Feature Fusion from Head to Tail for Long-Tailed Visual Recognition

Mengke Li[1,2], Zhikai Hu[3], Yang Lu[4], Weichao Lan[3], Yiu-ming Cheung[3], Hui Huang[2*]
[1] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[3] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
[4] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China
limengke@gml.ac.cn, cszkhu@comp.hkbu.edu.hk, luyang@xmu.edu.cn, {cswclan, ymc}@comp.hkbu.edu.hk,
hhzhiyan@gmail.com

Abstract

The imbalanced distribution of long-tailed data presents a considerable challenge for deep learning models, as it causes them to prioritize the accurate classification of head classes but largely disregard tail classes. The biased decision boundary caused by inadequate semantic information in tail classes is one of the key factors contributing to their low recognition accuracy. To rectify this issue, we propose to augment tail classes by grafting the diverse semantic information from head classes, referred to as head-to-tail fusion (H2T). We replace a portion of feature maps from tail classes with those belonging to head classes. These fused features substantially enhance the diversity of tail classes. Both theoretical analysis and practical experimentation demonstrate that H2T can contribute to a more optimized solution for the decision boundary. We seamlessly integrate H2T in the classifier adjustment stage, making it a plug-and-play module. Its simplicity and ease of implementation allow for smooth integration with existing long-tailed recognition methods, facilitating a further performance boost. Extensive experiments on various long-tailed benchmarks demonstrate the effectiveness of the proposed H2T. The source code is available at https://github.com/Keke921/H2T.

Introduction

Deep models have shown remarkable capabilities in diverse visual recognition tasks (Hu et al. 2019; Cheung, Li, and Zou 2021; Minaee et al. 2022; Lan et al. 2024), yet their performance heavily relies on data that is evenly distributed across categories. In contrast, real-world data typically follows long-tailed distributions (Reed 2001; Liu et al. 2019; Zhang et al. 2023), which hinders model performance, especially on minority classes (referred to as tail classes in long-tailed data). This challenge has become one of the bottlenecks limiting the advancement of deep models. In recent times, a plethora of methods have emerged to address the problem of severe class imbalance in long-tailed data from different aspects, such as class-balancing methods (Chawla et al. 2002; Huang et al. 2016; Ren et al. 2018; Cui et al. 2019; Lin et al. 2020; Huang et al. 2020; Hong et al. 2021; Park et al. 2021), re-margining methods (Cao et al. 2019; Li, Cheung, and Lu 2022; Li, Cheung, and Hu 2022), data augmentation (Kim, Jeong, and Shin 2020; Wang et al. 2021a; Li et al. 2021; Park et al. 2022), and ensembling learning (Wang et al. 2021b; Li et al. 2022a; Jin et al. 2023). While these methods demonstrated the capability to yield robust predictions, they primarily focus on training a new model to acquire a relatively balanced embedding space and/or assembling multiple diverse networks. However, they overlook acquiring a more optimal classifier (determining the decision boundary), which is crucial for fully releasing the potential of the acquired backbone.

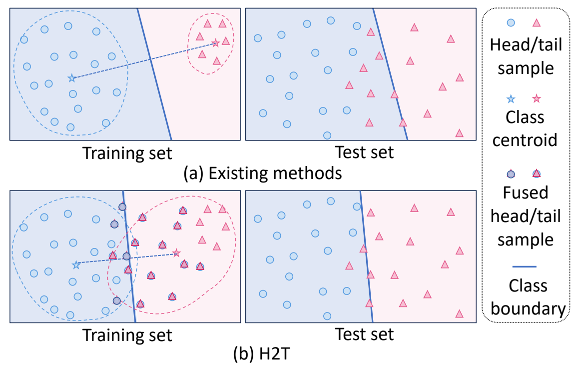

Figure 1: Comparison between decision boundaries produced by (a) existing methods and (b) proposed H2T.

This paper thereby aims to obtain a more effective classifier with the trained backbone. It is widely recognized that increasing class margins (Cao et al. 2019; Deng et al. 2019) and/or tightening the intra-class space (Wang et al. 2018, 2017) for the training set can improve the generalization performance of the model (Liu et al. 2021; Hu et al. 2023). However, while the clear margin is effective for balanced data, it may still have a biased decision boundary for long-tailed data. Take the binary case in the embedding space of the obtained backbone as an example, the decision boundary is usually the midline connecting the two class centroids. The head class has sufficiently sampled and its embedding space is fully occupied (Xiao et al. 2021). On the opposite, the tail class suffers from a scarcity of samples, leading to sparsely distributed semantic regions. The bias in the tail centroid persists due to head squeeze even if the existence

*Corresponding author.

of a clear margin. As a result, during the inference stage, a considerable number of samples that differ from the training set emerge, causing the erosion of well-defined margins. Consequently, numerous tail class samples are misclassified as head classes, as illustrated in Figure 1(a).

To enrich the sparse tail class semantics and calibrate the bias in tail classes, we propose a direct and effective solution, named head-to-tail fusion (H2T), which grafts partial semantics from the head class on the tail class. In particular, the fusion operation in H2T involves the direct replacement of certain features of tail samples with those of head samples. This is based on the assumption that predictions on tail classes that have rare instances are easily affected by head classes that appear frequently. Transferring the head semantics can effectively fill the tail semantic area and the category overlap, which compels the decision boundary to shift closer to a more optimal one, as shown in Figure 1(b). Ultimately, the generalization performance of the tail classes can be improved. To streamline the fusion operation, we design an easy-to-implement strategy that takes full use of the obtained features without retraining the backbone. Specifically, we randomly substitute several channels in the feature maps of the class-balanced data with those of the head-biased data. This makes the feature maps of tail classes with a high probability of being fused with head classes. We apply H2T to the classifier tuning stage, and with just a few lines of code and a few epochs of training, remarkable performance improvements can be achieved. In addition, H2T does not alter the structure of the backbone or increase the network parameters. This characteristic makes it highly adaptable and compatible with various existing techniques, facilitating its seamless incorporation into diverse existing methods. The main contributions of this paper are summarized as follows:

- We propose a novel H2T that borrows information from head classes and applies it to augment tail classes without additional data or network parameters, which can release the potential of the well-trained backbone and obtain better decision boundaries.
- We devise a simple fusion strategy that partially substitutes the feature maps of an instance-wise sampling branch with those of a class-balanced sampling branch. This approach enables H2T to be implemented with ease and without requiring any modifications to the original backbone structure.
- Considerable performance improvement is achieved in both single and multi-expert models by integrating the H2T module into the existing structures. Extensive experiments on popular benchmarks validate the efficacy of the proposed method.

