



OPEN ACCESS

EDITED BY

Honggang Qi,
University of Chinese Academy of Sciences,
China

REVIEWED BY

Tao Zhou,
Nanjing University of Science and Technology,
China
Yi-Jing Zhang,
Shanghai Jiao Tong University, China

*CORRESPONDENCE

Qingdu Li
✉ liqd@usst.edu.cn

RECEIVED 19 September 2023

ACCEPTED 31 October 2023

PUBLISHED 05 January 2024

CITATION

Liu N, Wang J, Zhu Y, Wan L and Li Q (2024)
Improving imbalance classification via
ensemble learning based on two-stage
learning. *Front. Comput. Neurosci.* 17:1296897.
doi: 10.3389/fncom.2023.1296897

COPYRIGHT

© 2024 Liu, Wang, Zhu, Wan and Li. This is an
open-access article distributed under the terms
of the [Creative Commons Attribution License
\(CC BY\)](#). The use, distribution or reproduction
in other forums is permitted, provided the
original author(s) and the copyright owner(s)
are credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted which
does not comply with these terms.

Improving imbalance classification via ensemble learning based on two-stage learning

Na Liu¹, Jiaqi Wang¹, Yongtong Zhu¹, Lihong Wan² and Qingdu Li^{1*}

¹Institute of Machine Intelligence, University of Shanghai for Science and Technology, Shanghai, China,

²Origin Dynamics Intelligent Robot Co., Ltd., Zhengzhou, China

The excellent performance of deep neural networks on image classification tasks depends on a large-scale high-quality dataset. However, the datasets collected from the real world are typically biased in their distribution, which will lead to a sharp decline in model performance, mainly because an imbalanced distribution results in the prior shift and covariate shift. Recent studies have typically used a two-stage learning method consisting of two rebalancing strategies to solve these problems, but the combination of partial rebalancing strategies will damage the representational ability of the networks. In addition, the two-stage learning method is of little help in addressing the problem of covariate shift. To solve the above two issues, we first propose a sample logit-aware reweighting method called (SLA), which can not only repair the weights of majority class hard samples and minority class samples but will also integrate with logit adjustment to form a stable two-stage learning strategy. Second, to solve the covariate shift problem, inspired by ensemble learning, we propose a multi-domain expert specialization model, which can achieve a more comprehensive decision by averaging expert classification results from multiple different domains. Finally, we combine SLA and logit adjustment into a two-stage learning method and apply our model to the CIFAR-LT and ImageNet-LT datasets. Compared with the most advanced methods, our experimental results show excellent performance.

KEYWORDS

imbalance, prior shift, covariate shift, reweighting, logit adjustment

1 Introduction

Benefiting from the development of computing resources in recent years, deep neural networks (DNNs) have been widely used in image classification (He et al., 2016), image segmentation (Zhou et al., 2019), object detection (Tian et al., 2019), etc. These successful application cases usually require large-scale high-quality labeled data, such as ImageNet (Russakovsky et al., 2015) and COCO (Lin et al., 2014), in which the sample distribution in the training and test dataset is almost consistent. However, training datasets collected from the real world generally have a biased distribution, i.e., the number of samples of each class varies greatly. Models trained by biased datasets will not only cause minority class samples to be misidentified as majority class samples but also confuse minority class samples with hard samples from the majority class, eventually leading to a sharp drop in network performance.

The prior shift and covariate shift resulting from an imbalanced distribution are the primary causes of the decline in network performance. Prior shift refers to the phenomenon

that the label distribution of one class in the training dataset and test dataset is inconsistent. Covariate shift mainly refers to the phenomenon that the data distribution of one class in the training dataset and test dataset is inconsistent. These shifts make the network parameters overfit to some majority class samples, resulting in the model's overconfidence in these examples and poor performance on the test dataset. For a long time, many studies have concentrated on developing rebalancing strategies to alleviate this overfitting, such as reweighting for the loss function (Ren et al., 2018; Cui et al., 2019), resampling for the training sample (Pouyanfar et al., 2018; Zhou et al., 2020), and logit adjustment for output logit (Menon et al., 2021; Xu et al., 2021). These strategies provide some good ideas for solving the problems caused by the imbalanced distribution. However, although reweighting and resampling can address class imbalance issues to some extent, the direct application of these methods will damage the deep feature representation ability of the network, making it difficult for the network parameters to reach their theoretical optimal solution (Zhou et al., 2020).

Adopting a two-stage learning strategy, typically using two separate rebalancing strategies in two training stages to decouple network feature representation learning and classifier learning, is a common way to overcome the issues mentioned above. However, some rebalancing strategies are incompatible, e.g., using resampling in the first stage and reweighting in the second stage. Reweighting promotes classifier learning, which encourages the classifier's decision boundary to move in the direction of classifying the minority classes as correctly as possible. Resampling ensures that the label distribution of the mini-batches sampled from the training dataset is consistent with the label distribution of the test dataset. Owing to the undersampling of the majority class samples and the oversampling of the minority class samples, some samples are not involved in the training process, resulting in a negative impact on feature representation learning. It is difficult to use the reweighting method to optimize the classifier when the separability of the feature is weak (Zhou et al., 2020). Based on the above analysis, we propose to use data augmentation instead of resampling in the first stage to maximize the representation ability of the network.

Our goal in this work is to design an efficient and useful two-stage learning method using currently available rebalancing strategies. Owing to the conflict between reweighting and resampling, we investigate the effects of the combination of logit adjustment and reweighting on DNNs. We discover that the network performance will be degraded when combining the existing classic reweighting methods with logit adjustment. This is because both logit adjustment and reweighting try to give minority class samples more attention while giving the majority class samples less attention, ultimately making the performance of the majority class drastically deteriorate. Additionally, because the confidence of majority class hard samples and minority class samples is extremely similar, the sample confidence-based reweighting method [such as focal loss (Lin et al., 2017)] will unfairly assign weights to these samples, which will increase the expected calibration error of the network (Guo et al., 2017). To this end, we propose a logit-aware reweighting method (called SLA) that could use the sample with

the largest logit of each class as the benchmark sample to assign appropriate weights to the remaining samples (Figure 1).

Furthermore, two-stage learning methods are ineffective at dealing with the covariate shift problem, which is an unavoidable but easily neglected issue in imbalanced image classification. It is hard to ensure that the distribution of the training and test dataset is entirely consistent. The minority class may have dramatically different numbers on the training and test datasets when the distribution of the training dataset is imbalanced, which exacerbates the inconsistency between the training data distribution and testing data distribution. In this situation, it is difficult to train a model with good generalizability using just a two-stage learning method. Inspired by ensemble learning, we propose a multi-domain expert specialization model to enhance the feature extract ability in a specific data distribution. In particular, in the first training stage, three different levels of data augmentation were employed to specialize the original data distribution into three distinct data distributions. Additionally, mixup was used to blend the original smaller feature distribution space into a larger feature space, thereby enhancing the model's feature extraction ability. At the same time, the model also includes a two-stage training loss strategy, which can promote the classifier to learn a more reliable decision boundary. Under the guidance of the two-stage learning method, our proposed model demonstrated excellent performance on existing imbalanced datasets.

In summary, our main contributions are as follows:

- (1) For two-stage learning methods, we indicate that the combination of existing reweighting methods and logit adjustment will lead to performance degradation for the majority class or cause significant calibration errors.
- (2) We propose a new reweighting method that can repair the weight of the majority of hard samples and minority samples calculated by the sample confidence-based reweighting method without significantly reducing the majority accuracy.
- (3) We propose a new ensemble learning framework that provides three deep specialized feature extractors for three different levels of data augmentation, which can significantly improve the representation ability of the network. Under the guidance of our proposed two-stage training loss strategy, it can significantly increase classification accuracy and reduce expected calibration error.

2 Related Work

2.1 Reweighting

The reweighting method assigns weights to each class or sample to alleviate the model performance degradation caused by imbalanced data. A weighting function that maps the loss function (or gradient) to each sample can be used to determine the weight. Through artificial prior knowledge or a simple neural network, the weighting function could be easily estimated.

