This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

Attentive Feature Augmentation for Long-Tailed Visual Recognition

Weiqiu Wang, Pingyu Wang, Zhicheng Zhao, Fei Su, Hongying Meng

***Abstract*—Deep neural networks have achieved a great success on many visual recognition tasks. However, training data with a long-tailed distribution dramatically degenerates the performance of recognition models. In order to relieve this imbalance problem, an effective Long-Tailed Visual Recognition (LTVR) framework is proposed based on learned balance and robust features under long-tailed distribution circumstance. In this framework, a plug-and-play Attentive Feature Augmentation (AFA) module is designed to mine class-related and variation-related features of original samples via attention mechanism. Then, those features are aggregated to synthesize fake features to cope with the imbalance of the original dataset. Moreover, a Lay-Back Learning Schedule (LBLS) is developed to ensure a good initialization of feature embedding. Extensive experiments are conducted with a two-stage training method to verify the effectiveness of the proposed framework on both feature learning and classifier rebalancing in the long-tailed image recognition task. Experimental results show that, when trained with imbalanced datasets, the proposed framework achieves superior performance over the state-of-the-art methods.**

***Index Terms*—Image Classification, Long-tailed Distribution, Data Augmentation, Data Synthesizing**

I. Introduction

Long Tailed Visual Recogniton refers to visual recognition with long-tailed label distribution dataset, where head classes (*i.e.*, the minority of classes) occupy the majority of samples, while tail classes (*i.e.*, the majority of classes) have the minority of samples. The conventional Deep Convolutional Neural Network (DCNN) models for visual recognition are developed on distribution-balanced datasets such as CIFAR10 [1], CIFAR100 [1], ImageNet-2012 [2] and MS COCO [3]. Even though these models have achieved a great success in many visual recognition tasks, they always suffer performance collapse on long-tailed data. Due to the extreme imbalance of long-tailed dataset and the lack of samples for most classes, most methods may learn biased and monotonous feature representations. Therefore, it is in urgent need of feature balancing, enhancing and enriching for Long Tailed Visual Recogniton.

The challenges caused by the long-tailed distribution have evoked a series of works on the visual recognition based on imbalanced datasets. As one of the most fundamental visual recognition task, image classification with long-tailed distribution has attracted an increasing research attention. The existing works for long-tailed image classification can be mainly divided into three categories: re-balancing methods [4], [5], [6], [7], [8], [9], [10], transfer learning methods [11], [12], [13] and ensemble methods [14], [15], [16]. For re-balancing methods, re-sampling and re-weighting are effective to relieve the imbalance problem of datasets. However, by under-sampling head classes or over-sampling tail classes, re-sampling methods may cause underfitting for head classes or overfitting for tail classes. Besides, re-weighting methods often require the sample distribution, which is nearly impossible for online and streaming data. Even worse, extensive experimental results have demonstrated that re-sampling and re-weighting methods are likely to learn suboptimal image representations [14], [15]. For transfer learning methods, they carefully design complicated modules to transfer knowledges, but might not be able to combine with other methods. For ensemble methods, multi-branch models contribute to mining the complementation among different branch models, but may not specifically improve the recognition performance of single branch model. Considering the drawbacks of these previous works, it is non-trivial to explore method for long-tailed feature learning of single-branch, which can not only relive the extreme imbalance without damaging feature embeddings, but also be convenient to combine with other methods.

In this paper, an effective Long-Tailed Visual Recognition (LTVR) framework is proposed to boost image classification and feature learning by generating new features with rich diversities. Specifically, a novel Attentive Feature Augmentation (AFA) module is designed in order to denoise, enhance and enrich feature embeddings, which consists of three main components: feature decomposition, attention addition and self-adapted feature generation. Firstly, original features extracted by the feature extractor are decoupled into class-related features and variation-related features via an attention mechanism. Secondly, the class-related features are added to original features to enhance the representation capabilities of original features. Finally, class-related and variation-related features are used to synthesize new features for each class, and the number of generated features is determined by the frequency of each class in a mini-batch. In addition, a straightforward Lay-Back Learning Schedule (LBLS) is proposed to ensure good initial feature embeddings for AFA module, which is vital for the following feature augmentation.

Extensive experiments are conducted on public benchmarks such as CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT. The elaborate performances on three subsets of

Weiqiu Wang, Pingyu Wang, Zhicheng Zhao and Fei Su are with Beijing Key Laboratory of Network System and Network Culture, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. (e-mail: wangweiqiu@bupt.edu.cn; applewangpingyu@bupt.edu.cn; zhaozc@bupt.edu.cn; sufei@bupt.edu.cn)

Hongying Meng is with the College of Engineering, Design, and Physical Sciences, Brunel University London, Uxbridge, United Kingdom. (e-mail: hongying.meng@brunel.ac.uk)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

these datasets are reported as well. The three subsets are Many-shot(more than 100 training samples for each class), Medium-shot(20∼100 training samples for each class) and Few-shot(less than 20 images for each class). In order to further analyze the effectiveness of our proposed method, the performances of our framework for two-stage training [15], containing feature learning and classifier adjusting stages are given. Experimental results show that when trained with imbalanced datasets, the proposed framework achieves superior performance over the state-of-the-art methods.

In summary, our contributions are below:

1) We propose Class-Aware Feature Decomposition (CAFD) method to decompose features into class-related and variation-related features through attention mechanism.
2) We introduce a Self-adapted Feature Generation (SFG) sub-module and achieve significant performance improvements. The sub-module sythesizes diverse fake features to enrich features for each class and relieve the class-imbalance between head and tail classes.
3) We design a simple but effective strategy, Lay-Back Learning Schedule (LBLS), to train the network ensuring the quality of initial feature embeddings.
4) The proposed end-to-end Long-Tailed Visual Recognition (LTVR) framework benefits for both feature learning and classifier rebalancing in long-tailed image classification. It achieves superior performance over the previous state-of-the-art methods.

The remaining sections are organized as follows. We firstly review previous works related to long-tailed visual recognition in Section II. Then, our prorosed Long-Tailed Visual Recognition (LTVR) framework is described in Section III. In Section IV, experimental results on three benchmarks are presented. Finally, Secion V shows the results of some ablation studies to verify the effectiveness of our proposed method further and Section VI is the conclusion.

II. Related Works

A. Long-tailed Recognition

Long-tailed distribution is prevailed in many manually-collected datasets [17], [18]. Therefore, there are many works focusing on solving the imbalance problem in the image recognition task. The existing long-tailed image recognition methods can be roughly divided into three categories: rebalancing methods, augmentation methods and ensemble methods.

Following the data processing pipeline, the rebalancing methods can be further divided into three types: re-sampling data [4], [5], [19], [20], [21], [22], re-weighting loss [6], [7], [8], [9], [10], [17], and re-adjusting classifier [23], [15], [24].

Re-sampling data methods try to rebalance the data distribution via under-sampling head classes [22], [5] or over-sampling tail classes [22], [21], [4]. The over-sampling manner samples tail data repeatedly, which enables the classifier to learn tail classes better. But duplicated tailed samples might lead to over-fitting upon tail classes [14]. The under-sampling manner decreases sampling frequencies for head classes, which may lose discriminative information on head classes.

In practice, re-weighting loss methods [6], [7], [8], [9], [10] assign larger weights to tail classes than head classes in loss terms. However, when the samples of tail classes are far from sufficient to recover the true distribution, the performance of these methods deteriorates [6].

Although re-sampling and re-weighting methods contribute to the classifier learning, they may hamper the feature learning [15], [14]. In addition, both sampling-based and loss-based re-balancing methods aim to focus on learning tail classes, but they may lead to overfitting upon tail classes [25]. Several works show that re-sampling only on classifier learning stage [15] or directly re-adjusting classifier weights are more effective [15], [24].

