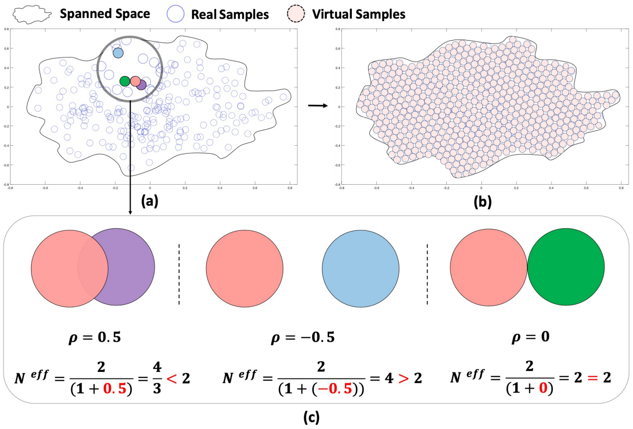
Spanned Space
Real Samples
Virtual Samples
(a)
(b)
$\rho = 0.5$
$\rho = -0.5$
$\rho = 0$
$N^{eff} = \frac{2}{(1+0.5)} = \frac{4}{3} < 2$
$N^{eff} = \frac{2}{(1+(-0.5))} = 4 > 2$
$N^{eff} = \frac{2}{(1+0)} = 2 = 2$
(c)

Figure 2. (a) Samples distribution of category “Chimpanzee” in the metric space with cosine distance. (b) The spanned space of N real correlational samples in category “Chimpanzee” rendered by N^{eff} virtual tangent samples. (c) For two samples, the effective area $1/\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{a}$ degenerates to $2/(1+\rho)$, where ρ is the correlation between two samples. When ρ= { 0.5, -0.5, 0 }, the effective area = { 4/3, 4, 2}, which can be smaller than, larger than, or equal to the statistical number 2.

The assumption of utilizing Eq.(8) to approximate Eq.(3) is reasonable. If N_c samples are entirely uncorrelated, i.e., they are independent and identically distributed, then $N_c^{eff} = N_c$, thus $\hat{Var}\left(\boldsymbol{\mu}_c\right)$ in Eq.(3) and Eq.(8) all degrade into $Var(\boldsymbol{\mu}_c)$ in Eq.(2). When samples are positively correlated, i.e., $Cov(\boldsymbol{f}_i, \boldsymbol{f}_j) > 0$, then $\hat{Var}(\boldsymbol{\mu}_c)$ in Eq .(3) is larger than $Var(\boldsymbol{\mu}_c)$ in Eq.(2). Accordingly, $N_c^{eff} < N_c$, resulting in $\hat{Var}\left(\boldsymbol{\mu}_c\right)$ in Eq.(8) is also larger than $Var\left(\boldsymbol{\mu}_c\right)$ in Eq.(2). Inversely, $\hat{Var}\left(\boldsymbol{\mu}_c\right)$ in both Eq.(3) and Eq.(8) are smaller than $Var\left(\boldsymbol{\mu}_c\right)$ in Eq.(2).

Since Eq.(7) is rewrited from Eq.(3), then, combining Eq.(7) and Eq.(8), we can obtain:

$$\sigma_c^2 \times \boldsymbol{a}_c^{\mathrm{T}}\boldsymbol{R}_c\boldsymbol{a}_c = \frac{\sigma_c^2}{N_c^{eff}}. \tag{9}$$

Therefore, the expected **effective area** for category l_c is:

$$N_c^{eff} = \left\{\frac{1}{\boldsymbol{a}_c^{\mathrm{T}}\boldsymbol{R}_c\boldsymbol{a}_c}, \forall i \in [1, N_c], \boldsymbol{a}_{ci} = 1/N_c\right\}. \tag{10}$$

Our effective area can reflect the distribution of samples. For example, for two samples in a category, Eq.(10) degenerates into $N^{eff} = 2/(1+\rho)$. As shown in Figure 2(c), if the correlation of the two samples is ρ=0.5, then, the effective area is 4/3, smaller than 2. While, when the correlation is ρ=-0.5, the effective area is 4, larger than the number 2. Note that our N_c^{eff} may not be an integer, which makes the N_c^{eff}-based reweighting more flexible than N_c.