Initially, Huang et al. (2016, 2019) used the reciprocal of class frequency as a weighting factor applied to class loss (Wang et al., 2017). Subsequently, Lin et al. (2017) extended the class frequency

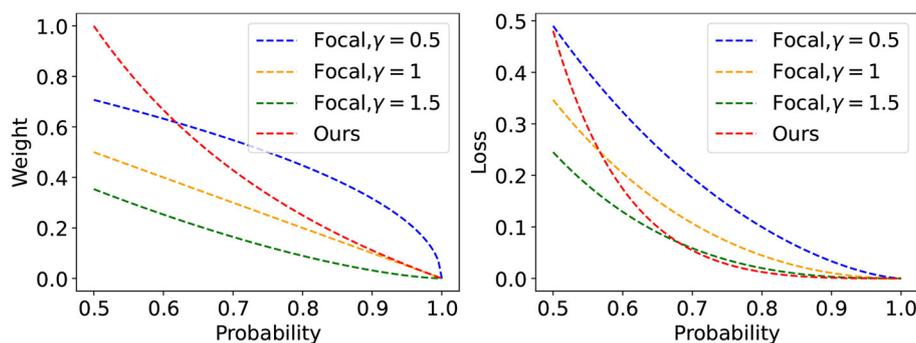


FIGURE 1

In the process of reweighting based on probability, some hard samples from the majority class will have similar weights to the samples from the minority class. As shown in the curve on the right of the above figure, our method can effectively focus on the hard sample of each class, and the loss of our proposed method rapidly decays in the low probability areas.

from a fixed prior to an adjustable parameter version. Khan et al. (2019) further extended the weighting method from the class level to the instance level (Cui et al., 2019). Although this approach is effective, the complex parameter adjustment rules are tedious and not universal. In addition, hard samples from the majority class are frequently weighted improperly because they share a lot of similarities with minority class samples in terms of loss values. To solve this problem, Ren et al. (2018) and Shu et al. (2019) proposed a robust weighted function mapping from samples to instance losses based on the meta-learner. However, it is difficult to estimate the parameters of the weighting network in the meta-learning method. The meta-learning method requires nested training, which costs a lot of time. Also, meta learners need a meta dataset that is close to the distribution of test dataset (Finn et al., 2017; Shu et al., 2019; Jamal et al., 2020; Li et al., 2021). Zhang and Pfister (2021) adjusted the process of meta-learning, which greatly reduced the training cost of meta-learning and alleviated the excessive dependence on metadata distribution. Although meta-learning is currently the best reweighting method for specific datasets, its demanding prerequisites and high training cost precluded us from using it to search for a weighting function.

2.2 Logit adjustment

The idea of logit adjustment was expressed earlier as margin loss. The essence of margin loss is to apply margin to logits of a specific class to obtain a greater classification interval (Liu et al., 2016, 2017; Wang et al., 2018). To address the imbalance image classification task, LDAM (Cao et al., 2019), EQL (Tan et al., 2020), and BALMS (Ren et al., 2020) suggest that minority classes need a large margin while majority classes need a small margin, and the margin is determined by an optimal trade-off boundary (Cao et al., 2019) or by using a meta learner (Jamal et al., 2020; Ren et al., 2020). Menon et al. (2021) summarizes the previous margin-based method and proposes the concept of logit adjustment. To find a suitable logit adjustment method more effectively and quickly, adding label distribution as prior information to the logit has become a stable improvement method (Hong et al., 2021; Menon et al., 2021; Xu et al., 2021; Aimar et al., 2022).

2.3 Two-stage learning

The two-stage training method usually defers the use of the rebalancing strategy, such as reweighting or resampling, to the second stage (Hong et al., 2021). By using a smaller learning rate, the classifier of the model can obtain a better decision boundary on the feature extracted by the feature extractor. Although the two-stage learning method can achieve decoupling training and improve the generalization performance of the model, combining two conflicting rebalancing strategies will lead to a decrease in model performance (Zhou et al., 2020). Therefore, it is important to carefully select and evaluate different rebalancing strategies to ensure that they are compatible with each other and can lead to improved overall performance. In this study, we found that the combination of logit adjustment and the existing reweighting method causes conflicts, making it difficult for the model to converge to the optimal solution. Based on the above findings, we propose a new reweighting method to address this issue.

3 Analysis

For a multi-class classification task, we assume a dataset with N samples, in which $X = \{x_1, x_2, \dots, x_N\}$ denotes the samples and $Y = \{y_1, y_2, \dots, y_N\}$ denotes the labels. The dataset can be defined as $D = \{(x_i, y_i), 1 \leq i \leq N\}$, where x_i denotes the i -th sample and $y_i \in \{0, 1\}^c$ is a c dimension vector. Our goal is to train a network that can minimize the misclassification error, i.e., $\min \sum_{i=1}^N \mathbb{P}(y_i \neq \arg \max p_{y_i}(x_i))$, where $p_{y_i}(x_i)$ represents the probability of x_i belonging to class y_i . In general, we use the *softmax cross-entropy* (CE) to represent this error,

$$\ell(y_i, f(x_i)) = -\log \frac{\exp(f_{y_i}(x_i))}{\sum_{j=1}^c \exp(f_{y_j}(x_i))} \quad (1)$$

where $f_{y_i}(x_i)$ and $f_{y_j}(x_i)$ represent the output logit of x_i belonging to classes y_i and y_j . For the class imbalance problem, the direct use of the CE loss function may lead to the bias toward majority classes during the training process and neglect the learning of minority classes, resulting in some minority class samples being mistakenly classified as the majority classes during the testing phase. To address

TABLE 1 Top-1 accuracy (%) and ECE (%) from the different combinations of logit adjustment and different reweighting methods.

Method	Many	Medium	Few	All
LA	64.9	50.3	29.5	49.5/3.8
LA + RW	30.8	39.5	23.7	31.8/36.9
LA + CB	44.8	44.1	42.0	43.8/4.4
LA + FL	62.5	49.9	32.8	49.5/3.2

We used the mixup $\alpha = 0.4$ on the CIFAR-100-LT dataset ($\rho = 100$).

this issue, most reweighting methods usually apply a learnable or pre-designed weighting factor w to modulate the CE loss function, which can improve the contribution of minority classes to the average loss and make network learning more focused on minority classes. The reweighting loss function can be expressed by the following equation,

$$\ell(y_i, f(x_i)) = -w_i \log \frac{\exp(f_{y_i}(x_i))}{\sum_{j=1}^c \exp(f_{y_j}(x_i))} \quad (2)$$

However, it is challenging to derive an explicit reweighting function without prior knowledge. In most reweighting methods, the weighting factor is naturally defined as a small weight for the majority class and a large weight for the minority class. Although this logical viewpoint is empirically correct, it does not consider the imbalanced distribution within the class; the samples of the same class can also be divided into the common sample and rare sample.