Unlike re-balancing methods with many variants, feature learning of long-tailed data lacks exploration. As the base of image recognition, feature learning becomes the bottleneck of the long-tailed visual recognition. When trained with long-tailed data, image recognition models are unable to learn balanced and robust features because of the dominance of head classes and the insufficient diversity of tail classes. In order to take full advantage of head classes, several works [11], [12], [26] transfer knowledges from head classes to tail classes to improve the representation capability of features from tail classes. For example, the representative approaches are transferring knowledges with memory module [27], [28], transferring feature distribution from head to tail [13], [17] and generating fake data [29], [4]. However, these methods usually design specific but complicated modules for feature transfer [15] and do not have an effective guide for the transfer process [25].

With inadequate data, the predicted distributions of tail classes are unreliable. One effective solution is to ensemble multiple models for better classification decisions. Binary Branches Network (BBN) [14] uses two branches to focus on head classes and tail classes, respectively. RIDE [15] dynamicly ensembles multiple branches by routing diverse distribution-aware experts. In addition, LFME [16] learns from multiple experts by self-paced knowledge distillation. In most cases, each single branch of the ensemble models performs worse than the baseline and the ensemble process is complicated or hard to optimize. In addition, there are also many other methods adopting different learning strategies to cope with the long-tailed visual recognition, *e.g.*, mixup [30], metric learning [31], meta learning [32]. However, they are beyond the scope of this article.

B. Data Augmentation

According to the law of large numbers, it requires many samples to estimate the mean value of real data distribution. Most deep learning methods are data-hungry. In many visual recognition tasks, data augmentation methods adopt linear transformation strategies and nonlinear generation modules to enhance the diversity of training samples, which contributes to boosting the generalization capabilities of recognition models. Specifically, these works can be divided into image-level augmentation and feature-level augmentation. Inspired by Generative Adversarial Networks (GANs) [33], several image-level

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

augmentaion works [34], [35] construct generation models to synthesize new images to enlarge the amount of training data. However, these generation models evidently increase the computation and memory costs, and there is no guarantee of visual quality of generated images. One more effective way is to conduct data augmentation in feature level [4], [36]. Our proposed AFA module adopts this way to balance long-tailed data.

C. Attention Mechanism

In image recognition tasks, attention mechanism is an efficient and effective way to gain better feature representations, focusing on the features related to final prediction and filtered some noise. There are many variants of attention mechanism in visual recognition and it can be divided into three types: spatial attention [37], [38], [39], channel attention [40] and the combination of them [41], [42]. Spatial attention mechanism aims to focus on the foreground of images, while channel attention explicitly models interdependencies between channels by adaptively recalibrating channel-wise features. CBAM [42] and GCNet [41] combine spatial attention and channel attention to strengthen the representation power of the model. It is worthy to note that self-attention mechanism is proved to be very effective in natural language processing (NLP) [43], [44]. For example, BERT [43] mainly adopts self-attention mechanism and achieves state-of-the-art performances on eleven NLP tasks, which gives rise to the surge of self-attention based networks for natural language processing. Recently, some works [45], [46] show that self-attention mechanism is beneficial to visual recognition. However, these works split images into patches, which is likely to destroy the spatial structure of visual objects and not convenient for various resolution images. Therefore, we take the image feature patterns as the counterpart of tokens in sentences rather than split images into several patches. Our proposed method constructs hierarchical channel attention to extract class-related features by fusing self-attention and global channel attention.

III. Proposed Method

In this section, we firstly introduce our Long-Tailed Visual Recognition (LTVR) framework addressing image classification with long-tailed distribution. Then, the key components of LTVR framework are described in detail. Challenges in long-tailed visual recognition mainly arise from the extreme imbalance data distribution and the lack of diverse samples for tail classes. Under these situations, the feature embeddings learned in recognition models are dominanted by head classes and the classifier is biased toward head classes as well, which make it hard to discriminate tail classes. Our proposed framework focuses on improving feature embeddings learning and rebalancing classifier at the same time by relieving the imbalance of data distribution and increasing the diverse features for tail classes. Specifically, in feature space, considering that visual features of different kinds of objects consist of different combinations of various feature patterns, our method is designed to decompose an image from feature pattern view instead of physical patch view [45]. We obtain different patterns of features via different convolution kernels, which naturally decompose an image into a group of feature patterns. Furthermore, our Attentive Feature Augmentation (AFA) module conducts feature denoising, feature enhancing and feature enriching via Class-Aware Feature Decomposition (CAFD), attentive feature addition and Self-adapted Feature Generation (SFG), respectively. Besides, to ensure the quality of initial feature embeddings for AFA module, the Lay-Back Learning Schedule (LBLS) is proposed. Table I shows the main notations used in this paper.

TABLE I: NOTATIONS AND DEFINITIONS.

Notations	Definitions
$\boldsymbol{X}_{ori}$	original features extracted by the feature extractor for a batch of input images
$\boldsymbol{X}_{class}$	class-related features decomposed from original features
$\boldsymbol{X}_{var}$	variation-related features decomposed from original features
$\boldsymbol{X}_{enh}$	enhanced features for input images
$\boldsymbol{X}_{gen}$	generated features based on class-related features and variation-related features
L_{ori}	the classification loss of enhanced features
L_{all}	the classification loss of the concatenation of enhanced features and generated features
L_{LBLS}	the classification loss computed by LBLS

A. Overall Framework

As shown in Fig. 1, the proposed LTVR framework consists of three main modules: Backbone Network, Attentive Feature Augmentation (AFA) and Lay-Back Learning Schedule (LBLS). Backbone Network includes the feature extractor and classifier head. Specifically, feature extractor is applied to extract original features $\boldsymbol{X}_{ori}$ (*i.e.*, real features). The subsequent AFA module decouples original features into class-related features $\boldsymbol{X}_{class}$ and variation-related features $\boldsymbol{X}_{var}$, which are used to synthesize new features $\boldsymbol{X}_{gen}$ (*i.e.*, fake features). Meanwhile, class-related features are added to original features to form enhanced features $\boldsymbol{X}_{enh}$. Finally, the concatenation of enhanced features $\boldsymbol{X}_{enh}$ and generated features $\boldsymbol{X}_{gen}$ are fed to the classifier head to obtain the final prediction. When training, the predictions are used to compute loss with Lay-Back Learning Schedule (LBLS) driving the network to learn. During inferencing, we drop the feature generation sub-module and LBLS module, just using the backbone network (*e.g.*, Resnet [47], ResNext [48]) and feature decomposition sub-module to output the predictions.

B. Backbone Network

Backbone network refers to the feature extractor and classifier head in Fig. 1. It is canonical residual network (*e.g.*, Resnet [47], ResNext [48]). Taking ResNext-50 as an example, we use the first three residual blocks as feature extractor to extract original features from input images. The last residual block and the last fully connected layer of ResNext-50 constitute a classifier head to make the final predictions. In other words, we insert AFA module between the third and the fourth residual block of backbone network to do feature denoising, enhancing and enriching.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

