Meta-Prototype Decoupled Training for Long-tailed Learning

Siming Fu[1], Huanpeng Chu[1], Xiaoxuan He[1], Hualiang Wang[1], Zhenyu Yang[2], and Haoji Hu[1⋆]

[1]College of Information Science and Electronic Engineering, Zhejiang University, China [2]Shenzhen TP-LINK Digital Technology Co., Ltd.
{fusimingchuhp,Xiaoxiao_He,hualiang_wang,haoji_hu}@zju.edu.cn,
yangzhenyu@tp-link.com.cn

Abstract. Long-tailed learning aims to tackle the crucial challenge that head classes dominate the training procedure under severe class imbalance in real-world scenarios. Supervised contrastive learning has turned out to be worth exploring research direction, which seeks to learn class-specific feature prototypes to enhance long-tailed learning performance. However, little attention has been paid to how to calibrate the empirical prototypes which are severely biased due to the scarce data in tail classes. Without the aid of correct prototypes, these explorations have not shown the significant promise expected. Motivated by this, we propose the meta-prototype contrastive learning to automatically learn the reliable representativeness of prototypes and more discriminative feature space via a meta-learning manner. In addition, on top of the calibrated prototypes, we leverage it to replace the mean of class statistics and predict the targeted distribution of balanced training data. By this procedure, we formulate the feature augmentation algorithm which samples additional features from the predicted distribution and further balances the over-whelming dominance severity of head classes. We summarize the above two stages as the meta-prototype decouple training scheme and conduct a series of experiments to validate the effectiveness of the framework. Our method outperforms previous work with a large margin and achieves state-of-the-art performance on long-tailed image classification and semantic segmentation tasks (e.g., we achieve 55.1% overall accuracy with ResNetXt-50 in ImageNet-LT).

Keywords: Meta-prototype · Decoupled training · Supervised contrastive learning · Feature augmentation.

1 Introduction

Most real-world data comes with a long-tailed nature: a few head classes contribute the majority of data, while most tail classes comprise relatively few data. An undesired phenomenon is models [31, 2] trained with long-tailed data perform better on head classes while exhibiting extremely low accuracy on tail ones.

⋆ Corresponding author

To address this problem, a large number of studies have been conducted in recent years, making promising progress in the field of deep long-tailed learning. Supervised contrastive learning (SCL) has been the main focus of many techniques for long-tailed learning. The mainstream insights work on supervised contrastive learning methods [17, 45] which seek to learn class-specific feature prototypes to enhance long-tailed learning performance. DRO-LT [21] innovatively explores the idea of feature prototypes to handle long-tailed recognition in an open world. Following that, TSC [15] converges the different classes of features to a target that is uniformly distributed over the hyper-sphere during training.

Nevertheless, when a class has only few samples, the distribution of training samples may not represent well the true distribution of the data. The shift between test distribution and training distribution causes the offset of the prototypes in tail classes [21]. The above works are all based on the empirical prototype under imbalanced data, limiting the effectiveness of feature representation. Therefore, the sub-optimal prototypes become an issue in learning high-quality representations for SCL methods, which confuse optimization for improved long-tailed learning.

To alleviate the above issues, we propose the supervised meta-prototype contrastive learning which calibrates the empirical prototype under the imbalanced setting. Specifically, we extend meta-learner to automatically restore the meta-prototypes of feature embeddings via two nested loops of optimization, guaranteeing the efficiency of the meta-prototype contrastive learning algorithm. Our major insight here is to parameterize the mapping function as a meta-network, which is theoretically a universal approximator for almost all continuous functions, and then use the meta-data (a small unbiased validation set) to guide the training of all the meta-network parameters. The meta-prototypes provide more meaningful feature prototypes which are designed to be robust against possible shifts of the test distribution and guide the SCL to obtain the discriminative feature representation space.

To further ease the dominance of the head classes in classification decisions, we develop the calibration feature augmentation algorithm based on the learned meta-prototype in classifier training stage. Specifically, we utilize it as the 'anchor' of corresponding class which represents the mean of the class statistics under the imbalanced setting. In contrast to the typical methods which generate the new feature samples based on the class statistics of imbalanced training data, our meta-prototype calibrates the bias and provides the reasonable feature distribution of new feature samples for tail classes. The newly generated feature are sampled from the calibrated distribution and help to find the correct classifier decision boundary via improving the performance of severely under-represented tail classes.

We summarize the above processes as the meta-prototype decoupled training framework which includes calibrating the empirical prototype for SCL in the representation learning stage and enhancing feature embedding for tail classes based on learned meta-prototype in the classifier learning stage. We extensively

validate our model on typical visual recognition tasks, including image classification on three benchmarks (CIFAR-100-LT [12], ImageNet-LT [18] and iNaturalist2018 [27]), semantic segmentation on ADE20K dataset [42]. The experimental results demonstrate our method consistently outperforms the state-of-the-art approaches on all the benchmarks.

Summary of Contributions:

- To the best of our acknowledge, we are the first in long-tailed learning to complete the meta-prototype to promote the representation quality of supervised prototype contrastive learning in the representation learning stage.
- On top of the learned meta-prototype, we develop the feature augmentation algorithm for tail classes to ease dominance of the head classes in classification decisions in the classifier learning stage.
- Our method outperforms previous works with a large margin and achieve state-of-the-art performance on long-tailed image classification and semantic segmentation tasks.