Related Work

This section mainly makes an overview of methods based on data augmentation which is the most relevant regime.

Classical Augmentation: Classical augmentation methods including flip, rotate, crop, padding, and color jittering *etc*. (Szegedy et al. 2015; He et al. 2016) have been widely applied in deep models. On long-tailed data, these methods can improve model robustness and prevent overfitting to specific input patterns to a certain extent. Recently, AutoAugment (Cubuk et al. 2019), Fast AutoAugment (Lim et al. 2019), Population based augmentation (Ho et al. 2019) and Randaugment (Cubuk et al. 2020), *etc*. have been proposed to determine the best data augmentation strategy for a dataset automatically. These methods have been proven to be effective in increasing the classification accuracy for DNNs and also have demonstrated their efficacy on long-tailed data (Ren et al. 2020; Cui et al. 2021; Li et al. 2022b).

MixUp Based Augmentation: MixUp (Zhang et al. 2018) is an effective data augmentation method that linearly combines a pair of row images and their labels. This method adds complexity control to the uncovered space in the data space through linearly interpolating discrete sample points, which reduces the model generalization error. Numerous studies (Zhang et al. 2021; Zhong et al. 2021; Li, Cheung, and Lu 2022) have experimentally proven that MixUp can significantly improve long-tailed recognition performance. CutMix (Yun et al. 2019), a variant of MixUp, augments data by cutting a patch from one input image and pasting it into another image within the training set. The ground truth labels are combined in proportion to the patch areas. Park *et al*. (Park et al. 2022) propose mixing the foreground patches from tail classes with the background images from head classes via CutMix, enabling the transfer of context-rich background information from head to tail.

New Images Based Augmentation: In addition to the aforementioned methods, researchers have proposed alternative approaches that generate new informative samples for tail classes. One such method is M2m (Kim, Jeong, and Shin 2020), which translates samples from head classes to tail classes using a pretrained auxiliary classifier. Wang *et al*. (2021a) utilize the learned encoded variation information from head classes to synthesize images for tail classes. Zada *et al*. (2022) directly exploit pure noise images as tail class samples. Zhang *et al*. (2021) utilize class activation maps (CAM) (Zhou et al. 2016) to separate image foreground and background, followed by augmenting the foreground by flipping, rotating, and scaling, *etc*. The new samples are then generated by overlaying the augmented foreground on the unchanged background.

Feature Level Augmentation: Data augmentation can also be performed in feature space. For example, Manifold MixUp (Verma et al. 2019) conducts the linear interpolation on the features of input images, which has been shown to yield better results in long-tail learning (Zhang et al. 2021; Zhou et al. 2020). Chu *et al*. (2020) use CAM to decompose features into class-generic and class-specific components. Tail classes are then augmented by combining their class-specific components with the class-generic components. Another promising technique is to transfer knowledge from head classes to tail classes. For example, Yin *et al*. (2019) and Liu *et al*. (2020) enrich tail classes with the intra-class variance of head classes to balance the class distributions in feature space. Recently, meta-learning-based augmentation has been leveraged to address class imbalance. Liu *et al*. (2022) design a meta-embedding that uses a memory bank to enrich tail classes. MetaSAug (Li et al. 2021) learns class-wise covariance matrices by minimizing loss on a small balanced validation set and subsequently uses the learned matrices to semantically augment tail classes.

Methodology

Preliminaries

We formally define the basic notations used in this paper before going into detail about our proposed method. We use $\{x, y\}$ to represent one input image and its corresponding label. The total number of classes is denoted by C, thus, we have $y \in \{0, 1, \cdots, C-1\}$. The training set includes N samples. Suppose that class i has n_i training samples. Then $N = \sum_i n_i$. For simplicity, we suppose $n_0 \geq n_1 \geq \cdots \geq n_{C-1}$. Feed x into the backbone, we can obtain its feature maps before the last pooling layer, denoted as $\mathcal{F} = [F_0, F_1, \cdots, F_{d-1}] \in \mathbb{R}^{w_F \times h_F \times d}$, where w_F and h_F represent the width and height of the feature map, respectively. d is the dimension of features in embedding space. The representation after the last pooling layer is $f \in \mathbb{R}^d$. The weight of the linear classifier is represented as $\mathbf{W} = [w_0, w_1, \cdots, w_{C-1}] \in \mathbb{R}^{d \times C}$, where w_i represents the classifier weight for class i. We use the subscripts h, m and t to indicate head, medium and tail classes, respectively. $z_i = w_i^T f$ represents the predicted logit of class i, where the subscript $i = y$ denotes the target logit and $i \neq y$ denotes the non-target logit.

Motivation

Head classes are significantly more numerous than tail classes in long-tailed data, resulting in a biased classifier, which leads to poor performance on the test set of tail classes. The straightforward solution involves increasing the importance or frequencies of tail classes (Cao et al. 2019; Zhou et al. 2020; Ando and Huang 2017). They can increase the performance of tail classes, nevertheless, they also entail an elevated risk of overfitting. To adjust the decision boundary and prevent overfitting, it is crucial to enhance the diversity of tail class samples (Zhang et al. 2023; Yang et al. 2022). Unfortunately, directly obtaining more samples is generally infeasible. We can consider enriching the tail class by maximizing the utilization of existing features. Typically, the misclassified samples are unseen instances from the training set, and these instances tend to be interspersed in the vicinity of the decision boundary. This pattern provides us an opportunity to augment the diversity within tail classes and populate class margins with semantically meaningful samples. This augmentation process contributes to an optimal decision boundary. To achieve this, we simulate the potential unseen samples by directly borrowing semantic information from head classes to augment tail classes. By doing so, the rich semantic information can be transferred from well-represented head classes to tail classes.