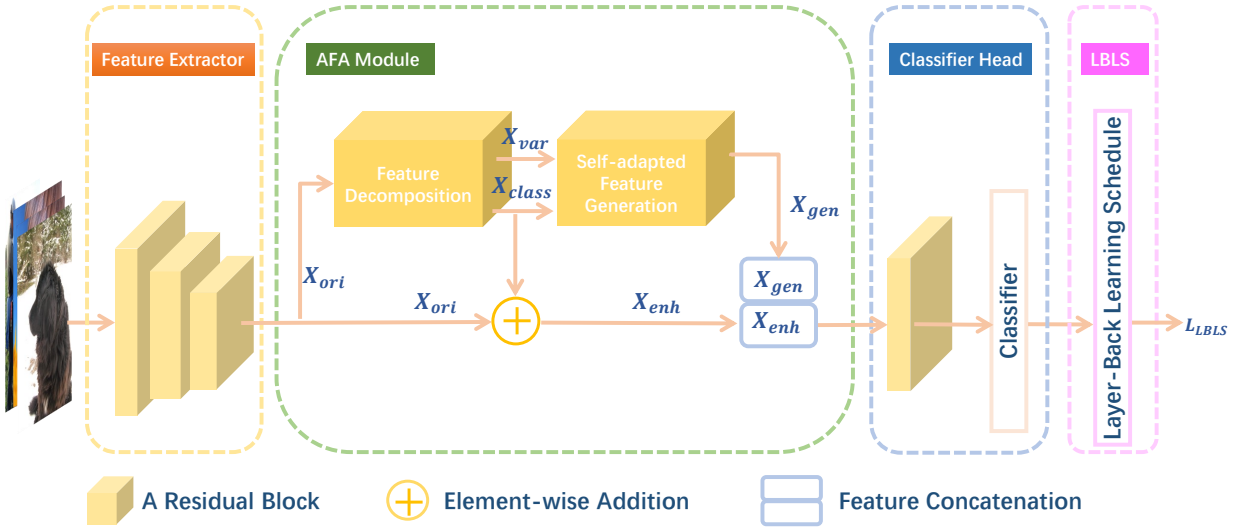


Fig. 1: Long-Tailed Visual Recognition (LTVR) framework. The Feature Decomposition, the Self-adapted Feature Generation sub-modules and the Lay-Back Learning Schedule (LBLS) are detailed in Section III-C1, Section III-C2 and Section III-D, respectively.

C. Attentive Feature Augmentation

1) Feature Decomposition: In long-tailed image classification, the first important issue is the lack of samples for tail classes. Therefore, it is necessary to transfer knowledge from head classes with sufficient samples to enhance the diversity of features for tail classes. However, knowledge extraction is intractable [29], [13]. As for the image classification, class-related features should not be transferred to other classes, while variation-related features can be viewed as common attributes shared between different classes. Motivated by the self-attention [44], we propose an Attentive Feature Augmentation (AFA) module to decouple image features into class-related features and variation-related features via an attention mechanism. From feature patterns decomposition aspect, each channel of the feature maps of an image is viewed as the counterpart to the words in a sentence. In practice, we firstly attend on original features by computing self-related coefficients of themselves. Specifically, for each batch, different linear transforms (*e.g.* fully connected layers or convolution layers with 1×1 convolution kernel) are conducted on flattened original features to get query features $\boldsymbol{Q} \in \mathbb{R}^{B\times D\times(H\times W)}$, key features $\boldsymbol{K} \in \mathbb{R}^{B\times D\times(H\times W)}$ and value features $\boldsymbol{V} \in \mathbb{R}^{B\times D\times(H\times W)}$. Here, (H, W) is the resolution of $\boldsymbol{X}_{ori} \in \mathbb{R}^{B\times D\times H\times W}$, D is the number of channels of original features and B is the batch size. In Equation 1, $softmax(\cdot)$ represents a softmax function. $\boldsymbol{d}_k$ denotes the embedding dimension of key features, which equals $(H \times W)$ in our method.

$$\boldsymbol{X}_{att} = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

Through Equation 1, features with different feature pattern as the center are obtained and denoted as $\boldsymbol{X}_{att} \in \mathbb{R}^{B\times D\times(H\times W)}$, which combines features from different channels. For example, feature of the first channel in $\boldsymbol{X}_{att}$ is the weighted combination of features of all channels of value features $\boldsymbol{V}$, and the weighted factors are correlation coefficients between the first channel of $\boldsymbol{Q}$ and all channels of $\boldsymbol{K}$, respectively.

Since key features usually consist of different visual patterns, we adopt SE-Block [40] to assign different weights to channels of $\boldsymbol{X}_{att}$, which decide their contributions to the model prediction. Besides, due to the instinct of the convolution operating within a local receptive field, we can capture the global information of $\boldsymbol{X}_{att}$ as well with the help of SE-Block as compensation. Therefore, re-weighted $\boldsymbol{X}_{att}$ is formulated as:

$$\widetilde{\boldsymbol{X}}_{att} = \boldsymbol{F}_{se}\left(\boldsymbol{X}_{att}\right), \tag{2}$$

where $F_{se}(\cdot)$ denotes the mapping function of the SE-Block.

After obtaining $\widetilde{\boldsymbol{X}}_{att}$ and $\boldsymbol{X}_{att}$, we aggregate these two features to predict the decomposition mask $\boldsymbol{M}$:

$$\boldsymbol{M} = softmax(\widetilde{\boldsymbol{X}}_{att} + \boldsymbol{X}_{att}). \tag{3}$$

As the mask $\boldsymbol{M}$ mainly captures the specific combination of feature patterns for each sample, this mask is adopted to extract class-related features via an attention mechanism:

$$\boldsymbol{X}_{class} = \boldsymbol{M} \odot \boldsymbol{V}. \tag{4}$$

Suppose class-related and variation-related features are linearly mixed together, we acquire variation-related features by subtracting class-related features from original features as follows:

$$\boldsymbol{X}_{var} = \boldsymbol{V} - \boldsymbol{X}_{class}. \tag{5}$$

In order to take full advantage of class-related information, we fuse class-related features and original features $\boldsymbol{X}_{ori}$ to reduce the noise of original features. The enhanced features $\boldsymbol{X}_{enh}$ can be formulated as:

$$\boldsymbol{X}_{enh} = \boldsymbol{X}_{ori} + \boldsymbol{X}_{class}. \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

2) Self-adapted Feature Generation: Another important issue in long-tailed visual recognition is the extreme imbalance of data distribution. With the extreme imbalance, the final feature representations of tail classes are biased to head classes [24], which reduces the discrimination of tail classes. Therefore, we intend to synthesize samples with smaller imbalance ratio to relieve the imbalance of original dataset, so that weaken the dominance of head classes. Moreover, these synthetic samples enrich the diversity of features for tail classes. In practice, we simply sample training images with instance-balanced sampling strategy [15], where every training sample has the same selection probability as follows:

$$p = \frac{1}{N} = \frac{1}{\sum_{i=1}^{C} n_i} \tag{7}$$

where N is the total number of samples in the training set, n_i denotes the number of training samples belonging to the i-th class and C is the number of classes. Consequently, the probability of sampling an image from class i is given by:

$$p_i = \frac{n_i}{\sum_{j=1}^{C} n_j} \tag{8}$$

It is easy to find that in the long-tailed recognition, the samples of head classes with more samples have larger probability to appear in a batch. Therefore, the frequency of each class in a batch can be used to control the synthesizing strategy, making the synthesize frequency biased towards to the classes with fewer samples in each batch. To be specific, the synthesizing control factor R_c is obtained by:

$$R_c = \frac{\left(\frac{1}{f_c}\right)^{\alpha}}{\sum_{c\in\mathbb{C}}\left(\frac{1}{f_c}\right)^{\alpha}}, \tag{9}$$

where f_c is the frequency of the c-th class in a mini-batch, α is the factor controlling the probability distribution of generated features, and $\mathbb{C}$ is the set of class indexes in a mini-batch. The number of the generated features for the c-th class is:

$$G_c = \lceil \gamma R_c B \rceil \qquad \forall c \in \mathbb{C}, \tag{10}$$

where γ is the ratio of generated features to original features and $\lceil \cdot \rceil$ is a ceiling operator.

According to the number of generated features in each class, we randomly select the i-th sample from the class-related features of one certain class c, and the selected features are denoted as $\boldsymbol{X}_{class}^{(i,c)}$. Correspondingly, we also randomly pick the j-th sample from variation-related features, denoted as $\boldsymbol{X}_{var}^{(j)}$. Then, these two features are aggregated to generate new features:

$$\boldsymbol{X}_G^{(g,c)} = \boldsymbol{X}_{class}^{(i,c)} + \boldsymbol{X}_{var}^{(j)}, \tag{11}$$

where $g \in [1, G_c]$, $0 \le i,\ j \le B$ and $i \ne j$. $\boldsymbol{X}_G^c$ is the generated features for class c appeared in the batch, $\boldsymbol{X}_G^c = [\boldsymbol{X}_G^{(1,c)}, \boldsymbol{X}_G^{(2,c)}, \ldots, \boldsymbol{X}_G^{(G_c,c)}]$. To sum up, the pseudo code of AFA module is presented in Algorithm1.