2 RELATED WORK

Supervised Contrastive Learning. Existing supervised contrastive learning-based methods for long-tailed learning seek to help alleviate the biased label effect. DRO-LT [22] extends standard contrastive loss and optimizes against the worst possible centroids within a safety hyper ball around the empirical centroid. KCL [10] develops a new method to explicitly pursue balanced feature space for representation learning. TSC [15] generates a set of targets uniformly distributed on a hypersphere and makes the features of different classes converge to these distinct and uniformly distributed targets during training. Hybrid-SC [30] explores the effectiveness of supervised contrastive learning. It introduces prototypical supervised learning to obtain better features and resolve the memory bottleneck. The above works are all based on the empirical prototype under imbalanced data, which limits the effectiveness of feature representation. To alleviate the above issue, we introduce the meta-prototype to calibrate the empirical prototype, further constructing a discriminative feature space.

Meta-learning. The recent development of meta-learning [7, 1] inspires researchers to leverage meta-learning to handle class imbalance. Meta-weight-net [24] introduces a method capable of adaptively learning an explicit weighting function directly from data. MetaSAug [14] proposes to augment tail classes with a variant of ISDA [32] by estimating the covariance matrices for tail classes. Motivated by these works, our method attempts to automatically estimate the meta-prototype of each class to calibrate the empirical prototype for high-quality feature representation.

Data Augmentation for Long-tailed Learning. In long-tail learning, transfer-based augmentation has been explored. Transfer-based augmentation seeks to transfer the knowledge from head classes to augment model performance on tail classes. TailCalibX [28] and GLAG [40] explore a direction that attempts to generate meaningful features by estimating the tail category's distribution.

RSG [29] dynamically estimates a set of feature centers for each class, and uses the feature displacement between head-class sample features and the nearest intra-class feature center to augment each tail sample feature. However, the estimated distribution of tail category and the intra-class feature center are biased or unreasonable due to the imbalanced size of training dataset. Our meta-prototype feature augmentation algorithm calibrates the bias and predicts likely shifts of the test distribution.

Decoupled Scheme for Long-tailed Learning. Decoupling [9] is a pioneering work that introduces a two-step training scheme. It empirically evaluates different sampling strategies for representation learning in the first step, and then evaluates different classifier training schemes by fixing the feature extractor trained in the second step. Decouple [9] and Bag of tricks [39] decouple the learning procedure into representation learning and classification, and systematically explore how different balancing strategies affect them for long-tailed recognition. BBN [43] further unifies the two stages to form a cumulative learning strategy. MiSLAS [41] proposes to enhance the representation learning with data mixup in the first stage. During the second stage, MiSLAS applies a label-aware smoothing strategy for better model generalization. In our paper, our method also adopts the two-stage decoupled training scheme, which leads to better long-tailed learning performance.

3 The proposed methods

3.1 Problem Definition

For long-tailed learning, considering $\mathcal{D}^{tra} = \{\boldsymbol{x}^i, y^i\}$, $i \in \{1, \cdots, K\}$ be the training set, where $\boldsymbol{x}^i$ denotes an image sample and y^i indicates its class label. Let K be the total numbers of classes, N_i be the number of samples in class i, where $\sum_{i=1}^{K} N_i = N$. A long-tail setup can be defined by ordering the number of samples per category, i.e. $N_1 \geq N_2 \geq \ldots \geq N_K$ and $N_1 \gg N_K$ after sorting of N_i. Under the long-tailed setting, the training dataset is imbalanced, leading to the poor performance on tail classes.

We train a network $\boldsymbol{\Psi}(\cdot; \boldsymbol{W})$ consisting of two components: (i) a backbone or representation network (CNN for images) that translates an image to a feature representation $\boldsymbol{z}_i = \boldsymbol{\Psi}(\boldsymbol{x}^i; \boldsymbol{w}^{\boldsymbol{E}}) \in \mathbb{R}^{1\times d}$ and (ii) a classifier $\boldsymbol{w}^C \in \mathbb{R}^{K\times d}$ at predicts the category specific scores (logits). As shown in Fig. 1, given a pair $(\boldsymbol{x}^i, y^i)$ sampled from a mini-batch $\mathcal{B} \subset \mathcal{D}^{tra}$, feature vector $\boldsymbol{z}_i$ is extracted by the feature extractor. $\boldsymbol{z}_i$ is projected onto the classifier to output the classification logit. Too few samples belonging to the tail classes result in inadequate learning of tail classes representations.

3.2 Supervised Meta-Prototype Contrastive Learning in The Representation Learning Stage

Supervised contrastive learning introduces cluster-based prototypes and encourages embeddings to gather around their corresponding prototypes. Our origi-