4. Training

4.1. Estimation of the Correlation Matrix

According to Eq.(10), we only need to calculate the correlation matrix $\boldsymbol{R}_c$ to obtain the effective area for category l_c. Ideally, if we can sample enough instances for each variable of l_c, the correlation $\rho_{\boldsymbol{f}_i,\boldsymbol{f}_j}$ can be calculated by:

$$\rho_{\boldsymbol{f}_i,\boldsymbol{f}_j} = \frac{E\left[(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}{\sqrt{E\left[(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_i-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}\sqrt{E\left[(\boldsymbol{f}_j-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}}. \tag{11}$$

However, in reality, it is impossible to collect numerous different instance sets to represent l_c, nor can we obtain sufficient instances for the variable in the i_{th} position. Therefore, in the case of limited samples of each variable, the correlation between the variables will change, resulting in an adaptive $\boldsymbol{R}_c$. During implementation, all we can utilize is one training dataset, i.e., only one sample is collected at the corresponding position of each variable. Therefore, we use the dataset at hand to estimate the $\boldsymbol{R}_c$ of this dataset.

We review the feature centralization of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, i.e., $\{\boldsymbol{f}_i-\boldsymbol{\mu}_c\}$ and $\{\boldsymbol{f}_j-\boldsymbol{\mu}_c\}$. Then, their cosine similarity is:

$$\begin{aligned} cos\left(\hat{\theta}\right) &= \frac{(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}}{\sqrt{(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_i-\boldsymbol{\mu}_c)^{\mathrm{T}}}\sqrt{(\boldsymbol{f}_j-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}}} \\ &= \frac{E\left[(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}{\sqrt{E\left[(\boldsymbol{f}_i-\boldsymbol{\mu}_c)(\boldsymbol{f}_i-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}\sqrt{E\left[(\boldsymbol{f}_j-\boldsymbol{\mu}_c)(\boldsymbol{f}_j-\boldsymbol{\mu}_c)^{\mathrm{T}}\right]}}. \end{aligned} \tag{12}$$

According to Eq.(12), we can see that when there only one instance is sampled for each variable, the form of correlation calculation is the same with the cosine similarity. Since it is impossible to obtain the correlation from the training dataset at hand, we have to use the cosine similarity instead, which also relates to the spanned space. So we can calculate the correlation matrix $\boldsymbol{R_c}$ with the cosine similarity between two centralized features. Then, the effective area for category l_c can be obtained with Eq.(10).

4.2. Online Effective Area Approximation

Theoretically, the effective area is calculated with $1/\boldsymbol{a}_c^T\boldsymbol{R}_c\boldsymbol{a}_c$. However, it is intractable to estimate it during implementation in a class-wise manner. There are two main reasons. Firstly, as the head class is usually large-scale, it is storage-consuming to maintain high dimensional features for its numerous samples. Secondly, the batch-wise calculation for the tail class is more reasonable. As head classes dominate the training process, features of tail samples are inaccurate. Then, the class-wise calculation will result in an inappropriately enlarged N^{eff} for the tail. The batch-wise calculation can obtain the sub-optimal N^{eff} which is near the statistical number N for the tail. So it is more reasonable to calculate the N^{eff} batch-by-batch. The simplified batch-wise based effective area calculation can be:

Algorithm 1 Effective area calculation in an epoch

Input: D_{train}, and the learned feature extractor Θ_f;
Output: $\{N_i^{eff}\}_{i=1}^C$;

```
Extract features for all the training samples with $\Theta_f$;
Calculate the prototype for each category;
for $b = 1$: $B$ do
    Sample a batch from $D_{train}$, containing a subset of categories $L_{sub}$.
    for each $l_c \in L_{sub}$ do
        Calculate each term in $\boldsymbol{R}_{cb}$ with Eq.(12);
        Calculate effective area $N_{cb}^{eff}$ with Eq.(13);
    end for
end for
Calculate the effective area for each class with Eq.(14);
```

$$N_{cb}^{eff} = 1/\boldsymbol{a}_{cb}^T\boldsymbol{R}_{cb}\boldsymbol{a}_{cb}, \tag{13}$$

where $\forall i \in [1, N_{cb}]$, $\boldsymbol{a}_{cb}^i = 1/N_{cb}$, N_{cb} is the number of samples from class l_c in batch b. Note that it is probable that a batch only contains 1 tail instance, then we set its batch-wise effective area to 1. Subsequently, we sum the effective area in all batches to obtain the overall effective area:

$$N_c^{eff} = \sum_{b=1}^{B} N_{cb}^{eff}, \tag{14}$$

where B is the number of batches in a training epoch. As we have mentioned above, the batch-wise effective area of minority categories is probably to be 1, so their overall effective area is almost equal to their statistical number N.

