Pattern Recognition Letters 168 (2023) 123–130

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Improving Tail-Class Representation with Centroid Contrastive Learning

Anthony Meng Huat Tiong[a,b,*], Junnan Li[a], Guosheng Lin[b], Boyang Li[b], Caiming Xiong[a], Steven C.H. Hoi[a]

[a] Salesforce Research, Singapore/United States
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

*Article history:*
Received 27 September 2022
Revised 11 January 2023
Accepted 7 March 2023
Available online 9 March 2023

Edited by: Jiwen Lu

*Keywords:*
Long-tailed classification
Imbalanced learning
Contrastive learning
Deep learning

ABSTRACT

In vision domain, large-scale natural datasets typically exhibit long-tailed distribution which has large class imbalance between head and tail classes. This distribution poses difficulty in learning good representations for tail classes. Recent developments have shown good long-tailed model can be learnt by decoupling the training into representation learning and classifier balancing. However, these works pay insufficient consideration on the long-tailed effect on representation learning. In this work, we propose interpolative centroid contrastive learning (ICCL) to improve long-tailed representation learning. ICCL interpolates two images from a class-agnostic sampler and a class-aware sampler, and trains the model such that the representation of the interpolative image can be used to retrieve the centroids for both source classes. We demonstrate the effectiveness of our approach on multiple long-tailed image classification benchmarks.



## 1. Introduction

In recent years, deep learning algorithms have achieved impressive results in various computer vision tasks [1–4]. However, long-tailed recognition remains as one of the major challenges. Different from most human-curated datasets where object classes have a balanced number of samples, the distribution of objects in real-world is a function of Zipf's law [5] where a large number of tail classes have few samples. Thus, models typically suffer a decrease in accuracy on the tail classes. Since it is resource-intensive to curate more samples for all tail classes, it is imperative to address the challenge of long-tailed recognition.

In the literature of long-tailed recognition, typical approaches address the class imbalance issue by either data re-sampling [6–8] or loss re-weighting techniques [9–11]. Re-sampling facilitates the learning of tail classes by shifting the skewed training data distribution towards the tail through undersampling or oversampling. Re-weighting modifies the loss function to encourage larger gradient contribution or decision margin of tail classes.

However, recent developments [12,13] discover that conventional re-sampling and re-weighting methods can lead to a sub-optimal long-tailed representation learning. In light of these findings, various approaches [12,13] propose to decouple representation learning and classifier balancing. BBN [13] demonstrates that the performance can be further improved by addressing the long-tailed effect during the representation learning.

In order to correctly classify tail-class samples, it is crucial to learn discriminative representations. In this work, we propose interpolative contrastive centroid learning (ICCL), a new two-stage framework to learn discriminative representations for tail classes. Specifically, inspired by Mixup [14], a data augmentation which linearly combines the training data in the input and label space, we create virtual training samples by interpolating two images from two samplers: a class-agnostic sampler which returns all images with equal probability, and a class-aware sampler which focuses more on tail-class images. We project images into a low-dimensional embedding space, and create class centroids as average embeddings. Given the interpolative embedding, we query the class centroids with a contrastive similarity matching, and train our model such that the embedding has higher similarities with the correct class centroids.

The intuition behind our method is to use the head classes to facilitate representation learning of the tail classes. For example, the head-class golden retriever may help the recognition of the

*Abbreviations:* ICCL, Interpolative contrastive centroid learning.

* Corresponding author.
*E-mail addresses:* anthonym001@e.ntu.edu.sg (A.M.H. Tiong), junnan.li@salesforce.com (J. Li), gslin@ntu.edu.sg (G. Lin), boyang.li@ntu.edu.sg (B. Li), cxiong@salesforce.com (C. Xiong), shoi@salesforce.com (S.C.H. Hoi).

https://doi.org/10.1016/j.patrec.2023.03.010
0167-8655/

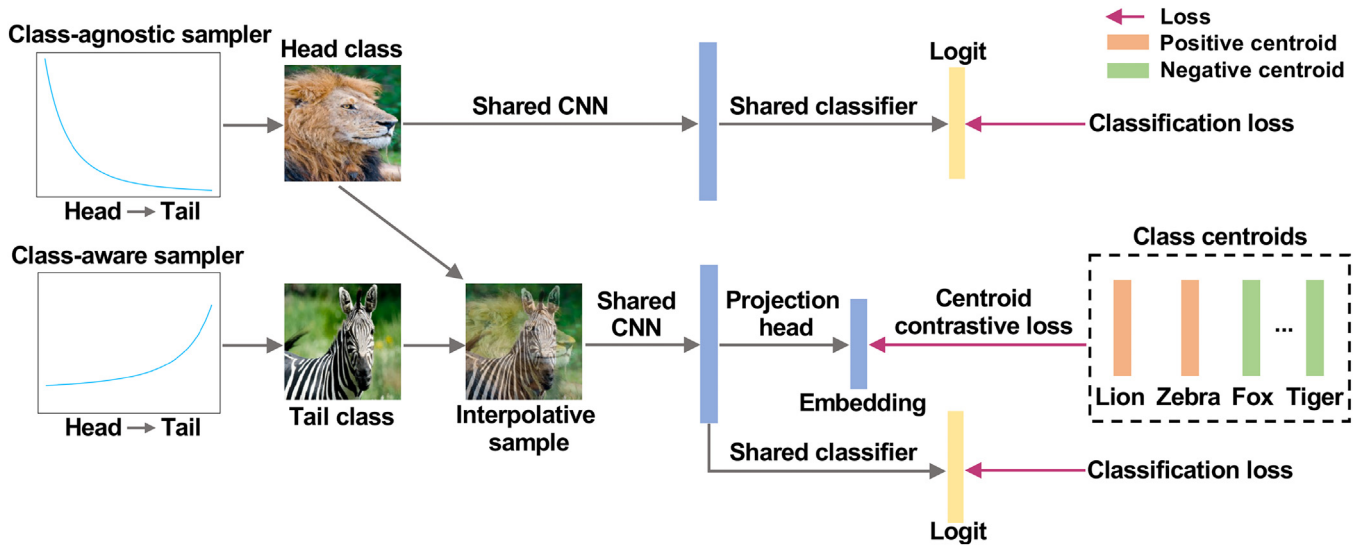


**Fig. 1.** ICCL framework. The uniform branch (top) focuses more on head-class samples and is trained with the standard cross entropy loss.The interpolative branch (bottom) focuses more on tail-class samples and is trained with an interpolative classification loss and an interpolative centroid contrastive loss. The model parameters are shared between the two branches.

tail-class poodle. Therefore, we adopt Mixup [14], which has the effect of creating a linearly interpolated feature space. The two samplers create an interpolation between a head and a tail. The interpolative sample also increases the tail-class training examples and provides additional supervision to learn tail-class representation. However, the model might be confused by the interpolative sample with either the head or the tail only. The proposed interpolative centroid contrastive loss encourages the tail centroids to be positioned discriminatively relative to the head classes by injecting class-balanced knowledge in the form of centroids. For the contrastive loss to work well, the head classes must be well positioned. Therefore, we adopt the regular classification loss (top branch in Figure 1) to perform representation learning for head classes. The interpolative classification loss (bottom branch) further improves the head-class alignment. These loss components reinforce one another.