Algorithm 1: Attentive Feature Augmentation (AFA)

Input : A batch of input features extracted by feature extractor: $\boldsymbol{X}_{ori} \in \mathbb{R}^{B\times D\times H\times W}$;
Generated ratio: γ;
Imbalance control factor: α;
Labels for input features: $\boldsymbol{Y} \in \mathbb{R}^{B}$

Output: Class-realated features: $\boldsymbol{X}_{class} \in \mathbb{R}^{B\times D\times H\times W}$;
Generated features: $\boldsymbol{X}_{gen} \in \mathbb{R}^{\left(\sum_{c\in\mathbb{C}}(\lceil\gamma R_c B\rceil)\right)\times D\times H\times W}$;
Labels for generated features: $\boldsymbol{Y}_{gen} \in \mathbb{R}^{\sum_{c\in\mathbb{C}}(\lceil\gamma R_c B\rceil)}$

$\widehat{\boldsymbol{X}}_{ori} \in \mathbb{R}^{B\times D\times(H\times W)} \leftarrow Reshape\left(\boldsymbol{X}_{ori}\right)$;
$\boldsymbol{Q} = \widehat{\boldsymbol{X}}_{ori}\boldsymbol{W}_Q;\ \boldsymbol{W}_Q \in \mathbb{R}^{B\times(H\times W)\times(H\times W)}$;
$\boldsymbol{K} = \widehat{\boldsymbol{X}}_{ori}\boldsymbol{W}_K;\ \boldsymbol{W}_K \in \mathbb{R}^{B\times(H\times W)\times(H\times W)}$;
$\boldsymbol{V} = \widehat{\boldsymbol{X}}_{ori}\boldsymbol{W}_V;\ \boldsymbol{W}_V \in \mathbb{R}^{B\times(H\times W)\times(H\times W)}$;
// $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ have the same dimensions with $\widehat{\boldsymbol{X}}_{ori}$
$\boldsymbol{X}_{att} = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V}$;
$\boldsymbol{X}_{att} \in \mathbb{R}^{B\times D\times(H\times W)}$;
$\ddot{\boldsymbol{X}}_{att} \in \mathbb{R}^{B\times D\times H\times W} \leftarrow Reshape\left(\boldsymbol{X}_{att}\right)$;
$\widetilde{\boldsymbol{X}}_{att} = \boldsymbol{F}_{se}\left(\ddot{\boldsymbol{X}}_{att}\right);\ \widetilde{\boldsymbol{X}}_{att} \in \mathbb{R}^{B\times D\times H\times W}$;
$\boldsymbol{M} = softmax\left(\widetilde{\boldsymbol{X}}_{att} + \ddot{\boldsymbol{X}}_{att}\right)$;
$\boldsymbol{X}_{class} = \boldsymbol{M} \odot \boldsymbol{V}$;
$\boldsymbol{X}_{var} = \boldsymbol{V} - \boldsymbol{X}_{class}$;
// To generate features for a batch
for c *in* $\mathbb{C}$ **do**
  $f_c \leftarrow$ the frequency of class c in labels $\boldsymbol{Y}$;
  $R_c = \frac{\left(\frac{1}{f_c}\right)^{\alpha}}{\sum_{c\in\mathbb{C}}\left(\frac{1}{f_c}\right)^{\alpha}}$;
  $G_c = \lceil \gamma R_c B \rceil$;
  for $g \leftarrow 1$ **to** G_c **do**
    // $0 \le i,\ j \le B$ and $i \ne j$
    $\boldsymbol{X}_G^{(g,c)} = \boldsymbol{X}_{class}^{(i,c)} + \boldsymbol{X}_{var}^{(j)}$;
  end
  for $g \leftarrow 1$ **to** G_c **do**
    $\boldsymbol{X}_G^c \in \mathbb{R}^{G_c\times D\times H\times W} \leftarrow Stack\left(\boldsymbol{X}_G^{(g,c)}\right)$;
  end
  $\boldsymbol{Y}_G^c \in \mathbb{R}^{G_c} \leftarrow c$
end
$\boldsymbol{X}_{gen} \leftarrow Stack\left(\boldsymbol{X}_G^c\right) \quad \forall c \in \mathbb{C}$;
$\boldsymbol{Y}_{gen} \leftarrow Stack\left(\boldsymbol{Y}_G^c\right) \quad \forall c \in \mathbb{C}$;
Return $\boldsymbol{X}_{class}$, $\boldsymbol{X}_{gen}$ *and* $\boldsymbol{Y}_{gen}$

D. Lay-Back Learning Schedule

The previous study of long-tailed visual recognition has found that re-weighting and re-sampling methods surprisedly damage feature learning [15], [14], [7]. Accordingly, it is necessary for image recognition models based on re-weighting and re-sampling methods to learn a good initial representation [7]. Similarly, the proposed AFA module also requires a good initialization to precisely decouple image features. In this work, we propose a new Lay-Back Learning Schedule

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

(LBLS) in order to ensure the quality of generated features. At the beginning of the training stage, recognition models with LBLS can be trained with original samples for some epochs to guarantee the quality of feature decomposition. Formally, the proposed LBLS can be easily implemented with loss L_{LBLS}:

$$L_{LBLS} = \begin{cases} L_{ori}, & \text{if } 0<e<t \\ L_{ori} + \lambda L_{all}, & \text{if } t \leq e \leq T \end{cases}, \tag{12}$$

where e means the current training epoch, t is the threshold epoch of using the loss of merged features with weight λ, T is the total number of training epochs, L_{ori} is the classification loss of the original samples, and the L_{all} is the classification loss of the concatenation of synthetic features and enhanced features. In the following experiments, we adopt Cross-Entropy [48] as our loss function for image classification.

IV. Experiments

To verify the effectiveness of the proposed method, extensive experiments are conducted on three datasets including Long-tailed CIFAR-10 and CIFAR-100 [1], [7] and ImageNet-LT [27]. Moreover, we report the results of two stage (the feature learning stage and the classifier adjusting stage) on the whole dataset and three subsets (many-shot, medium-shot, few-shot).

A. Dataset

Long-Tailed CIFAR-10 and CIFAR-100 [1], [7]: The canonical balanced CIFAR-10 and CIFAR-100 datasets contain the same 50,000 training images and 10,000 validation images with a resolution of 32×32. With further fine annotation on these images, CIFAR-100 has 100 classes while CIFAR-10 has 10 classes. We adopt the method in [7] to exert long-tailed distribution on the original balanced dataset with the controllable imbalance ratio β, which is the ratio of the most frequent class to the least frequent class, *e.g.*, $\beta = \frac{N_{max}}{N_{min}}$. The larger the imbalance ratio β is, the more difficult for feature learning is. In our experiments, we explore on the relative severe imbalanced CIFAR-10 and CIFAR-100 with imbalance ratio 100, which denote as CIFAR-10-IR100 and CIFAR-100-IR100, respectively.

ImageNet-LT [27]: ImageNet-LT is the subset of ImageNet-2012 [2], which is sampled by the Pareto distribution with the power value $\alpha = 6$ to satisfy the long-tailed requirement. ImageNet-LT totally has 115.8K images for 1000 categories and the number of samples per class ranges from 5 to 1280. The test set is balanced.

B. Implementation Details

Network Architecture: For all experiments, our backbone network ResNext-50 [48] is trained from scratch by default. Without specific description, we apply our AFA module between the third block and the fourth block of ResNext-50. In our experiments, the ratio of batch size and learning rate are consistent with the strong baseline [15].

Hyper-parameters Setting: Without any careful adjustment of hyper-parameters for different datasets, we adopt the same setting of hyperparameters for different datasets. In our experiments, the ratio of generated samples to original samples is set as 1.0, and the factor α is set as 4.