4.3. Reweighted Loss Function

When the effective area for each category is obtained, we can achieve fair training with the simple reweighted loss function as follows:

$$\mathcal{L}_{AREA} = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} log\frac{e^{\boldsymbol{z}_i^{y_i}}}{\sum_{j=1}^{C} e^{\boldsymbol{z}_i^j}}, \tag{15}$$

where w_{y_i} is the weight for class l_{y_i}, $\boldsymbol{z}_i = H(\boldsymbol{f}_i, \Theta_h)$ is the prediction score. For a category l_c, we simplify the relation between its weight w_c and the effective area N_c^{eff} as:

$$w_c = \frac{1/N_c^{eff}}{\Sigma_{i=1}^C 1/N_i^{eff}} \times C. \tag{16}$$

Our method contains two stages. In stage-I, we train a basic classifier with the standard cross-entropy loss. In

Datasets	Class Number	λ	Training Set	Min Class Number	Max Class Number	Test Set
CIFAR-10-LT	10	200, 100, 50, 20, 10, 1	11,203 $\sim$ 50,000	25 $\sim$ 500	5,000	10,000
CIFAR-100-LT	100	200, 100, 50, 20, 10, 1	9,502 $\sim$ 50,000	2 $\sim$ 500	500	10,000
ImageNet-LT	1,000	256	115,846	5	1,280	50,000
iNaturalist 2018	8,142	500	437,513	2	1,000	24,426

Table 1. Details of CIFAR-10/100-LT, ImageNet-LT, and iNaturalist 2018.

	CIFAR-10-LT					**CIFAR-100-LT**				
Imbalance ratio λ	200	100	50	20	10	200	100	50	20	10
Cross-Entropy†	65.68	70.36	74.81	82.23	86.39	34.84	38.32	43.85	51.14	55.71
Focal Loss(γ= 0.5)† [28]	64.00	70.23	76.72	82.89	86.81	35.00	38.69	44.12	51.10	55.70
Focal Loss(γ= 1.0)† [28]	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
Focal Loss(γ= 2.0)† [28]	64.88	69.59	76.52	83.23	86.32	34.75	38.39	43.70	51.02	55.00
L2RW‡ [44]	66.25	72.23	76.45	81.35	82.12	33.00	38.90	43.17	50.75	52.12
Meta-Weight-Net [48]	68.91	75.21	80.06	84.94	87.84	37.91	42.09	46.74	54.37	58.46
FSR [59]	67.76	-	79.17	-	87.40	35.44	-	42.57	-	55.45
Class-Balanced Loss [9]	68.89	74.57	79.27	84.36	87.49	36.23	39.60	45.32	52.59	57.99
CE-DRW [4]	-	76.34	79.97	-	87.56	-	41.51	45.29	-	58.12
CE-DRS [4]	-	75.61	79.81	-	87.38	-	41.61	45.48	-	58.11
LDAM [4]	-	73.35	-	-	86.96	-	39.6	-	-	56.91
Meta-Class-Weight ‡[19]	70.66	76.41	80.51	**86.46**	88.85	39.31	43.35	48.53	55.62	59.58
LDAM-DRW [4]	-	77.03	-	-	88.16	-	42.04	-	-	57.99
FSR-DF [59]	66.15	-	79.78	-	88.15	36.74	-	44.43	-	55.60
IB [41]	73.96	78.26	81.70	85.80	88.25	37.31	42.14	46.22	52.63	57.13
AREA	**74.99**	**78.88**	**82.68**	85.99	88.71	**43.85**	**48.83**	**51.77**	**57.02**	**60.77**

Table 2. Accuracy of ResNet-32 on CIFAR-10-LT and CIFAR-100-LT.

stage-II, we update the parameters of the network with the reweighted loss function in Eq.(15). More details about calculating the effective area are shown in Algorithm 1.

