Reviving Undersampling for Long-tailed Learning

Hao Yu[a,b], Yingxiao Du*[a,b], Jianxin Wu†[a,b]

[a] *National Key Laboratory for Novel Software Technology, Nanjing University, China*
[b] *School of Artificial Intelligence, Nanjing University, China*

Abstract

The training datasets used in long-tailed recognition are extremely unbalanced, resulting in significant variation in per-class accuracy across categories. Prior works mostly used average accuracy to evaluate their algorithms, which easily ignores those worst-performing categories. In this paper, we aim to enhance the accuracy of the worst-performing categories and utilize the harmonic mean and geometric mean to assess the model's performance. We revive the balanced undersampling idea to achieve this goal. In few-shot learning, balanced subsets are few-shot and will surely underfit, hence it is not used in modern long-tailed learning. But, we find that it produces a more equitable distribution of accuracy across categories with much higher harmonic and geometric mean accuracy, but with lower average accuracy. Moreover, we devise a straightforward model ensemble strategy, which does not result in any additional overhead and achieves improved harmonic and geometric mean while keeping the average accuracy almost intact when compared to state-of-the-art long-tailed learning methods. We validate the effectiveness of our approach on widely utilized benchmark datasets for long-tailed learning. Our code is at https://github.com/yuhao318/BTM/.

Keywords: Image Classification, Long-tailed Learning, Undersampling

*Equal contribution.

†Corresponding author, email address: `wujx2001@gmail.com` (Jianxin Wu). This work was partly supported by the National Natural Science Foundation of China under Grant 62276123.

1. Introduction

With the blessing of many balanced large-scale high-quality datasets, such as ImageNet [1] and Places [2], deep neural networks have made significant breakthroughs in many computer vision tasks. These large-scale datasets are balanced, i.e., the number of samples in each class will be close to each other. However, in many practical applications, the data tend to follow a long-tailed distribution, that is, the number of training images in each category is severely imbalanced. To solve the long-tailed recognition problem, researchers have proposed many long-tailed recognition algorithms and achieved high average accuracy on many long-tailed datasets.

Previous long-tailed classification algorithms tend to manually split all classes into "few", "medium", and "many" subsets based on the number of training samples in each class, and the accuracy within each subset is usually reported along with the overall test set accuracy. However, focusing on average accuracy alone is too crude, as some worst-performing classes have zero accuracies and are overshadowed by other classes. Furthermore, classes in the "few" subset do not necessarily perform worse than those in the "medium" or "many" subsets [3]. Although the average accuracy is widely used in long-tailed classification as an optimization target, the industrial community considers the accuracies of those worst categories more critical. Therefore, it is not enough to focus on improving the average accuracy—worst-performing categories need more attention. Because the harmonic and geometric mean of per-class accuracy are more sensitive to the worst categories, GML [3] applies these metrics to measure the performance of the worst categories. Since the harmonic mean is numerically unstable to be optimized, GML chooses to maximize the geometric mean of per-class recall.

In this paper, we believe that compared to the geometric mean, the harmonic mean can better reflect the performance of the worst categories. To help the worst-performing categories, we argue that we need to *revive undersampling*: using few-shot *balanced* subsets to train models for long-tailed learning. Balanced undersampling has never been popular or even used practically in long-tailed learning, because it obviously will cause severe underfitting. But, we find that on top of a regularly learned backbone network, *fine-tuning on a few-shot balanced subset can (surprisingly) improve the harmonic and geometric mean greatly*, while only slightly decreasing the average accuracy. Our next surprising finding is that we can ensemble several models fine-tuned on multiple balanced few-shot datasets by directly averaging the model weights. This model averaging not only improves harmonic and geometric mean, but also adds no extra inference cost because its final model is a single network instead of many networks. In addition, our training strategy can also slightly increase the accuracy of the "few" classes in general, which further demonstrates the effectiveness of our approach. We name our plug-and-play and efficient training strategy as Balanced Training and Merging (BTM). In particular, our contributions are as follows:

- We discover that balanced training drives the model to produce a more uniform recall distribution across categories, and averaging the fine-tuned models can further improve the harmonic and geometric mean.
- Based on our observations, we propose a novel plug-and-play training strategy, i.e., Balanced Training and Merging (BTM). With only a small number of samples and a little additional training overhead, BTM can significantly improve the worst-performing categories with no additional inference overhead.
- Our BTM is easy-to-implement, light-weight, and can be integrated with other long-tailed classification algorithms easily. We conduct abundant experiments to demonstrate its effectiveness.

2. Related work

In this section, we review long-tailed learning methods.

2.1. Re-sampling and re-weighting methods.

Re-sampling methods either over-sample minority categories [4] or under-sample majority categories [5]. Re-weighting methods [6], on the other hand, assign different weights to each category when defining the loss function. CMO [7] pastes an image from a minority class onto rich-context images from a majority class to over-sample the tailed classes. Zhang et al. [8] introduce representative feature extraction and effective sample modeling to mitigate the prior and representation gaps. WGCC [9] introduces a weight-guided class complementing framework to mitigate the gradient shift issue caused by un-sampled classes in long-tailed scenarios. DBN-Mix [10] combines two samples generated by a uniform sampler and a re-balanced sampler to augment the training dataset. RML [11] design a re-weighting scheme so that the augmented positive gradients of minority samples will be emphasized. Resampling has the potential risk of either over-fitting or under-fitting, while re-weighting makes the loss function hard to optimize. Our BTM is based on undersampling but *avoids under-fitting.*

2.2. Decoupling methods.

Decoupling methods are based on the observation that over-sampling negatively affects the learned feature representations, but is critical for learning an unbiased linear classifier. cRT [12] first trains a network using a plain cross-entropy loss and then re-trains the classifier using a balanced sampler. MiSLAS [13] further considers model calibration and uses mixup [14] and label-aware smoothing [13] in the first and second stage, respectively. GCL [15] adds different amplitude Gaussian perturbations to each class. PASCL [16] applies asymmetric supervised contrastive learning to encourage the model to distinguish between tail-class in-distribution samples and OOD samples. LPT [17] introduces several trainable prompts into the decoupling training. H2T [18] augments tail classes by grafting diverse semantic information from head classes in the second stage. Our BTM adds an additional plug-and-play balanced training stage to the two-stage approach, but *only requires little computational overhead and incurs no inference overhead.*

2.3. Ensemble methods.

BBN [19] uses two branches that use different sampling strategies during training. RIDE [20] attaches multiple heads to a single network and uses an additional loss function during training to increase the diversity of each head. During inference, special routing rules are applied to select appropriate heads for prediction. Chen et al. [21] transfer knowledge from head classes to get the target probability density of tail classes. SHIKE [22] applies the MoE architecture to fuse depth-wise features. MGKT [23] proposes a multi-scale feature fusion network, which aims to fully mine the rich information of the features. LCReg [24] learns a set of class-agnostic latent features shared by both head and tail classes, and then uses semantic data augmentation on the latent features to implicitly increase the diversity of the training sample. NCL++ [25] enforces consistent predictions among different experts and augmented copies, which reduces the learning uncertainties. Our BTM approach also merges multiple models trained on some randomly sampled few-shot datasets, but we focus on improving the accuracy of those worst-performing categories rather than the overall average accuracy.

