Global and Local Mixture Consistency Cumulative Learning for Long-tailed Visual Recognitions

Fei Du[1,2,3], Peng Yang[1,3], Qi Jia[1,3], Fengtao Nan[1,2,3], Xiaoting Chen[1,3], Yun Yang[1,3*]
[1]National Pilot School of Software, Yunnan University, Kunming, China
[2]School of Information Science and Engineering, Yunnan University, Kunming, China
[3]Yunnan Key Laboratory of Software Engineering
{dufei,yangpeng,jiaqi,fengtaonan,chenxiaoting}@mail.ynu.edu.cn yangyun@ynu.edu.cn

Abstract

In this paper, our goal is to design a simple learning paradigm for long-tail visual recognition, which not only improves the robustness of the feature extractor but also alleviates the bias of the classifier towards head classes while reducing the training skills and overhead. We propose an efficient one-stage training strategy for long-tailed visual recognition called Global and Local Mixture Consistency cumulative learning (GLMC). Our core ideas are twofold: (1) a global and local mixture consistency loss improves the robustness of the feature extractor. Specifically, we generate two augmented batches by the global MixUp and local CutMix from the same batch data, respectively, and then use cosine similarity to minimize the difference. (2) A cumulative head-tail soft label reweighted loss mitigates the head class bias problem. We use empirical class frequencies to reweight the mixed label of the head-tail class for long-tailed data and then balance the conventional loss and the rebalanced loss with a coefficient accumulated by epochs. Our approach achieves state-of-the-art accuracy on CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets. Additional experiments on balanced ImageNet and CIFAR demonstrate that GLMC can significantly improve the generalization of backbones. Code is made publicly available at https://github.com/ynu-yangpeng/GLMC.

1. Introduction

Thanks to the available large-scale datasets, *e.g.*, ImageNet [10], MS COCO [27], and Places [46] Database, deep neural networks have achieved dominant results in image recognition [15]. Distinct from these well-designed balanced datasets, data naturally follows long-tail distribution in real-world scenarios, where a small number of head classes occupy most of the samples. In contrast, dominant tail classes only have a few samples. Moreover, the tail classes are critical for some applications, such as medical diagnosis and autonomous driving. Unfortunately, learning directly from long-tailed data may cause model predictions to over-bias toward the head classes.

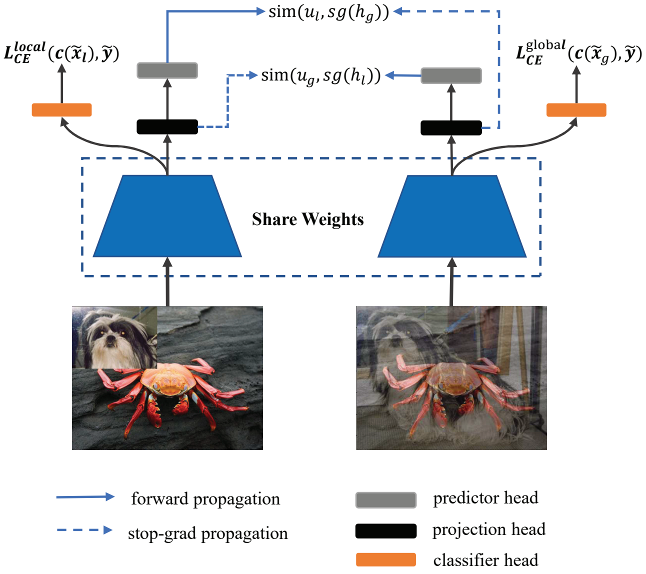
$\mathrm{sim}(u_l, sg(h_g))$
$L_{CE}^{local}(c(\tilde{x}_l), \tilde{y})$
$\mathrm{sim}(u_g, sg(h_l))$
$L_{CE}^{global}(c(\tilde{x}_g), \tilde{y})$
Share Weights
forward propagation
stop-grad propagation
predictor head
projection head
classifier head

Figure 1. An overview of our GLMC: two types of mixed-label augmented images are processed by an encoder network and a projection head to obtain the representation h_g and h_l. Then a prediction head transforms the two representations to output u_g and u_l. We minimize their negative cosine similarity as an auxiliary loss in the supervised loss. $sg(\cdot)$ denotes stop gradient operation.

There are two classical rebalanced strategies for long-tailed distribution, including resampling training data [7, 13, 35] and designing cost-sensitive reweighting loss functions [3, 20]. For the resampling methods, the core idea is to oversample the tail class data or undersample the head classes in the SGD mini-batch to balance training. As for

[*] Corresponding author

the reweighting strategy, it mainly increases the loss weight of the tail classes to strengthen the tail class. However, learning to rebalance the tail classes directly would damage the original distribution [45] of the long-tailed data, either increasing the risk of overfitting in the tail classes or sacrificing the performance of the head classes. Therefore, these methods usually adopt a two-stage training process [1,3,45] to decouple the representation learning and classifier fine-tuning: the first stage trains the feature extractor on the original data distribution, then fixes the representation and trains a balanced classifier. Although multi-stage training significantly improves the performance of long-tail recognition, it also negatively increases the training tricks and overhead.