We summarize our contribution as follows: (a) we introduce interpolative centroid contrastive learning for discriminative long-tailed representation learning. ICCL reduces intra-class variance and increases inter-class variance by optimizing the distance between sample embeddings and the class centroids; (b) ICCL improves tail-class representations by addressing class imbalance with class-aware sample interpolation and interpolative centroid loss; (c) ICCL achieves improved performance on multiple long-tailed recognition benchmarks. We also perform ablation studies to verify the effectiveness of each proposed component.

## 2. Related work

**Re-sampling.** Re-sampling aims to address the imbalance issue from the data level. Two main re-sampling approaches include oversampling and undersampling. Oversampling [6,15] increases the number of tail-class samples at the risk of overfitting the model, whereas undersampling might reduce the head classes diversity by decreasing their sample numbers [7]. Class-balanced sampling assigns equal sampling probability for all classes, and then selects their respective images uniformly [8].

**Re-weighting.** Re-weighting methods modify the loss function algorithmically to encourage larger gradient contribution or decision margin of tail classes [9–11]. Cui *et al.* [9] introduce class-balanced loss based on the effective class samples, which improves upon the approaches that assign class weight inversely proportional to their sample number [16]. Cao *et al.* [17] develop label dependent loss that promotes larger margins for tail class. Menon *et al.* [10] propose logit adjustment to softmax loss by considering the pairwise class relative margin.

Conventional re-sampling and re-weighting could affect the quality of the long-tailed representation [12,13]. Our method adopts two samplers, a class-agnostic and a class-aware, to create interpolative samples that address the imbalance issue during the representation learning.

**Decoupled strategy.** Several studies decouple the training into representation learning and classifier balancing [12,13,18]. Kang *et al.* [12] find that the long-tailed distribution has more negative impact on the classifier than the representation. They propose a two-stage strategy which firstly learns the representation in a class-agnostic manner, followed by rebalancing the classifier. BBN [13] introduces a re-balancing branch which focuses on tail classes and trains it with the conventional uniform branch simultaneously in a single-stage curriculum learning manner. Our method is based on the two-stage training approach. We show that long-tailed representation learning can be further improved by our proposed interpolative centroid contrastive learning even before balancing the classifier rebalancing.

**Contrastive learning.** Recently, contrastive learning approaches demonstrate strong performance in self-supervised representation learning [19–21]. Contrastive learning encourages the two augmented embeddings from the same image to have higher similarity in contrast to others. Several works [22–24] extend contrastive learning to long-tailed recognition. KCL [22] constructs the positive pairs by selecting the same number of examples for all classes in order to learn a balanced representation space. TSC [23] creates uniformly dispersed targets as anchors to align the feature of different classes. BCL [24] incorporates class semantics into the targets by using class prototypes. In contrast, ICCL operates on an interpolative sample consisting information of both classes. Our contrastive loss seeks to learn a representation for the interpolative sample, such that it can be used to retrieve the centroids for both source classes. The centroid retrieval is performed via non-parametric contrastive similarity matching in the low-dimensional space, thus it is different from Mixup [14] which operates on the parametric classifier.

**Mixup.** By performing convex combination of training samples, Mixup [14] regularizes the neural network. Chou *et al.* [25] pro-

pose larger mixing weight for tail classes to push the decision boundary towards the head classes. Ye *et al.* [26] balance the training by weakening the head-class learning through mixing their features with others. Several works apply Mixup to improve contrastive learning representation in *self-supervised* setting [27,28]. In contrast, our interpolative centroid contrastive loss is a new loss designed for *supervised* representation learning under long-tailed distribution.

## 3. Method

### 3.1. Overall framework

Our proposed long-tailed representation learning framework consists of a uniform and an interpolative branch as illustrated in Figure 1. The uniform branch follows the original long-tailed distribution to learn more generalizable representations from data-rich head-class samples, whereas the interpolative branch focuses more on modeling the tail-class to improve tail-class representation. Both branches share the same model parameters, which is different from BBN [13]. Our framework consists of: (a) a **CNN encoder** which transforms an image into a feature vector $\boldsymbol{g}_i \in \mathbb{R}^{d_g}$. The feature $\boldsymbol{g}_i$ is the output from the global average pooling layer; (b) a **MLP projection head** [19] which transforms the feature vector $\boldsymbol{g}_i$ into a low-dimensional normalized embedding $\boldsymbol{z}_i \in \mathbb{R}^{d_z}$; (c) a **linear classifier** which returns a class probability $\boldsymbol{p}_i$ given a feature vector $\boldsymbol{g}_i$; (d) **class centroids** $\boldsymbol{c}^k \in \mathbb{R}^{d_z \times K}$ which resides in the low-dimensional embedding space. Similar to MoPro [29], we compute the centroid of each class as the exponential-moving-average (EMA) of the low-dimensional embeddings for samples from that class. Specifically, the centroid for class $k$ is updated during training by:

$$\boldsymbol{c}^k \leftarrow m \cdot \boldsymbol{c}^k + (1-m) \sum_{k=1}^{K} \mathbb{1}_{y_i=k} \cdot \boldsymbol{z}_i, \tag{1}$$

where $m$ is the momentum coefficient; (e) a **class-agnostic and a class-aware sampler** which create interpolative samples.

