Enhancing Class-Imbalanced Learning with Pre-Trained Guidance through Class-Conditional Knowledge Distillation

Lan Li [1 2] Xin-Chun Li [1 2] Han-Jia Ye [1 2] De-Chuan Zhan [1 2]

Abstract

In class-imbalanced learning, the scarcity of information about minority classes presents challenges in obtaining generalizable features for these classes. Leveraging large-scale pre-trained models with powerful generalization capabilities as teacher models can help fill this information gap. Traditional knowledge distillation transfers the label distribution $p(\boldsymbol{y}|\boldsymbol{x})$ predicted by the teacher model to the student model. However, this method falls short on imbalanced data as it fails to capture the class-conditional probability distribution $p(\boldsymbol{x}|\boldsymbol{y})$ from the teacher model, which is crucial for enhancing generalization. To overcome this, we propose Class-Conditional Knowledge Distillation (CCKD), a novel approach that enables learning of the teacher model's class-conditional probability distribution during the distillation process. Additionally, we introduce Augmented CCKD (ACCKD), which involves distillation on a constructed class-balanced dataset (formed through data mixing) and feature imitation on the entire dataset to further facilitate the learning of features. Experimental results on various imbalanced datasets demonstrate an average accuracy improvement of 7.4% using our method.

1. Introduction

Real-world datasets often exhibit imbalances (Horn et al., 2018), with some classes having only a few samples. Models trained on such imbalanced data often face challenges in generalizing well to balanced test data, especially for rare classes (Liu et al., 2019). Improving recognition performance on imbalanced data presents a significant challenge for modern deep learning methods. Class-imbalanced learning (CIL) (Cui et al., 2019) focuses on addressing how to learn from highly imbalanced data, which have two primary challenges: (1) mitigating classifier bias resulting from imbalanced class distributions in training samples, and (2) improving the generalization of minority class samples due to their limited representation.

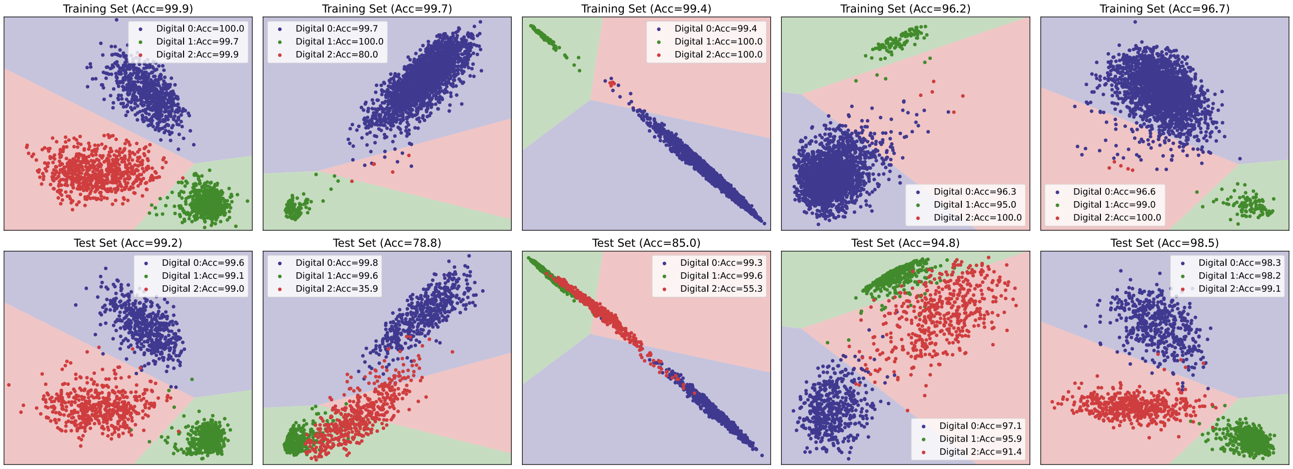
Specifically, deep neural networks try to learn the posterior probability of samples, denoted as $p(\boldsymbol{y}|\boldsymbol{x})$, which can be decomposed using Bayes' theorem as $p(\boldsymbol{y}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})$. The first challenge arises from the mismatch between the imbalanced distribution in the training samples and the typically uniform distribution in the test data, i.e., $p_{te}(\boldsymbol{y}) \neq p_{tr}(\boldsymbol{y})$. Models trained on such data tend to favor the majority class, leading to suboptimal performance on minority classes during testing. Current strategies involve adjusting class weights in the loss function (Cui et al., 2019; Cao et al., 2019; Menon et al., 2021; Kang et al., 2020) or resampling (Chawla et al., 2002; Drummond et al., 2003; He & Garcia, 2009; Afonin & Karimireddy, 2022).