In this paper, our goal is to design a simple learning paradigm for long-tail visual recognition, which not only improves the robustness of the feature extractor but also alleviates the bias of the classifier towards head classes while reducing the training skills and overhead. For improving representation robustness, recent contrastive learning techniques [8,18,26,47] that learn the consistency of augmented data pairs have achieved excellence. Still, they typically train the network in a two-stage manner, which does not meet our simplification goals, so we modify them as an auxiliary loss in our supervision loss. For head class bias problems, the typical approach is to initialize a new classifier for resampling or reweighting training. Inspired by the cumulative weighted rebalancing [45] branch strategy, we adopt a more efficient adaptive method to balance the conventional and reweighted classification loss.

Based on the above analysis, we propose an efficient one-stage training strategy for long-tailed visual recognition called Global and Local Mixture Consistency cumulative learning (GLMC). Our core ideas are twofold: (1) a global and local mixture consistency loss improves the robustness of the model. Specifically, we generate two augmented batches by the global MixUp and local CutMix from the same batch data, respectively, and then use cosine similarity to minimize the difference. (2) A cumulative head-tail soft label reweighted loss mitigates the head class bias problem. Specifically, we use empirical class frequencies to reweight the mixed label of the head-tail class for long-tailed data and then balance the conventional loss and the rebalanced loss with a coefficient accumulated by epochs.

Our method is mainly evaluated in three widely used long-tail image classification benchmark datasets, which include CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets. Extensive experiments show that our approach outperforms other methods by a large margin, which verifies the effectiveness of our proposed training scheme. Additional experiments on balanced ImageNet and CIFAR demonstrate that GLMC can significantly improve the generalization of backbones. The main contributions of our work can be summarized as follows:

- We propose an efficient one-stage training strategy called Global and Local Mixture Consistency cumulative learning framework (GLMC), which can effectively improve the generalization of the backbone for long-tailed visual recognition.
- GLMC does not require negative sample pairs or large batches and can be as an auxiliary loss added in supervised loss.
- Our GLMC achieves state-of-the-art performance on three challenging long-tailed recognition benchmarks, including CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets. Moreover, experimental results on full ImageNet and CIFAR validate the effectiveness of GLMC under a balanced setting.

2. Related Work

2.1. Contrastive Representation Learning

The recent renaissance of self-supervised learning is expected to obtain a general and transferrable feature representation by learning pretext tasks. For computer vision, these pretext tasks include rotation prediction [22], relative position prediction of image patches [11], solving jigsaw puzzles [30], and image colorization [23, 43]. However, these pretext tasks are usually domain-specific, which limits the generality of learned representations.

Contrastive learning is a significant branch of self-supervised learning. Its pretext task is to bring two augmented images (seen as positive samples) of one image closer than the negative samples in the representation space. Recent works [17, 31, 36] have attempted to learn the embedding of images by maximizing the mutual information of two views of an image between latent representations. However, their success relies on a large number of negative samples. To handle this issue, BYOL [12] removes the negative samples and directly predicts the output of one view from another with a momentum encoder to avoid collapsing. Instead of using a momentum encoder, Simsiam [5] adopts siamese networks to maximize the cosine similarity between two augmentations of one image with a simple stop-gradient technique to avoid collapsing.

For long-tail recognition, there have been numerous works [8, 18, 26, 47] to obtain a balanced representation space by introducing a contrastive loss. However, they usually require a multi-stage pipeline and large batches of negative examples for training, which negatively increases training skills and overhead. Our method learns the consistency of the mixed image by cosine similarity, and this method is conveniently added to the supervised training in an auxiliary loss way. Moreover, our approach neither uses negative pairs nor a momentum encoder and does not rely on large-batch training.

2.2. Class Rebalance learning

Rebalance training has been widely studied in long-tail recognition. Its core idea is to strengthen the tail class by oversampling [4, 13] or increasing weight [2, 9, 44]. However, over-learning the tail class will also increase the risk of overfitting [45]. Conversely, under-sampling or reducing weight in the head class will sacrifice the performance of head classes. Recent studies [19, 45] have shown that directly training the rebalancing strategy would degrade the performance of representation extraction, so some multi-stage training methods [1, 19, 45] decouple the training of representation learning and classifier for long-tail recognition. For representation learning, self-supervised-based [18, 26, 47] and augmentation-based [6, 32] methods can improve robustness to long-tailed distributions. And for the rebalanced classifier, such as multi-experts [24, 37], reweighted classifiers [1], and label-distribution-aware [3], all can effectively enhance the performance of tail classes. Further, [45] proposed a unified Bilateral-Branch Network (BBN) that adaptively adjusts the conventional learning branch and the reversed sampling branch through a cumulative learning strategy. Moreover, we follow BBN to weight the mixed labels of long-tailed data adaptively and do not require an ensemble during testing.

3. The Proposed Method

In this section, we provide a detailed description of our GLMC framework. First, we present an overview of our framework in Sec.3.1. Then, we introduce how to learn global and local mixture consistency by maximizing the cosine similarity of two mixed images in Sec.3.2. Next, we propose a cumulative class-balanced strategy to weight long-tailed data labels progressively in Sec.3.3. Finally, we introduce how to optionally use MaxNorm [1, 16] to fine-tune the classifier weights in Sec.3.4.