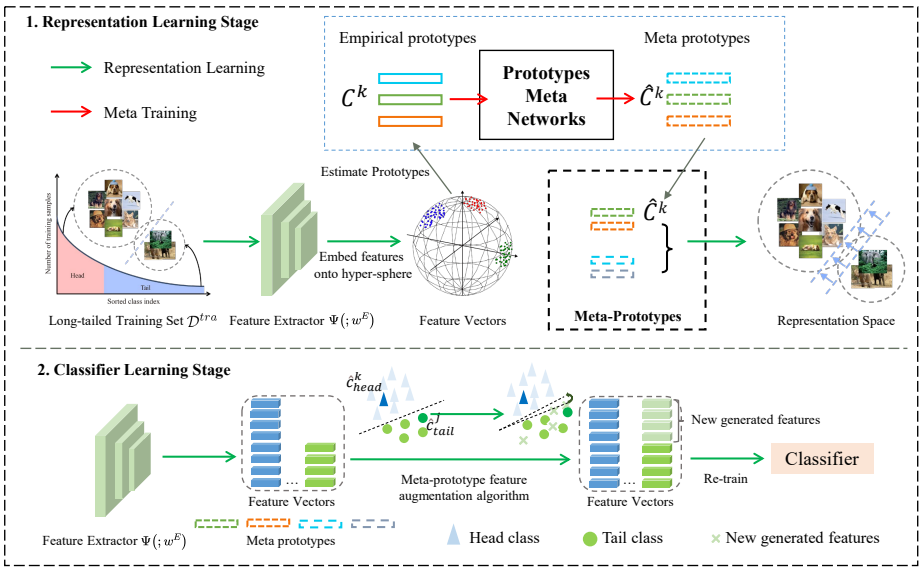


Fig. 1. Overview of our proposed method during the training period. **Upper box** introduces the meta-prototype, which consists of the following steps in sequence: sampling a mini-batch images $\mathcal{B}$ from training set $\mathcal{D}^{tra}$, learning features by the feature extractor $\boldsymbol{\Psi}(\cdot;\boldsymbol{w}^{\boldsymbol{E}})$, embedding features onto the hyper-sphere, estimating the prototypes for classes, and learning meta-prototypes for discriminative representation space. **Bottom box** introduces the meta-prototype feature augmentation algorithm which enriches the samples of tail classes to re-build the classifier decision boundaries.

nal feature prototypes follow the MoPro [13], adopting the exponential-moving-average (EMA) algorithm during training by:

$$\boldsymbol{c}_k \leftarrow m\boldsymbol{c}_k + (1-m)\boldsymbol{z}_i, \quad \forall i \in \{i \mid \hat{y}_i = k\}, \tag{1}$$

where $\boldsymbol{c}_k$ is the prototype for class k and m is momentum coefficient, usually set as 0.999. Then given the embedding z_i^f, the prototypes are queried with contrastive similarity matching. The prototype contrastive loss [13, 23] is defined as:

$$\mathcal{L}_{\mathrm{PC}} = -\log\left[\frac{\exp\left(z_i^f \cdot c^k/\tau\right)}{\sum_{j=1}^{K}\exp\left(z_i^f \cdot c^j/\tau\right)}\right], \tag{2}$$

where τ is a hyper-parameter and usually set as 0.07 [11]. The neural network is denoted as $f(\cdot, \mathbf{W})$, and $\mathbf{W}$ denotes all of its parameters. Generally, the optimal network parameter $\mathbf{W}^*$ can be extracted by minimizing the training loss:

$$\mathcal{L}^{\mathrm{train}}(\mathbf{W}; \boldsymbol{c}^k) = \mathcal{L}_{CE}(\mathbf{W}) + \lambda \cdot \mathcal{L}_{\mathrm{PC}}(\mathbf{W}, \boldsymbol{c}^k), \tag{3}$$

where λ denotes the weighting coefficient to balance the two loss terms and $\mathcal{L}_{CE}$ is the cross-entropy loss. As aforementioned, the empirical prototypes of

tail classes can be far away from the ground-truth prototypes due to the limited features of tail classes and large variances in data distribution between training and test datasets. Therefore, we aim to learn appropriate feature prototypes to perform reasonable feature representation learning.

The whole process of the meta-prototype constrastive learning is summarized in Algorithm 1. In the presence of imbalanced training data, our method calibrates the empirical prototypes by prototype meta network, denoted as $\mathcal{C}(\boldsymbol{c}^k;\Theta)$. $\boldsymbol{c}^k$ is the input of the meta network and Θ represents the parameters contained in it. The meta network consists of MLP, which maps the empirical prototype $\boldsymbol{c}^k$ into the meta-prototype $\hat{\boldsymbol{c}}^k$. The prototype meta network is an encoder-decoder network, where the encoder contains one linear layer with a ReLU activation function, and the decoder consists of a Linear-ReLU-Linear structure. The optimal parameter $\mathbf{w}$ is calculated by minimizing the following training loss:

$$\begin{aligned}\mathbf{W}^*(\Theta) &= \arg\min_{\mathbf{W}} \mathcal{L}^{\text{train}}(\mathbf{W};\boldsymbol{c}^k;\Theta) \\ &= \arg\min_{\mathbf{W}} \left\{\mathcal{L}_{CE}(\mathbf{W}) + \lambda \cdot \mathcal{L}_{\text{PC}}(\mathbf{W}, \mathcal{C}(\boldsymbol{c}^k;\Theta))\right\}.\end{aligned} \tag{4}$$

The parameters contained in the meta-network can be optimized by using the meta-learning idea. The optimal parameter Θ^* can be obtained by minimizing the following meta-loss:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}^{meta}\left(\mathbf{W}^*(\Theta)\right), \tag{5}$$

where $\mathcal{L}^{\text{meta}}(\mathbf{w}) = \mathcal{L}_{CE}\left(y_i^{(\text{meta})}, f\left(x_i^{(\text{meta})}, \mathbf{W}\right)\right)$ on meta-data. Specifically, following the meta-learning methods [14, 24] for long-tailed learning, we conduct a small amount of balanced meta-data set (i.e., with balanced data distribution) $\left\{x_i^{(\text{meta})}, y_i^{(\text{meta})}\right\}_{i=1}^{M}$ to represent the meta-knowledge of ground-truth sample-label distribution, where M is the number of meta-samples and $M \ll N$.