3.1 Compensation training classifier

From the perspective of data distribution, we can rapidly identify why the model trained from the training dataset often performs poorly in the test phase in imbalance image classification tasks. The training and test objectives can be expressed by the following probability,

$$\mathbb{P}^s(y|x) \propto \frac{\mathbb{P}^s(x, y)}{\mathbb{P}^s(x)} \propto \mathbb{P}^s(x|y)\mathbb{P}^s(y) \quad (3)$$

$$\mathbb{P}^t(y|x) \propto \frac{\mathbb{P}^t(x, y)}{\mathbb{P}^t(x)} \propto \mathbb{P}^t(x|y)\mathbb{P}^t(y) \quad (4)$$

where s represents the source domain (training dataset) and t represents the target domain (test dataset). According to Equations (3) and (4), we can further express it as a form of measuring the difference between the training and testing object (Jamal et al., 2020),

$$\mathbb{P}^s(y|x) = \mathbb{P}^t(y|x) \frac{\mathbb{P}^s(x|y)\mathbb{P}^s(y)}{\mathbb{P}^t(x|y)\mathbb{P}^t(y)} \quad (5)$$

$$\underbrace{\mathbb{P}^s(x|y) \ominus \mathbb{P}^t(x|y)}_{\text{Covariate shift}} \text{ and } \underbrace{\mathbb{P}^t(y) \ominus \mathbb{P}^s(y)}_{\text{Prior shift}} \quad (6)$$

Covariate shift is a common issue in deep learning tasks that refers to the situation in which the input data or feature distribution

differs between the training dataset and test dataset, leading to a poor generalization performance of the trained model on the test dataset. The network will inevitably suffer from this damage during training. For the imbalance image classification task, this damage will become more serious (Jamal et al., 2020). Prior shift refers to a common problem that arises when there is some difference in the label distribution between the training and test datasets. Specifically, it is caused by the difference in the distribution of the number of samples per class between the training and test datasets (Menon et al., 2021). This makes the algorithm learn a biased representation, resulting in decreased performance when applied in the test phase. Owing to the difficulty in estimating covariate shift, we will discuss strategies for mitigating this problem in Section 4.2, but temporarily ignore its impact here. In previous training processes, the softmax classifier was typically used for both training and testing. However, as indicated by Equation (6), two shifts between the training and test objectives exist. To address these problems, we can adjust the training loss as follows:

$$\ell(y_i, f(x_i)) = -\log \frac{\exp(f_{y_i}(x_i) + \log \mu_i)}{\sum_{j=1}^c \exp(f_{y_j}(x_i) + \log \mu_j)} \quad (7)$$

where $\mu_i = \frac{\mathbb{P}^{\text{train}}(y_i)}{\mathbb{P}^{\text{test}}(y_i)}$, μ is a factor to measure the label distribution difference between the training and test datasets. Furthermore, Equation (7) can be expressed as follows:

$$\ell(y_i, f(x_i)) = -\log \left[1 + \sum_{j \neq i} \frac{\mu_j}{\mu_i} \exp(f_{y_j}(x_i) - f_{y_i}(x_i)) \right] \quad (8)$$

If y_i represents the majority classes and $\mu_j < \mu_i$, the loss value calculated based on Equation (8) will decrease compared with CE. This will make the network tend to learn from minority classes during parameter updates, reducing the attention to majority classes, thereby improving the performance of the network. For convenience, we will use logit adjustment (LA) to represent the above training losses.

3.2 Mixed reweighting and LA

Compensating the output logit can effectively alleviate the learning bias caused by imbalanced data distribution. To further improve the effectiveness of boundary correction, we combine reweighting with LA into a new paradigm and explore effective combination strategies. Specifically, we conduct experiments using ResNet-32 trained on the CIFAR-100-LT dataset with different combinations of reweighting and LA. The reweighting methods, which include reweight (RW) (Wang et al., 2017), class-balanced loss (CB) (Cui et al., 2019), and focal loss (FL) (Lin et al., 2017), were introduced in the 180th epoch (out of a total of 200 epochs) for ResNet-32.

Table 1 presents the results obtained from the aforementioned settings. We can infer that (1) the combination of existing reweighting methods and LA will lead to a decline in overall accuracy, especially in the majority classes. This indicates that there is a conflict between the existing reweighting and LA, and there is an overlap between providing large margins and large weights for

the minority classes, which ultimately leads to a significant decline in the performance of the majority classes. (2) Although focal loss can maintain the accuracy of the majority classes to a certain extent, it is expected that the calibration error is still large. This is because focal loss assigns similar weights to the hard samples from majority classes and the samples from minority classes.

4 Method

4.1 Sample logit-aware reweighting

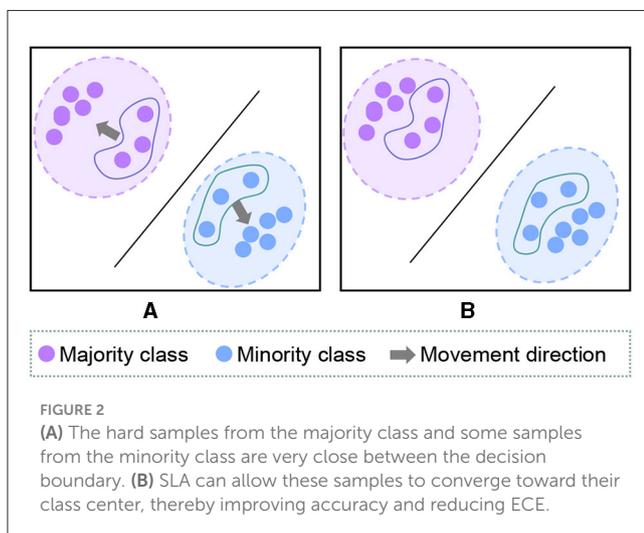
The purpose of the two-stage training method is to focus on obtaining a powerful feature extractor and classifier in the first stage and reduce the difference between the sample confidence and the overall class confidence in the second stage. From the perspective of sample confidence, assigning higher weights to samples with low confidence is an effective solution. However, when it comes to hard samples in the majority classes, their confidence levels are often indistinguishable from the samples in the minority classes. To overcome this issue, we propose a sample logit-aware reweighting method (called SLA in this study) that reduces the gap between the single sample confidence and the overall class average confidence, without significantly sacrificing accuracy. The sample confidence can be calculated as follows:

$$p_i = \frac{\exp(f_{y_i}(x_i) + \log \mu_i)}{\sum_{j=1}^c \exp(f_{y_j}(x_i) + \log \mu_j)} \quad (9)$$

where p_i represents the predicted probability that sample x_i belongs to the correct label after adjusting for the output logit. In addition, based on the idea of SLA, to make the weighting factor w_i pay more attention to hard samples based on the probability reweighting method, we use the sample with the maximum logit of each class to guide the learning of the remaining samples. The sample weight can be expressed as follows:

$$w_i = (1 - p_i)^\gamma \exp(f_{y_i}(x_*) - f_{y_i}(x_i)) \quad (10)$$

where x_* is the sample with the largest logit in all training samples belonging to y_i , and γ is a weighted rate adjustment factor.



Commonly, $f_{y_i}(x_i) = W_{y_i} z_i$, W is the weight matrix of the linear layer and z_i is the feature embedding of x_i . To obtain more stable sample weights, we calculate the cosine value by standardizing W_{y_i} and z_i .

$$\cos\theta(y_i) = \frac{W_{y_i}^T}{\|W_{y_i}\|} \cdot \frac{z_i}{\|z_i\|} \quad (11)$$

Therefore, after transforming the logit into the corresponding cosine representation (Figure 2), the final SLA reweighting formula can be expressed as follows:

$$w_i = (1 - p_i)^\gamma \exp(\tau \cos\theta_{y_*} - \tau \cos\theta_{y_i}) \quad (12)$$

where θ_{y_i} corresponds to the angle between z_i and W_{y_i} , θ_{y_*} corresponds to the z_* and W_{y_i} , and τ is a hyperparameter.

4.2 Multi-domain expert specialization model

The main objective of the first stage of training in the two-stage method is to enhance the feature extraction capability of the network. However, it is challenging for a single-channel feature extractor to learn robust parameters when the data distribution is extremely imbalanced, particularly when complex data augmentation techniques are applied. To address this problem, we propose a multi-domain expert specialization model for augmenting data across multiple domains (Algorithm 1).