3.1. Overall Framework

Our framework is divided into the following six major components:

- A stochastic mixed-label data augmentation module $Aug(x, y)$. For each input batch samples, $Aug(x, y)$ transforms x and their labels y in global and local augmentations pairs, respectively.

- An encoder (e.g., ResNet) $f(\tilde{x})$ that extracts representation vectors r from the augmented samples $\tilde{x}$.

- A projection $proj(x)$ that maps vectors r to lower dimension representations h. The projection is simply a fully connected layer. Its output has no activation function.

- A predictor $pred(x)$ that maps the output of projection to the contrastive space. The predictor also a fully connected layer and has no activation function.

- A linear conventional classifier head $c(x)$ that maps vectors r to category space. The classifier head calculates mixed cross entropy loss with the original data distribution.

- (optional) A linear rebalanced classifier head $cb(x)$ that maps vectors r to rebalanced category space. The rebalanced classifier calculates mixed cross entropy loss with the reweighted data distribution.

Note that only the rebalanced classifier $cb(x)$ is retained at the end of training for the long-tailed recognition, while the predictor, projection, and conventional classifier head will be removed. However, for the balanced dataset, the rebalanced classifier $cb(x)$ is not needed.

3.2. Global and Local Mixture Consistency Learning

In supervised deep learning, the model is usually divided into two parts: an encoder and a linear classifier. And the classifiers are label-biased and rely heavily on the quality of representations. Therefore, improving the generalization ability of the encoder will significantly improve the final classification accuracy of the long-tailed challenge. Inspired by self-supervised learning to improve representation by learning additional pretext tasks, as illustrated in Fig.1, we train the encoder using a standard supervised task and a self-supervised task in a multi-task learning way. Further, unlike simple pretext tasks such as rotation prediction, image colorization, etc., following the global and local ideas [39], we expect to learn the global-local consistency through the strong data augmentation method MixUp [42] and CutMix [41].

Global Mixture. MixUp is a global mixed-label data augmentation method that generates mixture samples by mixing two images of different classes. For a pair of two images and their labels probabilities (x_i, p_i) and (x_j, p_j), we calculate $(\tilde{x}_g, \tilde{p}_g)$ by

$$\begin{aligned} &\lambda \sim Beta(\beta, \beta), \\ &\tilde{x}_g = \lambda x_i + (1-\lambda) x_j, \\ &\tilde{p}_g = \lambda p_i + (1-\lambda) p_j. \end{aligned} \tag{1}$$

where λ is sampled from a $Beta$ distribution parameterized by the β hyper-parameter. Note that p are one-hot vectors.

Local Mixture. Different from MixUp, CutMix combines two images by locally replacing the image region with a patch from another training image. We define the combining operation as

$$\tilde{x}_l = \boldsymbol{M} \odot x_i + (\mathbf{1} - \boldsymbol{M}) \odot x_j. \tag{2}$$

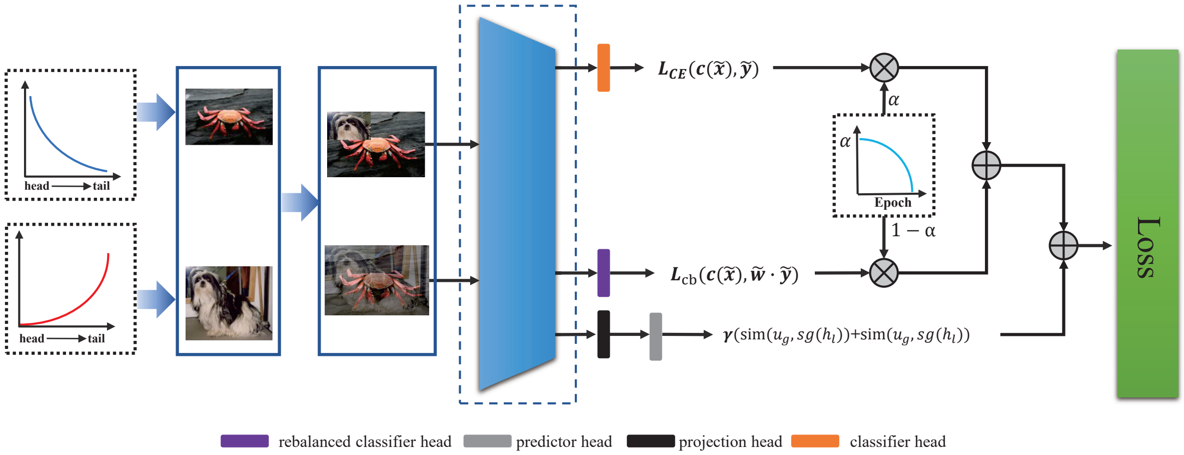
head
tail
head
tail
$L_{CE}(c(\tilde{x}), \tilde{y})$
α
α
Epoch
$1-\alpha$
$L_{cb}(c(\tilde{x}), \tilde{w}\cdot\tilde{y})$
$\gamma(\text{sim}(u_g, sg(h_l))+\text{sim}(u_g, sg(h_l)))$
Loss
rebalanced classifier head
predictor head
projection head
classifier head