Online approximation. To estimate the optimal feature prototypes for different classes, we adopt a double optimization loop, respectively, to guarantee the efficiency of the algorithm. We optimize the model in a meta-learning setup by i). updating equation of the network parameter can be formulated by moving the current $\mathbf{W}^{(t)}$ along the descent direction of the objective loss in Eq. 4 on a mini-batch training data by

$$\hat{\mathbf{W}}^{(t)}(\Theta) \leftarrow \mathbf{W}^{(t)} - \alpha \times \nabla_{\mathbf{W}^{(t)}} \mathcal{L}^{\text{train}}(\mathbf{W};\boldsymbol{c}^k;\Theta), \tag{6}$$

where α is the step size. ii). After receiving the updated network parameters $\hat{\mathbf{W}}^{(t)}(\Theta)$, the parameter Θ of the meta-network can then be readily updated by Eq. 5, i.e., moving the current parameter $\Theta^{(t)}$ along the objective gradient to be calculated on the meta-data by

$$\Theta^{(t+1)} = \Theta^{(t)} - \beta \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta^{(t)}} \mathcal{L}^{meta}\left(\hat{\mathbf{W}}^{(t)}(\Theta)\right), \tag{7}$$

Algorithm 1 The Meta-Prototype Contrastive Learning Algorithm.

Input: Training data $\mathcal{D}^{tra}$, meta-data set $\mathcal{D}^{meta}$, batch size n, m, max epochs T, epoch threshold T_{th}.
Output: Network parameter $\mathbf{W}^{(T)}$, meta-network parameter $\Theta^{(T)}$.

for epoch $= 0 : \mathrm{T_{th}} - 1$ **do**
  Update $\mathbf{W}$ by $\mathcal{L}_{CE}$.
  Update $\boldsymbol{c}_k$ by Eq. 1.
end for
Initialize meta network parameters $\Theta^{(0)}$.
for epoch $= \mathrm{T_{th}} : \mathrm{T} - 1$ **do**
  $\{\ \boldsymbol{x}^i,\ y^i\ \} \leftarrow$ SampleMiniBatch (D^{tra}, n).
  $\{\boldsymbol{x}^{(\mathrm{meta}\ )}, y^{(\mathrm{meta}\ )}\} \leftarrow$ SampleMiniBatch (D^{meta}, m).
  Formulate the network learning function $\hat{\mathbf{W}}^{(t)}(\Theta)$ by Eq. 6.
  Update $\Theta^{(t+1)}$ by Eq 7.
  Update $\mathbf{W}^{(t+1)}$ by Eq 8.
end for

where β is the step size. iii) Then, the updated $\Theta^{(t+1)}$ is employed to ameliorate the parameter $\mathbf{W}$ of the network, constituting a complete loop:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \alpha \times \nabla_{\mathbf{W}^{(t)}} \mathcal{L}^{\text{train}}\ (\mathbf{W}^{(t)}; \boldsymbol{c}^k; \Theta^{(t+1)}), \tag{8}$$

Since the updated meta-network $\mathcal{C}(\boldsymbol{c}^k; \Theta^{(t+1)})$ are learned from balanced meta-data, we could expect $\mathcal{C}(\boldsymbol{c}^k; \Theta^{(t+1)})$ contribute to learning better network parameters $\mathbf{W}^{(t+1)}$.

3.3 Meta-Prototype Feature Augmentation in The Classifier Training Stage

On the classifier training phase, the target of our work is to generate addition feature embeddings to further balance the over-whelming dominance severity of head classes in the representation space. It is natural to utilize the feature augmentation to calibrate the ill-defined decision boundary. Following the Joint Bayesian face model [3], typical feature augmentation methods [28, 40, 36] assume that the features $\mathbf{z}_i$ lies in a Gaussian distribution with a class mean μ_i and a covariance matrix Σ_i. The mean of a class is estimated as the arithmetic mean of all features in the same class by $\mu_k = \frac{1}{N_k} \sum_{i \in \mathcal{F}_k} \mathbf{z}_i$.

However, the means of Gaussian distribution for tail classes are biased due to sparse sample size of the tail categories and large variances for data distribution between train and test datasets. This bias causes the distribution of the generated data to deviate significantly from the data distribution of the validation set. It leads to significant performance drop, even the destruction of the original representational space. Therefore, as Fig. 2 illustrated, we leverage the meta-prototypes $\hat{c}_i$ as the 'anchor' to replace the typical class statistics μ_k to provide the reasonable feature distribution of new feature samples for tail classes.

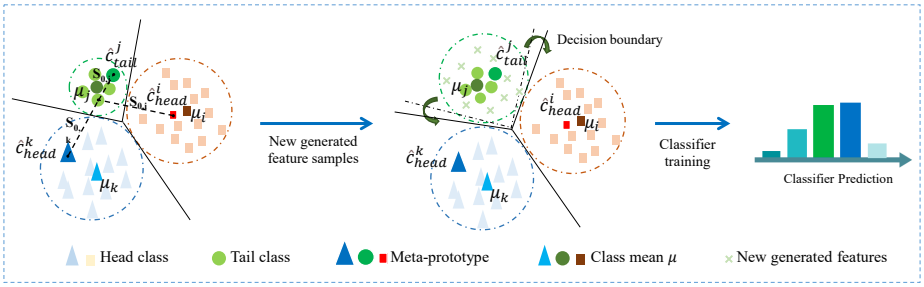


Fig. 2. Illustration of the feature augmentation process based on the learned meta-prototype $\hat{c}$. Tukey's Ladder of Power transformation function transfers the feature instance $\mathbf{z}_i$ into $\tilde{\mathbf{z}}_i$. Meta-prototypes replace the means μ of class statistics to calculate the neighbors $\mathcal{N}_i$ via $S_{i,k}$ and the calibrated distribution $\mu_{\tilde{\mathbf{z}}_i}$ and $\Sigma_{\tilde{\mathbf{z}}_i}$. Additional features for tail classes are sampled from the calibrated statistics so as to ease the dominance of the head classes in classification decisions.