### 3.2. Interpolative sample generation

We utilize two different samplers for interpolative sample generation: (a) a **class-agnostic sampler** which selects all samples with an equal probability regardless of the class, thus it returns more head-class samples. We denote a sample returned by the class-agnostic sampler as $(\boldsymbol{x}_i^h, y_i^h)$; (b) a **class-aware sampler** which focuses more on tail classes. It first samples a class and then select the corresponding samples uniformly with repetition. Let $n^k$ denotes the number of samples in class $k$, the probability $p(k)$ of sampling class $k$ is inversely proportional to $n^k$ as follows:

$$p(k) = \frac{(1/n^k)^\gamma}{\sum_{j=1}^{K}(1/n^j)^\gamma}, \tag{2}$$

where $\gamma$ is an adjustment parameter. When $\gamma = 0$, the class-aware sampler is equivalent to the balanced sampler in [8]. When $\gamma = 1$, it is the reverse sampler in [13]. We denote a sample returned by the class-aware sampler as $(\boldsymbol{x}_i^t, y_i^t)$.

An interpolative image $\boldsymbol{x}_i^f$ is formed by linearly combining two images from the class-agnostic and class-aware sampler, respectively.

$$\boldsymbol{x}_i^f = \lambda \boldsymbol{x}_i^h + (1-\lambda)\boldsymbol{x}_i^t, \tag{3}$$

where $\lambda \sim \mathcal{U}(0,1)$ is sampled from a uniform distribution. It is equivalent to the Beta$(\alpha,\alpha)$ used in Mixup [14] with $\alpha = 1$. Our contrastive learning trains the model such that the representation of the interpolative image is discriminative for both class $y_i^h$ and class $y_i^t$.

### 3.3. Interpolative centroid contrastive loss

Here we introduce the proposed interpolative centroid contrastive loss which aims to improve long-tailed representation learning. Given the low-dimensional embedding $\boldsymbol{z}_i^f$ for an interpolative sample $\boldsymbol{x}_i^f$, we use $\boldsymbol{z}_i^f$ to query the class centroids with contrastive similarity matching. Specifically, the probability that the $k$-th class centroid $\boldsymbol{c}^k$ is retrieved is given as:

$$p(\boldsymbol{c}^k|\boldsymbol{x}_i^f) = \frac{\exp(\boldsymbol{z}_i^f \cdot \boldsymbol{c}^k/\tau)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_i^f \cdot \boldsymbol{c}^j/\tau)}, \tag{4}$$

where $\tau$ is a scalar temperature parameter to scale the similarity. Equation 4 can be interpreted as a non-parametric classifier. Since the centroid is computed as the moving-average of $\boldsymbol{z}_i$, it does not suffer from the problem of weight imbalance as a parametric classifier does.

Since $\boldsymbol{x}_i^f$ is a linear interpolation of $\boldsymbol{x}_i^h$ and $\boldsymbol{x}_i^t$ (see Equation 3), our loss encourages the retrieval of the corresponding centroids of class $y_i^h$ and $y_i^t$. Thus, the interpolative centroid contrastive loss is defined as:

$$\mathcal{L}_{cc}^{it} = -\lambda \log(p(\boldsymbol{c}^{y_i^h}|\boldsymbol{x}_i^f)) - (1-\lambda)\log(p(\boldsymbol{c}^{y_i^t}|\boldsymbol{x}_i^f)). \tag{5}$$

The proposed centroid contrastive loss introduces valuable structural information into the embedding space. The numerator of $p(\boldsymbol{c}|\boldsymbol{x}_i^f)$ reduces the intra-class variance by pulling embeddings with the same class closer to the class centroid. The denominator of $p(\boldsymbol{c}|\boldsymbol{x}_i^f)$ increases the inter-class variance by pushing an embedding away from other classes' centroids. Therefore, more discriminative representations can be learned.

### 3.4. Overall loss

Given the classifier's output prediction probability $\boldsymbol{p}(\boldsymbol{x}_i^h)$ for an image $\boldsymbol{x}_i^h$, we define the classification loss on the uniform branch as the standard cross entropy loss:

$$\mathcal{L}_{ce} = -\log(p^{y_i^h}(\boldsymbol{x}_i^h)). \tag{6}$$

For an interpolative sample $\boldsymbol{x}_i^f$, the classification loss is

$$\mathcal{L}_{ce}^{it} = -\lambda \log(p_i^{y_i^h}(\boldsymbol{x}_i^f)) - (1-\lambda)\log(p_i^{y_i^t}(\boldsymbol{x}_i^f)). \tag{7}$$

During training, we jointly minimize the sum of losses on both branches:

$$\mathcal{L}_{total} = \sum_{i=1}^{n} \omega_u \mathcal{L}_{ce} + \omega_{it}(\mathcal{L}_{ce}^{it} + \mathcal{L}_{cc}^{it}), \tag{8}$$

where $\omega_u$ and $\omega_{it}$ are the weights for the uniform branch and the interpolative branch, respectively.

### 3.5. Classifier rebalancing

We rebalance our classifier after the representation learning stage. Specifically, we discard the projection head and fine-tune the linear classifier. The CNN encoder is either fixed or fine-tuned with a smaller learning rate. In order to rebalance the classifier towards tail classes, we employ our class-aware sampler. We denote the sampler's adjustment parameter as $\gamma'$. Due to more frequent sampling of tail-class samples, the classifier's logits distribution would shift towards the tail classes at the cost of lower accuracy on head classes. To control the trade-off between the head and tail, we introduce a distillation loss [36] using the classifier from the first stage as the teacher. The overall loss for classifier rebalancing consists of a cross-entropy classification loss and a KL-divergence distillation loss.

$$\mathcal{L}_{cb} = \sum_{i=1}^{n}(1-\omega_d)\mathcal{L}_{ce} + \omega_d \tau_d^2 \mathcal{L}_{KL}\big(\sigma(\boldsymbol{o}^T/\tau_d), \sigma(\boldsymbol{o}^S/\tau_d)\big), \tag{9}$$

where $\omega_d$ is the weight of the distillation loss, $\boldsymbol{o}^S$ and $\boldsymbol{o}^T$ are the class logits produced by the student classifier (2nd stage) and the teacher classifier (1st stage), respectively. $\tau_d$ is the distillation temperature and $\sigma$ is the softmax function.

For inference, we use a classification network consisting of the CNN encoder followed by the rebalanced classifier.

## 4. Experiments

### 4.1. Dataset and evaluation

We evaluate our method on three standard benchmark datasets for long-tail recognition as follows:

**CIFAR-LT.** CIFAR10-LT and CIFAR100-LT contain samples from the CIFAR10 and CIFAR100 [39] dataset, respectively. The class sampling frequency follows an exponential distribution. Following [13,17], we construct LT datasets with different imbalance ratios of 100, 50, and 10. Imbalance ratio is defined as the ratio of the maximum to the minimum class sampling frequency. The number of training images for CIFAR10-LT with an imbalance ratio of 100, 50 and 10 is 12k, 14k and 20k, respectively. Similarly, CIFAR100-LT has a training set size of 11k, 13k and 20k. Both test sets are balanced with the original size of 10k.