5. Discussion

Rationality of the $\rho_{\boldsymbol{f}_i,\boldsymbol{f}_j}$ Estimation: Theoretically, if each sample can be a random variable and their correlation coefficients can be obtained, then, we can calculate the intrinsic effective area for each category (i.e., the spanned size of a category, which can be estimated with sufficient samples). Fortunately, it is unnecessary to estimate the intrinsic effective area in implementation. Since we learn the model with a specific training dataset, all we have to do is to estimate the spanned space of the currently collected samples in each category to achieve fair training with this training set, rather than the original intrinsic effective area of each category. Therefore, obtaining the practical effective area depicted by the training set at hand is enough.

Adaptive Updating of the Weights: For a specific category, features of its samples are optimized in different training epochs. Then, the correlation matrix changes with the optimizing features. Accordingly, the effective area also varies. Therefore, our weights will be adaptively updated.

6. Experiment

Datasets We conduct our experiments on four long-tailed datasets: CIFAR-10/100-LT [9], ImageNet-LT [31], and iNaturalist 2018[1], and two balanced datasets: CIFAR-10/100. We set the imbalance ratio λ[2] as the ratio of the sample sizes between the most and least frequent classes (i.e., $\lambda = N_{max}/N_{min}$). Details are shown in Table 1.
Compared Methods. We compare AREA with the state-of-the-art reweighting methods. **(1) Baseline**: Cross-Entropy. **(2) Instance-level reweighting**: Focal Loss , L2RW, Meta-Weight-Net, FSR. **(3) Class-level reweighting**: Class-Balanced Loss, LDAM, CE-DRW and Meta-class-Weights. **(4) Hybrid methods**, which combine instance-level and class-level reweighting: LDAM-DRW, IB, and FSR-DF.

[1] https://github.com/visipedia/inat_comp

[2] $\lambda = 1$ stands for the balanced CIFAR-10/100.

Note that, the **most relevant** methods are CE-DRW and Class-Balanced Loss. CE-DRW is reweighted by inversing the samples' number N and Class-Balanced Loss is reweighted by the effective number. In the tables below, "†" and "‡" indicate the results reported in [9] and [5]. Methods without "†" and "‡" mean the results from original papers.

6.1. Results on CIFAR-10/100-LT

For CIFAR-10/100-LT, we use ResNet-32 [16] as our backbone, which is trained 200 epochs by standard SGD with a momentum of 0.9 and a weight decay 5×10^{-4}. For the sake of fairness, we follow LDAM [4] using simple data augmentation strategies including RandomCrop and RandomHorizontalFlip without extra augmentations. The initial learning rate is 0.1, and the linear warm-up learning rate schedule [14] is adopted. Besides, we decay the learning rate by 0.01 at the 160^{th} epoch and again at 180^{th} epoch. From the results in Table 2, we can conclude that:

Firstly, compared with the most relevant method CE-DRW and Class-Balanced Loss, our method outperforms a large margin by {-, 2.54%, 2.71%, -, 1.15%} and {6.1%, 4.31%, 3.41%, 1.63%, 1.22% } on CIFAR-10-LT under different λ, respectively. While the improvements are {-, 7.32%, 6.48%, -, 2.65% } and {7.62%, 9.23%, 6.45%, 4.43%, 2.78%} on CIFAR-100-LT. It shows that considering the correlation between samples is more reasonable.

Secondly, compared with the best reweighting method IB, our AREA improves the accuracy by {1.03%, 0.62%, 0.98%, 0.19%, 0.46% } on CIFAR-10-LT and {6.54%, 6.69%, 5.55%, 4.39%, 3.64%} on CIFAR-100-LT, which illustrates the effectiveness of our method.

Thirdly, the improvements of AREA on CIFAR-100-LT are more obvious than those on CIFAR-10-LT, and so is with a higher λ. It shows that AREA is more suitable to deal with large-scale long-tailed data. The more imbalanced the data, the better improvements our AREA can achieve.

6.2. Results on ImageNet-LT and iNaturalist 2018

For fairness, we use ResNet-50 as the backbone and train the network with batch size 256 for ImageNet-LT and iNaturalist 2018. The linear warm-up learning rate schedule is used for them. Concretely, for ImageNet-LT, we train the model 120 epochs by SGD with a momentum of 0.9 and a weight decay 2×10^{-4}. We set the initial learning rate to 0.1 and decay it by 0.1 at the 60^{th} and 80^{th} epoch. From the results in Table 3, we can see that the accuracy of AREA is improved to 49.53%, which is the best among all the reweighting methods. Besides, it is improved by 8.68% compared with the Class-Balanced Loss, which shows that AREA can obtain a much more unbiased classifier by comprehensively considering the correlation between samples.