Given a trained backbone (discussed in Sec. 3.2), we first pre-compute feature representations for the entire dataset. These features of true samples are denoted as $\mathcal{F} = \{\mathbf{z}_i\}_{i=1}^{N}$. $\mathcal{F}_k$ denotes features of images belonging to the category k. For each class, we sample $N_1 - N_K$ additional features, such that the resulting feature dataset is completely balanced and all classes have N_1 instances. Sampling is performed based on an instance specific calibrated distribution. Specifically, each $\mathbf{z}_{ik}$ (i^{th} feature from category k) is responsible for generating $s_{\text{new}} = \max\left\{[N_1/N_k - 1]_+, 1\right\}$ features, where $[\cdot]_+$ is the ceiling function.

Based on the learned meta-prototype, the features covariance for the corresponding class can be calculated as:

$$\Sigma_k = \frac{1}{N_k - 1} \sum_{i \in \mathcal{F}_k} \left(\mathbf{z}_i - \hat{\boldsymbol{c}}^k\right)\left(\mathbf{z}_i - \hat{\boldsymbol{c}}^k\right)^T, \tag{9}$$

where $\Sigma_k \in \mathbb{R}^{d \times d}$ denotes the full covariance of the Gaussian distribution for category k. Next, for each feature $\tilde{\mathbf{z}}_i$ belonging to tail classes k processed by Tukey's Ladder of Power transformation [26], we calculate the similarity degree with other classes k which have more training samples as $S_{i,k} = \tilde{\mathbf{z}}_i^{\top} \cdot \hat{\boldsymbol{c}}^k / \|\tilde{\mathbf{z}}_i^{\top}\| \cdot \|\hat{\boldsymbol{c}}^k\|$. We identify the set of M category indices that are neighbors $\mathcal{N}_i$ with the maximum cosine similarity. We calibrate the distribution of feature $\tilde{\mathbf{z}}_i$ as:

$$\begin{aligned} \mu_{\tilde{\mathbf{z}}_i} &= (1-\alpha) \cdot \tilde{\mathbf{z}}_i + \alpha \cdot \frac{1}{M} \sum_{k \in \mathcal{N}_i} \frac{e^{S_{i,k}}}{\sum_{j=1}^{M} e^{S_{i,j}}} \cdot \hat{c}^k \\ \Sigma_{\tilde{\mathbf{z}}_i} &= (1-\alpha)^2 \cdot \Sigma_i + \alpha^2 \cdot \frac{1}{M} \sum_{k \in \mathcal{N}_i} \frac{e^{S_{i,k}}}{\sum_{j=1}^{M} e^{S_{i,j}}} \cdot \Sigma_k + \beta, \end{aligned} \tag{10}$$

where α is the hyper-parameter to balance the degree of the calibration and β is an optional constant hyper-parameter to increase the spread of the calibrated

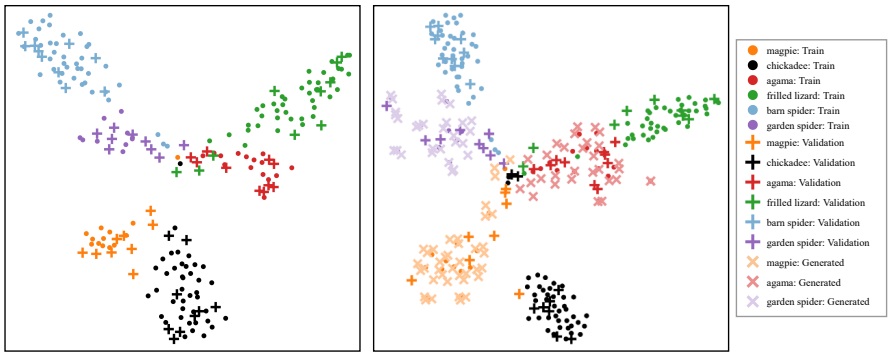


Fig. 3. t-SNE visualization of a few head and tail classes from ImageNet-LT. The plot on the left is before generation, and the plot on the right is after generation. We show 10 validation samples for each class and limit to 40 training + generated samples for ease of interpretation. Markers: · (dot) indicate training samples; + (plus) are validation samples; and × (cross) are generated features also shown with a lighter version of the base color. Best seen in colour.

distribution. We found that $\beta = 0.05$ works reasonably well for multiple experiments. We generate the new samples with the same associated class label and denote the new samples for category k as $\mathcal{F}_k^*$. This combined set of features is generated for all categories and used to train classifier. As shown in Fig. 3, we generate features using our meta-prototype feature augmentation and re-build the t-SNE visualization in the right plot. Compared with the left plot which is before generation, the right plot eases the interpretation and clarifies the feature boundaries. In addition, due to the meta-prototype, the newly generated features are close to validation samples, which further promote the performance of the classifier.

4 Experiments

4.1 Long-tailed Image Classification Task

Datasets and Setup. We perform experiments on long-tailed image classification datasets, including the CIFAR-100-LT [12], ImageNet-LT [18] and iNaturalist2018 [27].

- CIFAR-100-LT is based on the original CIFAR-100 dataset, whose training samples per class are constructed by imbalance ratio (The imbalance ratios we adopt in our experiment are 10, 50 and 100).
- ImageNet-LT is a long-tailed version of the ImageNet dataset by sampling a subset following the Pareto distribution with power value 6. It contains 115.8K images from 1,000 categories, with class cardinality ranging from 5 to 1,280.

Table 1. Top 1 accuracy of CIFAR-100-LT with various imbalance factors (100, 50, 10). RL, DT, and DA indicate representation learning, decouple training, and data augmentation, respectively.