**ImageNet-LT.** The training set consists of 1000 classes with 116k images sampled from the ImageNet [1] dataset. Following [37], the class sampling frequency bases on a Pareto distribution with a shape parameter of 6. The imbalance ratio is 256. The validation set consists of 20k images. The test set is ImageNet original test set with a size of 50k.

**iNaturalist 2018.** It is a real-world long-tailed dataset for fine-grained image classification of 8,142 species [40]. We utilize the official training and test datasets composing of 438k training and 24k test images.

For all datasets, we evaluate our models on the test sets and report the overall top-1 accuracy across all classes. For CIFAR-LT which is a relatively small dataset, we average the accuracy over 3 trials. For ImageNet-LT and iNaturalist 2018, we perform a single run. To further access the model's accuracy on different classes, we group the classes into splits according to their number of images [12,37]: many ($>100$ images), medium ($20-100$ images) and few ($<20$ images) for ImageNet-LT and iNaturalist 2018.

### 4.2. Implementation details

For fair comparison, we follow the same training setup of previous works using SGD optimizer with a momentum of 0.9. For all experiments, we fix class centroid momentum coefficient $m=0.99$, class-aware sampler adjustment parameter $\gamma=0$, $\gamma'=1$, distillation weight $\omega_d=0.5$ and distillation temperature $\tau_d=10$. Unless otherwise specified, for the hyperparameters, we set branch weights $\omega_u=1$, $\omega_{it}=1$, temperature $\tau=0.07$ and projected embedding size $d_z=128$ in the representation learning stage. $m$, $\tau$ and $d_z$ are based on typical values suggested in the contrastive learning literature [19,20]. These hyperparameters are the same for ImageNet-LT and iNaturalist 2018, but $\tau$ and $d_z$ are different for CIFAR-LT due to smaller network architecture and image size. In the classifier balancing stage, we freeze the CNN and fine-tune the classifier using the original learning rate$\times$0.1 with cosine decay for 10 epochs.

We also design a warm-up training curriculum. Specifically, in the first $T$ epochs, we train only the uniform branch using the cross-entropy loss $\mathcal{L}_{ce}$ and a (non-interpolative) centroid contrastive loss $\mathcal{L}_{cc}=-\log(p(\boldsymbol{c}^{y_i^h}|\boldsymbol{x}_i^h))$. After $T$ epochs, we activate the interpolative branch and optimize $\mathcal{L}_{total}$ in Equation 8. The warm-up provides good initialization for the representations and the centroids. $T$ is scheduled to be approximately halfway through the total number of epochs.

**Table 1**
Top-1 accuracy on CIFAR100-LT and CIFAR10-LT using ResNet-32 CIFAR. Here, * denotes results from [13]. ** denotes results from [23]. † denotes our reproduced results based on [12] setting. ‡ denotes models trained longer (400 epochs) with stronger augmentation (Cutout and AutoAugment) than the others.

| Dataset | CIFAR100-LT | | | CIFAR10-LT | | |
|---|---|---|---|---|---|---|
| Imbalance ratio | 100 | 50 | 10 | 100 | 50 | 10 |
| CE* | 38.3 | 43.9 | 55.7 | 70.4 | 74.8 | 86.4 |
| Focal Loss* [30] | 38.4 | 44.3 | 55.8 | 70.4 | 76.7 | 86.7 |
| Mixup* [14] | 39.5 | 45.0 | 58.0 | 73.1 | 77.8 | 87.1 |
| Manifold Mixup* [31] | 38.3 | 43.1 | 56.6 | 73.0 | 78.0 | 87.0 |
| CB-Focal* [9] | 39.6 | 45.2 | 58.0 | 74.6 | 79.3 | 87.1 |
| CE-DRW* [17] | 41.5 | 45.3 | 58.1 | 76.3 | 80.0 | 87.6 |
| CE-DRS* [17] | 41.6 | 45.5 | 58.1 | 75.6 | 79.8 | 87.4 |
| LDAM-DRW* [17] | 42.0 | 46.6 | 58.7 | 77.0 | 81.0 | 88.2 |
| cRT† [12] | 42.3 | 46.8 | 58.1 | 75.7 | 80.4 | 88.3 |
| LWS† [12] | 42.3 | 46.4 | 58.1 | 73.0 | 78.5 | 87.7 |
| BBN [13] | 42.6 | 47.0 | 59.1 | 79.8 | 82.2 | 88.3 |
| KCL** [22] | 42.8 | 46.3 | 57.6 | 77.6 | 81.7 | 88.0 |
| M2m [32] | 43.5 | - | 57.6 | 79.1 | - | 87.5 |
| TSC [23] | 43.8 | 47.4 | 59.0 | 79.7 | 82.9 | 88.7 |
| Logit adjustment [10] | 43.9 | - | - | 77.7 | - | - |
| De-confound-TDE [33] | 44.1 | 50.3 | 59.6 | 80.6 | 83.6 | 88.5 |
| ResLT [34] | 45.3 | 50.0 | 60.8 | 80.4 | 83.5 | 89.1 |
| PaCo‡ [35] | 52.0 | 56.0 | 64.2 | - | - | - |
| ICCL (ours) | **46.6** | **51.6** | **62.1** | **82.1** | **84.7** | **89.7** |

**CIFAR-LT.** We use a ResNet-32 [41] and follow the training strategies in [13]. We train the model for 200 epochs, $32\times32$ image resolution, 128 batch size and 2e−4 weight decay. We use standard data augmentation which consists of random horizontal flip and cropping with a padding size of 4. The learning rate increases to 0.1 within the first 5 epochs and decays at epoch 120 and 160 with a step size of 0.01. We set $\tau=0.3$ and $T=80$ and 100 for CIFAR100-LT and CIFAR10-LT, respectively. $\omega_u$ is set as 0 after warm-up. $d_z$ is 32. In the classifier balancing stage, we fine-tune the CNN encoder using cosine scheduling with an initial learning rate of 0.01.

**ImageNet-LT.** We train a ResNeXt-50 [42] model for 90 epochs, $224\times224$ image resolution, 256 batch size, 5e−4 weight decay and 0.1 learning rate with cosine decay. Similar to [12], we augment the data using random horizontal flip, cropping and color jittering. We set $T=40$.