2.4. Other methods.

Besides the methods mentioned above, some methods try to use self-supervised learning to tackle the long-tailed recognition problem. For example, PaCo [26] uses a balanced supervised contrastive loss [27]. OTmix [28] proposes an adaptive image-mixing method to incorporate both class-level and sample-level information. However, these works usually use a lot of additional training data. Recently, Du and Wu [3] propose GML to focus more on the worst categories and propose to use the harmonic and geometric mean of per-class accuracy instead of the overall accuracy on the whole test set as an alternative metric. Our work shares the same goal as GML but achieves higher harmonic and geometric mean. Later we will also show that *our BTM can be combined with GML to obtain better results.*

3. Method

We describe our framework in this section, starting by introducing the evaluation metrics we prefer, followed by novel questions and key observations we revealed in two-stage decoupling methods. Based on these observations, we propose our training pipeline, Balanced Training and Merging (BTM), a simple plug-and-play strategy to improve the worst-performing categories.

3.1. Harmonic Mean is the Preferred Evaluation Metric

For long-tailed learning, given a real number p and the per-class accuracy $\{x_1, x_2, \cdots, x_n\}$ on a balanced test set, the generalized mean with exponent p of these accuracies is

$$M_p(x_1, \dots, x_n) = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^p \right)^{1/p}. \tag{1}$$

For instance, when $p = -\infty$, $M_{-\infty}(x_1, \dots, x_n) = \min\{x_1, \dots, x_n\}$ is the minimum of per-class accuracy. When p is -1 and 0, $M_{-1}(x_1, \dots, x_n) = \frac{n}{\frac{1}{x_1} + \cdots + \frac{1}{x_n}}$ and $M_0(x_1, \dots, x_n) = \sqrt[n]{x_1 \cdot \cdots \cdot x_n}$ are the harmonic and geometric mean, respectively. In particular, the arithmetic mean accuracy, $M_1(x_1, \dots, x_n) = \frac{x_1 + \cdots + x_n}{n}$, is frequently-used in long-tailed learning.

Compared to average accuracy, worst-case accuracy may be more important [3]. For example, given the per-class accuracy $\{x_1, x_2\} = \{0.1, 0.9\}$, its arithmetic mean is $\frac{0.1+0.9}{2} = 0.5$. While this result seems well, it has a minimum accuracy of 0.1, which indicates this algorithm is unusable in real-world applications. Compared to the geometric mean $\sqrt{0.1 \times 0.9} = 0.3$, the harmonic mean $\frac{2}{\frac{1}{0.1}+\frac{1}{0.9}} = 0.18$ is more sensitive to the low recall values and has smaller absolute value, which is closer to $M_{-\infty} = 0.1$, the worst-case accuracy we want to maximize.

However, it is hard to optimize the minimum accuracy directly. The harmonic mean is defined using reciprocal, which makes it hard and numerically unstable to be optimized [3]. Note that even 1% improvement in harmonic mean is very difficult, and some state-of-the-art long-tail recognition algorithms have high average accuracy but very low harmonic mean (cf. Table 3 for more details). The previous work GML chooses to maximize the geometric mean over a mini-batch as a surrogate for the harmonic mean accuracy. In this paper, our BTM applies balanced fine-tuning of the pre-trained model, which can help the backbone to obtain a more even feature distribution and be conducive to balancing the accuracy between different classes. Therefore, compared with GML, our BTM is a more direct solution to improve the harmonic mean as well as the geometric mean.

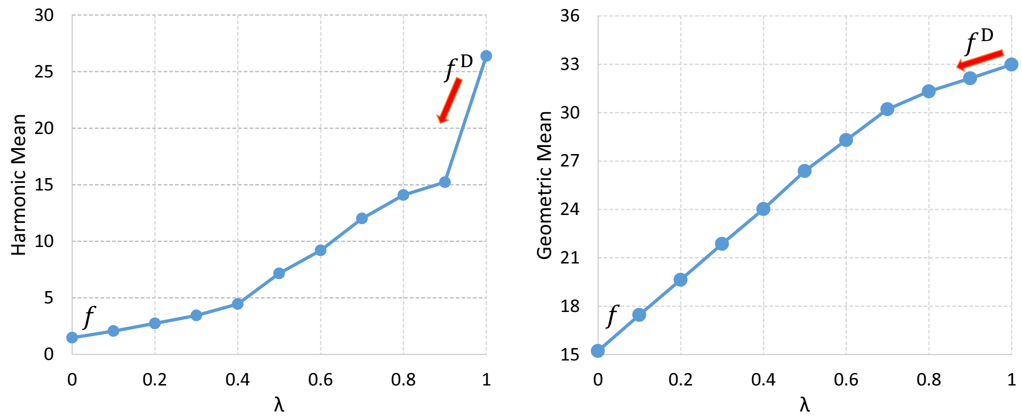
3.2. Can We Revive the Undersampling Strategy?

As is shown in GML, the per-class accuracy of models trained on an imbalanced dataset varies a lot from category to category. There are two recognized reasons for that. First, some categories are essentially more difficult than others. Second, there remains a large difference between the numbers of samples in different categories [5]. The first difficulty stems from the property of each category itself and is hard to handle. In this paper, we focus on the second difficulty and try to deal with the imbalanced data distribution with undersampling technology.