Type	Method	CIFAR-100-LT		
		100	50	10
Baseline	Softmax	38.3	43.9	55.7
RL	KCL [10]	42.8	46.3	57.6
	DRO-LT [21]	47.3	57.6	63.4
	TSC [15]	43.8	47.4	59.0
	Hybrid-SC [30]	46.7	51.9	63.1
DT	Decoupling [9]	43.3	47.4	57.9
	De-confound [25]	44.1	50.3	59.6
	MiSLAS [41]	47.0	52.3	63.2
	Bag of tricks [39]	47.8	51.7	-
DA	MetaSAug [14]	48.1	52.3	61.3
	TailCalibX [28]	46.6	50.9	61.1
	RSG [29]	44.6	48.5	-
	GLAG[40]	51.7	55.3	64.5
	Ours	**52.3**	**55.9**	**64.9**

– iNaturalist2018 is the largest dataset for long-tailed visual recognition. It contains 437.5K images from 8,142 categories. It is extremely imbalanced with an imbalance factor of 512.

Experimental Details. We implement all experiments in PyTorch. On CIFAR-100-LT, following [20], we use ResNet-32 [33] as the feature extractor for all methods. we conduct model training with SGD optimizer based on batch size 256, momentum 0.9 under three imbalance ratios (10, 50 and 100). For image classification on ImageNet-LT, following [25, 5, 8], we use ResNetXt-50 [33] as the feature extractor for all methods. We conduct model training with the SGD optimizer based on batch size 512, and momentum 0.9. In both training epochs (90 and 200 training epochs), the learning rate is decayed by a cosine scheduler [19] from 0.2 to 0.0. On iNaturalist2018 [27] dataset, we use ResNet-50 [33] as the feature extractor for all methods with 200 training epochs, with the same experimental parameters set for the other. Moreover, we use the same basic data augmentation (i.e., random resize and crop to 224, random horizontal flip, color jitter, and normalization) for all methods.

Comparison with State of the Arts. As shown in Tab. 1, to prove the versatility of our method, we employ our method on the CIFAR-100-LT dataset

Table 2. Results on ImageNet-LT in terms of accuracy (Acc) under 90 and 200 training epochs. In this table, CR, DT, and RL indicate class re-balancing, decouple training, and representation learning, respectively.

Type	Method	90 epochs				200 epochs			
		Many	Med.	Few	All	Many	Med.	Few	All
Baseline	Softmax	66.5	39.0	8.6	45.5	66.9	40.4	12.6	46.8
CR	Focal Loss [16]	66.9	39.2	9.2	45.8	67.0	41.0	13.1	47.2
	BALMS [20]	61.7	48.0	29.9	50.8	62.4	47.7	32.1	51.2
	LDAM [2]	62.3	47.4	32.5	51.1	60.0	49.2	31.9	51.1
	LADE [8]	62.2	48.6	31.8	51.5	63.1	47.7	32.7	51.6
	DisAlign [37]	62.7	52.1	31.4	53.4	-	-	-	-
DT	Decoupling [9]	62.4	39.3	14.9	44.9	60.9	36.9	13.5	43.0
	MiSLAS [41]	62.1	48.9	31.6	51.4	65.3	50.6	33.0	53.4
	De-confound [25]	63.0	48.5	31.4	51.8	64.9	46.9	28.1	51.3
	xERM_{TDE} [44]	-	-	-	-	68.6	50.0	27.5	54.1
RL	OLTR [17]	58.2	45.5	19.5	46.7	62.9	44.6	18.8	48.0
	DRO-LT [21]	-	-	-	-	64.0	49.8	33.1	53.5
	PaCo [5]	59.7	51.7	36.6	52.7	63.2	51.6	39.2	54.4
DA	RSG [29]	68.7	43.7	16.2	49.6	65.0	49.4	31.1	52.9
	SSP [34]	65.6	49.6	30.3	53.1	67.3	49.1	28.3	53.3
	Ours	**64.3**	**51.6**	**31.8**	**53.8**	**66.3**	**52.8**	**35.2**	**55.1**

with three imbalance ratios. We compare against the most relevant methods and choose methods that are recently published and representative of different types, such as class re-balancing, decouple training and data augmentation. Our method surpasses the DRO-LT [21] under various imbalance factors, especially on the largest imbalance factor (52.3% vs 47.3%). Furthermore, compared with the data augmentation methods [40], our model achieves competitive performance (52.3% vs 51.7% with 100 imbalance factor).

Tab. 2 shows the long-tailed results on ImageNet-LT. We adopt the performance data from the deep long-tailed survey [38] for various methods at 90 and 200 training epochs to make a fair comparison. Our approach achieves 53.8% and 55.1% in overall accuracy, which outperforms the state of the art methods by a significant margin at 90 and 200 training epochs, respectively. Compared with representation learning methods, our method surpasses SSP by 0.7% (53.8% vs 53.1%) at 90 training epochs and outperforms SSP by 1.8% (55.1% vs 53.3%) at 200 training epochs. In addition, our method obtains higher performance by 1.1% (53.8% vs 52.7%) and 0.7% (55.1% vs 54.4%) than PaCo at 90 and 200 training epochs, respectively.

Furthermore, Tab. 3 presents the experimental results on the naturally-skewed dataset iNaturalist2018. Compared with the improvement brought by

Table 3. Benchmarking on iNaturalists2018 in Top 1 accuracy (%). RL, DT, and DA indicate representation learning, decouple training, and data augmentation.