Methodology: Fusing Head Features to Tail

We fuse the features of head classes to the tail to exploit the abundant closet semantic information. This operation can enrich the tail classes and expand their embedding space distribution. The fusion process is formulated as:

$$\tilde{\mathcal{F}} = \mathcal{M}_p \otimes \mathcal{F}_t + \overline{\mathcal{M}}_p \otimes \mathcal{F}_h, \tag{1}$$

where $\mathcal{M}_p$ is the mask stacked with multiple 1 matrices $\mathbf{I}_M = \mathbb{1}^{w_F \times h_F}$ and 0 matrices $\mathbf{O}_M = \mathbb{0}^{w_F \times h_F}$. The total number of all the 1 and 0 matrices is d. $\overline{\mathcal{M}}p$ is the complement of $\mathcal{M}p$, that is, the indices of 0 matrices in $\overline{\mathcal{M}}p$ correspond to the 1 matrix in $\mathcal{M}p$, and vice versa. The subscript p of the mask matrices represents the fusion ratio. Specially, the number of $\mathbf{I}_M$ and $\mathbf{O}_M$ is $[d \times p]$ and $d - [d \times p]$, respectively. $[\cdot]$ means rounding operation. $\tilde{\mathcal{F}}$ is then passed through a pooling layer and classifier to predict the corresponding logits $\mathbf{z} = [z_0, z_1, \cdots, z_{C-1}]$. Different loss functions, such as CE loss, MisLAS (Zhong et al. 2021), or GCL (Li, Cheung, and Lu 2022), to name a few, can be adopted. The backbone ϕ can be the single model (He et al. 2016) as well as the multi-expert model (Wang et al. 2021b; Xiang, Ding, and Han 2020). We exploit the two-stage training (Kang et al. 2020) and apply H2T in stage II.

The selection of features to be fused poses a tedious task during training since visual recognition tasks often encompass a vast number of categories, and it cannot be guaranteed that each minibatch contains the required categories. We thereby devise a simple strategy to facilitate an easy-to-execute fusion process. This strategy involves sampling two versions of data: 1) class-balanced data $\mathcal{T}^B$ used to balance the empirical/structural risk minimization (ERM/SRM) of each class, which is fed into the fused branch, and 2) instance-wise data $\mathcal{T}^I$ has a high probability to obtain head class samples, which is fed into the fusing branch. The sampling rates p_i^B and p_i^I for class i within the sets $\mathcal{T}^B$ and $\mathcal{T}^I$, respectively, are calculated by

$$p_i^B = \frac{1}{C},\ p_i^I = \frac{n_i}{N}. \tag{2}$$

Balanced sampling data ensures that each class is sampled with equal probability $\frac{1}{C}$. The fewer the number of samples, the higher the probability of being resampled multiple times. The instance-wise sampling branch deals with the original data distribution, resulting in head-biased data. Therefore, the head classes have a higher probability of being sampled. Next, we can use the feature maps $\mathcal{F}^B$ and $\mathcal{F}^I$ obtained from $\mathcal{T}^B$ and $\mathcal{T}^I$ to substitute $\mathcal{F}_t$ and $\mathcal{F}_h$ in Eq. (1). By doing so, the features of the repeatedly sampled tail classes will be fused with the head class features with a higher probability. In contrast to the common practice of linearly combining a pair of inputs and their labels for augmentation, we only use the labels of the fused branch, namely the balance-sampled data, as the ground truth. Through the adoption of this method, the semantics of head classes are utilized to enrich tail classes. The proposed framework is illustrated in Figure 2 and Algorithm 1.

Rationale Analysis

Although the proposed H2T appears intuitive and straightforward on the surface, it is built on a foundation of rationale. We explore its theoretical rationality in depth in this section. For ease of analysis, we assume that the feature maps to be fused are rearranged in order without loss of generality. Then, after the pooling layer, the feature can be written as $f_i^T = [\dot{f}_i^T, \ddot{f}_i^T]$, where $\dot{f}_i$ and $\ddot{f}_i$ denote the portions

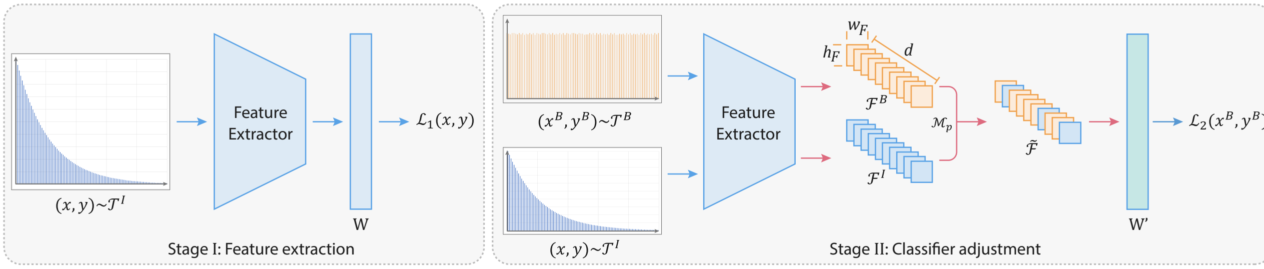

Figure 2: Framework of head-to-tail fusion (H2T).

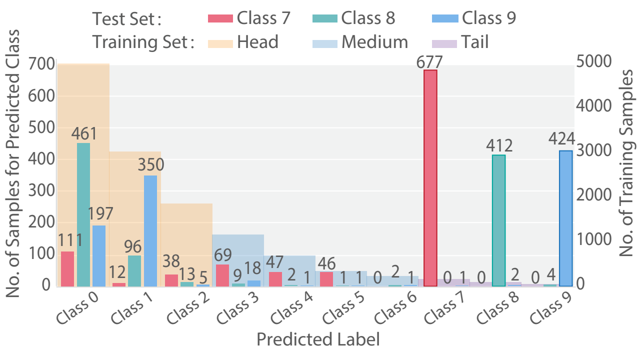

Figure 3: Frequency of predictive labels for tail classes samples. A large number of tail class samples are incorrectly recognized as head classes.

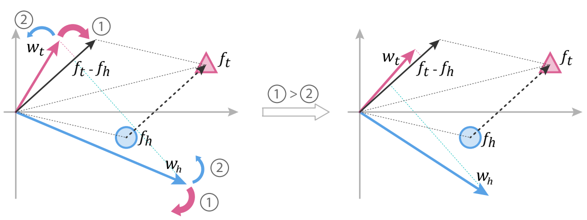

Figure 4: Rationale analysis of H2T. Forces ① and ② are generated by Eq. (9) and Eq. (7), respectively. ① > ② makes the tail sample to "pull" closer to w_t and "push" further away from w_h, leading to the adjustment of decision boundary and enlargement of the tail class space.

of features that are retained and to be fused, respectively. $w_i^T = [\dot{w}_i^T, \ddot{w}_i^T]$ is the corresponding classifier weights.