To solve the difficulty induced by imbalance, *the most natural solution is to have a balanced training set*. As oversampling leads to severe overfitting, undersampling seems a better choice. But, in long-tailed datasets, tail categories often have very limited (e.g., 5) training samples. Hence a *balanced under-sampled* dataset will be *few-shot*. Therefore, a key question is:

Question 1. *Can we improve the accuracy of the worst categories with the few-shot balanced dataset?*

We conducted a simple experiment to answer this question. Two-stage decoupling methods train the whole network in the first stage, and then



(a) (b)

Figure 1: a and b present the harmonic and geometric mean of interpolated models between the raw model f $(\lambda = 0)$ and the fine-tuned model f^D $(\lambda = 1)$, respectively.

fine-tune the classifier in the second stage. Here we take the ResNet-50 [29] pre-trained with *first stage* MiSLAS in the Places-LT [2, 30] dataset as the original model f, and randomly sample a 5-shot *balanced* dataset D from the training data. Then we fine-tune f using D for 30 epochs and obtain a model f^D. After fine-tuning, following [31], we *merge* the original and fine-tuned models by linear interpolation. Given $\lambda \in [0, 1]$, the interpolated model is

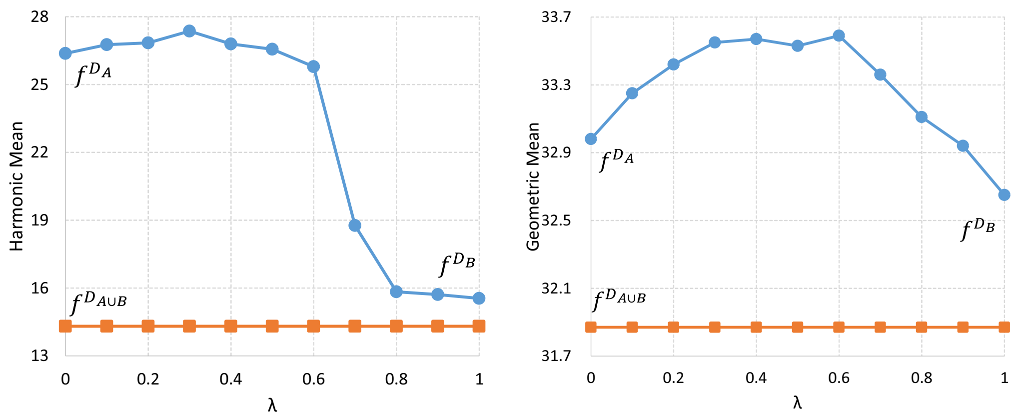
$$f_\lambda^D = \lambda f^D + (1 - \lambda) f \,. \tag{2}$$

Figure 1a and 1b show the harmonic and geometric mean of interpolated models. These curves roughly describe the performance of the worst-performing categories. Those results show that

Observation 1. *For the first-stage pre-trained model in long-tailed learning, fine-tuning with a few-shot balanced dataset can highly improve the accuracy in the worst-performing categories.*

Besides, we also find that with the decrease of λ, the harmonic and geometric mean of the interpolated model f_λ^D show a monotonically decreasing trend, which indicates

Observation 2. *The harmonic and geometric mean of these interpolated models present smooth and monotonic curves.*



(a) (b)

Figure 2: Blue curves in a and b present the harmonic and geometric mean of interpolated models between the fine-tuned model f^{D_A} $(\lambda = 0)$ and the fine-tuned model f^{D_B} $(\lambda = 1)$, respectively. Yellow curves mean the harmonic and geometric mean of $f^{D_{A\cup B}}$.

These two surprising findings answer Question 1: we *can* fine-tune the *whole* network with only *few* training samples. Even if only scarce training data is available, the fine-tuning optimization process hardly suffers from overfitting. That is, if we pay attention to the proper metric, we can *revive* undersampling. Then, a natural question is:

Question 2. *Can we further improve both worst-case and average accuracy with few-shot balanced undersampling?*

Note that Observation 2 gives us an insight into how interpolated models might behave if we have multiple models fine-tuned on *different* balanced datasets. Therefore, we fine-tuned f on different 5-shot balanced datasets D_A and D_B to obtain fine-tuned models f^{D_A} and f^{D_B}, respectively. Then we merge them by linear interpolation, too:

$$f^{D_{A \overset{\lambda}{\leftrightarrow} B}} = (1-\lambda) f^{D_A} + \lambda f^{D_B}. \tag{3}$$

Furthermore, we also merged the balanced data A and B into $A \cup B$, then fine-tuned f using $A \cup B$ to obtain $f^{D_{A\cup B}}$. Note that $D_{A\cup B}$ is *no longer balanced*. Figure 2 presents the experimental results. With different balanced tiny sets, the fine-tuned models f^{D_A} and f^{D_B} have higher harmonic and geometric mean than the original model. We also have the two observation,

Observation 3. *The harmonic and geometric mean of the original model f^{D_A} and f^{D_B} are both higher than the ones of $f^{D_{A\cup B}}$.*

This observation suggests that *balancing is the key to help the worst-performing categories.* $D_{A\cup B}$, despite having more training data, is imbalanced and performs worse than either D_A or D_B. And,

Observation 4. *With appropriate λ (e.g., $\lambda = 0.5$), the interpolated models $f^{D_{A \overset{\lambda}{+} B}}$ have higher harmonic and geometric mean than both f^{D_A} and f^{D_B}.*

This finding indicates that after balanced training, we can continue to *merge the fine-tuned models to achieve higher performance* in the worst-performing categories.

3.3. Balanced Training and Merging

Based on our questions and observations, we propose Balanced Training and Merging (BTM) to revive undersampling for long-tailed learning.

Previous decoupling methods are often divided into two steps: first, train a model using the original long-tailed training set, which does *not* take care of imbalance; second, freeze the backbone network and then only fine-tune the classifier of the model. Note that our previous observations are all for the pre-trained model in the first stage, so we insert our BTM algorithm into the first and second stages. The detailed information is shown in Algorithm 1.

We insert a plug-and-play BTM module between the first and second stages in any existing decoupling methods. Thus it can be widely applied to various decoupling methods. When merging the fine-tuned models, we directly set the weights of each model as $1/N_D$. This is designed to make our approach as flexible and simple as possible. If there already are the pre-trained weights of the first pre-train stage, we can skip the first step.

Our BTM module can be plugged into long-tailed learning methods other than the decoupling ones. However, if the pre-trained model in a long-tailed algorithm has *already considered and handled* the imbalance property, it is not suitable for BTM. We propose a simple modification correspondingly, which converts the 'Pre-train' stage into two stages ('Pre-train' and 'FC'):

- **Pre-train.** Follow the original training strategy *with* imbalance handling to obtain a pre-trained model.
- **FC.** Use the cross-entropy loss to fine-tune *only the classifier* of the model on *all* training data which are imbalanced.

Algorithm 1: The BTM framework.

Input: The whole training set D.

Output: A long-tailed learning model with more balanced precision distribution.

1. **Pre-train.** Follow the original first-stage training strategy and directly train a model *without* handling imbalance.
2. **Dataset sampling.** *Randomly* sample N_D few-shot balanced datasets from the whole training set D, and each balanced dataset contains only N_C samples for each category.
3. **BTM.** Fine-tune the *whole* model (including *both* backbone and classifier) on these N_D few-shot balanced datasets, then merge the fine-tuned N_D models using simple averaging.
4. **Post-train.** Freeze the merged model's backbone, then follow the original second-stage training strategy and only fine-tune the classifier.