Figure 2. An illustration of the cumulative class-balanced learning pipeline. We apply uniform and reversed samplers to obtain head and tail data, and then they are synthesized into head-tail mixture samples by MixUp and CutMix. The cumulative learning strategy adaptively weights the rebalanced classifier and the conventional classifier by epochs.

where $\boldsymbol{M} \in \{0,1\}^{W\times H}$ denotes the randomly selected pixel patch from the image x_i and pasted on x_j, $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is element-wise multiplication. Concretely, we sample the bounding box coordinates $B = (r_x, r_y, r_w, r_h)$ indicating the cropping regions on x_i and x_j. The box coordinates are uniformly sampled according to

$$\begin{aligned} r_x &\sim Uniform(0, W), r_w = W\sqrt{1-\lambda} \\ r_y &\sim Uniform(0, H), r_h = H\sqrt{1-\lambda} \end{aligned} \tag{3}$$

where λ is also sampled from the $Beta(\beta, \beta)$, and their mixed labels are the same as MixUp.

Self-Supervised Learning Branch. Previous works require large batches of negative samples [17, 36] or a memory bank [14] to train the network. That makes it difficult to apply to devices with limited memory. For simplicity, our goal is to maximize the cosine similarity of global and local mixtures in representation space to obtain contrastive consistency. Specifically, the two types of augmented images are processed by an encoder network and a projection head to obtain the representation h_g and h_l. Then a prediction head transforms the two representations to output u_g and u_l. We minimize their negative cosine similarity:

$$sim(u_g, h_l) = -\frac{u_g}{\|u_g\|}\cdot\frac{h_l}{\|h_l\|} \tag{4}$$

where $\|\cdot\|$ is l_2 normalization. An undesired trivial solution to minimize the negative cosine similarity of augmented images is all outputs "collapsing" to a constant. Following SimSiam [5], we use a stop gradient operation to prevent collapsing. The SimSiam loss function is defined as:

$$\pounds_{sim} = sim(u_g, sg(h_l)) + sim(u_l, sg(h_g)) \tag{5}$$

this means that h_l and h_g are treated as a constant.

Supervised Learning Branch. After constructing the global and local augmented data pair $(\tilde{x}_g; \tilde{p}_g)$ and $(\tilde{x}_l; \tilde{p}_l)$, we calculate the mixed-label cross-entropy loss:

$$\pounds_c = -\frac{1}{2N}\sum_{i=1}^{N}(\tilde{p}_g^i(logf(\tilde{x}_g^i)) + \tilde{p}_l^i(logf(\tilde{x}_l^i))) \tag{6}$$

where N denote the sampling batch size and $f(\cdot)$ denote predicted probability of $\tilde{x}$. Note that a batch of images is augmented into a global and local mixture so that the actual batch size will be twice the sampling size.

3.3. Cumulative Class-Balanced Learning

Class-Balanced learning. The design principle of class reweighting is to introduce a weighting factor inversely proportional to the label frequency and then strengthen the learning of the minority class. Following [44], the weighting factor w_i is define as:

$$w_i = \frac{C\cdot(1/r_i)^k}{\sum_{i=1}^{C}(1/r_i)^k} \tag{7}$$

where r_i is the i-th class frequencies of the training dataset, and k is a hyper-parameter to scale the gap between the head and tail classes. Note that $k = 0$ corresponds to no reweighting and $k = 1$ corresponds to class-balanced method [9]. We change the scalar weights to the one-hot vectors form and mix the weight vectors of the two images:

$$\tilde{w} = \lambda w_i + (1-\lambda)w_j. \tag{8}$$

Formally, given a train dataset $D = \{(x_i, y_i, w_i)\}_{i=1}^{N}$, the rebalanced loss can be written as:

$$\pounds_{cb} = -\frac{1}{2N}\sum_{i=1}^{N}\tilde{w}^i(\tilde{p}_g^i(logf(\tilde{x}_g^i)) + \tilde{p}_l^i(logf(\tilde{x}_l^i))) \tag{9}$$

Table 1. Top-1 accuracy (%) of ResNet-32 on CIFAR-10-LT and CIFAR-100-LT with different imbalance factors [100, 50, 10]. GLMC consistently outperformed the previous best method only in the one-stage.