On the one hand, we expect $z_t > z_h$ for tail class, so that

$$\begin{aligned} w_t^T f_t &> w_h^T f_t \quad \Rightarrow \\ \dot{w}_t^T \dot{f}_t + \ddot{w}_t^T \ddot{f}_t &> \dot{w}_h^T \dot{f}_t + \ddot{w}_h^T \ddot{f}_t. \end{aligned} \tag{3}$$

However, as shown in Figure 3, numerous tail class samples are incorrectly classified into head classes. Therefore, the trained model actually predicts $z_h > z_t$, namely,

$$\dot{w}_t^T \dot{f}_t + \ddot{w}_t^T \ddot{f}_t < \dot{w}_h^T \dot{f}_t + \ddot{w}_h^T \ddot{f}_t. \tag{4}$$

After fusing the head to the tail in stage II, the feature of tail class is $\tilde{f}_t = [\dot{f}_t, \ddot{f}_h]$, and the corresponding logit is $\tilde{z}_t$. Our training goal is still to make the target logit larger than

Algorithm 1: H2T

Input: Training set, fusion ration p ;
Output: Trained model;
Initialize the model ϕ randomly ;
for $iter = 1$ *to* E_0 **do**
  Sampling batches of data $(x, y) \sim \mathcal{T}^I$ from the instance-wise sampling data;
  Obtain the feature map $\mathcal{F} = \phi_\theta(x)$;
  Calculate logits $z = \mathbf{W}^T\mathbf{f}$ and loss $\mathcal{L}_1(x, y)$;
  $\phi = \phi - \alpha\nabla_\phi \mathcal{L}_1((x, y); \phi)$.
end
for $iter = E_0 + 1$ *to* E_2 **do**
  Sample batches of data $(x^B, y^B) \sim \mathcal{T}^B$ and $(x, y) \sim \mathcal{T}^I$;
  Obtain feature maps $\mathcal{F}^B = \phi_\theta(x^B)$ and $\mathcal{F}^I = \phi_\theta(x)$;
  Fuse feature maps by Eq. 1 to obtain $\tilde{\mathcal{F}}$;
  Input $\tilde{\mathcal{F}}$ to pooling layer to obtain $\tilde{\mathbf{f}}$, then calculate the logits by $\tilde{z} = \mathbf{W}^T\tilde{\mathbf{f}}$ and the loss by $\mathcal{L}_2(x^B, y^B)$;
end
Froze the parameters of representation learning ϕ^r, and finetune the classifier parameters ϕ^c: $\phi^c = \phi^c - \alpha\nabla_{\phi^c}\mathcal{L}_2((x^B, y^B); \phi^c)$.

the non-target class, namely, $\tilde{z}_t > \tilde{z}_h$. Therefore, we have

$$\begin{aligned} w_t^T \tilde{f}_t &> w_h^T \tilde{f}_t \Rightarrow \\ \begin{bmatrix} \dot{w}_t \\ \ddot{w}_t \end{bmatrix}^T \begin{bmatrix} \dot{f}_t \\ \ddot{f}_h \end{bmatrix} &> \begin{bmatrix} \dot{w}_h \\ \ddot{w}_h \end{bmatrix}^T \begin{bmatrix} \dot{f}_t \\ \ddot{f}_h \end{bmatrix} \Rightarrow \\ \dot{w}_t^T \dot{f}_t + \ddot{w}_t^T \ddot{f}_h &> \dot{w}_h^T \dot{f}_t + \ddot{w}_h^T \ddot{f}_h. \end{aligned} \tag{5}$$

Adding the last line in Eq. (5) to Eq. (4), we can obtain that

$$\ddot{w}_h^T(\ddot{f}_t - \ddot{f}_h) > \ddot{w}_t^T(\ddot{f}_t - \ddot{f}_h). \tag{6}$$

We use $\ddot{\theta}_*$ ($*$ is h or t) to represent the angle between $\ddot{w}_*$ and $\ddot{f}_t - \ddot{f}_h$. Eq. (6) is further simplified to

$$|\ddot{w}_h| \cos\ddot{\theta}_h > |\ddot{w}_t| \cos\ddot{\theta}_t. \tag{7}$$

On the other hand, similar to Eq. (3), for logits of head classes, we have

$$\dot{w}_h^T \dot{f}_h + \ddot{w}_h^T \ddot{f}_h > \dot{w}_t^T \dot{f}_h + \ddot{w}_t^T \ddot{f}_h. \tag{8}$$

Add the last line in Eq. (5) to Eq. (8), we obtain that

$$|\dot{w}_t| \cos \dot{\theta}_t > |\dot{w}_h| \cos \dot{\theta}_h. \tag{9}$$

Eq. (7) forces w_h to move closer to tail class samples while pushing w_t further away from them, thus increasing the distribution span of head classes. Conversely, Eq. (9) exerts an opposite force. Figure 4 geometrically interprets the rationale of H2T. Notably, when the fusion ratio p is small, Eq. (9) exerts a greater force, enabling the classifier to be well-calibrated. As p grows, the item $\ddot{w}_t^T \ddot{f}_h > \ddot{w}_h^T \ddot{f}_h$ in Eq. (6) progressively becomes more dominant. Under this case, even though Eq. (7) encourages w_t to move away from tail class samples, which is larger than the force of Eq. (9), $\ddot{w}_t^T \ddot{f}_h > \ddot{w}_h^T \ddot{f}_h$ attracts w_t towards head class samples, thus broadens the distribution span of tail classes. Consequently, the classifier increases its performance in the tail class, regardless of the value of p.

Experiments

Datasets and Metrics

We evaluate H2T on four widely-used benchmarks: CIFAR100-LT (Cao et al. 2019), ImageNet-LT, iNaturalist 2018 (Van Horn et al. 2018), and Places-LT. CIFAR100-LT is a small-scale dataset that is sub-sampled from the balanced version CIFAR100 (Krizhevsky, Hinton et al. 2009). We used the imbalance ratios ($\rho = \frac{n_{max}}{n_{min}}$) 200, 100, and 50 (Chu et al. 2020). The original versions of imageNet-2012 (Russakovsky et al. 2015) and Places 365 (Zhou et al. 2017) are also balanced datasets. We use the same settings as Liu *et al*. (Liu et al. 2019) to obtain the long-tailed versions. iNaturalist is collected from all over the world and is naturally heavily imbalanced. The 2018 version (Van Horn et al. 2018) is utilized in our experiment. We also report the comparison results on CIFAR10-LT (Cao et al. 2019) in the Appendix (Li et al. 2023b). As for the metrics, besides top-1 classification accuracy, following Liu *et al*. (Liu et al. 2019), the accuracy on three partitions: head ($n_i > 100$), medium ($20 < n_i \leq 100$) and tail ($n_i \leq 20$) are also compared.