Note that other stages remain unchanged and are omitted from the above list. In the 'FC' stage, we only need to fine-tune the classification layer, so our lightweight framework only requires few computational resources. The main reason for this design is to insert our BTM algorithm into existing models with very low cost, since BTM is designed for one-stage models (i.e. classifier does not handle imbalance well). If BTM is used directly on an already trained model, the harmonic and geometric mean increases are not significant (cf. Table 8 for more details). No matter whether imbalance is handled or not in the 'Pre-train' stage, the backbone is always useful in our BTM framework. But, the 'FC' stage needs to prepare an FC that does *not* handle imbalance, which is handled in the next 'BTM' stage.

In summary, compared with previous long-tailed classification algorithms, BTM only adds a balance training and merging step, so our method is simple, plug-and-play and easy to deploy online. BTM training strategy only involves several balanced few-shot datasets, so the training overhead can be *ignored*. Besides, BTM has no effect on the model structure and generates no additional inference overhead. To the best of our knowledge, although balanced undersampling and direct weight fusion have been explored in many machine learning tasks, they have not been successfully utilized in long-tailed learning yet. Our BTM approach is the first attempt to introduce those technologies

Dataset name	# Categories	# Training	# Test	Imbalance Ratio
CIFAR100-LT [32]	100	10,847	10,000	100
ImageNet-LT [1, 30]	1,000	115,846	50,000	256
Places-LT [2, 30]	365	62,500	36,500	996
iNaturalist2018 [33]	8,142	437,513	24,426	500

Table 1: Some statistics of the benchmark datasets used.

for improving the worst-performing categories.

4. Experiments

We conducted extensive experiments in this section. First, we introduce the datasets, evaluation metrics, and implementation details. Then we compare our method with various baseline and state-of-the-art methods. Finally, we will present ablation studies.

4.1. Datasets, Metrics, and Implementation Details

We use widely used long-tailed recognition datasets, i.e., CIFAR100-LT [32], Places-LT [2, 30], ImageNet-LT [1, 30] and iNaturalist2018 [33]. Statistics about them can be found in Table 1. The original CIFAR100 [32], Places [2] and ImageNet [1] are balanced datasets. We follow previous work [30] to construct the long-tailed version by down-sampling the original training set using a Pareto distribution.

Following GML [3], we focus on improving the worst-performing categories in long-tailed recognition. Besides the conventional average accuracy, we compute the accuracy for each category and report their harmonic and geometric mean. These two metrics are more sensitive to small numbers than conventional accuracy, which are believed to better reflect the fairness of a model. Following previous work [30, 13], we use ResNet-32 [29] for CIFAR100-LT, ResNet-152 [29] on Places-LT, ResNeXt-50 [34] or ResNet-50 on ImageNet-LT and ResNet-50 on iNaturalist2018. We choose to apply our method to PaCo/GPaCo [26, 35] and MiSLAS [13], which are two current state-of-the-art one/two stage long-tail recognition methods.

For the two-stage methods like MiSLAS and H2T, we directly use the pre-train weights of the first 'Pre-train' stage. Then we apply our BTM and 'Post-train' stage. We first train our model 30 epochs in the BTM stage.

After that, we follow the original second-stage training strategy and fine-tune the classifier 10 epochs in the 'Post-train' stage. For the fine-tuning process in the BTM stages, we follow the data augmentation strategy of the 'Pre-train' stage, we set batch size as 256 and use SGD optimizer and set the momentum and weight decay as 0.9 and 5×10^{-4}. The initial learning rate is 5×10^{-3}, 5×10^{-3}, 1×10^{-3} and 5×10^{-4} for CIFAR100-LT, Places-LT, ImageNet-LT and iNaturalist2018, respectively. The cosine learning rate schedule and traditional cross-entropy loss are used.

For the other methods like PaCo/GPaCo, OTmix and PASCL, we first re-train the classifier for 10 epochs with all training data in the 'FC' stage. Then we fine-tune the whole model for 30 epochs with the balanced dataset in the BTM stage. In the final 'Post-train' stage, we train the classifier for 40 epochs. We use the same training strategy for the first two steps for simplicity, and the training hyperparameters are same as those of MiSLAS. In the final 'Post-train' stage, we apply the re-weighting and re-sampling training strategy. The loss function is label-aware smoothing loss and we train the classifier with a cosine learning rate schedule.

4.2. Comparison with Other Methods

Now we present the comparison of our methods with various baseline and state-of-the-art methods. In particular, for categories that have zero accuracy, we substitute it with a small number (10^{-3}) otherwise the harmonic and geometric mean will be zero. In these tables, "H-Mean" stands for harmonic mean, "G-Mean" for geometric mean and "L-Recall" for the lowest recall across all categories. "H-Acc.", "M-Acc." and "T-Acc." represent the accuracies in head, middle and tail (i.e., "many" ,"medium" and "few") subsets, respectively. Note that we do not report the lowest recall in both ImageNet-LT and iNaturalist2018 datasets, because their lowest recall is zero across all algorithms. We run our BTM algorithm three times and report the mean and standard deviation of the fine-tuned model.

CIFAR100-LT. Table 2 shows the comparison results on CIFAR100-LT. We apply our method to MiSLAS, OTmix and H2T. Our BTM method improves harmonic and geometric mean by large margins while maintaining overall accuracy. We also list the target objective (worst category's accuracy) in the 'L-Recall' column, where BTM shows clear advantages, too.

Places-LT. Table 3 shows the comparison results on Places-LT. We apply our method to GPaCo and MiSLAS on this dataset. GPaCo is an extension of PaCo, which simplifies some training settings and achieves better results.

Methods	H-Mean	G-Mean	L-Recall	Acc.	H-Acc.	M-Acc.	T-Acc.
MiSLAS [13]	30.9	40.2	5.0	47.0	**61.4**	49.1	26.7
OTmix [28]	17.3	33.9	1.0	46.4	70.2	46.9	16.1
H2T [18]	31.5	41.1	4.0	**47.8**	60.5	**50.5**	**28.8**
MiSLAS + BTM	**36.3**$_{\pm 1.07}$	42.9$_{\pm .31}$	**8.0**$_{\pm .47}$	47.1$_{\pm .22}$	61.0$_{\pm .32}$	48.9$_{\pm .22}$	27.2$_{\pm .45}$
OTmix + BTM	32.1$_{\pm 1.22}$	35.6$_{\pm .71}$	7.0$_{\pm .82}$	46.3$_{\pm .51}$	69.5$_{\pm .62}$	46.6$_{\pm .33}$	16.1$_{\pm .42}$
H2T + BTM	34.2$_{\pm 2.07}$	**43.5**$_{\pm .82}$	8.3$_{\pm .67}$	47.3$_{\pm .31}$	60.1$_{\pm .45}$	50.2$_{\pm .29}$	28.6$_{\pm .33}$

Table 2: Results on the CIFAR100-LT dataset with imbalance ratio 100.