Method	**ImageNet-LT**
Cross-Entropy‡	38.88
Focal Loss‡[28]	30.50
Class-Balanced Loss‡[9]	40.85
OLTR‡ [32]	40.36
LDAM‡ [4]	41.86
LDAM-DRW [4]	45.74
Meta-Class-Weight‡ [19]	44.92
NCM [22]	44.30
Decoupling [22]	47.30
AREA	**49.53**

Table 3. Results on ImageNet-LT.

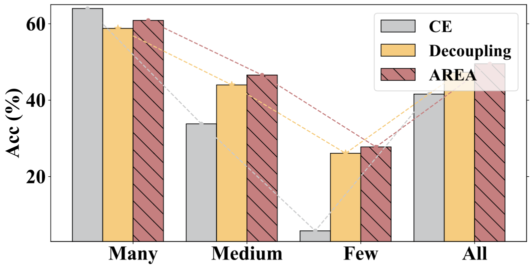

Figure 3. Detailed results on ImageNet-LT.

Method	**iNaturalist 2018**
Cross-Entropy‡	57.30
Focal Loss‡[28]	58.03
Class-Balanced Loss‡[9]	61.12
CE-DRW [4]	63.73
CE-DRS‡ [4]	63.56
LDAM‡ [4]	64.58
LDAM-DRW [4]	68.00
NCM [22]	63.10
Decoupling [22]	67.60
FSR [59]	65.52
IB [41]	65.39
AREA	**68.36**

Table 4. Results on iNaturalist 2018.

To detailly analyze the results, we further report the accuracy of three groups of categories that contain varied numbers of training data on ImageNet-LT following the research in [31]: Many-shot (>100), Medium-shot (20∼100), and Few-shot (<20). As shown in Figure 3, our AREA can achieve good improvements {2.09%, 2.58%, 1.66%} on all three groups compared with Decoupling, which is also the two-stage methods. Besides, the AREA can outperform the CE by a large margin {12.78%, 21.96%} on Medium and Few, while with a little drop of 3.11% on Many.

For iNaturalist 2018, we train the model by SGD with a momentum of 0.9 and a weight decay 1×10^{-4}. We set the learning rate initialized to 0.05 and decay it by 0.1 at

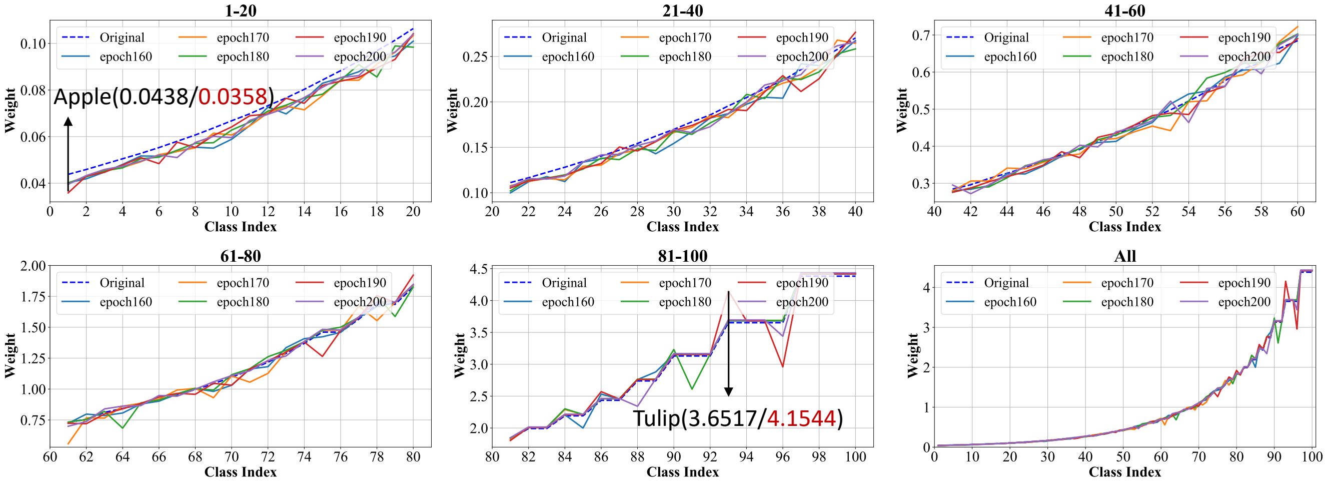

Figure 4. N-based weights vs N^{eff}-based weights on CIFAR-100-LT with λ=100. The “Original” means N-based weight.