The second challenge stems from the inadequacy of information in the few samples of the minority class to accurately represent its distribution. This results in a significant mismatch between the learned $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ and the true conditional probability distribution of samples. Common approaches include incorporating prior knowledge to enrich the minority class distribution, using methods like sample interpolation (Ando & Huang, 2017; Ye et al., 2020), data augmentation (Ahn et al., 2023) and generating adversarial samples (Kim et al., 2020). However, this challenge remains akin to "making bricks without straw."

To tackle the second challenge, we propose leveraging a large-scale pre-trained model to compensate for the lacking information about minority classes. With the development of deep neural networks, large-scale pre-trained models have demonstrated remarkable generalization capabilities. For example, models such as CLIP (Radford et al., 2021) demonstrate outstanding zero-shot classification performance on new tasks after being pre-trained on extensive image-text datasets. Upon fine-tuning on task-specific samples, the refined model demonstrates robust classification abilities. However, these models are often large, posing deployment

[1] School of Artificial Intelligence, Nanjing University, China [2] National Key Laboratory for Novel Software Technology, Nanjing University, China. Correspondence to: Han-Jia Ye <yehj@lamda.nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Teacher (b) KD (c) Weighted KD (d) CCKD (e) ACCKD

Figure 1. Experiment on the classification of digits 0, 1, and 2 in MNIST. We employed a neural network to extract two-dimensional features from samples and used cross-entropy loss for classification. The classification regions of each class in the feature space, as well as the distributions of training and testing samples, are presented with different colors. The teacher model is trained on data with class sample counts of (700, 700, 700). Various distillation methods perform knowledge distillation on the teacher model using data with class sample counts of (2000, 100, 5).

difficulties and limiting their applicability in specific scenarios (Zhou, 2023). Therefore, we consider leveraging pre-trained models to improve the generalization of minority class samples in CIL.

A direct approach to leveraging pre-trained models is knowledge distillation (Hinton et al., 2015), which involves minimizing the Kullback-Leibler divergence between the outputs of a teacher model and a student model. This allows the student model to learn from the teacher model's outputs. However, when applied to imbalanced data, this approach encounters specific challenges. As shown in Figure 1, a toy experiment was conducted on a class-imbalanced subset dataset constructed from MNIST (LeCun et al., 1998). Figure 1(b) and 1(c) illustrate the results obtained after being trained using distillation loss and class-weighted distillation loss, respectively. It is evident that both methods exhibit poor generalization on minority classes in terms of learned features and corresponding classifiers.

We attribute this situation to the fact that the student model fails to acquire the knowledge embedded in the $p^t(\boldsymbol{x}|\boldsymbol{y})$ of the teacher model. Learning a more accurate $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ is paramount for improving the generalization capability of minority classes in imbalanced scenarios. Therefore, we propose that, on imbalanced data, the student model should learn the $p^t(\boldsymbol{x}|\boldsymbol{y})$ of the teacher model during the distillation process, and we introduce a corresponding method to achieve this. As illustrated in Figure 1(d), this approach significantly enhances the generalization ability of minority class samples.

We also observe that the cause of the divergence between the learned $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ and the teacher model is the significant disparity in the number of samples used for training the teacher model compared to the student model. The teacher model is pre-trained on large-scale datasets, with far more samples than those available for training the student model. Consequently, there is a lack of sufficient data to transfer the information from the teacher model, especially for the minority classes. Therefore, we further propose a method to address this issue by constructing a class-balanced dataset through mixing on the training samples. We introduce a loss specifically designed for learning the $p^t(\boldsymbol{x}|\boldsymbol{y})$ on this balanced dataset. Finally, we incorporate a feature imitation loss to further enhance the learning of the student model. Figure 1 (e) illustrates the effectiveness of our approach.

In summary, our main contributions are as follows: 1. We are the first to propose the use of pre-trained models to facilitate learning on imbalanced data. 2. We suggest that on imbalanced data, the student model should acquire the knowledge of $p^t(\boldsymbol{x}|\boldsymbol{y})$ provided by the pre-trained model. We propose a corresponding learning approach to address this paradigm shift effectively. 3. We additionally present a method to strengthen the learning of $p^t(\boldsymbol{x}|\boldsymbol{y})$ by constructing a synthetic class-balanced dataset and incorporating a feature imitation loss for added efficacy.