4.2.1 Multiple data augment header with mixup

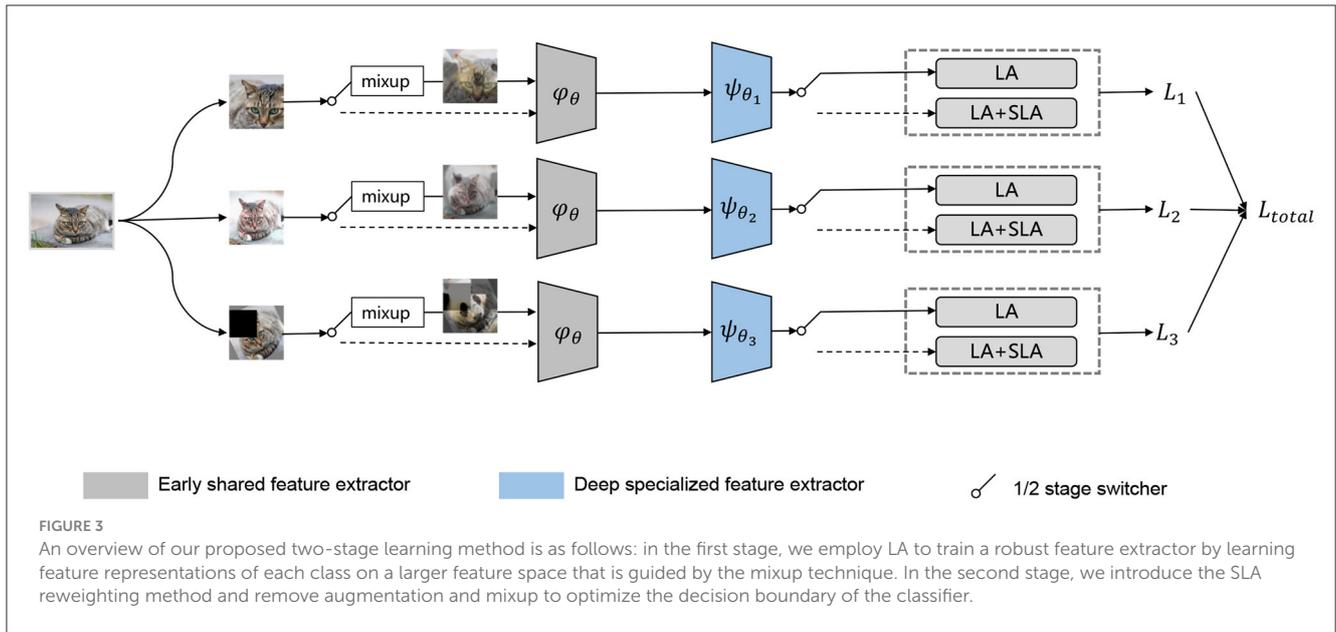
Before inputting data into the network, multiple data augmentation techniques (including mixup) should be applied to the data. The purpose of data augmentation is to expand the domain boundary of the source domain data as much as possible

```

Input: Training data  $D = \{(x_i, y_i)\}_{i=1}^N$ , batch size  $n$ .
Output: Optimized network parameters  $\theta$ .
1: Initialization for  $\theta$ 
2: while  $t \leq \text{MaxEpoch}$  do
3:    $\{(x_i, y_i)\}_{i=1}^n \leftarrow \text{Sample a minibatch from } D$ .
4:   Getting  $\{(\tilde{x}_i, \tilde{y}_i)\}$  using Equation (14)
5:   Getting each expert loss using Equation (16)
6:   Calculate total loss using Equation (18)
7:   if  $t == \text{SwitchEpoch}$  then
8:     Create a list  $S$  to store  $\cos\theta_{y_*}$  of each class.
9:   end if
10:  if  $t > \text{SwitchEpoch}$  then
11:    Find the sample with the largest logit score
    for each class and calculate  $\cos\theta_{y_*}$  using Equation
    (11)
12:    Update the list  $S_t \rightarrow S_{t+1}$ .
13:  end if
14:  Use SGD to update network parameters  $\theta$ ;
15: end while

```

Algorithm 1. The training process of our proposed method.



(Figure 3), thereby alleviating the severe covariate shift caused by the imbalanced distribution.

$$T_k(x_i) = \begin{cases} \text{mixup}(\text{Aug}_k(x_i)) & \text{if 1-stage} \\ \text{Aug}_k(x_i) & \text{if 2-stage} \end{cases} \quad (13)$$

where T_k represents the result of applying the k -th data augmentation function (Aug) to input x_i . To make better use of data augmentation, we apply the mixup strategy based on the augmented data during the first stage of training. By using mixup, the resulting data can be represented as if it was sampled from a new sampling space: $D_l = \{(\tilde{x}_i, \tilde{y}_i), 1 \leq i \leq N'\}$. After combining two augmented samples using mixup, the newly generated sample $\{\tilde{x}, \tilde{y}\}$ can be expressed as follows:

$$\begin{aligned} \tilde{x} &= \epsilon T(x_i) + (1 - \epsilon)T(x_j) \\ \tilde{y} &= \epsilon y_i + (1 - \epsilon)y_j \end{aligned} \quad (14)$$

where $\epsilon \sim \text{Beta}(\alpha, \alpha)$ with $\alpha \in (0, 1)$, which allows for flexible adjustment of the mixing ratio during training. By introducing this sampling procedure, the model can be trained on a new sample space that comprises mixtures of the original augmented inputs, allowing it to learn more robust representations and improve its ability to generalize to new samples.

4.2.2 Early shared and deep special feature extractor

During the feature extraction process in the early layers of CNN, the network tends to learn low-level features such as points and lines. As a result, we opt for utilizing the same early shared feature extractor for different enhanced data during the first stage. However, during deep feature extraction, the varying enhancement of three levels of data augmentation requires specialized deep feature extractors to extract professional features. To achieve this goal, we employ three distinct deep feature extractors, with their outputs expressed as

$$f^k(x_i) = \psi_{\theta_k}(\varphi_{\theta}(T_k(x_i))) \quad (15)$$

where $k \in [1, 3]$, $f^k(x_i)$ represents the output logit after x_i passes through the early shared feature extractor φ_{θ} and k -th deep special feature extractor ψ_{θ_k} .

4.2.3 Two-stage training loss strategy

As analyzed in Section 3.1, the two-stage training method requires training a better feature extractor in the first stage. Therefore, we only compensate the classifier and do not use any reweighting method during the first stage of training. Hence, the model should use a reweighting method in the following training process to optimize the decision boundary of the classifier to reduce ECE.

$$L_k(y_i, f^k(x_i)) = -w_i^k \log \frac{\exp(f_{y_i}^k(x_i) + \log \mu_i)}{\sum_{j=1}^c \exp(f_j^k(x_i) + \log \mu_j)} \quad (16)$$

Equation (16) represents the loss function L_k for the k -th expert, and w_i^k can be expressed in the following form:

$$w_i^k = \begin{cases} 1 & \text{if 1-stage} \\ (1 - p_i^k)^{\gamma} \exp(\tau \cos \theta_{y_*} - \tau \cos \theta_{y_i}) & \text{if 2-stage} \end{cases} \quad (17)$$

Here, γ and τ are hyperparameters, p_i^k is the predicted probability of the k -th expert of the sample x_i belonging to its true class after compensating the out logit, and $f_{y_i}^k(x_i)$ is the output from the k -th expert belongs to y_i class from the k -th expert. Thus, the final loss function can be expressed as the weighted sum of losses obtained by three experts. We use ϵ_k to indicate the degree of attention given to the k experts; increasing ϵ_k can make the model more inclined to learn from expert k -th. To make the results of other ensemble learning methods more comparable and ensure the fairness of the comparison, we set ϵ_k to 1 in all the experiments

conducted in this study. The final expression for the total loss function is represented by Equation (18).

$$L_{total} = \epsilon_1 L_1 + \epsilon_2 L_2 + \epsilon_3 L_3 \quad (18)$$

4.2.4 Test time prediction

Considering we used a loss function in the training stage that was the weighted sum of individual losses from multiple experts, we employ the weighted average logit output of three experts during the test process as our final prediction to minimize empirical risk. The probability that x_i belongs to a certain class can be calculated using the following formula:

$$\hat{p}_i = \arg \max \left\{ \frac{1}{3} \sum_{k=1}^3 \epsilon_k \frac{\exp(f_{y_i}^k(x_i))}{\sum_{j=1}^c \exp(f_{y_j}^k(x_i))}, i \in [1, c] \right\} \quad (19)$$

5 Experiments

5.1 Datasets

5.1.1 CIFAR-10-LT and CIFAR-100-LT

The CIFAR-10 and CIFAR-100 datasets are common image classification datasets that contain 50,000 training images and 10,000 test images with 10 or 100 classes (Krizhevsky et al., 2009). Following Cao et al. (2019), we create the long-tailed distribution version by randomly removing training samples and keeping the distribution of the test dataset balanced. We use the imbalance ratio ρ to represent the imbalance degree of the dataset, where $\rho = N_{max}/N_{min}$, $N_{max}(N_{min})$ is the number of the most (least) frequent class. In this study, we used the imbalance ratio of 10, 50, 100, and 200 to carry out experiments.