CIFAR100	200	100	50	20	10
AREA-fixed weights	42.23	47.73	50.67	56.58	60.21
AREA	**43.85**	**48.83**	**51.77**	**57.02**	**60.77**

Table 5. Adaptively updated vs fixed weights after stage-I.

Method	**CIFAR-10**	**CIFAR-100**
Cross-Entropy	92.61	68.80
AREA	**92.84** (+0.23)	**69.70** (+0.90)

Table 6. Results on balanced CIFAR-10/100.

160^{th} and 180^{th} epoch. As shown in Table 4, our AREA achieves the best performance of 68.36%. Compared with the most relevant Class-Balanced Loss and CE-DRW, the improvements are 7.24% and 4.63%, respectively. It shows that AREA can better describe the size of categories in the feature space compared with the statistical number N.

6.3. N-Based Weight vs N^{eff}-Based Weight

To explicitly exhibit the difference between the N-based weight and the N^{eff}-based weight, we visualize them on CIFAR-100-LT with $\lambda = 100$. The “Original” in figures means N-based weight. According to the results shown in Figure 4, we have the following observations:

Firstly, most N^{eff}-based weights are not equal to the corresponding N-based weights, making up a large overall gap. For example, the N^{eff}-based weight of “Tulip” is 4.1544, which is larger than the N-based weight 3.6517. For “Apple”, the N^{eff}-based weight is 0.0358, which is smaller than the N-based weight 0.0438.

Secondly, our N^{eff}-based weights can be adaptively adjusted in different training epochs. This is because as the training procedure proceeded, the features are optimized. Correspondingly, the correlations of samples are updated, resulting in adaptive effective areas in different epochs. Although the N^{eff}-based weights in different epochs look similar in Figure 4, a subtle gap will generate a large weight ratio change. For instance, the N^{eff}-based weight ratio of “Tulip/Apple” in epoch 160 is 91.75 (3.67/0.04) and 115.39 (4.154/0.036) in epoch 190. To show the effectiveness of adaptation, we conduct an ablation study in Table 5.

Thirdly, to explicitly explain the effectiveness of AREA compared with the most relevant CE-DRW, we further analyze the weight ratio of the N-based and N^{eff}-based. For example, the weight ratio of “Tulip/Apple” in the N-based line is 83.3721(3.6517/0.0438). However, it can be 115.9137(4.1544/0.0358) under N^{eff}-based, which makes the classifier give the tail category “Tulip” more attention. So this improved weight ratio can alleviate the skewness of the hyperplane to minority categories.

Furthermore, we also visualize the effective area on CIFAR-100-LT with different λ in the Appendix. It shows that the N^{eff} of most majority categories are much larger than the statistical number N because of the high diversity. More analysis and visualization of the N^{eff} and weights can be found in the Appendix.

6.4. Results on Balanced CIFAR-10/100

Inevitably, statistically balanced datasets may be imbalanced in the feature space. Therefore, we conduct experiments on balanced CIFAR-10/100. The results are shown in Table 6, we can see that even for the balanced CIFAR-10/100, our method can achieve improvements of 0.23%, and 0.90% compared with cross-entropy loss, respectively.

Figure 5 depicts the difference between the N^{eff}-based and N-based weights on balanced CIFAR-100. To show it clearly, we reorder the classes according to the N^{eff}-based weights. We can see that although the N-based weights are all 1 on the balanced dataset, the N^{eff}-based weights vary

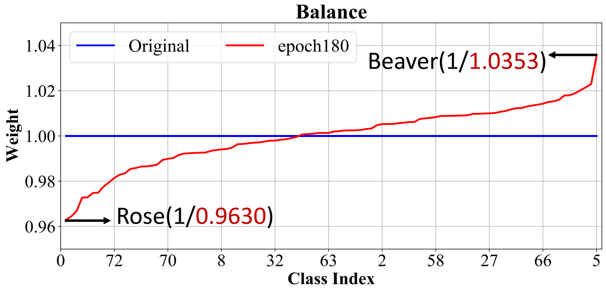

Figure 5. N-based vs N^{eff}-based weights.

because of the different sample distributions. For example, the N^{eff}-based weight of “Beaver” (1.0353) is larger than its N-based one, while the N^{eff}-based of “Rose” (0.9630) is smaller than N-based due to the high diversity of its samples. Details of the experimental settings and analysis of N^{eff} on balanced CIFAR are provided in the Appendix.