**iNaturalist 2018.** Following the training strategies in [12], we train a ResNet-50 model for 90 epochs and 200 epochs using 0.2learning rate with cosine decay, $224\times224$ image resolution, 512 batch size and 1e−4 weight decay. The data augmentation comprises of only horizontal flip and cropping. $T$ is set as 40 and 100 epochs for training epochs of 90 and 200, respectively.

### 4.3. Results

In this section, we present the results.The proposed ICCL achieves improved performance on all benchmarks across baseline methods except PaCo [35]. It is worth noting that PaCo utilizes stronger augmentation (RandAugment, Cutout and AutoAugment) and longer training (400 epochs) compared with ICCL and other methods.

**CIFAR-LT.** Table 1 demonstrates that ICCL surpasses existing methods across different imbalance ratios for both CIFAR100-LT and CIFAR10-LT. The performance of ICCL outperforms ResLT by 1.3% on the more challenging CIFAR100-LT with an imbalance ratio of 100.

**ImageNet-LT.** Table 2 presents the ImageNet-LT results, where ICCL outperforms the existing methods. For ImageNet-LT, we also propose an improved set of hyperparameters which increases the

**Table 2**
Top-1 accuracy on ImageNet-LT using ResNeXt-50. * denotes results from [33]. † denotes our reproduced results using improved settings. ‡ denotes models trained longer (400 epochs) with stronger augmentation (RandAugment) than the others. The trade-off between head class (i.e. many) and tail class (i.e. medium and few) accuracy is adjustable without affecting the overall accuracy (see Fig. 4).

| Method | Overall | Many | Medium | Few |
|---|---|---|---|---|
| OLTR* [37] | 41.9 | 51.0 | 40.8 | 20.8 |
| Focal Loss* [30] | 43.7 | 64.3 | 37.1 | 8.2 |
| NCM [12] | 47.3 | 56.6 | 45.3 | 28.1 |
| $\tau$-norm [12] | 49.4 | 59.1 | 46.9 | 30.7 |
| cRT [12] | 49.6 | 61.8 | 46.2 | 27.4 |
| LWS [12] | 49.9 | 60.2 | 47.2 | 30.3 |
| De-confound-TDE [33] | 51.8 | 62.7 | 48.8 | 31.6 |
| ResLT [34] | 52.9 | 63.0 | 50.5 | 35.5 |
| DisAlign [38] | 53.4 | 62.7 | 52.1 | 31.4 |
| PaCo‡ [35] | 58.2 | 67.5 | 56.9 | 36.7 |
| cRT† [12] | 52.4 | **64.3** | 49.1 | 30.7 |
| LWS† [12] | 52.5 | 63.0 | 49.6 | 32.8 |
| De-confound-TDE† [33] | 52.4 | 63.5 | 49.2 | 32.2 |
| ICCL (ours) | **54.0** | 60.7 | **52.9** | **39.0** |

**Table 3**
Top-1 accuracy on iNaturalist 2018 using ResNet-50 for 90 epochs and 200 epochs. * denotes results from [13] which uses 90 and 180 epochs. ‡ denotes models trained for 400 epochs. The trade-off between head class (i.e. many) and tail class (i.e. medium and few) is adjustable without affecting the overall accuracy (see Fig. 4).

| Method | 90 Epochs | | | | 200 Epochs | | | |
|---|---|---|---|---|---|---|---|---|
| | Overall | Many | Medium | Few | Overall | Many | Medium | Few |
| CB-Focal [9] | 61.1 | - | - | - | - | - | - | - |
| CE-DRS* [17] | 63.6 | - | - | - | - | - | - | - |
| CE-DRW* [17] | 63.7 | - | - | - | - | - | - | - |
| LDAM-DRW [17] | 68.0 | - | - | - | - | - | - | - |
| LDAM-DRW* [17] | 64.6 | - | - | - | 66.1 | - | - | - |
| NCM [12] | 58.2 | 55.5 | 57.9 | 59.3 | 63.1 | 61.0 | 63.5 | 63.3 |
| cRT [12] | 65.2 | 69.0 | 66.0 | 63.2 | 68.2 | **73.2** | 68.8 | 66.1 |
| $\tau$-norm [12] | 65.6 | 65.6 | 65.3 | 65.9 | 69.3 | 71.1 | 68.9 | 69.3 |
| LWS [12] | 65.9 | 65.0 | 66.3 | 65.5 | 69.5 | 71.0 | 69.8 | 68.8 |
| Logit adjustment [10] | 66.4 | - | - | - | - | - | - | - |
| BBN [13] | 66.4 | 49.4 | **70.8** | 65.3 | 69.7 | 61.7 | **73.6** | 66.9 |
| KCL [22] | 68.6 | - | - | - | - | - | - | - |
| DisAlign [38] | 69.5 | 61.6 | **70.8** | 69.9 | 70.2 | 68.0 | 71.3 | 69.4 |
| TSC [23] | 69.7 | **72.6** | 70.6 | 67.8 | - | - | - | - |
| ResLT [34] | - | - | - | - | 70.2 | 68.5 | 69.9 | 70.4 |
| PaCo‡ [35] | - | - | - | - | 73.0 | 69.5 | 73.4 | 73.0 |
| ICCL (ours) | **70.5** | 67.6 | 70.2 | **71.6** | **72.5** | 72.1 | 72.3 | **72.9** |

accuracy for existing methods. Specifically, different from the original hyperparameters used in [12], we use a smaller batch size of 256 and a learning rate of 0.1. Furthermore, we find it is better to use original learning rate $\times 0.1$ for classifier balancing. For fair comparison, we re-implement Decouple methods [12] and De-confound-TDE [33] using our settings and obtain better accuracy than those reported in the original papers. However, ICCL still achieves the best overall accuracy of 54.0% with noticeable accuracy gains on medium and few classes.

**iNaturalist 2018.** On the real-world large-scale iNaturalist 2018 dataset, ICCL achieves good improvements compared with existing methods as shown in Table 3. For 90 and 200 epochs, our method achieves the best overall accuracy of 70.5% and 72.5% respectively.

### 4.4. Ablation study

Here we perform extensive ablation study to examine the effect of each component and hyperparameters of ICCL, and provide analysis on what makes ICCL successful.