Methods	H-Mean	G-Mean	L-Recall	Acc.	H-Acc.	M-Acc.	T–Acc.
CE	0.7	12.1	0.0	28.7	44.2	26.8	6.7
BSCE [6]	5.6	29.3	0.0	37.2	39.7	38.3	30.1
PaCo [26]	2.5	27.9	0.0	40.5	36.8	46.5	33.2
MiSLAS [13]	28.8	35.3	3.0	40.1	39.3	43.0	35.8
GPaCo [35]	10.9	35.0	0.0	**41.7**	**39.5**	**47.2**	33.0
MiSLAS + BTM	**29.7**$_{\pm .53}$	35.6$_{\pm .12}$	**4.0**$_{\pm .00}$	40.2$_{\pm .16}$	39.2$_{\pm .13}$	43.1$_{\pm .18}$	**36.0**$_{\pm .11}$
GPaCo + BTM	29.4$_{\pm 2.55}$	**35.9**$_{\pm .43}$	2.3$_{\pm .47}$	40.5$_{\pm .21}$	38.4$_{\pm .33}$	46.3$_{\pm .12}$	33.2$_{\pm .23}$

Table 3: Results on the Places-LT dataset.

For example, compared with MiSLAS, GPaCo has a higher accuracy rate and lower harmonic and geometric mean. For MiSLAS, the conventional accuracy even increases along with harmonic and geometric mean. It is worth noting that our method also consistently outperforms the original model on the "few" category in general.

ImageNet-LT. Table 4 shows the results on ImageNet-LT. We improved the harmonic and geometric meanwhile the overall accuracy remained almost unchanged. In particular, we improved the harmonic mean more than the geometric mean. Generally speaking, BTM successfully improves the worst-performing categories and does no harm to the overall accuracy.

iNaturalist2018. Table 5 summarizes the results of the experiments conducted on the iNaturalist2018 dataset. The average accuracy with applying BTM remains almost unchanged. Compared to ImageNet-LT and Places-LT, iNaturalist2018 has a much larger scale. Furthermore, since each category only has three test images, all current methods have a very low harmonic mean of recall on this dataset. Therefore, it is difficult to improve the

Methods	H-Mean	G-Mean	Acc.	H-Acc.	M-Acc.	T-Acc.
CE	1.3	23.3	43.9	65.0	37.1	8.1
BSCE [6]	13.7	42.3	50.5	60.9	48.0	29.8
cRT [12]	13.8	41.4	49.6	59.3	47.1	30.9
DiVE [36]	12.8	45.5	53.6	64.6	50.9	32.0
RIDE [20]	17.3	47.6	55.7	67.4	52.3	34.8
PaCo [26]	21.8	51.3	**58.3**	**66.2**	52.6	55.1
PaCo + BTM	**22.7**$_{\pm.51}$	**51.6**$_{\pm.26}$	58.0$_{\pm.26}$	65.9$_{\pm.17}$	**52.7**$_{\pm.24}$	**55.4**$_{\pm.16}$
MiSLAS [13]	17.9	45.8	**52.7**	**62.7**	50.5	34.7
OTmix [28]	3.4	34.3	49.2	57.5	42.6	47.6
PASCL [16]	5.1	35.7	45.5	51.4	41.4	42.8
MiSLAS + BTM	20.3$_{\pm1.36}$	**46.2**$_{\pm.21}$	52.4$_{\pm.43}$	62.5$_{\pm.32}$	**50.6**$_{\pm.11}$	34.8$_{\pm.06}$
OTmix + BTM	18.2$_{\pm1.33}$	35.5$_{\pm.35}$	48.7$_{\pm.35}$	56.0$_{\pm.32}$	42.4$_{\pm.22}$	**47.8**$_{\pm.28}$
PASCL + BTM	**20.7**$_{\pm.81}$	37.4$_{\pm.55}$	45.2$_{\pm.22}$	50.5$_{\pm.31}$	41.5$_{\pm.28}$	42.9$_{\pm.22}$

Table 4: Results on the ImageNet-LT dataset. Note that MiSLAS, OTmix and PASCL use ResNet-50 instead of ResNeXt-50 as the backbone, so we list them in separate rows.

harmonic and geometric mean on the iNaturalist2018 dataset. Nevertheless, we also obtain 0.3% and 0.6% improvements, respectively. Besides, although all algorithms obtain zero lowest recall values, our improvement in harmonic and geometric mean still demonstrates the effectiveness of BTM.

4.3. Ablation Studies

We conducted several ablation studies. If not otherwise specified, we used the default training setting.

Results of Different Weight Merging Strategies. In the default settings, we set the merging ratio for each model weight to $1/N_D$. In addition to the "Average Merging" strategy, we also explore adaptive fusion ratio strategies. In particular, the fusion coefficients are proportional to each model's harmonic and geometric mean in the balanced training dataset, and the sum of these coefficients is one. We call these strategies "Adaptive Ratio with H&G-Mean". In addition, we also follow the averaging weights strategy of greedy soups [37] and use the harmonic and geometric mean as the criterion. That is, we use one model only when merging the model is better than not merging. We call these methods "Greedy Soup with H&G-Mean". We conduct the experiments on Places-LT with our BTM method applied to

Methods	H-Mean	G-Mean	Acc.	H-Acc.	M-Acc.	T-Acc.
BSCE [6]	1.5	43.9	67.7	68.0	67.5	67.4
BBN [19]	1.5	45.4	69.7	52.8	74.2	68.6
DiVE [36]	1.9	49.6	71.1	70.8	70.2	67.8
MiSLAS [13]	2.0	51.3	**71.6**	**73.2**	**72.4**	70.4
MiSLAS + BTM	**2.3**$_{\pm.07}$	**51.9**$_{\pm.37}$	71.3$_{\pm.14}$	71.1$_{\pm.21}$	72.3$_{\pm.19}$	**70.8**$_{\pm.23}$

Table 5: Results on the iNaturalist2018 dataset.