6.5. Combining with SOTA

Our simple and non-parametric AREA is orthogonal to the SOTA methods based on hybrid rebalancing methods [61, 58, 37, 17, 49], data augmentations [7, 10, 40], ensemble [51, 24] and supervised contrastive learning [54, 8, 50, 62, 26]. Although the performance of AREA is not so good as those methods, which benefit from complex, multiple models and strong augmentations, our AREA can be effectively and flexibly plugged into them to achieve further improvements. We combine it with the current SOTA methods, i.e., two supervised contrastive methods PaCo [8] and BCL[62], one ensemble method NCL[24] and one data augmentation method GLMC [10]. They are weighted based on the samples’ number N and we use effective area N^{eff} to replace N. The results in Table 7 show that combining AREA with SOTA methods can achieve better performance.

7. Conclusion and Limitations

In this paper, we have derived a simple and non-parametric formula to estimate the size of the spanned space (i.e., the effective area) for a category based on the statistical number and the relations of its samples. By simply assigning the inverse of the effective area for reweighting, our method has achieved state-of-the-art performance on four long-tailed datasets compared with other reweighting methods. Furthermore, it has also shown a good improvement on balanced datasets, indicating that the statistically balanced dataset may be imbalanced in the spanned space.

Nonetheless, AREA has some limitations. Firstly, it is calculated based on high-level representations, so better representations can further promote the accuracy of effective area estimation. Secondly, the AREA is designed for the supervised scenario. Therefore, how to apply it in unsupervised and semi-supervised learning needs to be further explored. Despite these limitations, we believe that AREA moves an important step forward to estimate the size of a category considering its sample distributions, and will inspire more talented and interesting work. In the future, we will work on better representation learning for long-tailed distribution and consider the possibility to extend AREA in unsupervised and semi-supervised learning.

CIFAR-100-LT	Reference	100	50	10
Hybrid rebalancing				
BBN [61]	CVPR20	42.56	47.02	59.12
Difficulty-Net+cRT [49]	WACV23	45.41	50.50	60.86
MiSLAS [60]	CVPR21	47.00	52.30	63.20
Weight Balancing [1]	CVPR22	53.35	57.71	68.67
GML-CE [12]	CVPR23	41.06	-	-
SuperDisco [11]	CVPR23	50.90	57.20	65.90
CR-CE [33]	CVPR23	40.50	45.10	57.40
Augmentation				
FSA [7]	ECCV20	48.51	52.17	65.29
RISDA [5]	AAAI22	50.16	53.84	62.38
OPeN [56]	ICML22	51.50	56.30	-
CMO [40]	CVPR22	47.20	51.70	58.40
GLMC [10]	CVPR23	57.11	62.32	72.33
Contrastive learning				
SSD [27]	ICCV21	46.00	50.50	62.30
PaCo [8]	ICCV21	52.00	56.00	64.20
TSC [26]	CVPR22	43.80	47.40	59.00
BCL [62]	CVPR22	51.93	56.59	64.87
Ensemble				
RIDE (3 experts)* [51]	ICLR21	48.60	51.40	59.80
NCL (ensemble) [24]	CVPR22	54.20	58.20	65.55
SHIKE(3E) [21]	CVPR23	56.30	59.80	-
AREA	in this paper	48.83	51.77	60.77
PaCo[8]+**AREA**		52.37	56.58	65.13
BCL[62]+**AREA**		52.03	56.67	65.01
NCL[24] +**AREA**		55.17	58.68	65.81
GLMC [10] +**AREA**		**57.95**	**61.08**	**73.20**

Table 7. Combining AREA with current SOTA methods.

8. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62006221, 62106258, 62006242, 62202459 and 61925602, the National Key R&D Program of China under Grant 2022YFB3103500, the Grant No.XDC02050200, the China Postdoctoral Science Foundation under Grant 2022M713348 and 2022TQ0363, and Young Elite Scientists Sponsorship Program by BAST (NO.BYESS2023304).