5.1.2 ImageNet-LT

ImageNet (Russakovsky et al., 2015) is a large-scale dataset for object classification. Based on this, Liu et al. (2019) made ImageNet-LT by sampling a subset following the Pareto distribution with power value $\alpha = 0.6$ from ImageNet, which contains $\sim 115.8K$ images with 1,000 classes. This choice is crucial because it controls the proportion of frequent and infrequent categories in the long-tailed distribution. In addition, the Pareto distribution has a characteristic long tail, which is desirable as it can generate more extreme long-tail datasets that are closer to real-world scenarios. The number of samples for the most frequent class is 1,280 images, whereas the number of samples for the least frequent class is only five images, i.e., the imbalance ratio $\rho = 256$.

5.2 Evaluation protocol

5.2.1 Expected calibration error

The purpose of model calibration is to ensure that the predictive confidence of the model for one sample is consistent with the true empirical risk probability. Therefore, we use the expected calibration error (ECE) to measure the calibration degree of the

network. To compute ECE, we group all N predictions into B interval bins of equal size. The ECE can be defined as:

$$ECE = \sum_{b=1}^B \frac{|T_b|}{N} |acc(T_b) - conf(T_b)| \quad (20)$$

where T_b is the set of samples with a network prediction belonging to Bin- b , $acc(\cdot)$ is the accuracy of T_b , and $conf(\cdot)$ is the predicted confidence of T_b .

5.3 Implementation details

For CIFAR-10-LT and CIFAR-100-LT datasets, we used ResNet-32 as the benchmark network. We used three different levels of data augmentation; the specific details are shown in Appendix. Following most practices, we set the batch size as 128 and the weight decay as $5e-4$. We used the SGD optimizer, and the initial learning rate was 0.1. For all experiments on the main result, the hyperparameter α was set to 0.2, and τ was set to 1. For a fair comparison, we trained 200 and 400 epochs, respectively, based on the above settings. During the training of 200 epochs, the learning rate was decreased by a factor 10 at epochs 160 and 180. During the training of 400 epochs, the learning rate was decreased by a factor 10 at epochs 320 and 360. The 1/2 stage switching time was set to epochs 160 and 320.

For ImageNet-LT, we adopted ResNet-50 and ResNetx-50 as the benchmark networks. As with CIFAR-LT, three different levels of data augmentation were employed. The batch size was set to 128 for ResNet-50 and 64 for ResNetx-50 with the weight decay as $5e-4$. We used the SGD optimizer, and the initial learning rate was set at 0.025. We used a cosine annealing learning rate schedule. For all experiments on the main result, the parameter α was set to 0.1, and τ was set to 1. During the training of 180 epochs, the learning rate changed periodically according to the law of the cosine annealing learning rate schedule. The 1/2 stage switching time was set to epoch 160.

5.4 Main results

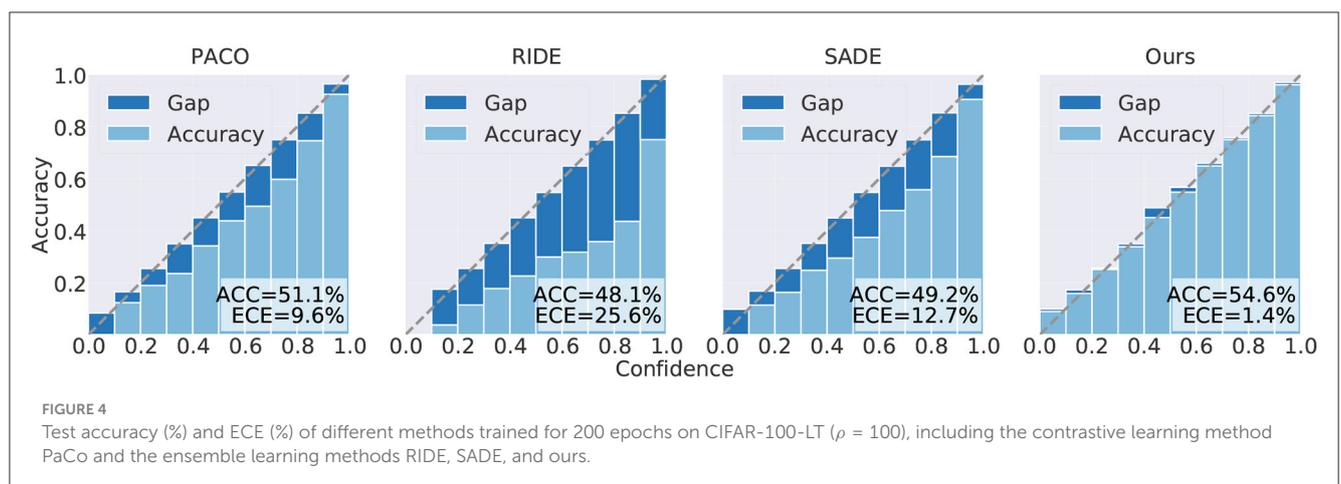
5.4.1 Result for CIFAR-LT

Table 2 presents a comparison of the results obtained by our proposed method and other various methods on CIFAR-LT. All experiments trained for 200 epochs. First, we observed that our method outperformed existing methods across all class imbalance ratios. Specifically, our proposed method achieved improvements of 4.7, 4.3, 3.2, and 1.4% on CIFAR-10-LT, and 3.9, 4.2, 4.2, and 4.1% on CIFAR-100-LT for imbalance ratios of 200, 100, 50, and 10, respectively, when compared with the state-of-the-art method. Second, it is worth noting that our method maintained a significant performance gap compared with other methods regardless of the class imbalance ratio, which demonstrates the effectiveness of our method. Furthermore, we observed that, compared with existing multi-expert methods, the accuracy gap between our proposed method and theirs gradually decreased with a decrease in the imbalance ratio. This phenomenon can be explained by the fact that when the imbalance ratio is small, data from the minority classes

TABLE 2 Test accuracy (%) on CIFAR-100-LT for various methods with different imbalance ratios ρ .

Method	Dataset Backbone Imbalance ratio	CIFAR-10-LT				CIFAR-100-LT			
		ResNet-32				ResNet-32			
		200	100	50	10	200	100	50	10
CE		65.7	70.4	74.8	83.4	38.3	38.2	43.9	56.9
CB-Focal (Cao et al., 2019)		–	74.6	79.2	86.8	–	39.6	45.2	58.0
MW-NET (Shu et al., 2019)		–	75.2	80.0	87.8	–	42.1	46.7	58.4
LDAM + DRW (Cao et al., 2019)		–	77.0	81.2	88.2	–	42.0	46.6	58.7
BBN (Zhou et al., 2020)		–	79.8	82.4	88.1	–	42.5	47.2	59.4
LA (Menon et al., 2021)		–	79.9	83.4	89.3	–	43.9	49.8	59.8
Mixup (Zhang et al., 2018)		–	73.3	77.6	87.2	–	39.6	45.1	58.4
Remix + DRW (Chou et al., 2020)		–	79.8	–	89.1	–	46.8	–	61.3
MiSLAS (Zhong et al., 2021)		–	82.1	85.7	90.0	–	47.0	52.3	63.2
RIDE (Zhang et al., 2018)		77.9	81.5	83.4	85.9	44.8	48.5	51.0	57.8
ACE (Cai et al., 2021)		–	81.2	84.3	–	–	49.4	50.7	–
SADE (Zhang et al., 2022)		78.6	82.4	85.6	90.5	46.2	50.4	54.2	63.8
Ours		83.3	86.7	88.9	91.9	50.1	54.6	58.4	67.9

All experiments used ResNet-32 as the backbone and trained for 200 epochs.



already cover a large data distribution space in the training dataset, thus weakening the effect of data augmentation on alleviating covariate shift caused by an imbalanced distribution. At the same time, we compared the SLA of different methods and the results showed that our proposed method achieved lowest SLA in addition to achieving considerable accuracy (Figure 4).