**Loss components.** For representation learning, ICCL introduces the interpolative centroid contrastive loss $\mathcal{L}_{cc}^{it}$ and the interpolative cross-entropy loss $\mathcal{L}_{ce}^{it}$ as shown in Equation 8. In Table 4, we evaluate the contribution of each loss components on the ImageNet-LT dataset. We consider many split as the head classes ($>100$ images per class), medium and few splits as the tail classes ($\leq 100$ images per class). We employ the same classifier balancing technique as described in Section 3.4. We observe that $\mathcal{L}_{ce}^{it}$ and $\mathcal{L}_{cc}^{it}$ improve the overall accuracy individually and collectively. By comparing with $\mathcal{L}_{ce}^{it}$ only, which is equivalent to Mixup [14], we demonstrate the superiority of our loss formulation. Additionally, having a warm-up stage before incorporating interpolative losses provides an extra accuracy boost, especially for the tail classes. This aligns with the observation in [17] which suggests that adopting a deferred schedule before re-sampling is better for representation learning.

**Table 4**
Ablation study on different components of ICCL on ImageNet-LT. Head denotes the many split, whereas tail includes the medium and few splits. The proposed $\mathcal{L}_{ce}^{it}$, $\mathcal{L}_{cc}^{it}$, and warm-up all contribute to accuracy improvement. Using only $\mathcal{L}_{ce}^{it}$ is equivalent to Mixup [14].

| $\mathcal{L}_{ce}$ | $\mathcal{L}_{ce}^{it}$ | $\mathcal{L}_{cc}^{it}$ | Warm-up | Overall | Head | Tail |
|---|---|---|---|---|---|---|
| ✓ | | | | 51.3 | 60.6 | 45.5 |
| | ✓ | | | 51.6 | 58.3 | 47.4 |
| | ✓ | | ✓ | 51.7 | 57.9 | 47.9 |
| ✓ | | ✓ | ✓ | 52.4 | 59.9 | 47.7 |
| ✓ | ✓ | | ✓ | 53.4 | 61.1 | 48.6 |
| ✓ | ✓ | ✓ | | 53.6 | 61.2 | 48.8 |
| ✓ | ✓ | ✓ | ✓ | 54.0 | 60.7 | 49.8 |

**Table 5**
Effect of sampling $\lambda$ from different Beta$(\alpha, \beta)$ distribution (see right figure) on CIFAR100-LT. $\lambda$ determines the weighting of the two samples for a given interpolative sample.

| Beta$(\alpha, \beta)$ | CIFAR100-LT |
|---|---|
| (0.2, 1.0) | 43.6 |
| (0.2, 0.2) | 43.8 |
| (0.6,0.6) | 45.4 |
| (1.0,1.0) | 46.6 |
| (2.0, 2.0) | 46.8 |

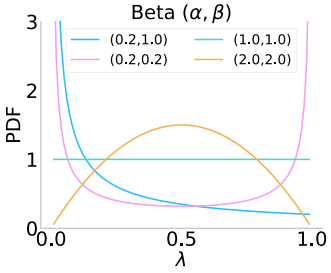


**Table 6**
Adjustment parameter $\gamma$ of the interpolative branch class-aware sampler. Focus excessively on head-class (uniform) or tail-class samples ($\gamma = 1$) leads to worse performance.

| Sampler | CIFAR100-LT | CIFAR10-LT | ImageNet-LT | iNaturalist |
|---|---|---|---|---|
| Uniform | 44.7 | 79.9 | 52.8 | 69.4 |
| $\gamma = 0$ | 46.6 | 82.1 | 54.0 | 70.5 |
| $\gamma = 0.5$ | 46.2 | 81.6 | 54.1 | 70.2 |
| $\gamma = 1.0$ | 46.2 | 81.1 | 53.1 | 70.1 |

**Interpolation weight** $\lambda$. In Equation 3, we sample the interpolation weight $\lambda \in [0, 1]$ from a uniform distribution, which is equivalent to Beta$(1, 1)$. We vary the beta distribution and study its effect on CIFAR100-LT with an imbalance ratio of 100. The resulting accuracy and the corresponding beta distribution are shown in Table 5. Sampling from Beta$(0.2, 1.0)$ is more likely to return a small $\lambda$, thus the interpolative samples contain more information about images from the class-aware sampler. As we fix $\alpha = \beta$ and increase them from 0.2 to 2, the accuracy increases. Good performance can be achieved with Beta$(1.0, 1.0)$ and Beta$(2.0, 2.0)$, where the sampled $\lambda$ is less likely to be an extreme value.

**Class-aware sampler adjustment parameter** $\gamma$. We further investigate the influence of $\gamma$ on representation learning. When $\gamma = 0$ and 1, the class-aware sampler is equivalent to class-balanced sampler [8] and reverse sampler [13] respectively. We include a class-agnostic uniform sampler as the baseline. Table 6 shows that the interpolative branch sampler should neither focus excessively on the tail classes ($\gamma = 1$) nor on the head classes (uniform). When using either of these two samplers, the resulting interpolative image might be less informative due to excessive repetition of tail-class samples or redundant head-class samples.

**Table 7**
Effect of classifier rebalancing on ImageNet-LT. ICCL learns better tail-class representation which leads to higher tail-class (i.e. medium and few) accuracy after classifier rebalancing.

| Method | Rebalancing | Overall | Many | Medium | Few |
|---|---|---|---|---|---|
| CE [12] | - | 46.7 | 68.1 | 40.2 | 9.0 |
| Mixup [14] | - | 49.5 | 64.1 | **44.8** | **24.4** |
| ICCL (ours) | - | **50.5** | **68.5** | 44.4 | 20.8 |
| CE [12] | ✓ | 52.4 | **64.3** | 49.1 | 30.7 |
| Mixup [14] | ✓ | 51.6 | 58.3 | 50.5 | 36.5 |
| ICCL (ours) | ✓ | **54.0** | 60.7 | **52.9** | **39.0** |

**Table 8**
Ablation study for classifier rebalancing parameters. ICCL benefits from using a reverse sampler ($\gamma' = 1$) and knowledge distillation ($\omega_d = 0.5$), especially for the more complex ImageNet-LT and iNaturalist datasets.

| $\gamma'$ | $\omega_d$ | CIFAR100-LT | CIFAR10-LT | iNaturalist | ImageNet-LT | |
|---|---|---|---|---|---|---|
| | | ICCL | ICCL | ICCL | ICCL | cRT |
| 0 | 0 | 45.3 | 77.5 | 69.5 | 53.7 | 52.4 |
| 0 | 0.5 | 45.0 | 77.6 | 69.5 | 53.2 | 52.2 |
| 1 | 0 | 47.1 | 82.3 | 70.2 | 53.6 | 49.6 |
| 1 | 0.5 | 46.6 | 82.1 | 70.5 | 54.0 | 51.3 |

**Rebalancing classifier.** In Table 7, we show the effect of classifier rebalancing, which improves the ICCL, CE [12] and Mixup [14] methods. By learning better tail-class representation, ICCL achieves higher overall accuracy compared to [12] and [14] both before and after classifier rebalancing.