Basic Settings

SGD with a momentum of 0.9 is adopted for all datasets. Stage II is trained with 10 epochs. For CIFAR100-LT, we refer to the settings in Cao *et al*. (Cao et al. 2019) and Zhong *et al*. (Zhong et al. 2021). The backbone network is ResNet-32 (He et al. 2016). Stage I trains for 200 epochs. The initial learning rate is 0.1 and is decayed at the 160^{th} and 180^{th} epochs by 0.1. The batch size is 128. For imageNet-LT and iNaturalist 2018, we use the commonly used ResNet-50. For Places-LT, we utilize the ResNet-152 pre-trained on imageNet. We reproduce the prior methods that provide the hyper-parameters for a fair comparison. For those methods that do not provide hyperparameters or official codes, we directly report their results presented in the original paper.

Comparisons to Existing Methods

Compared Methods: For single model, we compare the proposed H2T with the following three kinds of methods: *two-stage methods*, i.e., LDAM-DRW (Cao et al. 2019), decoupling representation (DR) (Kang et al. 2020), MisLAS (Zhang et al. 2021), and GCL (Li, Cheung, and Lu 2022); *decision boundary adjustment method*, i.e., Adaptive Bias Loss (ABL) (Jin, Lang, and Lei 2023); *data augmentation methods*, i.e., MBJ (Liu, Li, and Sun 2022), and CMO (Park et al. 2022). We report the results of CE loss and balanced softmax cross-entropy loss (BSCE) (Ren et al. 2020) with CMO (abbreviated as CE+CMO and BSCE+CMO, respectively). The results of H2T accompanied by DR (CE loss is utilized), MisLAS, and GCL are reported. For multi-expert models, we compare BBN (Zhou et al. 2020), RIDE (Wang et al. 2021b), ACE (Cai, Wang, and Hwang 2021), and ResLT (Cui et al. 2023). Both RIDE and ResLT use 3-expert architectures. For Places-LT, we report the results of RIDE with linear and cosine classifiers.

Imbalance Ratio	200	100	50
Single Model			
CE loss	35.99	39.55	45.40
LDAM-DRW (Cao et al. 2019)	38.91	42.04	47.62
DR+MU (Kang et al. 2020)	41.73	45.68	50.86
MisLAS (Zhong et al. 2021)	43.45	46.71	52.31
MBJ* (Liu et al. 2022)	-	45.80	52.60
GCL (Li et al. 2022)	**44.76**	48.61	**53.55**
CE+CMO* (Park et al. 2022)	-	43.90	48.30
BSCE+CMO* (Park et al. 2022)	-	46.60	51.40
ABL* (Jin et al. 2023)	-	46.80	52.10
DR+H2T	43.95	**47.73**	52.95
MisLAS+H2T	43.84	47.62	52.73
GCL+H2T	<u>**45.24**</u>	<u>**48.88**</u>	<u>**53.76**</u>
Multi-Expert Model			
BBN (Zhou et al. 2020)	37.21	42.56	47.02
RIDE (Wang et al. 2021b)	45.84	50.37	**54.99**
ACE* (Cai et al. 2021)	-	49.40	50.70
RIDE+CMO* (Park et al. 2022)	-	**50.00**	53.00
ResLT (Cui et al. 2023)	44.26	48.73	53.81
RIDE+H2T	<u>**46.64**</u>	<u>**51.38**</u>	<u>**55.54**</u>
ResLT+H2T	**46.18**	49.60	54.39

Table 1: Comparison results on CIFAR100-LT. The backbone is ResNet-32. $\star$ denotes the results quoted from the corresponding papers. The best and the second-best results are shown in underline bold and bold, respectively.

Comparison Results: The results on CIFAR100-LT are shown in Table 1. H2T can further improve both single and multi-expert models. Applying H2T to the vanilla training of DR with CE loss (DR+H2T) enhances the performance by a significant margin, surpassing most recent methods. The biggest improvement exceeds 2%. H2T can also further boost the performance of two-stage SOTA methods, i.e., MisLAS and GCL, albeit with less significant improvements than for DR. H2T also bolsters the efficacy of multi-expert models. For example, RIDE and ResLT integrating with H2T outperform the original methods by 1.01% and 0.87% on CIFAR100-LT with $\rho = 100$.

The results on large-scale datasets are summarized in Tables 2, 3, and 4. The baseline (CE loss) achieves high accuracy on frequent classes while the performance on rare classes is unsatisfactory. Comparatively, all methods, except CMO and multi-expert models, exhibit substantial improve-

Method	Head	Med	Tail	All
Single Model				
CE loss	64.91	38.10	11.28	44.51
LDAM-DRW (2019)	58.63	48.95	30.37	49.96
DR (2020)	62.93	49.77	33.26	52.18
MisLAS (2021)	62.53	49.82	34.74	52.29
MBJ* (2022)	61.60	48.40	39.00	52.10
GCL (Li et al. 2022)	62.24	48.62	52.12	**54.51**
CE+CMO* (2022)	67.00	42.30	20.50	49.10
BSCE+CMO* (2022)	62.00	49.10	36.70	52.30
ABL* (2023)	62.60	50.30	36.90	53.20
DR+H2T	63.26	50.43	34.11	52.74
MisLAS+H2T	62.42	51.07	35.36	52.90
GCL+H2T	62.36	48.75	52.15	**54.62**
Multi-Expert Model				
BBN (2020)	-	-	-	48.30
RIDE (2021b)	69.59	53.06	30.09	55.72
ACE* (2021)	-	-	-	54.70
RIDE+CMO* (2022)	66.40	53.90	35.60	**56.20**
ResLT (2023)	59.39	50.97	41.29	52.66
RIDE+H2T	67.55	54.95	37.08	**56.92**
ResLT+H2T	62.29	52.29	35.31	53.39

Table 2: Comparison results on imageNet-LT.