	Method	CIFAR-10-LT			CIFAR-100-LT		
		IF=100	50	10	100	50	10
	CE	70.4	74.8	86.4	38.3	43.9	55.7
rebalance classifier	BBN [45]	79.82	82.18	88.32	42.56	47.02	59.12
	CB-Focal [9]	74.6	79.3	87.1	39.6	45.2	58
	LogitAjust [29]	80.92	-	-	42.01	47.03	57.74
	weight balancing [1]	-	-	-	53.35	57.71	68.67
augmentation	Mixup [42]	73.06	77.82	87.1	39.54	54.99	58.02
	RISDA [6]	79.89	79.89	79.89	50.16	53.84	62.38
	CMO [32]	-	-	-	47.2	51.7	58.4
self-supervised pretraining	KCL [18]	77.6	81.7	88	42.8	46.3	57.6
	TSC [25]	79.7	82.9	88.7	42.8	46.3	57.6
	BCL [47]	84.32	87.24	91.12	51.93	56.59	64.87
	PaCo [8]	-	-	-	52	56	64.2
	SSD [26]	-	-	-	46	50.5	62.3
ensemble classifier	RIDE (3 experts) + CMO [32]	-	-	-	50	53	60.2
	RIDE (3 experts) [37]	-	-	-	48.6	51.4	59.8
one-stage training	**ours**	**92.34**	**94.37**	**94.92**	**55.88**	**61.08**	**70.74**
finetune classifier	**ours + MaxNorm [1]**	**94.18**	**95.13**	**95.7**	**57.11**	**62.32**	**72.33**

where $f(\tilde{x})$ and $\tilde{w}$ denote predicted probability and weighting factor of mixed image $\tilde{x}$, respectively. Note that the global and local mixed image have the same mixed weights.

Cumulative Class-Balanced Learning. As illustrated in Fig.2, we use the bilateral branches structure to learn the rebalance branch adaptively. But unlike BBN [45], our cumulative learning strategy is imposed on the loss function instead of the fully connected layer weights and uses reweighting instead of resampling for learning. Concretely, the loss $\pounds_c$ of the unweighted classification branch is multiplied by α, and the rebalanced loss $\pounds_{cb}$ is multiplied by $1-\alpha$. α automatically decreases as the current training epochs T increase:

$$\alpha = 1 - (\frac{T}{T_{max}})^2 \tag{10}$$

where T_{max} is the maximum training epoch.

Finally, the total loss is defined as a combination of loss L_{sup}, L_{cb}, and L_{sim}:

$$\pounds_{total} = \alpha \pounds_c + (1-\alpha)\pounds_{cb} + \gamma \pounds_{sim} \tag{11}$$

where γ is a hyperparameter that controls L_{sim} loss. The default value is 10.

3.4. Finetuning Classifier Weights

[1] investigate that the classifier weights would be heavily biased toward the head classes when faced with long-tail data. Therefore, we optionally use MaxNorm [1, 16] to finetune the classifier in the second stage. Specifically, MaxNorm restricts weight norms within a ball of radius δ:

$$\Theta^* = \underset{\Theta}{argmin} F(\Theta; D),\ s.t. ||\theta_k||_2^2 \le \delta^2 \tag{12}$$

this can be solved by applying projected gradient descent (PGD). For each epoch (or iteration), PGD first computes an updated θ_k and then projects it onto the norm ball:

$$\theta_k \leftarrow min(1, \delta/||\theta_k||_2) * \theta_k \tag{13}$$

4. Experiments

In this section, we evaluate the proposed GLMC on three widely used long-tailed benchmarks: CIFAR-10-LT, CIFAR-100-LT, and ImageNet-LT. We also conduct a series of ablation studies to assess each component of GLMC's importance fully.

4.1. Experiment setup

Datasets. Following [40], we modify the balanced CIFAR10, CIFAR100, and ImageNet-2012 dataset to the uneven setting (named CIFAR10-LT, CIFAR100-LT, and ImageNet-LT) by utilizing the exponential decay function $n = n_i\mu^i$, where i is the class index (0-indexed), n_i is the original number of training images and $\mu \in (0, 1)$. The imbalanced factor β is defined by $\beta = N_{max}/N_{min}$, which reflects the degree of imbalance in the data. CIFAR10-LT and CIFAR100-LT are divided into three types of train datasets, and each dataset has a different imbalance factor [100,50,10]. ImageNet-LT has a 256 imbalance factor. The

Table 2. Top-1 accuracy (%) on ImageNet-LT dataset. Comparison to the state-of-the-art methods with different backbone. † denotes results reproduced by [47] with 180 epochs.

Method	Backbone	ImageNet-LT			
		Many	Med	Few	All
CE	ResNet-50	64	33.8	5.8	41.6
CB-Focal [9]	ResNet-50	39.6	32.7	16.8	33.2
LDAM [3]	ResNet-50	60.4	46.9	30.7	49.8
KCL [18]	ResNet-50	61.8	49.4	30.9	51.5
TSC [25]	ResNet-50	63.5	49.7	30.4	52.4
RISDA [6]	ResNet-50	-	-	-	49.3
BCL (90 epochs) [47]	ResNeXt-50	67.2	53.9	36.5	56.7
BCL (180 epochs) [47]	ResNeXt-50	67.9	54.2	36.6	57.1
PaCo† (180 epochs) [8]	ResNeXt-50	64.4	55.7	33.7	56.0
Balanced Softmax† (180 epochs) [34]	ResNeXt-50	65.8	53.2	34.1	55.4
SSD [26]	ResNeXt-50	66.8	53.1	35.4	56
RIDE (3 experts) + CMO [32]	ResNet-50	66.4	53.9	35.6	56.2
RIDE (3 experts) [37]	Swin-S	66.9	52.8	37.4	56
weight balancing + MaxNorm [1]	ResNeXt-50	62.5	50.4	41.5	53.9
ours	ResNeXt-50	**70.1**	52.4	30.4	56.3
ours + MaxNorm [1]		60.8	**55.9**	**45.5**	56.7
ours + BS [34]		64.76	55.67	42.19	**57.21**

most frequent class includes 1280 samples, while the least contains only 5.