At the same time, we performed long-term training for 400 epochs on CIFAR-100-LT ($\rho = 100$), and the corresponding results are presented in Table 3. Compared with those in Table 2, our proposed method demonstrated continued improvement in accuracy beyond 200 epochs. This is attributed to the inclusion of multiple data augmentation headers in our network architecture, which significantly enhances the representation ability of the network's feature extractor and mitigates the representation difficulties introduced by covariate shift, leading to enhanced overall accuracy. More importantly, the performance of our

proposed method in the few classes is far better than that of other methods. This is because we have assigned a specialized feature extractor for each level of data augmentation, which can prevent the representation coupling caused by different levels of data augmentation.

5.4.2 Result for ImageNet-LT

Tables 4, 5 present the comparison results between our proposed method and existing methods on the long-tailed dataset ImageNet-LT. Compared with the multi-expert model RIDE (Wang et al., 2021) and SADE (Zhang et al., 2022), our method introduces a multiple data augmentation header with mixup based on the deep specialized feature extractor, leading to an improved performance on minority classes by effectively maintaining the

TABLE 3 Test accuracy (%) on CIFAR-100-LT ($\rho = 100$) for different methods.

Methods	Many	Medium	Tail	All	ECE
BALMS (Ren et al., 2020)	–	–	–	50.8	–
PaCo (Cui et al., 2021)	62.9	53.5	35.6	51.7	9.3
BCL (Zhu et al., 2022)	69.7	53.8	35.5	53.9	–
RIDE (Wang et al., 2021)	66.8	53.6	23.5	49.6	34.8
SADE (Zhang et al., 2022)	66.4	51.7	29.0	50.4	17.9
Ours	69.3	55.5	39.3	55.4	1.4

All experiments used ResNet-32 as the backbone and trained for 400 epochs.

TABLE 4 Test accuracy (%) on ImageNet-LT on ResNet-50 and ResNetx-50 for various methods.

Backbone	ResNet-50	ResNetx-50
CE	47.1	48.2
MiSLAS (Zhong et al., 2021)	52.7	–
UniMix (Xu et al., 2021)	48.4	–
PaCo (Cui et al., 2021)	57.0	58.2
BCL (Zhu et al., 2022)	56.0	57.1
LA (Menon et al., 2021)	51.2	–
RIDE (Wang et al., 2021)	54.9	56.4
ACE (Cai et al., 2021)	54.8	56.5
SADE (Zhang et al., 2022)	–	58.8
Ours	57.8	59.9

model's strong representation ability from the first stage to the second stage via our proposed two-stage adjustment strategy. In contrast to other methods based on contrastive learning, such as PaCo (Cui et al., 2021) and BCL (Zhu et al., 2022), we all use various data augmentation methods. However, our proposed multi-channel deep feature extraction strategy can learn the optimal representation of different degrees of data augmentation to maximize their effectiveness. This is the main difference between our approach and others. By exploiting the different levels of data augmentation, we achieve better performance.

To further verify the effectiveness of our proposed reweighting method, we report the test accuracy (%) and ECE (%) on the combination of LA and different reweighting methods on ImageNet-LT using ResNet-50. All experiments used the same model structure and experimental settings as the multi-domain expert specialization model we proposed. Table 6 presents the results of our experiments, which demonstrate that our reweighting method outperformed other reweighting techniques in the minority classes, while only slightly compromising performance in the majority classes. The results suggest that appropriate reweighting methods can alleviate the overfitting of model parameters to most classes caused by the long-tailed distribution.

TABLE 5 Test accuracy (%) on ImageNet-LT on ResNetx-50 for various methods.

Methods	Many	Medium	Few	ALL
LADE (Hong et al., 2021)	65.1	48.9	33.4	53.0
BL Softmax (Ren et al., 2020)	65.8	53.2	34.1	55.4
PaCo (Cui et al., 2021)	64.4	55.7	33.7	56.0
BCL (Zhu et al., 2022)	67.9	54.2	36.6	57.1
RIDE (Wang et al., 2021)	68.0	52.9	35.1	56.3
SADE (Zhang et al., 2022)	67.0	56.4	42.6	58.7
Ours	67.1	54.7	56.5	59.9

TABLE 6 Test accuracy (%) and ECE (%) on ImageNet-LT on ResNet-50 for different reweighting methods.

Methods	Many	Medium	Tail	ALL	ECE
LA + RW	69.3	54.1	32.7	53.4	5.2
LA + CB	35.2	44.1	51.5	43.1	6.6
LA + FL	60.7	53.9	42.8	53.1	3.8
LA + Ours	65.4	52.2	54.8	57.8	3.3

On the other hand, inappropriate reweighting methods will lead to biased models or significant performance decreases in the majority classes.