2. Related Work

Class-imbalanced learning. The datasets with class imbalances can lead models to learn biases toward training data, causing a significant decrease in performance on balanced test data. The reasons for this include the classifier's inclination towards the majority class and the model's difficulty in learning the generalized feature distribution from minority class samples. Some direct solutions involves resampling (Chawla et al., 2002; Han et al., 2005) and reweighting (Huang et al., 2016; Wang et al., 2017; Cui et al., 2019) at the sample level. However, these methods can compromise the generalization of feature distribution from minority class samples, as overfitting may arise due to the limited information on minority classes resulting from increased weights on a small number of minority samples. Kang et al. (2020) and Zhou et al. (2020) highlighted the need to decouple the feature learning and classifier learning stages, separately rebalancing the classifier layer. Therefore, some approaches using two-stage training (Zhong et al., 2021; Alshammari et al., 2022), modifying the classifier in the second stage, have demonstrated notable enhancements in performance. Another category of methods modifies the loss function by meta-learning (Jamal et al., 2020; Chao et al., 2020) or incorporating class weights at the classifier level (Ren et al., 2020; Menon et al., 2021; Ye et al., 2020; Li et al., 2024) rather than the sample level.

In response to the challenge of poor generalization in minority class features, two main categories of methods are proposed: ensemble learning and sample augmentation. Ensemble learning (Wang et al., 2021) methods recommend training multiple diverse expert models to capture diverse representations, thereby enhancing feature generalization. Sample augmentation methods primarily involve enhancing minority class samples in the input (Kim et al., 2020; Ahn et al., 2023; Shi et al., 2023a) or feature space (Ye et al., 2020), enabling the learned feature distribution on these samples to better approximate the true distribution. Additionally, Liu et al. (2022) found that self-supervised representations are more robust when dealing with class imbalance than supervised representations. Some studies have already developed supervised contrastive learning methods (Kang et al., 2021; Cui et al., 2021; Zhu et al., 2022) specifically designed for imbalanced datasets. Differing from these approaches, our main emphasis is on exploring how to leverage information provided by a pre-trained model to tackle this problem. Additionally, as pre-trained models have shown promising zero-shot classification performance, recent research (Wang et al., 2024; Shi et al., 2023b; Dong et al., 2022) has also focused on fine-tuning these models on imbalanced data.

Knowledge distillation. Knowledge Distillation (KD) is a technique used for transferring knowledge between different models (Hinton et al., 2015), primarily applied in model compression. Recently, He et al. (2021); Zhang et al. (2023); Xiang et al. (2020) have employed KD to harness the additional supervision for class-imbalanced learning. However, their approach involved training the teacher network from scratch on imbalanced data, which is less scalable compared to using a single large pre-trained model.

3. Methodology

3.1. Preliminaries and background

Assume a training dataset $D_{tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, consisting of images of size d and their corresponding labels $y_i \in \{1, \dots, C\}$, where N represents the dataset size and $\boldsymbol{x}_i \in \mathbb{R}^d$. The one-hot encoding process converts each label y_i into a vector $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \dots, y_{i,C}) \in \mathbb{R}^C$. In this vector, the k-th component $y_{i,k}$ is set to 1 if $k = y_i$ and 0 otherwise. Define $D_c \subset D$ as the subset of class c, meaning $D_c = \{\boldsymbol{x}, y) \,|\, y = c, (\boldsymbol{x}, y) \in D\}$. For simplicity, let us assume $|D_1| \geq |D_2| \geq \dots \geq |D_C|$, where $|D|$ indicates the cardinality of the dataset. We set N_{max} as $|D_1|$ and N_{min} as $|D_C|$. CIL is designed for training models in situations where the class distribution of the training data, denoted as $p_{tr}(\boldsymbol{y})$, significantly differs from the testing data, represented by $p_{te}(\boldsymbol{y})$. Specifically, $p_{tr}(\boldsymbol{y})$ shows a highly imbalanced distribution, whereas $p_{te}(\boldsymbol{y})$ is balanced.

In deep learning, the challenges arising from class imbalance manifest in two aspects: the predictions of the model tend to be biased towards the majority and exhibit significant learning biases for features of the minority classes. Specifically, deep models seek to minimize classification loss on the training set to learn the posterior probability, $p_{tr}(\boldsymbol{y}|\boldsymbol{x})$, for a sample $\boldsymbol{x}$. This probability can be decomposed using Bayes' theorem as $p_{tr}(\boldsymbol{y}|\boldsymbol{x}) \propto p_{tr}(\boldsymbol{x}|\boldsymbol{y})p_{tr}(\boldsymbol{y})$. While making predictions $p_{te}(\boldsymbol{y}|\boldsymbol{x})$ on the test data, the model assumes $p_{tr}(\boldsymbol{y}) = p_{te}(\boldsymbol{y})$ and $p_{tr}(\boldsymbol{x}|\boldsymbol{y}) = p_{te}(\boldsymbol{x}|\boldsymbol{y})$. The disparities in $p(\boldsymbol{y})$ and $p(\boldsymbol{x}|\boldsymbol{y})$ respectively give rise to these two challenges. The first challenge can be addressed through various methods, such as incorporating a correction term for $p_{tr}(\boldsymbol{y})$ in the training loss.