Classifier Adjustment: The previous works [15], [14] have found that in the long-tailed image classification, the L_2 norms of the classifier of neural network is consistent with the long-tailed distribution of the dataset, which means the more samples of a class, the larger magnitude of the class weight is, and vice versa. Therefore, classifier weight adjusting is important. The simplest way is to retrain the classifier with class-balanced sampling [15]. With adjusting classifier, we can explore the improvements of the feature learning brought by our proposed method excluding the effect of re-balancing classifier, which is necessary but not reported in most of the previous work in long-tailed image classification.

Training/Testing Configurations: We apply the effective two-stage training on all experiments. Without specific declaration, for the first training stage, the instance-balanced sampling is adopted as default, which means random sampling from datasets without any specific design. In the second stage, we sample the dataset by a class-balanced sampling strategy to retrain the classifier only, aiming to get a relative balanced classifier. We set the epoch $t = 30$ to add the loss of the concatenate features with weight factor $\lambda = 2$ in the first stage and the total number of training epochs of the first stage is 90. We retrain the classifier for 10 epochs. We use SGD optimizer with weight decay 0.0005, momentum 0.9 and cosine learning rate scheduler [49] gradually decaying from initial learning rate to 0. To make fair comparison, the training hyperparameters are the same with the strong baseline [15], which is also the state-of-the-art result of the single-branch model in long-tailed image classification.

C. Comparison with State-of-the-Art Methods

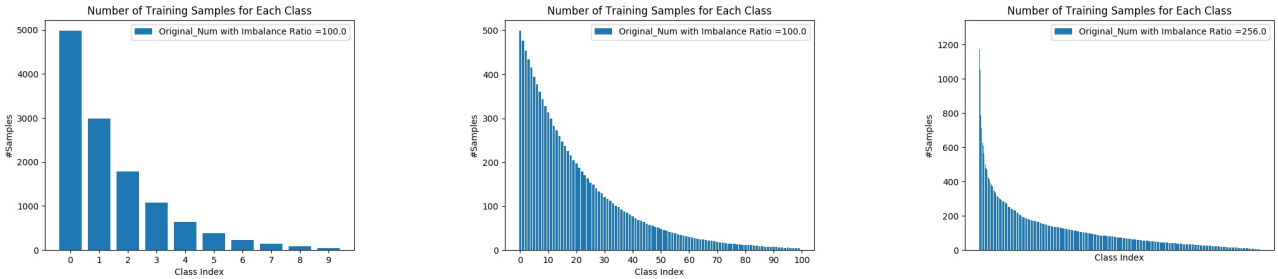
We evaluate our proposed LTVR framework on three long-tailed benchmarks including CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT for image classification. In long-tailed visual recognition, the common setting is training on long-tailed datasets and testing on the corresponding balanced test datasets. We report top-1 accuracy on all benchmarks, denoted as *ALL*. To explore the elaborate performance of the model on these benchmarks, we split the test dataset into three subsets according to the number of training samples for each class: Many-shot(more than 100 training samples), Medium-shot(20$\sim$100 training samples) and Few-shot(less than 20 images) [27]. Top-1 accuracy on the three subsets are denoted as *Many*, *Medium* and *Few*, respectively.

In Table II, we compare our proposed framework with current state-of-the-art methods on the long-tailed CIFAR10 with imbalance ratio 100. Following the work [7], we get the distribution of the number of training samples of CIFAR10_LT as shown in Fig. 2a. Specifically, the numbers of training samples of all classes are larger than 20, therefore, top-1 accuracy on Few-shot subset is zero. The first two rows are the first stage results of baseline and our proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology

TABLE II: Results on **CIFAR-10** with imbalance ratio=100. "Baseline" represents the canonical backbone Resnext-50 in end-to-end training style. "LTVR" denotes that our proposed framework. "Crt" means retraining the classifier only with class-balanced sampling for the dataset. Therefore, we denote the two-stage training results of the baseline and our framework as "Baseline + Crt" and "LTVR + Crt", respectively.

Method	All	Many	Mediumt	Few
Baseline	68.3	75.7	38.6	0
LTVR	69.2	77.1	37.4	0
Baseline(PB re-sampling [15])	71.8	76.7	52.3	0
Baseline(CB re-sampling [15])	66.0	70.4	48.4	0
Baseline(SR re-sampling [15])	72.3	78.0	49.5	0
Baseline(PB re-sampling [15])+Crt	73.1	77.1	56.8	0
Baseline(CB re-sampling [15])+Crt	66.4	73.3	39.0	0
Baseline(SR re-sampling [15])+Crt	75.2	79.6	57.9	0
Baseline + Crt	75.8	77.0	71.2	0
LTVR + Crt	78.2	77.3	81.8	0



(a) CIFAR-10-IR100 (b) CIFAR-100-IR100 (c) ImageNet_LT

Fig. 2: The number of training samples for each class in CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT. Followed the previous work [7], the imbalance ratio of dataset is obtained by $\frac{N_{max}}{N_{min}}$, where N_{max} and N_{min} are the frequencies of the most and the least frequent class, respectively.

TABLE III: Results on **CIFAR-100** with imbalance ratio=100. The set of notations is same with the ones in table II.

Method	All	Many	Medium	Few
Baseline	37.9	66.5	37.5	5.0
LTVR	42.0	69.7	41.6	10.1
Baseline(PB re-sampling [15])	41.1	64.3	42.6	12.5
Baseline(CB re-sampling [15])	36.8	60.9	36.9	8.6
Baseline(SR re-sampling [15])	39.2	65.3	38.3	9.8
Baseline(PB re-sampling [15])+Crt	41.3	63.9	43.1	13.0
Baseline(CB re-sampling [15])+Crt	36.8	60.7	36.3	9.6
Baseline(SR re-sampling [15])+Crt	41.9	63.8	43.2	14.7
Baseline + Crt	44.7	60.6	45.8	24.9
LTVR + Crt	46.8	63.5	47.4	26.5

LTVR framework. In order to explore the improvements of feature learning, we adopt two-stage training schedule in our experiments. With re-training the classifier with class-balanced sampling, namely, Crt [15], we can get a relative balanced classifier. Under the circumstance, the previous re-sampling and re-weighting methods cease to be effective, even perform worse than the basic instance-sampling method, which is shown in Table II. We denote the baseline with re-sampling methods [15] (progressive-balanced sampling, class-balanced sampling, square-root sampling) as Baseline(PB re-sampling), Baseline(CB re-sampling), Baseline(SR re-sampling), respectively. For the first training stage, the performances of the PB re-sampling and SR re-sampling are better than our default baseline with instance-sampling, but the performance of CB re-sampling is worse than our default baseline. It shows that it is possible to improve the performance on long-tailed visual recognition with appropriate design of re-sampling. But when stepping further with Crt, the performance gains of re-sampling methods disappear. However, our proposed LTVR framework performs better continuously, even combined with Crt. Compared to state-of-the-art method Crt, our module still can improve the performance of model by a large margin 2.4%. By analyzing the results of the experiments, we can conclude two key points: (1) Our proposed method can balance the classifier to some extent. (2) With our proposed module, the feature learning is improved as well based on the simple instance-sampling.

The results of experiments on CIFAR-100-IR100 are given in Table III. The number of training samples for each class is illustrated in Fig. 2b. The first two rows of Table III are the results of baseline and our LTVR framework in the first feature learning stage. And the last rows of Table III are the results of

TABLE IV: Results on **ImageNet_LT** with imbalance ratio=100. To make fair comparison, all models were using the ResNeXt-50 as backbone. Results marked with ♠ are copied from Decouple [15]. Results marked with ♣ are copied from Decouple [24].