Network architectures. For a fair comparison with recent works, we follow [1, 8, 47] to use ResNet-32 [15] on CIFAR10-LT and CIFAR100-LT, ResNet-50 [15] and ResNeXt-50-32x4d [38] on ImageNet-LT. The main ablation experiment was performed using ResNet-32 on the CIFAR100 dataset.

Evaluation protocol. For each dataset, we train them on the imbalanced training set and evaluate them in the balanced validation/test set. Following [1, 28], we further report accuracy on three splits of classes, Many-shot classes (training samples > 100), Medium-shot (training samples $20 \sim 100$) and Few-shot (training samples ≤ 20), to comprehensively evaluate our model.

Implementation. We train our models using the PyTorch toolbox [33] on GeForce RTX 3090 GPUs. All models are implemented by the SGD optimizer with a momentum of 0.9 and gradually decay learning rate with a cosine annealing scheduler, and the batch size is 128. For CIFAR10-LT and CIFAR100-LT, the initial learning rate is 0.01, and the weight decay rate is 5e-3. For ImageNet-LT, the initial learning rate is 0.1, and the weight decay rate is 2e-4. We also use random horizontal flipping and cropping as simple augmentation.

4.2. Long-tailed Benchmark Results

Compared Methods. Since the field of LTR is developing rapidly and has many branches, we choose recently published representative methods of different types for comparison. For example, SSD [26], PaCo [8], KCL [18], BCL [47], and TSC [25] use contrastive learning or self-supervised methods to train balanced representations. RIDE [37] combines multiple experts for prediction; RISDA [6] and CMO [32] apply strong data augmentation techniques to improve robustness; Weight Balancing [1] is a typical two-stage training method.

Table 3. Top-1 accuracy (%) on full ImageNet dataset with ResNet-50 backbone.

Method	Augmentation	Top-1 acc
vanilla	Simple Augment	76.4
vanilla	MixUp [42]	77.9
vanilla	CutMix [41]	78.6
Supcon [21]	RandAugment	78.4
PaCo [8]	Simple Augment	78.7
PaCo [8]	RandAugment	79.3
ours	MixUp + CutMix	**80.2**

Results on CIFAR10-LT and CIFAR100-LT. We conduct extensive experiments to compare GLMC with state-of-the-art baselines on long-tailed CIFAR10 and CIFAR100 datasets by setting three imbalanced ratios: 10, 50, and 100. Table 1 reports the Top-1 accuracy of various methods on CIFAR-10-LT and CIFAR-100-LT. We can see that our GLMC consistently achieves the best results on all datasets, whether using one-stage training or a two-stage finetune classifier. For example, on CIFAR100-LT (IF=100), Our method achieves **55.88%** in the first stage, outperforming

Algorithm 1 Learning algorithm of our proposed GLMC

Input: Training Dataset $D = \{(x_i, y_i, w_i)\}_{i=1}^{N}$
Parameter: $Encoder(\cdot)$ denotes feature extractor; $proj(\cdot)$ and $pred(\cdot)$ denote projection and predictor; $c(\cdot)$ and $cb(\cdot)$ denote convention classifier and rebalanced classifier; T_{max} is the Maximum training epoch; $sg(\cdot)$ denotes stop gradient operation.

```
for $T = 1$ in $T_{max}$ do
    $\alpha \leftarrow 1 - (\frac{T}{T_{max}})^2$
    for $(x, y, w)$ in $D$ do
        $\lambda \leftarrow Beta(\beta, \beta)$
        $(\tilde{x}_g, \tilde{p}_g, \tilde{w}_g) \leftarrow MixUp(x, y, w, \lambda)$
        $(\tilde{x}_l, \tilde{p}_l, \tilde{w}_l) \leftarrow CutMix(x, y, w, \lambda)$
// Generate global and local mixed augmented data.
        $r_g, r_l \leftarrow Encoder(\tilde{x}_g), Encoder(\tilde{x}_l)$
        $h_g, h_l \leftarrow proj(r_g), proj(r_l)$
// Map representation $r_g$ and $r_l$ to vector $h_g$ and $h_l$.
        $u_g, u_l \leftarrow pred(h_g), pred(h_l)$
// Map representation $h_g$ and $h_l$ to contrastive space $u_g$ and $u_l$.
        $\pounds_{sim} \leftarrow sim(u_g, sg(h_l)) + sim(u_l, sg(h_g))$
// Calculate global and local mixture similarity.
        $p_g^c, p_l^c \leftarrow Sofmatx(c(r_g)), Sofmatx(c(r_l))$
// Calculate the classification probability of the convention branch
        $\pounds_c \leftarrow \pounds(\tilde{p}, p_g^c) + \pounds(\tilde{p}, p_l^c)$
// Calculate the classification loss
        $p_g^{cb}, p_l^{cb} \leftarrow Sofmatx(cb(r_g)), Sofmatx(cb(r_l))$
// Calculate the classification probability of the rebalance branch
        $\pounds_{cb} \leftarrow \pounds(\tilde{p}, p_g^{cb}) + \pounds(\tilde{p}, p_l^{cb})$
// Calculate the rebalanced classification loss
        $\pounds_{total} = \alpha\pounds_c + (1 - \alpha)\pounds_{cb} + \gamma\pounds_{sim}$
// Calculate the total loss
        Update model parameters by minimizing $\pounds_{total}$
    end for
end for
```