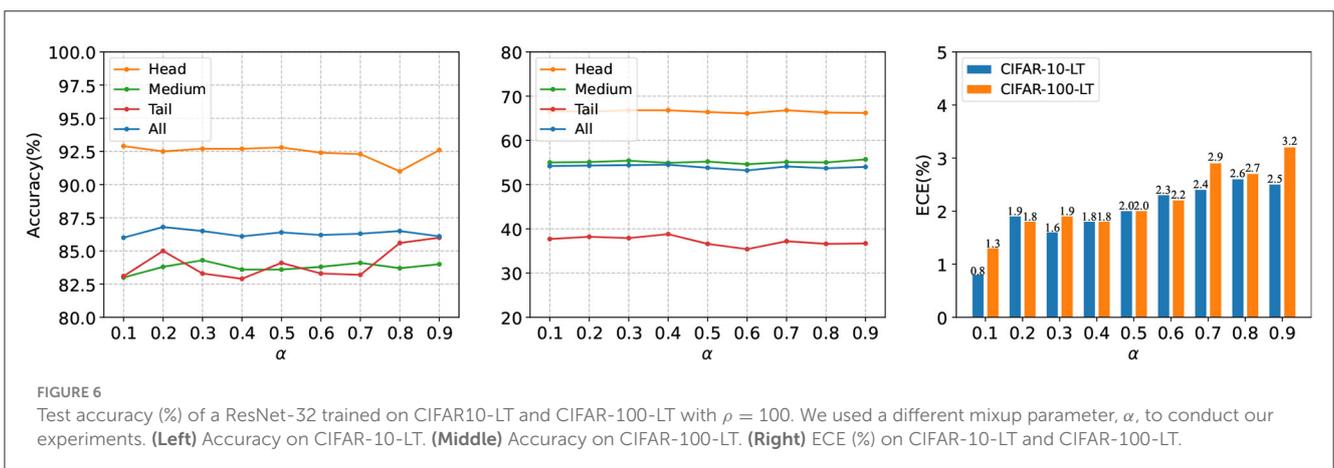
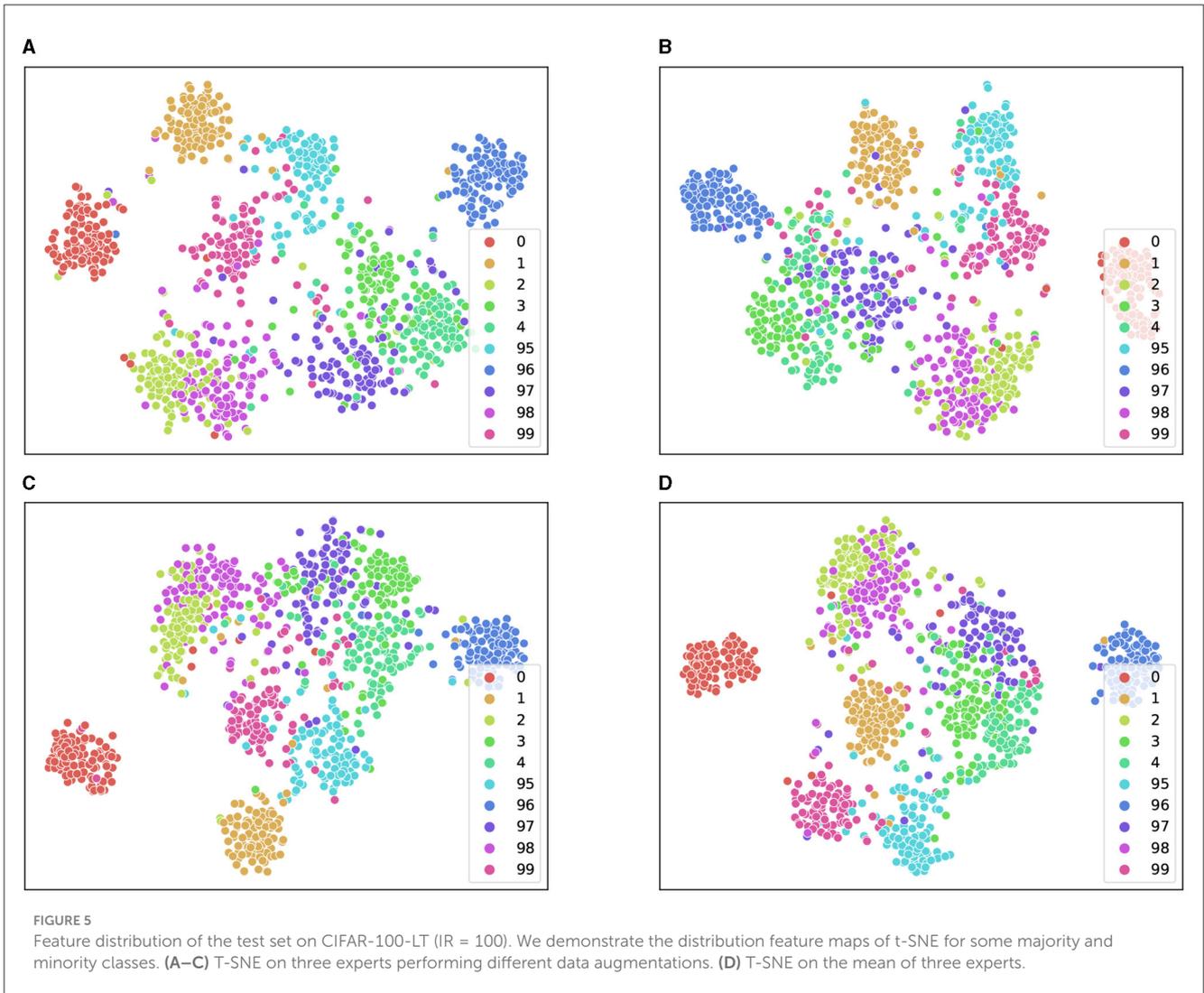
5.5 Feature distribution

To gain further insights into the effectiveness of our proposed method, we visualized the extracted features using t-SNE. As depicted in Figure 5, feature-1 and feature-2 correspond to the features obtained after dimensionality reduction. We observed that strong data augmentation could enhance feature separability but at the expense of increasing intraclass distance. By leveraging the domain expertise of three different experts and averaging their augmented features, we were able to obtain distinctive features that preserve intraclass similarity while improving interclass discrimination. This allowed us to achieve a clear decision boundary between different classes, even when using a simple linear classifier.

5.6 Ablation study

5.6.1 The effect of different mixup parameters α

To study the influence of the change of mixup parameters (α) on our proposed method, we conducted a thorough ablation experiment on CIFAR-LT with $\rho = 100$ to find out the optimal parameter range. Figure 6 shows the result. We observed that (1) when α is > 0.4 , the accuracy of the tail class fluctuates greatly; this phenomenon is obvious when the number of classes is small. The main reason for this is that with the increase in α , the value of u tends to be uniformly distributed due to the drastic change in mixing degree between different epochs and the lack of tail class



data; this large randomness destroys the stability training of the tail class. (2) With an increase in α , the ECE of the results shows an increasing trend, which indicates that adjusting parameter α in our proposed method plays a crucial role in reducing the ECE of the model.

5.6.2 The effect of the different hyperparameter τ

As reported in Table 7, we explored how hyperparameter τ influences the model. We can easily find that when the imbalance factor is fixed, the accuracy and ECE will decrease as the τ increases. The main reason for this phenomenon is that increasing τ enhances

TABLE 7 Ablation study of different imbalance ratios and τ .

Dataset	Imbalance ratio	$1/\tau$					
		1	0.9	0.8	0.7	0.6	0.5
CIFAR-10-LT	200	83.3/1.4	83.2/1.3	83.6/1.2	83.3/1.2	83.2/0.9	83.0/0.8
	100	86.7/1.9	86.5/1.8	86.4/1.2	86.3/1.7	86.0/1.2	86.0/1.3
	50	88.9/2.4	88.3/1.9	88.6/2.5	88.4/2.0	88.8/2.2	88.5/2.0
	10	91.9/2.0	91.7/1.8	91.7/1.9	91.9/1.8	92.2/2.0	91.5/1.7
CIFAR-100-LT	200	50.1/1.8	50.1/1.6	49.9/1.6	50.2/1.4	50.1/1.2	50.0/1.0
	100	54.6/1.8	54.0/2.0	54.0/1.7	54.6/1.6	54.2/1.6	54.0/1.5
	50	58.4/1.7	58.7/2.1	58.5/1.9	58.8/1.7	58.6/1.6	58.0/1.6
	10	67.9/2.4	67.6/2.1	67.0/2.1	67.7/2.0	67.3/2.0	67.0/1.8

We chose several values from 1 to 0.5 for $1/\tau$ to perform our ablation experiment.

TABLE 8 Ablation study of various combinations of the module to verify the effectiveness of different modules.

Module			CIFAR-10-LT			CIFAR-100-LT		
MU	SLA	TL	100	50	10	100	50	10
		✓	84.8/5.6	87.7/3.9	91.6/3.2	53.2/6.5	58.0/6.3	67.0/4.0
✓		✓	84.7/5.4	88.2/3.4	91.5/2.8	54.1/4.0	59.2/3.0	67.3/2.8
	✓	✓	86.4/3.2	88.3/1.3	91.7/1.2	53.6/3.3	58.2/1.7	66.9/1.2
✓	✓	✓	86.7/1.9	88.9/2.4	91.9/2.0	54.6/1.8	58.4/1.7	67.9/2.4

We conducted a thorough ablation experiment. MU, using mixup in the first stage of learning; SLA, using SLA in the second stage of learning; TL, using the two-stage learning method.

the effect of SLA, which changes the decision boundary while reducing intraclass spacing. As the decision boundary no longer tends to reduce overall empirical risk, this will reduce some of the model's performance.

5.6.3 The effect of different modules

Table 8 present the results of our ablation investigation into the use of mixup in the first stage (MU), reweighting in second-stage learning (SLA), and two-stage learning (TL). As expected, we observed a decrease in accuracy and an increase in ECE for all datasets as the imbalance ratio increased. Combining MU or SLA modules with TLs consistently led to improved accuracy and reduced ECE. Notably, our proposed SLA method demonstrated a more positive impact on TL than MU under multi-data augmentations, thereby proving its effectiveness. Additionally, when all three modules were combined, our proposed algorithm maximized the model's generation ability while maintaining low ECE, despite not being optimal.