Method	All	Many	Medium	Few
♠Cross Entropy(CE) [48]	44.4	65.9	37.5	7.7
♣Focal Loss [50]	43.7	64.3	37.1	8.2
♣Cosine [51], [52]	47.6	67.3	41.3	14.0
♣Capsule [27], [53]	46.5	67.1	40.0	11.2
♣De-confound [24]	48.6	67.9	42.7	14.7
♣Consine-TDE [24]	50.5	61.8	47.1	30.4
♣Capsule-TDE [24]	50.6	62.3	46.9	30.6
♣De-confound-TDE [24]	51.8	62.7	48.8	31.6
♣OLTR [27]	41.9	51.0	40.8	20.8
♠Cross Entropy(CE) + NCM [15]	47.3	56.6	45.3	28.1
♠Cross Entropy(CE) + Crt [15]	49.6	61.8	46.2	27.4
♠Cross Entropy(CE) + T-norm [15]	49.4	59.1	46.9	30.7
♠Cross Entropy(CE) + LWS[15]	49.9	60.2	47.2	30.3
LTVR	48.9	68.2	42.9	15.3
LTVR + Crt	**52.6**	**64.8**	**49.6**	**28.5**

the state-of-the-art method and our framework combined with the state-of-the-art method in the second classifier adjusting stage. We also report the results of baseline with re-sampling methods for the first and second training stage, which can get the same conclusion stated above. Both for the first feature learning stage and the second classifier adjusting stage, our proposed method gains large margins by 4.1% and 2.7% on the whole dataset, respectively. Besides, with our proposed method, the performance is improved on Many-shot, Medium-shot and Few-shot simultaneously compared with the state-of-the-art method for both the first feature learning stage and the second classifier adjusting stage. As for the ImageNet_LT dataset, the results are shown in Table IV. The number of training samples for each class is shown in Fig. 2c, from which we can see that the imbalance of ImgeNet_LT is severer than the CIFAR-10-IR100 and CIFAR-100-IR100. The results of different state-of-the-art methods in Table IV are separated into four parts. All methods take Resnext-50 [48] as the backbone. The first part of Table IV, denoted as Cross Entropy(CE), is the baseline trained with Cross-Entropy loss for the first feature learning stage. The results in the second part of the table are the methods trained only for single stage. Focal loss [50], the effective method to mine the hard samples, is useless under the long-tailed distribution. The effects of normalized classifiers(the cosine classifier [51], [52] and the capsule classifier [27], [53]) are similar to the second classifier adjusting stage to rebalance the classifier weights [24]. The De-confound and the *-TDE series methods in Table IV introduce the causal effect into the long-tailed visual recognition, which need to put restrictions on both the classifier weights and the features. Even more, these methods need extra memory to save the head direction of learned features and extra computing cost to calculate the projection on the head direction for features of each sample [24]. OLTR [27] method is carefully designed with memory module, which sacrifices too much head performance loss as the expense to obtain the tail performance gain. The results of the effective two-stage methods for long-tailed classification are shown in the third part of Table IV. We take Crt [15] method for our second classifier adjusting stage. Compared to these state-of-the-art methods, it is obvious that our method can rebalance the classifier to some extent, which brings the gain for the first feature learning stage. And the learned feature representations of our framework are better as well, which can be concluded from the gain of LTVR + Crt compared to Cross Entropy(CE) + Crt.

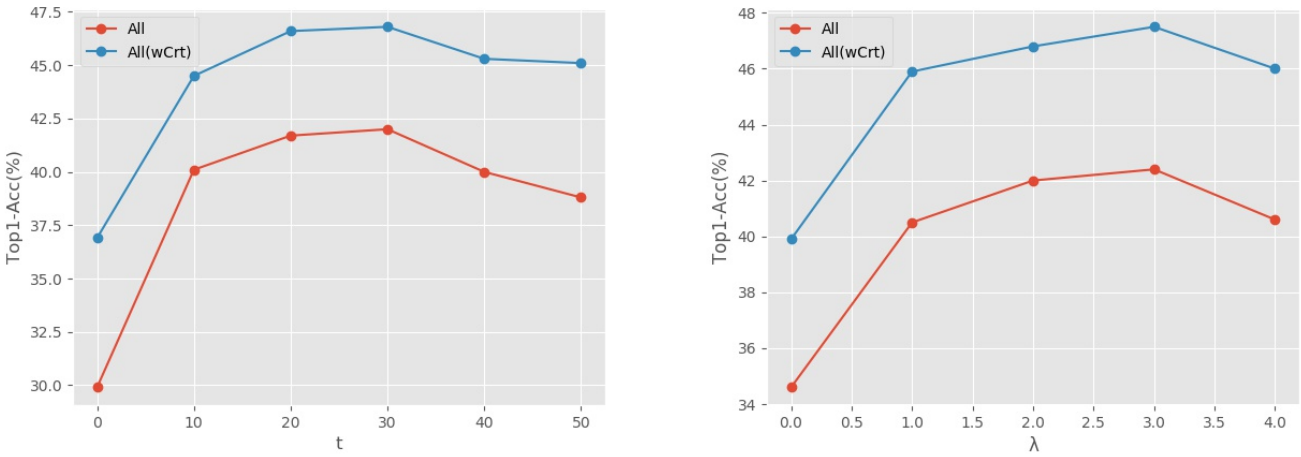
TABLE V: Results of ablation study of attention mechanism in Feature Decomposition sub-module on **CIFAR-100** with imbalance ratio=100. The set of notations is same with the ones in table II. "LTVR(CBAM)" represents replacing Feature Decomposition sub-module in LTVR framework with CBAM. Similarly, "LTVR(GC Blocks)" means replacing Feature Decomposition sub-module in LTVR framework with GC Blocks.

Method	All	Many	Medium	Few
Baseline	37.9	66.5	37.5	5.0
LTVR(CBAM)	34.0	61.7	31.5	4.5
LTVR(GC Blocks)	34.0	61.9	30.7	5.2
LTVR	42.0	69.7	41.6	10.1
Baseline + Crt	44.7	60.6	45.8	24.9
LTVR(CBAM) + Crt	38.9	51.5	39.8	23.2
LTVR(GC Blocks) + Crt	39.0	52.8	38.8	23.1
LTVR + Crt	46.8	63.5	47.4	26.5

D. Ablation Study

Feature Decomposition: To explore the effectiveness of our proposed attentive feature decomposition mechanism, we take CBAM [42] and GC blocks [41] to decompose original features, respectively. CBAM [42] and GC blocks [41] are both effective attention mechanism in visual recognition by combining spatial attention and channel attention. The results are reported in Table V. Our proposed attention mechanism performs better than CBAM and GC blocks in feature de-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology



(a) Initial Epoch t (b) Weight λ

Fig. 3: Analyzing the sensitivity of hyper-parameters t and λ on CIFAR-100-IR100. "All" and "All(wCrt)" denote top1-accuracy (%) on the whole test set in the first feature learning stage and the second classifier adjusting stage, respectively.

TABLE VI: Results of ablation study of Self-adapted Feature Generation Module on **CIFAR-100** with imbalance ratio=100. The set of notations is same with the ones in table II. "LTVR(w/o Gen)" represents our proposed framework without the Self-adapted Feature Generation Module.

Method	All	Many	Medium	Few
Baseline	37.9	66.5	37.5	5.0
LTVR(w/o Gen)	38.0	67.4	36.7	5.1
LTVR	42.0	69.7	41.6	10.1
Baseline + Crt	44.7	60.6	45.8	24.9
LTVR(w/o Gen) + Crt	44.8	60.4	45.5	25.7
LTVR + Crt	46.8	63.5	47.4	26.5

TABLE VII: Ablation study of attention addition on **CIFAR-100** with imbalance ratio=100. The set of notations is same with the ones in table II.

Method	All	Many	Medium	Few
Baseline	37.9	66.5	37.5	5.0
LTVR(w/o AttAdd)	41.2	68.9	40.1	10.2
LTVR	42.0	69.7	41.6	10.1
Baseline + Crt	44.7	60.6	45.8	24.9
LTVR(w/o AttAdd) + Crt	45.8	62.6	46.1	25.9
LTVR + Crt	46.8	63.5	47.4	26.5

composition. It shows that decomposing features form feature pattern view is effective.