the two-stage weight rebalancing strategy (53.35%). After finetuning the classifier with MaxNorm, our method achieves **57.11%**, which accuracy increased by **+3.76%** compared with the previous SOTA. Compared to contrastive learning families, such as PaCo and BCL, GLMC surpasses previous SOTA by **+5.11%**, **+5.73%**, and **+7.46%** under imbalance factors of 100, 50, and 10, respectively. In addition, GLMC does not need a large batch size and long training epoch to pretrain the feature extractor, which reduces training skills.

Results on ImageNet-LT. Table 2 compares GLMC with state-of-the-art baselines on ImageNet-LT dataset. We report the Top-1 accuracy on Many-shot, Medium-shot, and Few-shot groups. As shown in the table, with only one-stage training, GLMC significantly improves the performance of the head class by 70.1%, and the overall performance reaches 56.3%, similar to PaCo (180 epochs). After finetuning the classifier, the tail class of GLMC can reach 45.5% (+ MaxNorm [1]) and 42.19% (+ BS [34]), which significantly improves the performance of the tail class.

Table 4. Top-1 accuracy (%) on full CIFAR-10 and CIFAR-100 dataset with ResNet-50 backbone.

Method	CIFAR-10	CIFAR-100
vanilla	94.85	75.28
MixUp [42]	95.95	77.99
CutMix [41]	95.41	78.03
SupCon [21]	96	76.5
PaCo [8]	-	79.1
ours	**97.23**	**83.05**

4.3. Full ImageNet and CIFAR Recognition

GLMC utilizes a global and local mixture consistency loss as an auxiliary loss in supervised loss to improve the robustness of the model, which can be added to the model as a plug-and-play component. To verify the effectiveness of GLMC under a balanced setting, we conduct experiments on full ImageNet and full CIFAR. They are indicative to compare GLMC with the related state-of-the-art methods (MixUp [42], CutMix [41], PaCo [8], and SupCon [8]). Note that under full ImageNet and CIFAR, we remove the cumulative reweighting and resampling strategies customized for long-tail tasks.

Results on Full CIFAR-10 and CIFAR-100. For CIFAR-10 and CIFAR-100 implementation, following PaCo and SupCon, we use ResNet-50 as the backbone. As shown in Table 4, on CIFAR-100, GLMC achieves **83.05%** Top-1 accuracy, which outperforms PaCo by **3.95%**. Furthermore, GLMC exceeds the vanilla cross entropy method by **2.13%** and **7.77%** on CIFAR-10 and CIFAR-100, respectively, which can significantly improve the performance of the base model.

Results on Full ImageNet. In the implementation, we transfer hyperparameters of GLMC on ImageNet-LT to full ImageNet without modification. The experimental results are summarized in Table 3. Our model achieves **80.2%** Top-1 accuracy, outperforming PaCo and SupCon by **+0.9%** and **+1.8%**, respectively. Compared to the positive/negative contrastive model (PaCo and SupCon). GLMC does not need to construct negative samples, which can effectively reduce memory usage during training.

4.4. Ablation Study

To further analyze the proposed GLMC, we perform several ablation studies to evaluate the contribution of each

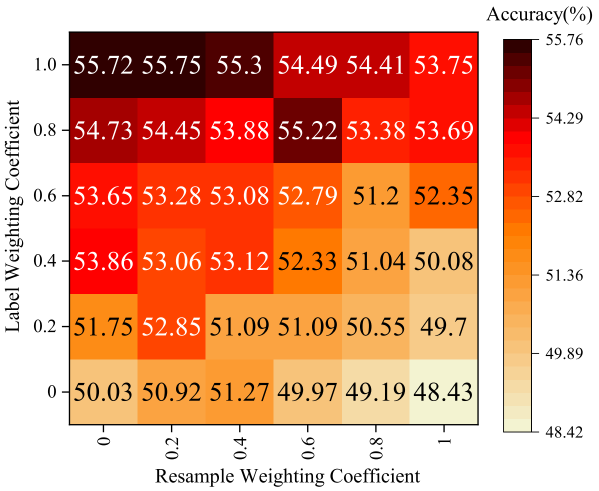

Figure 3. Confusion matrices of different label reweighting and resample coefficient k on CIFAR-100-LT with an imbalance ratio of 100.

component more comprehensively. All experiments are performed on CIFAR-100 with an imbalance factor of 100.