The second challenge arises from the scarcity of samples in minority classes, making it difficult for the learned $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ to generalize on the test data. This discrepancy is evident in the distinct feature distributions of minority class samples between the training and the test sets, preventing the classifier from effectively distinguishing minority class samples in the test data. This problem is difficult to address due to a lack of relevant knowledge about the distribution of minority classes during training. The ability of the model to accurately learn $p_{tr}(\boldsymbol{x}|\boldsymbol{y})$ determines the upper bounds of its capabilities.

To address the problem of inaccurate $p(\boldsymbol{x}|\boldsymbol{y})$, the introduction of prior knowledge is necessary. Given the remarkable generalization capabilities demonstrated by pre-trained models such as CLIP, we propose resolving this problem by distilling relevant knowledge from them. We denote the pre-trained model, serving as the teacher model, as f_t, and the model to be trained, the student model, as f_s.

For the pre-trained model $f_t = g_t \cdot h_t$, where g_t represents the feature encoder and h_t represents the classifier, we consider three approaches to construct the teacher model: Zero-shot (Radford et al., 2021), NCM (Nearest Class Mean) (Kang et al., 2020), and Fine-tuning (Chen et al., 2022). For the zero-shot and NCM pre-training models, g_t directly uses the CLIP image encoder, and h_t consists of a cosine classifier in both cases. For the zero-shot approach, we concatenate the prompt template and the class label to create a new input, which is then processed by the text encoder of CLIP. The output from it serves as the parameters for h_t. In the NCM approach, the parameters of h_t are obtained by calculating the feature mean for each class of training samples. Additionally, f_t can be obtained by fine-tuning the CLIP image encoder or through parameter-efficient fine-tuning. The student model $f_s = g_s \cdot h_s$ similarly includes both an image encoder g_s and a classifier h_s. The output logits for a sample $\boldsymbol{x}$ on the student model and the teacher model are denoted as $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$, respectively.

In summary, our primary focus is on addressing the challenge of leveraging large-scale pre-trained models to assist in learning from imbalanced data.

3.2. What to Distill, $p(\boldsymbol{y}|\boldsymbol{x})$ or $p(\boldsymbol{x}|\boldsymbol{y})$?

Knowledge distillation uses knowledge embedded in the teacher model to help train a student network. For a given training example $\boldsymbol{x}$, KD transfers the knowledge by minimizing the distance, typically measured using the Kullback-Leibler (KL) divergence, between $p^t(\boldsymbol{y}|\boldsymbol{x})$ and $p^s(\boldsymbol{y}|\boldsymbol{x})$ on the training dataset. This process is accomplished by minimizing the distillation loss $\mathcal{L}_{dis}$:

$$\mathcal{L}_{kd} = \mathrm{KL}\left(p^t(\boldsymbol{y}|\boldsymbol{x}) \parallel p^s(\boldsymbol{y}|\boldsymbol{x})\right). \tag{1}$$

KD typically assumes a close match between the class distribution of the training data and test data. However, in the context of CIL, there exists a systematic disparity between the distributions of training and test samples, thereby invalidating this assumption. The model exhibits better fitting performance on majority classes, while experiencing comparatively higher losses on classes with fewer samples.

To overcome this challenge, a direct strategy involves balancing the distillation loss across different classes. This can be achieved by amplifying the distillation loss on minority class samples, such as reweighting, to ensure the convergence of average losses across diverse class samples. Nevertheless, as illustrated in Figure 1, this method diminishes the generalization capability of features.

The observed phenomenon can be attributed the insufficient generalization of the model caused by learning $p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})$ on imbalanced data. According to Bayes' theorem, $p^t(\boldsymbol{y}|\boldsymbol{x})$, is determined by $p(\boldsymbol{y})$ and $p^t(\boldsymbol{x}|\boldsymbol{y})$. While $p(\boldsymbol{y})$ reflects the class preferences of the teacher model, $p(\boldsymbol{x}|\boldsymbol{y})$ characterizes the distribution of sample $\boldsymbol{x}$ given label $\boldsymbol{y}$, signifying the teacher model's grasp of the intrinsic data structure, which is crucial for the generalization. When $p_{tr}(\boldsymbol{y}) \neq p(\boldsymbol{y})$, forcefully minimizing the distance between $p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})$ and $p^t(\boldsymbol{y}|\boldsymbol{x})$ inevitably introduces discrepancies between $p^s(\boldsymbol{y}|\boldsymbol{x})$ and $p^t(\boldsymbol{y}|\boldsymbol{x})$, resulting in the loss of crucial information embedded in the teacher model.