Method	Head	Med	Tail	All
Single Model				
CE loss	76.10	69.05	62.44	66.86
LDAM-DRW (2019)	-	-	-	68.15
DR (2020)	72.88	71.15	69.24	70.49
MisLAS (2021)	72.52	72.08	70.76	71.54
MBJ* (2022)	-	-	-	70.00
GCL (2022)	66.43	71.66	72.47	71.47
CE+CMO* (2022)	76.90	69.30	66.60	68.90
BSCE+CMO* (2022)	68.80	70.00	72.30	70.90
ABL* (2023)	-	-	-	71.60
DR+H2T	71.73	72.32	71.30	**71.81**
MisLAS+H2T	69.68	72.49	72.15	**72.05**
GCL+H2T	67.74	71.92	72.22	71.62
Multi-Expert Model				
BBN (2020)	-	-	-	69.70
RIDE (2021b)	76.52	74.23	70.45	72.80
ACE*(2021)	-	-	-	**72.90**
RIDE+ CMO* (2022)	68.70	72.60	73.10	72.80
ResLT* (2023)	64.85	70.64	72.11	70.69
RIDE+H2T	75.69	74.22	71.36	**73.11**
ResLT with H2T	68.41	72.31	72.09	71.88

Table 3: Comparison results on iNaturalist 2018.

ments in the medium and tail classes but at the expense of reduced accuracy in head classes. CMO leverages the background of head classes to augment tail classes without reducing the number of training samples in head classes, thus enhancing model performance for both head and tail classes. However, CMO+CE shows less competitive results in improving tail class performance. ABL also adjusts the decision boundary, but its effectiveness is not as pronounced as H2T. It is worth noting that all comparison methods necessitate training from scratch. H2T can outperform the comparison methods on most datasets by re-fining the classifier with only basic CE loss and several training epochs. For example,

Method	Head	Med	Tail	All
Single Model				
CE loss	46.48	25.66	8.09	29.43
DR (2020)	41.66	37.79	32.77	37.40
MisLAS (2021)	41.95	41.88	34.65	40.38
MBJ* (2022)	39.50	38.20	35.50	38.10
GCL (2022)	38.64	42.59	38.44	40.30
ABL* (2023)	41.50	40.80	31.40	39.40
DR+H2T	41.96	42.87	35.33	**40.95**
MisLAS+H2T	41.40	43.04	35.95	**41.03**
GCL+H2T	39.34	42.50	39.46	40.73
Multi-Expert Model				
LFME* (2020)	39.30	39.60	24.20	36.20
RIDE (2021b) (LC)	44.79	40.69	31.97	40.32
RIDE (2021b) (CC)	44.38	40.59	32.99	40.35
ResLT* (2023)	40.30	44.40	34.70	41.00
RIDE+H2T (LC)	42.99	42.55	36.25	**41.38**
RIDE+H2T (CC)	42.34	43.21	35.62	**41.30**

Table 4: Comparison results on Places-LT. CC, cosine classifier. LC, linear classifier.

on iNaturalist 2018, DR+H2T achieves 71.81%, which is the highest among all other comparison methods. Furthermore, H2T can deliver consistent performance improvements for both single and multi-expert models, particularly for tail and medium classes. Nevertheless, the accuracy improvements of cosine classifier (CC) are less pronounced than those of linear classifier (LC). For example, we can observe that H2T improves GCL using CC much less than DR and MisLAS that use LC. One reason is that the soft margin in GCL allowing samples to fall in the margin between classes can alleviate the decision boundary bias compared with hard margins such as LDAM (Li et al. 2023a). This phenomenon also confirms our motivation. Table 4 further compares the improvements of H2T on RIDE with LC (by 1.06%) and CC (0.95%). The margin on LC is more profound.

Further Analysis

This section visualizes the decision boundary of H2T on each class and investigates the effects of the sampler for the fusing branch and the fusion ratio p. All experiments are performed with DR+H2T.

Visualization of Decision Boundary: Figure 5 shows the t-SNE visualization of the distribution in embedding space and the decision boundary, which demonstrates our motivation (i.e., H2T can enrich tail classes and calibrate the decision boundary). For a more convenient and clearer presentation, the experiment is conducted on CIFAR10-LT and we show the most easily misclassified classes shown in Figure 3 (i.e., Class 0 and Class 8). More visualization results for other classes can be found in the Appendix (Li et al. 2023b). Features are extracted from class-balanced sampling data. We can see that the distribution of Class 8 is sparser than that of Class 0 by DR. H2T enriches the diversity of tail classes without external information, achieving a more optimal decision boundary. It is worth noting that, the significance of a clear margin differs between balanced and imbalanced datasets. In a balanced dataset, a clear margin is superior because the classifier is unbiased for each class. How-

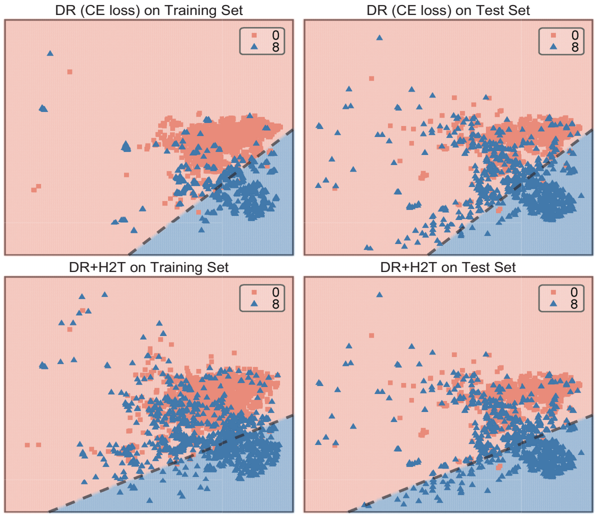

Figure 5: Decision boundary comparison of Class 0 and 8 without H2T v.s. with H2T (namely, DR and DR+H2T).

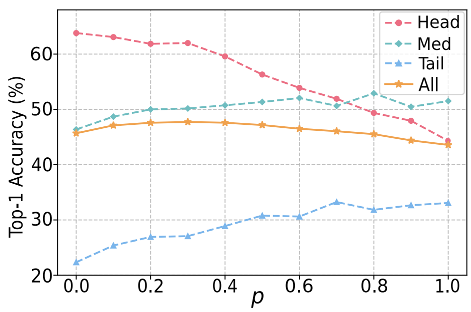

Figure 6: Change of accuracy on different splits w.r.t. p.

ever, in the case of imbalanced training data, a clear margin provides the classifier more room to squeeze tail classes so that correctly classifies head classes. Therefore, we use H2T to fill the marge with semantic samples, which has multiple advantages: 1) prevent over-squeezing for tail classes, thus achieving a more reasonable decision boundary, 2) simulate potentially unseen samples, thus improving the generalization performance of the model on the test set. This is distinctly different from the perception that clear margins can improve performance on balanced datasets.