The effect of rebalancing intensity. As analyzed in Sec. 3.3, we mitigate head classes bias problems by reweighting labels and sampling weights by inverting the class sampling frequency. See Fig.3, we set different reweighting and resampling coefficients to explore the influence of the rebalancing strategy of GLMC on long tail recognition. One can see very characteristic patterns: the best results are clustered in the upper left, while the worst are in the lower right. It indicates that the class resampling weight is a very sensitive hyperparameter in the first-stage training. Large resampling weight may lead to model performance degradation, so it should be set to less than 0.4 in general. And label reweighting improves long tail recognition significantly and can be set to 1.0 by default.

The effect of mixture consistency weight γ. We investigate the influence of the mixture consistency weight γ on the CIFAR100-LT (IF=100) and plot the accuracy-weight curve in Fig.4. It is evident that adjusting γ is able to achieve significant performance improvement. Compared with the without mixture consistency (γ=0), the best setting (γ=10) can improve the performance by +2.41%.

The effect of each component. GLMC contains two essential components:(1) Global and Local Mixture Consistency Learning and (2) Cumulative Class-Balanced reweighting. Table 5 summarizes the ablation results of GLMC on CIFAR100-LT with an imbalance factor of 100. Note that both settings are crossed to indicate using a standard cross-entropy training model. We can see that both components significantly improve the baseline method. Analyzing each element individually, Global and Local Mixture Consistency Learning is crucial, which improves performance by an average of 11.81% (38.3% → 50.11%).

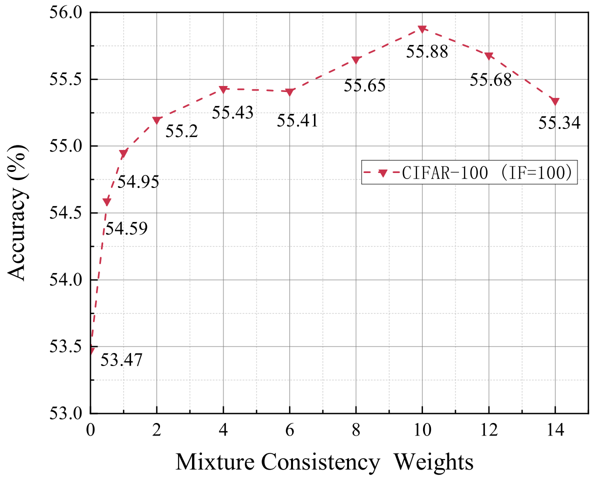

Figure 4. Different global and local mixture consistency weights on CIFAR-100-LT (IF = 100) .

Table 5. Ablations of the different key components of GLMC architecture. We report the accuracies (%) on CIFAR100-LT (IF=100) with ResNet-32 backbone. Note that all model use one-stage training.

Global and Local Mixture Consistency	Cumulative Class-Balanced	Accuracies(%)
×	×	38.3
×	✓	44.63
✓	×	50.11
✓	✓	55.88

5. Conclusion

In this paper, we have proposed a simple learning paradigm called Global and Local Mixture Consistency cumulative learning (GLMC). It contains a global and local mixture consistency loss to improve the robustness of the feature extractor, and a cumulative head-tail soft label reweighted loss mitigates the head class bias problem. Extensive experiments show that our approach can significantly improve performance on balanced and long-tailed visual recognition tasks.

Acknowledgements. The authors would like to acknowledge the financial support provided by the Natural Science Foundation of China (NSFC) under Grant: No.61876166, Yunnan Basic Research Program for Distinguished Young Youths Project under Grant: 202101AV070003, and Yunnan Basic Research Program under Grant: 202201AS070131.

References

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6907, 2022. 2, 3, 5, 6, 7

[2] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019. 3

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3, 6

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 3

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2, 4

[6] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 356–364, 2022. 3, 5, 6

[7] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020. 1

[8] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021. 2, 5, 6, 7

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 3, 4, 5, 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2

[13] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005. 1, 3

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 6

[16] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 3, 5

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 2, 4

[18] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020. 2, 3, 5, 6

[19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019. 3

[20] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017. 1

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 6, 7

[22] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[23] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6874–6883, 2017. 2

[24] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6979, 2022. 3

[25] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6918–6928, 2022. 5, 6

[26] Tianhao Li, Limin Wang, and Gangshan Wu. Self supervision to distillation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 630–639, 2021. 2, 3, 5, 6

[27] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[28] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 6

[29] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 5

[30] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 3, 5, 6

[33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[34] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 6, 7

[35] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. 1

[36] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020. 2, 4

[37] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. 3, 5, 6

[38] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6

[39] Yun Yang and Jianmin Jiang. Hybrid sampling-based clustering ensemble with global and local constitutions. *IEEE Transactions on Neural Networks and Learning Systems*, 27(5):952–965, 2016. 3

[40] Cao Yue, M Long, J Wang, Zhu Han, and Q Wen. Deep quantization network for efficient image retrieval. In *Proc. 13th AAAI Conf. Artif. Intell.*, pages 3457–3463, 2016. 5

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3, 6, 7

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3, 5, 6, 7

[43] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[44] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021. 3, 4

[45] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min 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*, pages 9719–9728, 2020. 2, 3, 5

[46] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 1

[47] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2022. 2, 3, 5, 6