Type	Method	iNaturalist			
		Many	Med.	Few	All
Baseline	Softmax	72.2	63.0	57.2	61.7
RL	Focal Loss [16]	-	-	-	61.1
	DRO-LT [21]	-	-	-	69.7
	OLTR [17]	59.0	64.1	64.9	63.9
	TSC [15]	72.6	70.6	67.8	69.7
	DisAlign [37]	69.0	71.1	70.2	70.6
DT	Decoupling [9]	65.6	65.3	65.5	65.6
	BBN [43]	49.4	70.8	65.3	66.3
DA	MetaSAug [14]	-	-	-	68.7
	GLAG[40]	-	-	-	69.2
	Ours	**72.8**	**71.7**	**70.0**	**71.0**

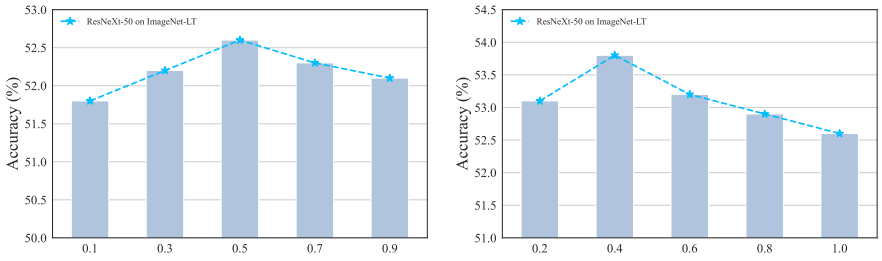
representation learning, decouple training and data augmentation approaches, our method achieves competitive result (71.0%) consistently.

4.2 Semantic Semgnetaion on ADE20K Dataset

To further validate our method, we apply our strategy to segmentation networks and report our performance on the semantic segmentation benchmark, ADE20K.
Dataset and Setup. ADE20K is a scene parsing dataset covering 150 fine-grained semantic concepts and it is one of the most challenging semantic segmentation datasets. The training set contains 20,210 images with 150 semantic classes. The validation and test set contain 2,000 and 3,352 images respectively.
Experimental Details. We evaluate our method using two widely adopted segmentation models (OCRNet [35] and DeepLabV3+ [4]) based on different backbone networks. We initialize the backbones using the models pre-trained on ImageNet [6] and the framework randomly. All models are trained with an image size of 512×512 and 80K/160K iterations in total. We train the models using the Adam optimizer with the initial learning rate of 0.01, weight decay of 0.0005, and momentum of 0.9. The learning rate dynamically decays exponentially according to the 'ploy' strategy.
Comparison with State of the Arts. The numerical results and comparison with other peer methods are reported in Tab. 4. Our method achieves 1.1% and 0.5% improvement in mIoU using OCRNet with HRNet-W18 when the iterations are 80K and 160K, respectively. Moreover, our method outperforms the baseline with large margin at 0.9% and 1.1% in mIoU using DeeplabV3+ with ResNet-50 when the iterations are 80K and 160K, respectively. Even with a stronger backbone, ResNet-101, our method also achieves 0.8% mIoU and 0.9% improvement

Table 4. Performance of semantic segmentation on ADE20K. R-50 and R-101 denote ResNet-50 and ResNet-101, respectively.

Framework	Method	Backbone	80K iteration		160K iteration	
			mIoU	mAcc	mIoU	mAcc
OCRNet	Baseline	HRNet-W18	39.2	49.0	40.8	50.9
	Ours		**40.3**	**51.9**	**41.3**	**52.8**
DeepLabV3+	Baseline	R-50	43.8	54.5	44.9	55.0
	DisAlign [37]		-	-	45.7	**57.3**
	Ours		**44.7**	**55.1**	**46.0**	57.0
	Baseline	R-101	46.1	56.2	46.4	56.7
	DisAlign [37]		-	-	47.1	59.5
	Ours		**46.9**	**57.1**	**47.3**	**59.9**



(a) λ in Eq. 3 (b) α in Eq. 10

Fig. 4. Ablation study on λ in Eq. 3 and α in Eq. 10.

than the baseline. Compared with DisAlign, our method still outperforms it on both in both mIoU and mAcc with various backbones consistently.

4.3 Ablation Study

We conduct ablation study on the ImageNet-LT dataset to further understand the hyper-parameters of our methods and the effect of each proposed component. All of them have trained with ResNetXt-50 by 90 epochs for a fair comparison.

λ in Meta Training Loss. One major hyper-parameter in our method is λ in Eq. 3, which adjusts the degree of adjustment in meta training loss. We set the hyper-parameter $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We study the sensitivity of the accuracy to the values of λ. Fig. 4(a) quantifies the effect of the trade-off parameter λ on the validation accuracy. It shows that combining the $\mathcal{L}_{PC}$ and $\mathcal{L}_{CE}$ with optimal λ is 0.5 gives the best results.

α in Meta-Prototype Feature Generation. In Eq. 10, we introduce a class-wise confidence score α which controls the degree of distribution calibration. We initialize α to 0.2 for each tail class and it changes adaptively during training.

Table 5. Ablation study on ImageNet-LT for different decouple methods.

CE	DRO-LT	KCL	**MPCL**	TailCalibX	**MPFA**	Many	Med.	Few	All
✓	✗	✗	✗	✗	✗	66.5	39.0	8.6	45.5
✓	✓	✗	✗	✗	✗	65.0	48.8	25.8	51.9
✓	✗	✓	✗	✗	✗	62.4	49.0	29.5	51.5
✓	✗	✗	✓	✗	✗	64.6	50.1	27.5	52.7
✓	✗	✗	✓	✓	✗	63.7	51.2	31.0	53.2
✓	✗	✗	✓	✗	✓	**64.3**	**51.6**	**31.8**	**53.8**

We set the hyper-parameter α in the interval from 0.2 to 1 with a stride of 0.2 and take the five sets of values to conduct ablation experiments as shown in Fig. 4(b). Overall, the larger α means more confidence to transfer the knowledge from head to tail classes. The optimal α for ImageNet-LT is 0.4.