Methods		H-Mean	G-Mean	L-Recall	Acc.
Average Merging	Merge	26.3	34.3	2.0	39.8
	PT	**29.8**	**35.6**	4.0	40.3
Adaptive Ratio with H-Mean	Merge	26.7	34.4	2.0	39.9
	PT	29.7	35.5	4.0	40.3
Adaptive Ratio with G-Mean	Merge	26.3	34.3	2.0	39.9
	PT	29.8	35.6	4.0	**40.5**
Greedy Soup with H-Mean	Merge	26.8	34.3	2.0	39.8
	PT	29.5	35.4	4.0	40.2
Greedy Soup with G-Mean	Merge	26.8	34.4	2.0	39.9
	PT	29.7	35.6	4.0	40.3

Table 6: Results of different weight merging strategies.

MiSLAS. Table 6 shows the results of merging (refer to "Merge") and post-training (refer to "PT") with different strategies. It can be seen that after merging, these strategies of adaptively adjusting the ratios and models can achieve higher harmonic and geometric mean than direct average merging. But after post-training, these temporary small advantages are quickly wiped out. The simplest average merging strategy achieves the highest harmonic and geometric mean instead. Therefore, BTM directly uses the average merging strategy for flexibility and simplicity.

Effects of the Size of the Sampled Few-Shot Datasets. In the default training settings, we randomly sample 10 few-shot datasets to perform the balanced training. And for each dataset, all categories have 10 training images so the sampled dataset is balanced. In this subsection, we study the effects of the size of the sampled few-shot datasets by varying the number of datasets sampled and the number of training images for each category.

$N_{\mathcal{D}}$	$N_{\mathcal{C}}$	H-Mean	G-Mean	L-Recall	Acc.	H-Acc.	M-Acc.	T-Acc.
2	10	28.4	35.0	2.0	39.8	38.6	42.7	35.8
4	10	29.1	35.1	3.0	40.0	40.0	42.9	35.8
8	10	28.2	34.8	2.0	39.9	38.7	42.8	35.8
20	10	15.2	32.6	0.0	38.3	38.7	41.1	32.6
10	5	28.5	35.1	2.0	40.1	39.1	**43.1**	35.8
10	10	**29.7**	**35.6**	4.0	**40.2**	**39.2**	**43.1**	**36.0**
10	20	29.6	35.3	**5.0**	40.1	**39.2**	43.0	35.8

Table 7: Effects of the size of the sampled datasets. $N_{\mathcal{D}}$ stands for the number of few-shot datasets sampled and $N_{\mathcal{C}}$ stands for the number of training images for each category.

When	Backbone	Classifier	H-Mean	G-Mean	L-Recall	Acc.
Between Stage1&2	✓		29.0	35.2	4.0	40.2
Between Stage1&2		✓	29.4	35.3	4.0	40.1
Between Stage1&2	✓	✓	**29.8**	**35.6**	4.0	**40.3**
After Stage2	✓		29.0	35.4	2.0	40.2
After Stage2		✓	28.5	35.3	2.0	40.2
After Stage2	✓	✓	28.5	35.3	2.0	40.2

Table 8: When and how to perform the balanced training.

The results on Places-LT with MiSLAS are shown in Table 7. As we can see from the table, when we fix $N_{\mathcal{C}} = 10$, the performance can be improved at the beginning when we increase $N_{\mathcal{D}}$ but later drops significantly when we set $N_{\mathcal{D}} = 20$. One possible reason for this phenomenon is that the model is overfitting because we use the same tail-class examples too many times. When $N_{\mathcal{D}} = 20$ the accuracy of the head classes decreases much less than that of the tail classes. On the other hand, when we fix $N_{\mathcal{D}} = 10$ and vary $N_{\mathcal{C}}$, the performance does not change much. Generally speaking, $N_{\mathcal{D}} = 10$ and $N_{\mathcal{C}} = 10$ seem to be a good choice in the Places-LT dataset. For simplicity, we follow this setting across all experiments.

When and How to Perform the Balanced Training. Currently we add the balanced training between the first and second stages of decoupled two-stage methods and we fine-tune the whole model using our sampled few-shot datasets. Here we explore some other possible design choices. Specifically, we try to only fine-tune the backbone or classifier or add the balanced training after the second stage. The results on Places-LT with MiSLAS are

Methods	H-Mean	G-Mean	L-Recall	Acc.
MisLas	30.9	40.2	5.0	47.0
MisLas + GML	36.5	40.9	11.00	46.5
MisLas + BTM	36.3	42.9	8.0	47.1
MisLas + BTM + GML	**36.7**	41.3	8.0	46.9

Table 9: Combining GML with our method in the CIFAR100-LT dataset.

Methods	H-Mean	G-Mean	L-Recall	Acc.
MisLas	28.8	35.3	3.0	40.1
MisLas + GML	28.8	34.9	3.0	39.7
MisLas + BTM	29.7	**35.6**	4.0	**40.2**
MisLas + BTM + GML	**29.9**	35.4	4.0	39.9

Table 10: Combining GML with our method in the Places-LT dataset.

shown in Table 8. As we can see from the table, fine-tuning either the backbone or classifier can improve the harmonic and geometric mean of per-class accuracy, but the final results are inferior to fine-tuning the whole model. Since the scale of our sampled datasets is small, fine-tuning the whole model would not cause much training overhead, we choose to fine-tune the whole model in order to achieve better performance. As for when to perform the balanced training, we can see that adding the balanced training after the second stage achieves inferior performance compared to adding it between the first and second stages.

Combining BTM with GML. GML [3] is the pioneering work in long-tailed recognition that aims at improving the performance of the worst categories. Since their method is also a plug-in, here we try to combine our method with GML. Specifically, in the third stage of our method, we use GML to fine-tune the classifier. Those results are shown in Table 9, 10 and 11. As we can see from those results, although GML can improve the harmonic mean, there is a noticeable drop in accuracy. BTM, on the other hand, does little harm to the overall accuracy. This may be because GML modifies the loss and fully fine-tunes the model with unbalanced samples, forcing a more balanced distribution of the model's accuracy. We used undersampling technology to fine-tune the model, essentially solving the problem of preci-

Methods	H-Mean	G-Mean	Accuracy
PaCo	21.8	51.3	**58.3**
PaCo + GML	31.1	50.8	55.6
PaCo + BTM	22.7	51.6	58.0
PaCo + BTM + GML	**31.3**	**51.4**	56.3

Table 11: Combing GML with our method in the ImageNet-LT dataset.