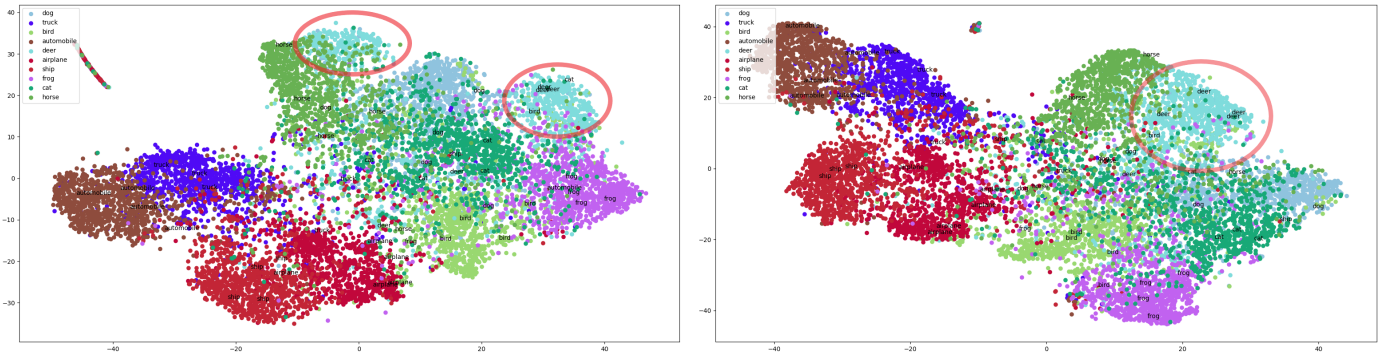
Self-adapted Feature Generation Module: We compare the performances of our proposed method with and without the Feature Generation Module in the first feature learning stage and the second classifier adjusting stage on CIFAR-100-IR100. As shown in Table VI, the Self-adapted Feature Generation Module plays an important role in our proposed AFA module, which can relieve the imbalance of the dataset and rebalance the classifier to some extent. In addition, with the supervision of the synthetic samples, Feature Decomposition sub-module can learn better as well.

Attention Addition: As the product of Feature Decomposition sub-module in our AFA module, class-related features $\boldsymbol{X}_{class}$ are not only the key components of generated feature but also beneficial to original feature. To make full use of it, we add class-related features $\boldsymbol{X}_{class}$ to original features as augmentation. The results in Table VII verify the effectiveness of the augmentation with the addition of class-related features. Adding class-related features to original features can augment the key features related to the final prediction and weaken the noises in samples to gain better feature representations. Besides, in the gradient backward process of the neural network, the addition of class-related features is also a pathway for gradients to backpropagate to Feature Decomposition sub-module for optimization. We denotes our proposed LTVR framework without the addition of class-related features as "LTVR(w/o AttAdd)". Without the addition of class-related features, the performance of our LTVR framework drops in both the first learning stage and the second classifier adjusting stage.

Effectiveness for Lay-Back Learning Schedule: We study the importance of Lay-Back Learning Schedule (LBLS) by adding the loss of synthetic samples at the beginning of the training process. The results of the experiments are shown in Table VIII, denoted as LTVR(w/o layback) and LTVR(w/o layback) + Crt for the first feature learning stage and the second classifier adjusting stage, respectively. It is obvious that the performance of the recognition model drops dramatically without Lay-Back Learning Schedule. It shows that good initial feature representations are crucial for the long-tailed image classification, which is also consistent with the observation of LDAM [7].

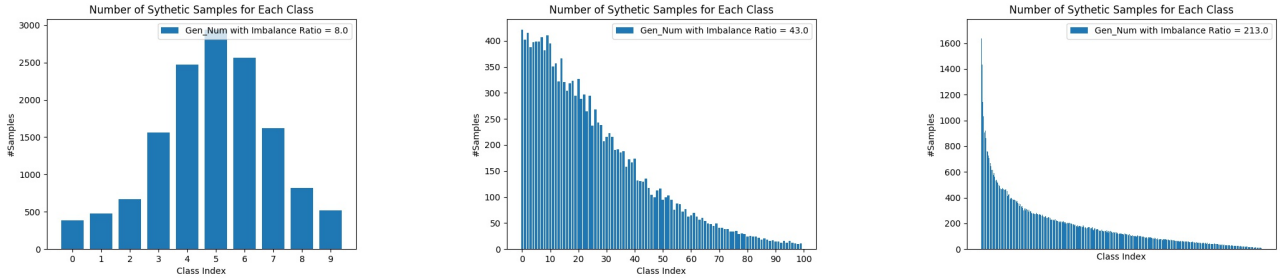
Sensitivity of Hyper-parameters: We also explore the sensitivity of the hyper-parameters t and λ for LBLS, which is illustrated in Fig. 3a and Fig. 3b, respectively. In the first t epochs of the first training stage, the model only learns from original samples, which is associated with the quality of initial feature representations for AFA module. It can be seen that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/tcsvt.2022.3161427, IEEE Transactions on Circuits and Systems for Video Technology



(a) Baseline: features of the test set of CIFAR-10-IR100 (b) Ours: features of the test set of CIFAR-10-IR100

Fig. 4: **Feature embedding.** Feature embedding visualizations of Baseline (denoted as "Baseline" in Table III) and our method (denoted as "LTVR" in Table III). Features embeddings gained by our method are semantically separable compared to Baseline suggesting that feature learning is improved by our proposed method.



(a) CIFAR-10-IR100 (b) CIFAR-100-IR100 (c) ImageNet_LT

Fig. 5: The number of synthetic training samples for each class in CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT.

TABLE VIII: Results of ablation study of the effectiveness of Lay-Back Learning Schedule (LBLS) on **CIFAR-100** with imbalance ratio=100. The set of notations is same with the ones in table II. "LTVR(w/o LBLS)" means the LTVR framework without using Lay-Back Learning Schedule.

Method	All	Many	Medium	Few
Baseline	37.9	66.5	37.5	5.0
LTVR(w/o LBLS)	29.9	59.7	24.7	1.3
LTVR	42.0	69.7	41.6	10.1
Baseline + Crt	44.7	60.6	45.8	24.9
LTVR(w/o LBLS) + Crt	36.9	50.5	36.7	21.2
LTVR + Crt	46.8	63.5	47.4	26.5

with the initial epoch t increasing, the performance of model is improved. However, feature decomposition sub-module needs to be trained with synthetic samples for sufficient iterations as well. So when the initial epoch t occupies a large part of total epochs of the first feature leaning stage, the performance of the recognition model decreases. As for the weight factor λ, Fig. 3b shows the necessity of the loss of the concatenation of synthetic features and enhanced features. It is a trade-off between original samples and synthetic samples.

V. Discussion

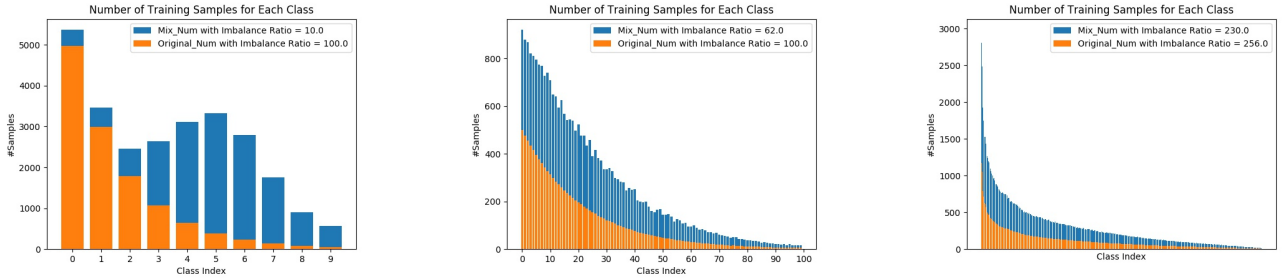
A. Feature Visualization

In this part, we compare the feature embeddings learned by Baseline and our proposed LTVR framework by "t-SNE" [54]. Specifically, we extract the feature embeddings of the test set of CIFAR10_LT(refer to CIFAR-10 dataset with imbalance ratio 100) from the last hidden layer of the model. These features are then projected to 2-dimensional space using t-SNE [54]. The feature embeddings of Baseline and our framework are visualized in Fig. 4a and Fig. 4b, respectively. It shows that the feature embeddings of our method are more semantically separable, which means the center features of each class has larger Euclidean distance with each other in the 2D-dimensional visualized space. In addition, the feature embeddings of our method has smaller intra-class distances which is notably illustrated by "deer" class marked by red circle in Fig. 4a and Fig. 4b. Therefore, the effects of our method for feature embeddings learning meet our expectations for image classification, which expects larger inter-class distances and smaller intra-class distances.