6 Conclusion

In this study, we addressed the problem of poor model performance due to prior shift and covariate shift caused by imbalanced distribution. To investigate the impact of logit adjustment and reweighting on model performance, we employed the two-stage learning method, which is currently a popular research direction. Our analysis revealed that combining existing reweighting methods and logit adjustment not only reduces

model performance but also increases ECE. Therefore, we proposed a sample logit-aware reweighting method that assigns more suitable weights to hard samples from majority classes and samples from minority classes. Additionally, to tackle the covariate shift problem, we introduced a multi-domain expert specialization model designed to enhance the feature extraction ability of the model. Through experiments conducted on various datasets, we demonstrated the effectiveness of our proposed method. Furthermore, ablation experiments reinforced our findings and emphasized that our proposed model outperforms current state-of-the-art methods. Overall, our study highlights the necessity of addressing prior and covariate shift in imbalanced datasets and provides an effective solution to improve model performance.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>; <https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz>; <https://image-net.org/>.

Author contributions

NL: Formal analysis, Software, Writing—original draft. JW: Writing—review & editing. YZ: Data curation, Writing—review & editing. LW: Writing—review & editing. QL: Funding acquisition, Supervision, Writing—review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This study was supported by the National Natural Science Foundation of China under grant 92048205.

Conflict of interest

LW was employed by Origin Dynamics Intelligent Robot Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial

relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Aimar, E. S., Jonnarth, A., Felsberg, M., and Kuhlmann, M. (2022). Balanced product of experts for long-tailed recognition. *arXiv [preprint]*. doi: 10.1109/CVPR52729.2023.01912
- Cai, J., Wang, Y., and Hwang, J.-N. (2021). "Ace: ally complementary experts for solving long-tailed recognition in one-shot," in *ICCV* (Piscataway, NJ).
- Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). "Learning imbalanced datasets with label-distribution-aware margin loss," in *NeuIPS* (Cambridge).
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). "A simple framework for contrastive learning of visual representations. In International conference on machine learning," in *PMLR* (New York, NY), 1597–1607.
- Chou, H.-P., Chang, S.-C., Pan, J.-Y., Wei, W., and Juan, D.-C. (2020). "Remix: rebalanced mixup," in *ECCV* (Berlin).
- Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: learning augmentation policies from data. *arXiv [preprint]*. doi: 10.1109/CVPR.2019.00020
- Cui, J., Zhong, Z., Liu, S., Yu, B., and Jia, J. (2021). "Parametric contrastive learning," in *ICCV* (Piscataway, NJ).
- Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). "Class-balanced loss based on effective number of samples," in *CVPR* (Piscataway, NJ).
- Finn, C., Abbeel, P., and Levine, S. (2017). "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML* (New York, NY).
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). "On calibration of modern neural networks," in *ICML* (New York, NY).
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ), 770–778.
- Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., and Chang, B. (2021). "Disentangling label distribution for long-tailed visual recognition," in *CVPR* (Piscataway, NJ): IEEE.
- Huang, C., Li, Y., Loy, C. C., and Tang, X. (2016). "Learning deep representation for imbalanced classification," in *CVPR* (Piscataway, NJ).
- Huang, C., Li, Y., Loy, C. C., and Tang, X. (2019). Deep imbalanced learning for face recognition and attribute prediction. *IEEE Transact. Patt. Anal. Mach. intell.* doi: 10.1109/TPAMI.2019.2914680
- Jamal, M. A., Brown, M., Yang, M.-H., Wang, L., and Gong, B. (2020). "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *CVPR* (Piscataway, NJ).
- Khan, S., Hayat, M., Zamir, S. W., Shen, J., and Shao, L. (2019). "Striking the right balance with uncertainty," in *CVPR* (Piscataway, NJ).
- Krizhevsky, A., Sutskever, I., and Hinton, G. (2009). *Learning Multiple Layers of Features From Tiny Images*. Toronto, ON.
- 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* (Piscataway, NJ).
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection," in *ICCV* (Piscataway, NJ).
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: common objects in context," in *ECCV* (Berlin).
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). "Sphereface: deep hypersphere embedding for face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ).
- Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). "Large-margin softmax loss for convolutional neural networks," in *ICML* (New York, NY).
- 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* (Piscataway, NJ).
- Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. (2021). "Long-tail learning via logit adjustment," in *ICLR* (Washington, DC).
- Pouyanfar, S., Tao, Y., Mohan, A., Tian, H., Kaseb, A. S., Gauen, K., et al. (2018). "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (IEEE), 112–117.
- Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., Li, H. (2020). "Balanced meta-softmax for long-tailed visual recognition," in *NeuIPS* (Cambridge).
- Ren, M., Zeng, W., Yang, B., and Urtasun, R. (2018). "Learning to reweight examples for robust deep learning," in *ICML* (New York, NY).
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comp. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y
- Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., et al. (2019). "Meta-weight-net: learning an explicit mapping for sample weighting," in *NeuIPS* (Cambridge).
- Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., et al. (2020). "Equalization loss for long-tailed object recognition," in *CVPR* (Piscataway, NJ).
- Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., and Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved yolo-v3 model. *Comp. Electron. Agric.* 157, 417–426. doi: 10.1016/j.compag.2019.01.012
- Wang, F., Cheng, J., Liu, W., and Liu, H. (2018). Additive margin softmax for face verification. *IEEE Sign. Process. Lett.* doi: 10.1109/LSP.2018.2822810
- Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. X. (2021). "Long-tailed recognition by routing diverse distribution-aware experts," in *ICLR* (Washington, DC).
- Wang, Y.-X., Ramanan, D., and Hebert, M. (2017). "Learning to model the tail," in *NeuIPS* (Cambridge).
- Xu, Z., Chai, Z., and Yuan, C. (2021). "Towards calibrated model for long-tailed visual recognition from prior perspective," in *NeuIPS* (Cambridge).
- Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). *mixup: Beyond Empirical Risk Minimization*. Vancouver, BC: Washington DC.
- Zhang, Y., Hooi, B., Lanqing, H., and Feng, J. (2022). "Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition," in *Advances in Neural Information Processing Systems*. eds S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc.).
- Zhang, Z., and Pfister, T. (2021). "Learning fast sample re-weighting without reward data," in *ICCV* (Piscataway, NJ).
- Zhong, Z., Cui, J., Liu, S., and Jia, J. (2021). "Improving calibration for long-tailed recognition," in *CVPR* (Piscataway, NJ): IEEE.
- 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* (Piscataway, NJ).
- Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2019). Unet++: redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transact. Med. Imaging* 39, 1856–1867. doi: 10.1109/TMI.2019.2959609
- Zhu, J., Wang, Z., Chen, J., Chen, Y.-P. P., and Jiang, Y.-G. (2022). "Balanced contrastive learning for long-tailed visual recognition," in *CVPR* (Piscataway, NJ).

Appendix

Three different image augmentations

CIFAR-LT: For small augmentation, we used random crop (which randomly crops 32×32 pixels in the image) with a padding of four pixels and a random horizontal flip. For medium augmentation, we used RandomResizedCrop (the crop size of which is the same as random crop) with scale (0, 2, 1), random horizontal flip, and the same setting with SimCLR (Chen et al., 2020), which includes random gray, random GaussianBlur, and random ColorJitter. For strong augmentation, we used random

crop (which randomly crops 32×32 pixels in the image) with a padding of four pixels, random horizontal flip, and CIFAR-Policy (Cubuk et al., 2018).

ImageNet-LT: For small augmentation, we used RandomResizedCrop (which randomly crop the image and resized it to 224×224 pixels in the image), random horizontal flip, and random ColorJitter. For medium augmentation, we used RandomResizedCrop, random horizontal flip, and the same setting with SimCLR (Chen et al., 2020). For strong augmentation, we used RandomResizedCrop and random horizontal flip with ImageNet-Policy (Cubuk et al., 2018).