Methods		H-Mean	G-Mean	L-Recall	Accuracy
Original Model	Stage1	1.47	15.22	0.00	29.62
	Stage2	28.75	35.30	3.00	40.12
Balanced Training	Model1	23.85	32.92	1.00	38.70
	Model2	24.99	33.00	2.00	38.58
	Model3	25.17	33.26	1.00	38.69
	Model4	25.49	33.14	2.00	38.92
	Model5	24.87	32.93	1.00	38.55
	Model6	15.31	33.04	1.00	38.89
	Model7	24.69	32.64	2.00	38.42
	Model8	24.04	32.91	1.00	38.74
	Model9	24.54	33.09	1.00	38.84
	Model10	25.79	33.30	2.00	38.95
Merge		26.34	34.26	2.00	39.84
Post-train		**29.80**	**35.60**	**4.00**	**40.25**

Table 12: Results of single fine-tuned and merged models in Places-LT with MiSLAS.

sion distribution caused by unbalanced datasets. Furthermore, compared to using GML alone, combining our method with GML can further improve the harmonic mean, and our BTM even achieves a higher geometric mean. This is because in the final fine-tuning classifier stage, since the backbone has already been corrected by our BTM algorithm, further use GML to fine-tuning classifier will result in higher harmonic and geometric mean. This proves that our proposed balanced training is indeed very helpful in improving the performance of the worst categories, and we can further combine our BTM with pioneering work to obtain better performances.

Results of Single Fine-tuned Models and the Merged Models. In this section, we report the results of each balanced fine-tuned, and merged

Methods		Harmonic Mean	Geometric Mean	Accuracy
Original Model	Stage1	0.93	21.28	45.51
	Stage2	17.68	45.78	**52.68**
Balanced Training	Model1	2.51	36.53	50.37
	Model2	2.70	36.28	50.33
	Model3	2.70	36.30	50.23
	Model4	2.77	36.24	50.05
	Model5	2.63	36.25	50.29
	Model6	2.78	36.60	50.48
	Model7	2.70	36.38	50.27
	Model8	2.76	36.07	50.16
	Model9	2.63	35.89	49.98
	Model10	2.63	35.98	49.97
Merge		2.81	36.78	50.97
Post-train		**21.11**	**45.85**	52.59

Table 13: Results of single fine-tuned and merged models in ImageNet-LT with MiSLAS.

model during training. For simplicity, we only report the results of MiSLAS and PaCo/GPaCo. In particular, MiSLAS is a two-stage decoupling method. we directly use the first-stage pre-training model and balanced fine-tune ten models based on it. After the balanced training stage, we merge those ten models and fine-tune the classifier. The results on the Places-LT, ImageNet-LT and iNaturalist2018 datasets are in Table 12, Table 13 and Table 14. It can be seen that compared with the first-stage pre-trained model, each model of balanced training has higher harmonic and geometric mean, and the merging strategy has further improved the results. After post-training, the accuracy of our final model produces a more even distribution of accuracy than the original model. We also report the PaCo/GPaCo's detailed results of each balanced fine-tuned and merged model during training. We first apply the 'FC' stage and then balanced train the weights. The results on the ImageNet-LT and Places-LT datasets are in Table 15 and Table 16 respectively. We can still come to similar conclusions.

Visualization of the Per-Class Accuracy. Since our goal is to improve the performance of the worst categories, here we visualize the change of per-class accuracy after applying our method to MiSLAS in the Places-LT dataset, and the result is shown in Figure 3. Our proposed balanced training makes the distribution of per-class accuracy more uniform, thus improves the

Methods		Harmonic Mean	Geometric Mean	Accuracy
Original Model	Stage1	1.19	39.50	66.87
	Stage2	2.03	51.27	**71.57**
Balanced Training	Model1	1.99	50.22	70.67
	Model2	1.90	49.55	70.59
	Model3	1.96	49.91	70.56
	Model4	1.96	49.87	70.50
	Model5	1.97	50.10	70.70
	Model6	1.97	49.96	70.58
	Model7	1.93	49.69	70.53
	Model8	1.93	49.89	70.72
	Model9	1.98	50.31	70.91
	Model10	1.97	49.97	70.52
Merge		2.02	50.69	70.96
Post-train		**2.18**	**52.02**	71.43

Table 14: Results of single fine-tuned and merged models in iNaturalist2018 with MiSLAS.

worst categories and leads to a higher harmonic mean.

5. Conclusions, Limitations and Future Work

In this paper, we presented a straightforward plug-and-play training strategy to tackle the worst-category problem in long-tailed learning, which has been paid more attention by researchers in recent years. By reviving (few-shot) balanced undersampling, our BTM training strategy can be easily integrated with various long-tailed algorithms, requiring minimal training overhead and imposing no additional inference burden. Across multiple widely used long-tailed datasets, BTM consistently achieves notable and stable improvements in both harmonic and geometric mean accuracy, while maintaining comparable average accuracy.

Although our method can significantly improve the accuracy balance across categories, we observed that for some large long-tailed datasets such as ImageNet and iNaturalist2018, the minimum recall remains zero even with the help of BTM. As a result, an intriguing direction for future research is how can we further enhance the minimum recall. Additionally, though our BTM will substantially increase harmonic and geometric mean, it will slightly decrease arithmetic accuracy in some scenarios. Especially on "many" subsets,

Methods		H-Mean	G-Mean	Accuracy
Original Model	Pre-train	21.75	51.29	**58.32**
	FC	2.19	33.82	52.12
Balanced Training	Model1	16.55	47.34	56.03
	Model2	11.33	47.17	56.08
	Model3	10.20	46.92	55.94
	Model4	11.29	46.64	55.54
	Model5	10.21	46.69	55.73
	Model6	10.01	46.09	55.41
	Model7	12.56	46.96	55.80
	Model8	9.28	46.79	55.95
	Model9	7.87	46.21	55.69
	Model10	12.65	47.33	56.02
Merge		11.52	48.46	57.03
Post-train		**22.85**	**51.45**	58.15

Table 15: Results of single fine-tuned and merged models in ImageNet-LT with PaCo.

there is a high probability that the accuracy will decline. Therefore, another interesting direction to explore is the simultaneous improvement of average accuracy alongside harmonic and geometric mean. Although our method is robust to model hyperparameters, how to accurately select the best hyperparameters is still a problem worth exploring. Besides, to further improve the persuasiveness of our BTM, it is also an interesting future direction to give a reasonable theoretical explanation for the algorithm.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[2] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (6) (2018) 1452–1464.

[3] Y. Du, J. Wu, No one left behind: Improving the worst categories in long-tailed learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15804–15813.