**Classifier rebalancing parameters.** In the classifier rebalancing stage, we fix the sampler adjustment parameter $\gamma' = 1$, and the distillation weight $\omega_d = 0.5$. We study their effects in Table 8. For the ICCL approach, using a reverse sampler ($\gamma' = 1$) is better than a balanced sampler ($\gamma' = 0$). Further, the distillation loss tends to benefit more complex ImageNet-LT and iNaturalist than CIFAR-LT datasets. For the baseline cRT [12], applying the reverse sampler and distillation does not give accuracy improvement compared to the default setting (52.4).

**Alternative rebalancing methods.** In Table 9, we compare different rebalancing approaches for ICCL. Across all datasets, ICCL classifier rebalancing achieves better performance than naive class-balanced sampling [12] and post-hoc logit adjustment [10].

**Weight norm visualization.** The $L_2$ norms of the weights for the linear classification layer suggest how balanced the classifier is. A high weight norm for a particular class indicates that the classifier has a high preference for that class. Figure 2 depicts the weight norm of ICCL and cRT [12] after the representation learning and classifier balancing stage. In both stages, the weight norms of the ICCL classifier are more balanced than cRT. Furthermore, we plot

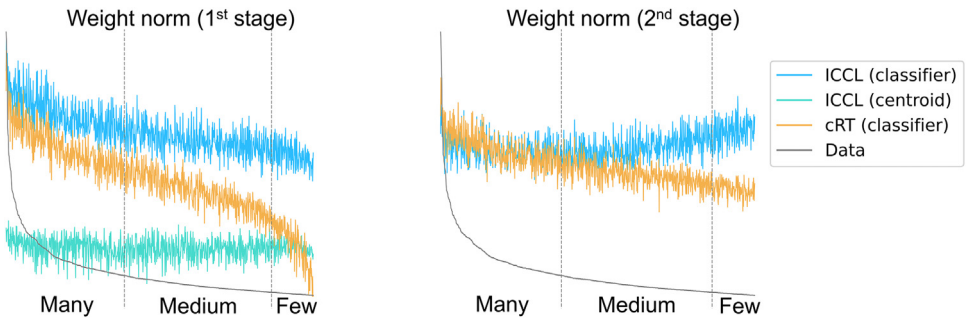


**Fig. 2.** Visualization of classifer's weight norm and centroid's norm of ICCL after the representation learning (left) and classifier balancing stage (right) on ImageNet-LT. Comparing with cRT [12], the weight norm of our ICCL classifier is more balanced. Additionally, the class centroids have intrinsically balanced norm.

**Table 9**
Comparison between ICCL classifier rebalancing with naive class-balanced sampling and post-hoc logit adjustment. ICCL classifier rebalancing is better than the alternatives.

| Method | CIFAR100-LT | CIFAR10-LT | iNaturalist | ImageNet-LT |
|---|---|---|---|---|
| Balanced sampling [12] | 45.3 | 77.5 | 69.5 | 53.7 |
| Logit adjustment [10] | 43.4 | 76.2 | 69.7 | 53.7 |
| ICCL (ours) | **46.6** | **82.1** | **70.5** | **54.0** |

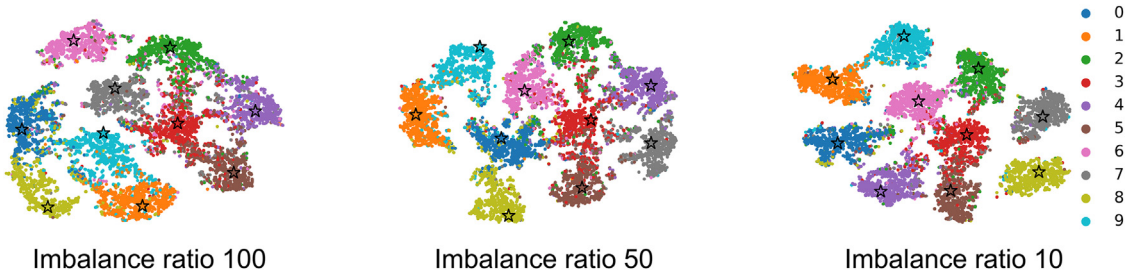


**Fig. 3.** Visualization of ICCL projected embeddings on CIFAR10-LT with an imbalance ratio of 100, 50 and 10. * denotes the class centroids. ICCL learns a representation that forms compact clusters.

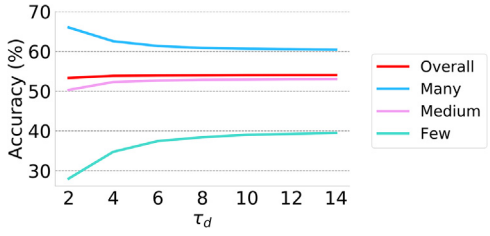


**Fig. 4.** Effect of distillation temperature $\tau_d$ on ImageNet-LT. ICCL's overall accuracy is not sensitive to $\tau_d$ variation.

the norms of our class centroids $\boldsymbol{c}^k$, which shows that the centroids are intrinsically balanced across different classes.

**Embedding visualization.** In Figure 3, we visualize the projected embeddings trained using ICCL on CIFAR10-LT with an imbalance ratio of 100, 50 and 10 using t-SNE. The class centroids facilitate the learning of a compact representation that is beneficial for good classification.

**Distillation loss.** In Table 10, we study the effectiveness of distillation loss in controlling the trade-off between different class splits during the classifier rebalancing stage. For ImageNet-LT and iNaturalist, the distillation loss improves the many class accuracy and achieves better overall accuracy.

**Distillation temperature** $\tau_d$. In Figure 4, we study how $\tau_d$ affects the accuracy of ICCL on ImageNet-LT. We find that the overall accuracy is not sensitive to changes in $\tau_d$. As $\tau_d$ increases, the teacher's logit distribution becomes more flattened. Therefore, the accuracy for medium and few class improves, whereas the accuracy for many class decreases.