The Influence of Sampler for Fusing Branch: We compare different sampling strategies for the fusing branch, including class-balanced sampling (CS), reverse sampling (RS) that is tail-biased, and instance-wise sampling (IS) that is head-biased. The results are shown in Table 5. DR uses cRT to fine-tune the classifier with the class-balanced sampling data, which enlarges the medium and tail class expands by pushing the decision boundary in the direction of compressing head classes. Additionally, DR uses CE loss that does not consider class margin. These lead to a significant boost in medium and tail classes while reducing the classification accuracy on head classes. RS augments classes of all scales using head class samples with the lowest probability. CS makes all classes augmented with other classes in an equal probability. IS allows more features from the medium and tail classes to be fused with head features. All sampling strategies adjust the decision boundaries towards narrowing the head class distribution. RS has less damage to the performance of head classes while the improvement brought about by IS to the tail and medium classes is more prominent. As the classifier tuning stage cannot change the embedding distribution, the performance on medium and tail classes is further improved but with performance degradation of head classes.

Method	Head	Med	Tail	All
CE loss†	67.86	38.51	5.66	39.55
DR w.o. H2T†	63.67	46.63	22.21	45.68
BS+RS	**63.56**	48.60	24.07	46.87
BS+BS	62.72	49.14	25.93	47.30
BS+IS	62.00	**50.17**	**27.07**	**47.73**

Table 5: Impact of different samplers.

The Impact of Fusion Ratio: Figure 6 shows the change of accuracy on different splits with respect to p. As p increases, the performance of medium and tail classes improves while that of head classes decreases, which is consistent with the Rationale Analysis section. The class extension of the medium class and the tail class is expanded by eroding that of head classes through H2T fusion. Since $\ddot{w}_t^T \ddot{f}_h > \ddot{w}_h^T \ddot{f}_h$ plays a disproportionate role, resulting in excessive head class samples being incorrectly labeled as medium and tail classes as p increases. If $p > 0.3$, the accuracy on head classes drops dramatically. Even replacing all feature maps with another class still yields a relatively satisfactory accuracy (43.57%), surpassing CE loss by 4.02%. $p = 0.3$ obtains the best overall performance (47.73%) on CIFAR100-LT. By adjusting p based on specific scenarios, one can control the accuracy of different classes, which is particularly useful when additional accuracy requirements exist for different scale classes. More visualization results for the fused embeddings and the selection strategy of fused features can be found in the Appendix (Li et al. 2023b).

Concluding Remarks

This paper has proposed a simple but effective H2T for augmenting tail classes by fusing the feature maps of head class samples to tail. By virtue of this fusion operation, our proposed H2T has two-fold advantages: 1) it produces relatively abundant features to augment tail classes, and 2) generates two opposing forces that restrain each other, preventing excessive sacrifices in head class accuracy. We have further designed a strategy that fuses the features of two branches with different sampling rates for easy implementation. Extensive experiments have shown that the proposed H2T further improves upon SOTA methods.

Despite its effectiveness, H2T has an underlying assumption that representation learning has reached its optimum on the available data. It only adjusts the decision boundary without altering the feature distribution. Ultimately, improvements in the tail classes are always accompanied by sacrificing the head class performance. Our future work will focus on obtaining a more reasonable embedding space distribution to overcome this limitation.

Acknowledgments

This work was supported in parts by NSFC (62306181, U21B2023, 62376233), DEGP Innovation Team (2022KCX TD025), NSFC-RGC (N_HKBU214/21), RGC (12201321, 12202622), and Shenzhen Science and Technology Program (KQTD20210811090044003, RCJC20200714114435012).

References

Ando, S.; and Huang, C. 2017. Deep Over-sampling Framework for Classifying Imbalanced Data. In *ECML/PKDD*, 770–785.

Cai, J.; Wang, Y.; and Hwang, J.-N. 2021. ACE: Ally Complementary Experts for Solving Long-Tailed Recognition in One-Shot. In *ICCV*, 112–121.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 1567–1578.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of artificial intelligence research*, 16: 321–357.

Cheung, Y.-M.; Li, M.; and Zou, R. 2021. Facial Structure Guided GAN for Identity-Preserved Face Image De-Occlusion. In *ICMR*, 46–54.

Chu, P.; Bian, X.; Liu, S.; and Ling, H. 2020. Feature Space Augmentation for Long-Tailed Data. In *ECCV*, volume 12374, 694–710.

Cubuk, E. D.; Zoph, B.; Mané, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Strategies From Data. In *CVPR*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 3008–3017.

Cui, J.; Liu, S.; Tian, Z.; Zhong, Z.; and Jia, J. 2023. ResLT: Residual Learning for Long-Tailed Recognition. *IEEE TPAMI*, 45(3): 3695–3706.

Cui, J.; Zhong, Z.; Liu, S.; Yu, B.; and Jia, J. 2021. Parametric contrastive learning. In *CVPR*, 715–724.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *CVPR*, 9268–9277.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *CVPR*, 4690–4699.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Ho, D.; Liang, E.; Chen, X.; Stoica, I.; and , P. 2019. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. In *ICML*, volume 97, 2731–2741.

Hong, Y.; Han, S.; Choi, K.; Seo, S.; Kim, B.; and Chang, B. 2021. Disentangling Label Distribution for Long-Tailed Visual Recognition. In *CVPR*, 6626–6636.

Hu, Z.; Cheung, Y.-m.; Li, M.; Lan, W.; Zhang, D.; and Liu, Q. 2023. Joint Semantic Preserving Sparse Hashing for Cross-Modal Retrieval. *IEEE TCSVT*, 1–15.

Hu, Z.; Liu, X.; Wang, X.; Cheung, Y.-m.; Wang, N.; and Chen, Y. 2019. Triplet Fusion Network Hashing for Unpaired Cross-Modal Retrieval. In *ICMR*, 141–149.

Huang, C.; Li, Y.; Loy, C. C.; and Tang, X. 2016. Learning Deep Representation for Imbalanced Classification. In *CVPR*.

Huang, C.; Li, Y.; Loy, C. C.; and Tang, X. 2020. Deep Imbalanced Learning for Face Recognition and Attribute Prediction. *IEEE TPAMI*, 42(11): 2781–2794.

Jin, L.; Lang, D.; and Lei, N. 2023. An Optimal Transport View of Class-Imbalanced Visual Recognition. *International Journal of Computer Vision*, 1–19.

Jin, Y.; Li, M.; Lu, Y.; Cheung, Y.-m.; and Wang, H. 2023. Long-Tailed Visual Recognition via Self-Heterogeneous Integration With Knowledge Excavation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23695–23704.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2020. Decoupling representation and classifier for long-tailed recognition. In *ICLR*.