Methods		H-Mean	G-Mean	Accuracy
Original Model	Pre-train	10.93	35.00	**41.68**
	FC	1.86	16.58	30.24
Balanced Training	Model1	10.06	31.10	38.03
	Model2	10.24	31.94	38.64
	Model3	10.16	31.64	38.43
	Model4	20.49	32.02	38.52
	Model5	13.89	32.18	38.57
	Model6	13.80	32.31	38.85
	Model7	8.26	31.40	38.24
	Model8	8.18	31.77	38.73
	Model9	10.50	31.97	38.40
	Model10	14.03	31.74	38.07
Merge		10.61	32.85	39.43
Post-train		**29.36**	**35.86**	40.72

Table 16: Results of single fine-tuned and merged models in Places-LT with GPaCo.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

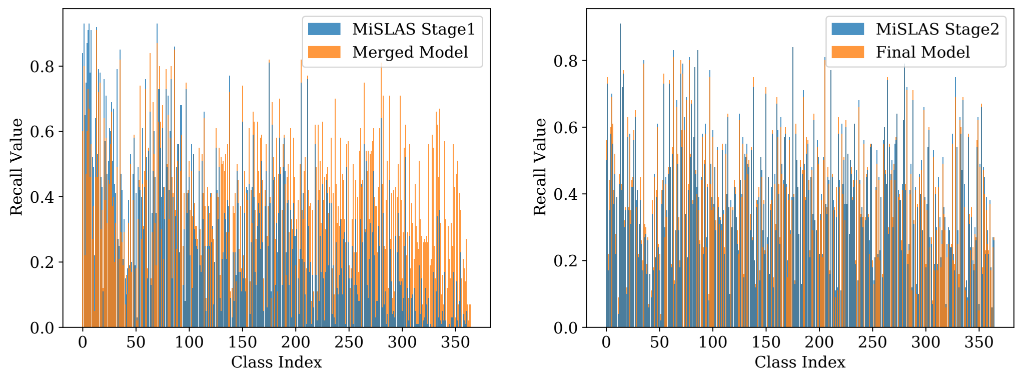
[5] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[6] J. Ren, C. Yu, S. Sheng, X. Ma, H. Zhao, S. Yi, H. Li, Balanced meta-softmax for long-tailed visual recognition, in: Advances in Neural Information Processing Systems 33, 2020, pp. 4175–4186.

[7] S. Park, Y. Hong, B. Heo, S. Yun, J. Y. Choi, The majority can help the minority: Context-rich minority oversampling for long-tailed classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 6877–6886.

[8] M.-L. Zhang, X.-Y. Zhang, C. Wang, C.-L. Liu, Towards prior gap and representation gap for long-tailed recognition, Pattern Recognition 133 (2023) 109012.

[9] X. Zhao, J. Xiao, S. Yu, H. Li, B. Zhang, Weight-guided class complementing for long-tailed image recognition, Pattern Recognition 138 (2023) 109374.



(a) (b)

Figure 3: Visualization of the change in the distribution of per-class recall (i.e., accuracy). (a) shows that by performing balanced training on our sampled few-shot datasets and later merging all models together, we are able to greatly improve the performance of the model. (b) is the comparison of per-class accuracy between our final model and MiSLAS.

[10] J. S. Baik, I. Y. Yoon, J. W. Choi, Dbn-mix: Training dual branch network using bilateral mixup augmentation for long-tailed visual recognition, Pattern Recognition 147 (2024) 110107.

[11] L. Xiang, J. Han, G. Ding, Margin-aware rectified augmentation for long-tailed recognition, Pattern Recognition 141 (2023) 109608.

[12] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: International Conference on Learning Representations, 2020.

[13] Z. Zhong, J. Cui, S. Liu, J. Jia, Improving calibration for long-tailed recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16489–16498.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[15] M. Li, Y.-m. Cheung, Y. Lu, Long-tailed visual recognition via gaussian clouded logit adjustment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6929–6938.

[16] H. Wang, A. Zhang, Y. Zhu, S. Zheng, M. Li, A. J. Smola, Z. Wang, Partial and asymmetric contrastive learning for out-of-distribution detection in long-

tailed recognition, in: International Conference on Machine Learning, 2022, pp. 23446–23458.

[17] B. Dong, P. Zhou, S. Yan, W. Zuo, Lpt: Long-tailed prompt tuning for image classification, in: International Conference on Learning Representations, 2023.

[18] M. Li, H. Zhikai, Y. Lu, W. Lan, Y.-m. Cheung, H. Huang, Feature fusion from head to tail for long-tailed visual recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2024, pp. 13581–13589.

[19] B. Zhou, Q. Cui, X.-S. Wei, Z.-M. Chen, BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9719–9728.

[20] X. Wang, L. Lian, Z. Miao, Z. Liu, S. Yu, Long-tailed recognition by routing diverse distribution-aware experts, in: International Conference on Learning Representations, 2021.

[21] J. Chen, B. Su, Transfer knowledge from head to tail: Uncertainty calibration under long-tailed distribution, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 19978–19987.

[22] Y. Jin, M. Li, Y. Lu, Y.-m. Cheung, H. Wang, Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 23695–23704.

[23] W. Zhao, H. Zhao, Hierarchical long-tailed classification based on multi-granularity knowledge transfer driven by multi-scale feature fusion, Pattern Recognition 145 (2024) 109842.

[24] W. Liu, Z. Wu, Y. Wang, H. Ding, F. Liu, J. Lin, G. Lin, LCReg: Long-tailed image classification with latent categories based recognition, Pattern Recognition 145 (2024) 109971.

[25] Z. Tan, J. Li, J. Du, J. Wan, Z. Lei, G. Guo, Ncl++: Nested collaborative learning for long-tailed visual recognition, Pattern Recognition 147 (2024) 110064.

[26] J. Cui, Z. Zhong, S. Liu, B. Yu, J. Jia, Parametric contrastive learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 715–724.

[27] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: Advances in Neural Information Processing Systems 33, 2020, pp. 18661–18673.

[28] J. Gao, H. Zhao, Z. Li, D. Guo, Enhancing minority classes by mixing: An adaptative optimal transport approach for long-tailed classification, in: Advances in Neural Information Processing Systems 36, 2023, pp. 60329–60348.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[30] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S. X. Yu, Large-scale long-tailed recognition in an open world, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2537–2546.

[31] G.-H. Wang, J. Wu, Practical network acceleration with tiny sets: Hypothesis, theory, and algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (12) (2024) 9272–9285.

[32] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images (2009).

[33] Y. Cui, Y. Song, C. Sun, A. Howard, S. Belongie, Large scale fine-grained categorization and domain-specific transfer learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4109–4118.

[34] S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.

[35] J. Cui, Z. Zhong, Z. Tian, S. Liu, B. Yu, J. Jia, Generalized parametric contrastive learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[36] Y.-Y. He, J. Wu, X.-S. Wei, Distilling virtual examples for long-tailed recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 235–244.

[37] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al., Model soups: averaging weights of multiple fine-tuned models improves accuracy

without increasing inference time, in: International Conference on Machine Learning, PMLR, 2022, pp. 23965–23998.