References

[1] Shaden Alshammari, Yuxiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *CVPR*, pages 6887–6897, 2022. 2, 9

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

[3] Jonathon Byrd and Zachary Chase Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881, 2019. 2

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pages 1565–1576, 2019. 2, 3, 6, 7

[5] Xiaohua Chen, Yucan Zhou, Dayan Wu, Wanqian Zhang, Yu Zhou, Bo Li, and Weiping Wang. Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification. In *AAAI*, pages 356–364, 2022. 2, 3, 7, 9

[6] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 3

[7] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *ECCV*, pages 694–710, 2020. 9

[8] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *ICCV*, pages 695–704, 2021. 9

[9] 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, 4, 6, 7

[10] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *CVPR*, 2023. 9

[11] Ying-Jun Du, Jiayi Shen, Xiantong Zhen, and Cees G. M. Snoek. Superdisco: Super-class discovery improves visual recognition for the long-tail. In *CVPR*, 2023. 9

[12] Yingxiao Du and Jianxin Wu. No one left behind: Improving the worst categories in long-tailed learning. 2023. 9

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018. 3

[14] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 7

[15] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 7

[17] 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. 9

[18] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016. 3

[19] 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 *CVPR*, pages 7610–7619, 2020. 2, 6, 7

[20] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002. 2

[21] Yan Jin, Mengke Li, Yang Lu, Yiu-ming Cheung, and Hanzi Wang. Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation. In *CVPR*, 2023. 9

[22] 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. 7

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012. 1

[24] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *CVPR*, pages 6949–6958, 2022. 9

[25] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *CVPR*, pages 5212–5221, 2021. 2, 3

[26] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogério Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *CVPR*, pages 6908–6918, 2022. 9

[27] Tianhao Li, Limin Wang, and Gangshan Wu. Self supervision to distillation for long-tailed visual recognition. In *ICCV*, pages 610–619, 2021. 9

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2, 3, 6, 7

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[30] Jialun Liu, Wenhui Li, and Yifan Sun. Memory-based jitter: Improving visual recognition on long-tailed data with diversity in memory. In *AAAI*, pages 1720–1728, 2022. 3

[31] 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

[32] 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. 7

[33] Yanbiao Ma, Licheng Jiao, Fang Liu, Shuyuan Yang, Xu Liu, and Lingling Li. Curvature-balanced feature manifold learning for long-tailed classification. In *CVPR*, 2023. 9

[34] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, pages 185–201, 2018. 3

[35] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, pages 89–96, 2011. 2

[36] Yang Lu Mengke Li, Yiu-ming Cheung. Long-tailed visual recognition via gaussian clouded logit adjustment. In *CVPR*, pages 6929–6938, 2022. 2

[37] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR 2021*, 2021. 9

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 3

[39] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *CVPR*, pages 6877–6886, 2022. 3

[40] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *CVPR*, pages 6877–6886, 2022. 9

[41] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *ICCV*, pages 735–744, 2021. 3, 6, 7

[42] Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *NeurIPS*, pages 3236–3246, 2017. 3

[43] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020. 2

[44] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340, 2018. 3, 6

[45] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999. 3

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 1

[47] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, pages 467–482, 2016. 2

[48] 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 *NeurIPS*, pages 1917–1928, 2019. 2, 3, 6

[49] Saptarshi Sinha and Hiroki Ohashi. Difficulty-net: Learning to predict difficulty for long-tailed recognition. In *WACV*, pages 6433–6442, 2023. 3, 9

[50] 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. 9

[51] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X. Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *ICLR*, 2021. 9

[52] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, pages 7032–7042, 2017. 3

[53] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, pages 162–178, 2020. 2

[54] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020. 9

[55] Zexian Yang, Dayan Wu, Wanqian Zhang, Bo Li, and Weiping Wang. Handling label uncertainty for camera incremental person re-identification. In *ACM MM*, 2023. 1

[56] Shiran Zada, Itay Benou, and Michal Irani. Pure noise to the rescue of insufficient data: Improving imbalanced classification by training on random noise images. In *ICML*, pages 25817–25833, 2022. 9

[57] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3

[58] 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. 9

[59] Zizhao Zhang and Tomas Pfister. Learning fast sample reweighting without reward data. In *ICCV*, pages 705–714, 2021. 2, 3, 6, 7

[60] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, pages 16489–16498, 2021. 9

[61] 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 9716–9725, 2020. 9

[62] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *CVPR*, pages 6898–6907, 2022. 9

[63] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018. 3