B. The Number of Synthetic Samples

With our self-adapted feature synthesizing strategy, there is no need to know the original data distribution (refer to the number of training samples for each class), which is usually required by the re-sampling and re-weighting methods. However, it is hard to get the data distribution of online and streaming data, which limits the applied scenarios of the re-sampling and re-weighting methods. Using our strategy to synthesize samples, the data distribution of the synthetic samples relies on the batch frequency of each class, which can dynamically adapt to the different dataset under the most common instance-balanced sampling circumstance.

The data distributions of synthetic samples for the three experimental datasets, CIFAR-10-IR100, CIFAR-100-IR100



(a) CIFAR-10-IR100 (b) CIFAR-100-IR100 (c) ImageNet_LT

Fig. 6: Comparison of original and mixed number of training samples for each class in CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT.

and ImageNet_LT, are shown in Fig. 5a, Fig. 5b, and Fig. 5c. The imbalance ratios of the data distribution of the synthetic samples for the three datasets are illustrated on the upper right corner of these figures. Furthermore, we visualize the data distributions of the original CIFAR-10-IR100, CIFAR-100-IR100 and ImageNet_LT dataset and the three datasets with synthetic samples, shown in Fig. 6a, Fig. 6b and Fig. 6c. With this visualization, we can directly see the rebalance effect of our synthetic strategy from the aspect of the number of training samples for each class.

We denote the number of original training samples and synthetic training samples of dataset as "Original_Num" and "Gen_Num", respectively. The sum of the original training samples and the synthetic training samples as "Mix_Num". As illustrated in Fig. 5 and Fig. 6, the synthetic samples have smaller imbalance ratio than the original dataset so that they can relieve the imbalance of the dataset when combined with the original dataset. With the synthetic samples, the imbalance ratio of CIFAR-10-IR100 is reduced from 100.0 to 10.0, the imbalance ratio of CIFAR-100-IR100 is dropped from 100.0 to 62.0, and the imbalance ratio of ImageNet_LT is reduced from 256.0 to 230.0. We also explore to synthesize more data for tail classes, however, the performance is worse than our strategy. Tail classes with few samples lack diversities to sythesize too many diverse fake features. Under this situation, too many synthetic samples for tail classes may lead to overfitting on limited effective training samples.

VI. Conclusion

To cope with the series of challenges in long-tailed image classification, we propose the LTVR framework to relieve the imbalance of the dataset and improve the feature learning as well. The proposed LTVR framework explores the feature initialization, feature denoising, feature enhancing and feature enriching. Extensive experimental results demonstrate that the proposed framework achieves superior performance over the state-of-the-art methods. From our experiments, it is also found that a good initial feature learning is crucial for the follow-up feature augmentation and feature synthesizing. In the future work, we will try to apply our proposed method on more long-tailed visual recognition tasks and dig more into feature decomposition methods and imbalance relieving methods to persistently promote the performance of deep learning in long-tailed visual recognition.

VII. Acknowledgement

This work is supported by Chinese National Natural Science Foundation under Grants 62076033, U1931202.

References

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

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, Ieee, 2009, pp. 248–255. doi:10.1109/cvprw.2009.5206848.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.

[5] C. Drummond, R. C. Holte, et al., C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, in: Workshop on learning from imbalanced datasets II, Vol. 11, Citeseer, 2003, pp. 1–8.

[6] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

[7] K. Cao, C. Wei, A. Gaidon, N. Arechiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, in: Advances in Neural Information Processing Systems, 2019, pp. 1567–1578.

[8] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE transactions on neural networks and learning systems 29 (8) (2017) 3573–3587.

[9] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, J. Yan, Equalization loss for long-tailed object recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11662–11671.

[10] C. Huang, Y. Li, C. C. Loy, X. Tang, Learning deep representation for imbalanced classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5375–5384.

[11] S. Bengio, Sharing representations for long tail computer vision problems, in: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 2015, pp. 1–1.

[12] Y. Cui, Y. Song, C. Sun, A. Howard, S. Belongie, Large scale fine-grained categorization and domain-specific transfer learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4109–4118.

[13] J. Liu, Y. Sun, C. Han, Z. Dou, W. Li, Deep representation learning on long-tailed data: A learnable embedding augmentation perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2970–2979.

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

[15] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, arXiv preprint arXiv:1910.09217 (2019).

[16] L. Xiang, G. Ding, J. Han, Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification, in: European Conference on Computer Vision, Springer, 2020, pp. 247–263.

[17] Y.-X. Wang, D. Ramanan, M. Hebert, Learning to model the tail, in: Advances in Neural Information Processing Systems, 2017, pp. 7029–7039.

[18] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE transactions on pattern analysis and machine intelligence 40 (6) (2017) 1452–1464.

[19] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE, 2008, pp. 1322–1328.

[20] Y. Zou, Z. Yu, B. Vijaya Kumar, J. Wang, Unsupervised domain adaptation for semantic segmentation via class-balanced self-training, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 289–305.

[21] J. Byrd, Z. Lipton, What is the effect of importance weighting in deep learning?, in: International Conference on Machine Learning, PMLR, 2019, pp. 872–881.

[22] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intelligent data analysis 6 (5) (2002) 429–449.

[23] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, S. Kumar, Long-tail learning via logit adjustment, arXiv preprint arXiv:2007.07314 (2020).

[24] K. Tang, J. Huang, H. Zhang, Long-tailed classification by keeping the good and removing the bad momentum causal effect, arXiv preprint arXiv:2009.12991 (2020).

[25] X. Wang, L. Lian, Z. Miao, Z. Liu, S. X. Yu, Long-tailed recognition by routing diverse distribution-aware experts, arXiv preprint arXiv:2010.01809 (2020).

[26] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4004–4012.

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

[28] L. Zhu, Y. Yang, 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, 2020, pp. 4344–4353.

[29] P. Chu, X. Bian, S. Liu, H. Ling, Feature space augmentation for long-tailed data, arXiv preprint arXiv:2008.03673 (2020).

[30] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[31] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, Range loss for deep face recognition with long-tailed training data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5409–5418. doi:10.1109/ICCV.2017.578.

[32] D. Ha, A. Dai, Q. V. Le, Hypernetworks, arXiv preprint arXiv:1609.09106 (2016).

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[34] Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature generating networks for zero-shot learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5542–5551.

[35] Y.-X. Wang, R. Girshick, M. Hebert, B. Hariharan, Low-shot learning from imaginary data, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7278–7286.

[36] B. Li, F. Wu, S.-N. Lim, S. Belongie, K. Q. Weinberger, On feature normalization and data augmentation, arXiv preprint arXiv:2002.11102 (2020).

[37] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.

[38] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, J. Jia, Psanet: Point-wise spatial attention network for scene parsing, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 267–283.

[39] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773.

[40] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[41] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, Gcnet: Non-local networks meet squeeze-excitation networks and beyond, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.

[42] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

[43] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.
URL https://openreview.net/forum?id=YicbFdNTTy

[46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[48] S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[49] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).

[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[51] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.

[52] H. Qi, M. Brown, D. G. Lowe, Low-shot learning with imprinted weights, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5822–5830.

[53] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, arXiv preprint arXiv:1710.09829 (2017).

[54] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (11) (2008).