**Table 10**
Effect of distillation loss on class split accuracy on ImageNet-LT and iNaturalist 2018. Distillation loss improves the many class accuracy and achieves better overall accuracy.

| Dataset | Distillation loss | Overall | Many | Medium | Few |
|---|---|---|---|---|---|
| ImageNet-LT | - | 53.6 | 57.5 | 53.5 | 43.1 |
| | ✓ | 54.0 | 60.7 | 52.9 | 39.0 |
| iNaturalist | - | 70.2 | 57.5 | 70.7 | 72.9 |
| | ✓ | 70.5 | 67.6 | 70.2 | 71.6 |

## 5. Conclusion

In this work, we propose an interpolative centroid contrastive learning technique for long-tailed representation learning. By utilizing class centroids and interpolative losses, we strengthen the discriminative power of the learned representations, leading to improved classification accuracy. We demonstrate the effectiveness of our approach with improvements on multiple long-tailed classification benchmarks. Specifically, on the real-world large-scale iNaturalist 2018, ICCL achieves good improvement over competing works. Through extensive ablation studies, we verify and provide insights into the design choices of our framework. For ICCL and existing methods, we observe that an increase in tail-class accuracy often leads to an undesirable drop in head-class accuracy. For future works, we aim to develop methods that can simultaneously improve all classes.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

Anthony Meng Huat Tiong is supported by Salesforce and the Singapore Economic Development Board under the Industrial Postgraduate Programme. Boyang Li is supported by the Nanyang Associate Professorship and the National Research Foundation Fellowship (NRF-NRFF13-2021-0006), Singapore. Any opinions, findings, conclusions, orrecommendations expressed in this material arethose of the authors and do not reflect the views of the funding agencies.

## References

[1] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: A large-scale hierarchical image database, CVPR, 2009.

[2] H. Li, Y. Liu, H. Zhang, B. Li, Evaluating and Mitigating Static Bias of Action Representations in the Background and the Foreground, arXiv Preprint, arXiv:2211.12883. 2022.
[3] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked Autoencoders Are Scalable Vision Learners, CVPR, 2022.
[4] C. Liu, H. Yu, B. Li, Z. Shen, Z. Gao, P. Ren, X. Xie, L. Cui, C. Miao, Noise-resistant Deep Metric Learning with Ranking-based Instance Selection, CVPR (2021).
[5] D.M. Powers, Applications and explanations of zipfs law, NEMLAP-CONLL, 1998.
[6] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, JAIR 16 (2002).
[7] C. Drummond, R.C. Holte, et al., C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, Workshop on learning from imbalanced datasets II, volume 11, Citeseer, 2003.
[8] L. Shen, Z. Lin, Q. Huang, Relay backpropagation for effective learning of deep convolutional neural networks, ECCV, Springer, 2016.
[9] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, CVPR, 2019.
[10] A.K. Menon, S. Jayasumana, A.S. Rawat, H. Jain, A. Veit, S. Kumar, Long-tail learning via logit adjustment, ICLR (2021).
[11] Y. Kim, Y. Lee, M. Jeon, Imbalanced image classification with complement cross entropy, Pattern Recognit. Lett. 151 (2021).
[12] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, ICLR, 2020.
[13] B. Zhou, Q. Cui, X.-S. Wei, Z.-M. Chen, BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition (2020).
[14] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, ICLR, 2018.
[15] Y.-g. Kim, Y. Kwon, M.C. Paik, Valid oversampling schemes to handle imbalance, Pattern Recognit. Lett. 125 (2019) 661–667.
[16] C. Huang, Y. Li, C.C. Loy, X. Tang, Learning deep representation for imbalanced classification, CVPR, 2016.
[17] K. Cao, C. Wei, A. Gaidon, N. Arechiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, NeurIPS, 2019.
[18] Y. Ma, M. Kan, S. Shan, X. Chen, Learning deep face representation with long-tail data: An aggregate-and-disperse approach, Pattern Recognit. Lett. 133 (2020).
[19] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, ICML, 2020.
[20] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, CVPR, 2020.
[21] J. Li, P. Zhou, C. Xiong, S.C. Hoi, Prototypical contrastive learning of unsupervised representations, ICLR, 2021.
[22] B. Kang, Y. Li, S. Xie, Z. Yuan, J. Feng, Exploring balanced feature spaces for representation learning, ICLR, 2020.
[23] T. Li, P. Cao, Y. Yuan, L. Fan, Y. Yang, R.S. Feris, P. Indyk, D. Katabi, Targeted supervised contrastive learning for long-tailed recognition, CVPR, 2022.
[24] J. Zhu, Z. Wang, J. Chen, Y.-P.P. Chen, Y.-G. Jiang, Balanced contrastive learning for long-tailed visual recognition, CVPR, 2022.
[25] H.-P. Chou, S.-C. Chang, J.-Y. Pan, W. Wei, D.-C. Juan, Remix: Rebalanced mixup, ECCV, Springer, 2020.
[26] H.-J. Ye, D.-C. Zhan, W.-L. Chao, Procrustean training for imbalanced deep learning, ICCV, 2021.
[27] V. Verma, M.-T. Luong, K. Kawaguchi, H. Pham, Q.V. Le, Towards domain-agnostic contrastive learning, ICML, 2021.
[28] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, H. Lee, i-mix: A domain-agnostic strategy for contrastive representation learning, ICLR, 2021.
[29] J. Li, C. Xiong, S.C. Hoi, Mopro: Webly supervised learning with momentum prototypes, 2021.
[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, ICCV, 2017.
[31] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, Y. Bengio, Manifold mixup: Better representations by interpolating hidden states, ICML, PMLR, 2019.
[32] J. Kim, J. Jeong, J. Shin, M2m: Imbalanced classification via major-to-minor translation, CVPR, 2020.
[33] K. Tang, J. Huang, H. Zhang, Long-tailed classification by keeping the good and removing the bad momentum causal effect, NeurIPS, 2020.
[34] J. Cui, S. Liu, Z. Tian, Z. Zhong, J. Jia, Reslt: Residual learning for long-tailed recognition, TPAMI (2022).
[35] J. Cui, Z. Zhong, S. Liu, B. Yu, J. Jia, Parametric contrastive learning, ICCV, 2021.
[36] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, NeurIPS Workshop, 2014.
[37] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, S.X. Yu, Large-scale long-tailed recognition in an open world, CVPR, 2019.
[38] S. Zhang, Z. Li, S. Yan, X. He, J. Sun, Distribution alignment: A unified framework for long-tail visual recognition, CVPR, 2021.
[39] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Technical Report, 2009.
[40] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, CVPR, 2018.
[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[42] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, CVPR, 2017.