Kim, J.; Jeong, J.; and Shin, J. 2020. M2m: Imbalanced classification via major-to-minor translation. In *CVPR*, 13896–13905.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Tech Report*.

Lan, W.; Cheung, Y.-M.; Jiang, J.; Hu, Z.; and Li, M. 2024. Compact Neural Network via Stacking Hybrid Units. *IEEE TPAMI*, 46(1): 103–116.

Li, B.; Han, Z.; Li, H.; Fu, H.; and Zhang, C. 2022a. Trustworthy Long-Tailed Classification. In *CVPR*, 6970–6979.

Li, J.; Tan, Z.; Wan, J.; Lei, Z.; and Guo, G. 2022b. Nested Collaborative Learning for Long-Tailed Visual Recognition. In *CVPR*, 6949–6958.

Li, M.; Cheung, Y.-m.; and Hu, Z. 2022. Key Point Sensitive Loss for Long-tailed Visual Recognition. *IEEE TPAMI*, in Press: 1–14.

Li, M.; Cheung, Y.-m.; and Lu, Y. 2022. Long-Tailed Visual Recognition via Gaussian Clouded Logit Adjustment. In *CVPR*, 6929–6938.

Li, M.; Cheung, Y.-m.; Lu, Y.; Hu, Z.; Lan, W.; and Huang, H. 2023a. Adjusting Logit in Gaussian Form for Long-Tailed Visual Recognition. *arXiv preprint arXiv:2305.10648*.

Li, M.; Hu, Z.; Lu, Y.; Lan, W.; Cheung, Y.-m.; and Huang, H. 2023b. Feature Fusion from Head to Tail for Long-Tailed Visual Recognition. arXiv:2306.06963.

Li, S.; Gong, K.; Liu, C. H.; Wang, Y.; Qiao, F.; and Cheng, X. 2021. MetaSAug: Meta Semantic Augmentation for Long-Tailed Visual Recognition. In *CVPR*, 5212–5221.

Lim, S.; Kim, I.; Kim, T.; Kim, C.; and Kim, S. 2019. Fast AutoAugment. In *NeurIPS*, 6662–6672.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE TPAMI*, 42(2): 318–327.

Liu, J.; Li, W.; and Sun, Y. 2022. Memory-Based Jitter: Improving Visual Recognition on Long-Tailed Data with Diversity in Memory. In *AAAI*, 1720–1728. AAAI Press.

Liu, J.; Sun, Y.; Han, C.; Dou, Z.; and Li, W. 2020. Deep Representation Learning on Long-Tailed Data: A Learnable Embedding Augmentation Perspective. In *CVPR*.

Liu, X.; Hu, Z.; Ling, H.; and Cheung, Y.-M. 2021. MTFH: A Matrix Tri-Factorization Hashing Framework for Efficient Cross-Modal Retrieval. *IEEE TPAMI*, 43(3): 964–981.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-Scale Long-Tailed Recognition in an Open World. In *CVPR*, 2537–2546.

Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; and Terzopoulos, D. 2022. Image Segmentation Using Deep Learning: A Survey. *IEEE TPAMI*, 44(7): 3523–3542.

Park, S.; Hong, Y.; Heo, B.; Yun, S.; and Choi, J. Y. 2022. The Majority Can Help The Minority: Context-rich Minority Oversampling for Long-tailed Classification. In *CVPR*, 6887–6896.

Park, S.; Lim, J.; Jeon, Y.; and Choi, J. Y. 2021. Influence-Balanced Loss for Imbalanced Visual Classification. In *ICCV*, 735–744.

Reed, W. J. 2001. The Pareto, Zipf and other power laws. *Economics letters*, 74(1): 15–19.

Ren, J.; Yu, C.; sheng, s.; Ma, X.; Zhao, H.; Yi, S.; and Li, h. 2020. Balanced Meta-Softmax for Long-Tailed Visual Recognition. In *NeurIPS*, volume 33, 4175–4186.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In *ICML*, volume 80, 4331–4340.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3): 211–252.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. J. 2018. The INaturalist Species Classification and Detection Dataset. In *CVPR*, 8769–8778.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 6438–6447.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017. NormFace: L_2 Hypersphere Embedding for Face Verification. In *ACM MM*, 1041–1049.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 5265–5274.

Wang, J.; Lukasiewicz, T.; Hu, X.; Cai, J.; and Zhenghua, X. 2021a. RSG: A Simple but Effective Module for Learning Imbalanced Datasets. In *CVPR*, 3784–3793.

Wang, X.; Lian, L.; Miao, Z.; Liu, Z.; and Yu, S. X. 2021b. Long-tailed Recognition by Routing Diverse Distribution-Aware Experts. In *ICLR*.

Xiang, L.; Ding, G.; and Han, J. 2020. Learning From Multiple Experts: Self-paced Knowledge Distillation for Long-Tailed Classification. In *ECCV*, 247–263.

Xiao, L.; Zhang, X.; Jing, L.; Huang, C.; and Song, M. 2021. Does head label help for long-tailed multi-label text classification. *AAAI*, 35(16): 14103–14111.

Yang, L.; Jiang, H.; Song, Q.; and Guo, J. 2022. A survey on long-tailed visual recognition. *IJCV*, 130(7): 1837–1872.

Yin, X.; Yu, X.; Sohn, K.; Liu, X.; and Chandraker, M. 2019. Feature transfer learning for face recognition with under-represented data. In *CVPR*, 5704–5713.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 6023–6032.

Zada, S.; Benou, I.; and Irani, M. 2022. Pure Noise to the Rescue of Insufficient Data: Improving Imbalanced Classification by Training on Random Noise Images. In *ICML*, volume 162, 25817–25833.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond empirical risk minimization. *ICLR*.

Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; and Feng, J. 2023. Deep Long-Tailed Learning: A Survey. *IEEE TPAMI*, 1–20.

Zhang, Y.; Wei, X.-S.; Zhou, B.; and Wu, J. 2021. Bag of Tricks for Long-Tailed Visual Recognition with Deep Convolutional Neural Networks. In *AAAI*, 3447–3455.

Zhong, Z.; Cui, J.; Liu, S.; and Jia, J. 2021. Improving Calibration for Long-Tailed Recognition. In *CVPR*, 16489–16498.

Zhou, B.; Cui, Q.; Wei, X.-S.; and Chen, Z.-M. 2020. BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition. In *CVPR*, 9719–9728.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*, 2921–2929.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6): 1452–1464.