Effectiveness of MPCL and MPFA. Tab. 5 verifies the critical roles of our adaptive modules for meta-prototype contrastive learning (MPCL) and meta-prototype feature augmentation (MPFA). The baseline only performs decoupled training pipelines without using any components of our methods. In representation learning stage, our MPCL module significantly surpasses the performance over the DRO-LT and KCL (52.7% vs 51.9% vs 51.5%). Moreover, in classifier training stage, our MPFA module further boosts the performance, especially in the tail classes (53.8% vs 53.2%). The results suggest the effectiveness of both the MPCL and MPFA components in improving the training performance.

5 Conclusion

In this paper, we have proposed a novel meta-prototype decoupled training framework to tackle the long-tail challenge. Our decoupled training framework includes calibrating the empirical prototype for SCL in the representation learning stage and enhancing feature embedding for tail classes based on learned meta-prototype in the classifier learning stage. The first module of our method completes the meta-prototype to promote the representation quality of supervised prototype contrastive learning. The second module leverages the learned meta-prototype to provide the reasonable feature distribution of new feature samples for tail classes. We sample features from the calibrated distribution to ease the dominance of the head classes in classification decisions. The experimental results show that our method achieves state-of-the-art performances for various settings on long-tailed learning.

Acknowledgments This work is supported by the National Natural Science Foundation of China (U21B2004), the Zhejiang Provincial key RD Program of China (2021C01119) , and the Zhejiang University-Angelalign Inc. R & D Center for Intelligent Healthcare.

References

1. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems **29** (2016)
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. Advances in neural information processing systems **32** (2019)
3. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: A joint formulation. In: European conference on computer vision. pp. 566–579. Springer (2012)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
5. Cui, J., Zhong, Z., Liu, S., Yu, B., Jia, J.: Parametric contrastive learning (2021)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
8. Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., Chang, B.: Disentangling label distribution for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6626–6636 (2021)
9. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition (2019)
10. Kang, B., Li, Y., Xie, S., Yuan, Z., Feng, J.: Exploring balanced feature spaces for representation learning. In: International Conference on Learning Representations (2021)
11. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in Neural Information Processing Systems **33**, 18661–18673 (2020)
12. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
13. Li, J., Xiong, C., Hoi, S.C.: Mopro: Webly supervised learning with momentum prototypes. arXiv preprint arXiv:2009.07995 (2020)
14. Li, S., Gong, K., Liu, C.H., Wang, Y., Qiao, F., Cheng, X.: Metasaug: Meta semantic augmentation for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5212–5221 (2021)
15. Li, T., Cao, P., Yuan, Y., Fan, L., Yang, Y., Feris, R.S., Indyk, P., Katabi, D.: Targeted supervised contrastive learning for long-tailed recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6918–6928 (2022)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
17. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2537–2546 (2019)

18. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
19. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
20. Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al.: Balanced meta-softmax for long-tailed visual recognition. Advances in Neural Information Processing Systems **33**, 4175–4186 (2020)
21. Samuel, D., Chechik, G.: Distributional robustness loss for long-tail learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
22. Samuel, D., Chechik, G.: Distributional robustness loss for long-tail learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9495–9504 (2021)
23. Samuel, D., Chechik, G.: Distributional robustness loss for long-tail learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9495–9504 (2021)
24. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. Advances in neural information processing systems **32** (2019)
25. Tang, K., Huang, J., Zhang, H.: Long-tailed classification by keeping the good and removing the bad momentum causal effect. Advances in Neural Information Processing Systems **33**, 1513–1524 (2020)
26. Tukey, J.W., et al.: Exploratory data analysis, vol. 2. Reading, MA (1977)
27. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
28. Vigneswaran, R., Law, M.T., Balasubramanian, V.N., Tapaswi, M.: Feature generation for long-tail classification. In: Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing. pp. 1–9 (2021)
29. Wang, J., Lukasiewicz, T., Hu, X., Cai, J., Xu, Z.: Rsg: A simple but effective module for learning imbalanced datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3784–3793 (2021)
30. Wang, P., Han, K., Wei, X.S., Zhang, L., Wang, L.: Contrastive learning based hybrid networks for long-tailed image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 943–952 (2021)
31. Wang, X., Lian, L., Miao, Z., Liu, Z., Yu, S.: Long-tailed recognition by routing diverse distribution-aware experts. In: International Conference on Learning Representations (2021)
32. Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G., Wu, C.: Implicit semantic data augmentation for deep networks. Advances in Neural Information Processing Systems **32** (2019)
33. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. arXiv preprint arXiv:1611.05431 (2016)
34. Yang, Y., Xu, Z.: Rethinking the value of labels for improving class-imbalanced learning. Advances in Neural Information Processing Systems **33**, 19290–19301 (2020)
35. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing (2018)

36. Zang, Y., Huang, C., Loy, C.C.: Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3457–3466 (2021)
37. Zhang, S., Li, Z., Yan, S., He, X., Sun, J.: Distribution alignment: A unified framework for long-tail visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2361–2370 (2021)
38. Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. arXiv preprint arXiv:2110.04596 (2021)
39. Zhang, Y., Wei, X.S., Zhou, B., Wu, J.: Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 3447–3455 (2021)
40. Zhang, Z., Xiang, X.: Long-tailed classification with gradual balanced loss and adaptive feature generation (2022)
41. Zhong, Z., Cui, J., Liu, S., Jia, J.: Improving calibration for long-tailed recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16489–16498 (2021)
42. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
43. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: 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. pp. 9719–9728 (2020)
44. Zhu, B., Niu, Y., Hua, X.S., Zhang, H.: Cross-domain empirical risk minimization for unbiased long-tailed classification. In: AAAI Conference on Artificial Intelligence (2022)
45. Zhu, L., Yang, Y.: Inflated episodic memory with region self-attention for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4344–4353 (2020)