Therefore, we consider allowing the student model to learn $p^t(\boldsymbol{x}|\boldsymbol{y})$ of the teacher model. A direct approach is to minimize the KL divergence of them:

$$\mathrm{KL}\left(p^t(\boldsymbol{x}|\boldsymbol{y}) \parallel p^s_{tr}(\boldsymbol{x}|\boldsymbol{y})\right). \tag{2}$$

However, since the distribution of $\boldsymbol{x}$ is continuous and difficult to compute, direct optimization of Equation (2) is not feasible. Therefore, we address this challenge by learning the student model's $p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})$ such that $p^t(\boldsymbol{x}|\boldsymbol{y}) = p^s_{tr}(\boldsymbol{x}|\boldsymbol{y})$. According to Bayes' theorem, we have $\frac{p^t(\boldsymbol{y}|\boldsymbol{x})}{p^t(\boldsymbol{y})} \propto \frac{p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})}{p^s_{tr}(\boldsymbol{y})}$. Combining this with Equation (1), we obtain:

$$\mathcal{L}_{cckd} = \mathrm{KL}\left(q(\boldsymbol{y}|\boldsymbol{x}) \parallel \mathrm{norm}(\frac{p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})}{p^s_{tr}(\boldsymbol{y})})\right), \tag{3}$$

where $q(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{norm}(\frac{p^t(\boldsymbol{y}|\boldsymbol{x})}{p^t(\boldsymbol{y})})$, norm represents the normalization of the distribution, and $p^s_{tr}(\boldsymbol{y})$ represents the class distribution of the training samples, i.e.,

$$p^s_{tr}(\boldsymbol{y}) = \frac{1}{|D_{tr}|} \sum_{\boldsymbol{x} \sim D_{tr}(\boldsymbol{x})} q(\boldsymbol{y}|\boldsymbol{x}). \tag{4}$$

The model assumes the same class-conditional probabilities between the training and test data when making predictions on the test data, that is:

$$\frac{p^s_{tr}(\boldsymbol{y}|\boldsymbol{x})}{p^s_{tr}(\boldsymbol{y})} \propto \frac{p^s_{te}(\boldsymbol{y}|\boldsymbol{x})}{p^s_{te}(\boldsymbol{y})}. \tag{5}$$

Substituting Equation 5 into Equation 3, we have:

$$\mathcal{L}_{cckd} = \mathrm{KL}\left(q(\boldsymbol{y}|\boldsymbol{x}) \parallel \mathrm{norm}(\frac{p^s_{te}(\boldsymbol{y}|\boldsymbol{x}) p^s_{tr}(\boldsymbol{y})}{p^s_{te}(\boldsymbol{y})})\right). \tag{6}$$

$p^s_{te}(\boldsymbol{y}|\boldsymbol{x}) = [s_1, s_2, \dots, s_C]$ and $p^t(\boldsymbol{y}|\boldsymbol{x}) = [t_1, t_2, \dots, t_C]$ are obtained by applying the softmax function to the logits $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$, respectively:

$$s_i = \frac{e^{z_i/\tau}}{\sum_{k=1}^{C} e^{z_k/\tau}}, t_i = \frac{e^{\hat{z}_i/\tau}}{\sum_{k=1}^{C} e^{\hat{z}/\tau}}. \tag{7}$$

By applying the softmax function in equation 7 to equation 6, we derive Proposition 3.1:

Proposition 3.1. *On class-imbalanced training set D_{tr}, the student model can learn the class-conditional probability distribution of the teacher model by $\mathcal{L}_{cckd}$:*

$$\begin{aligned} \mathcal{L}_{cckd} = -\sum_{i=1}^{C} \Bigg(& \frac{e^{(\hat{z}_i - \log(p^t_i))/\tau}}{\sum_{j=1}^{C} e^{(\hat{z}_i - \log(p^t_j))/\tau}} \\ & \times \log \frac{e^{(z_i + \log(p^s_{tr,i}) - \log(p^s_{te,i}))/\tau}}{\sum_{j=1}^{C} e^{(z_j + \log(p^s_{tr,j}) - \log(p^s_{te,j}))/\tau}} \Bigg). \end{aligned} \tag{8}$$

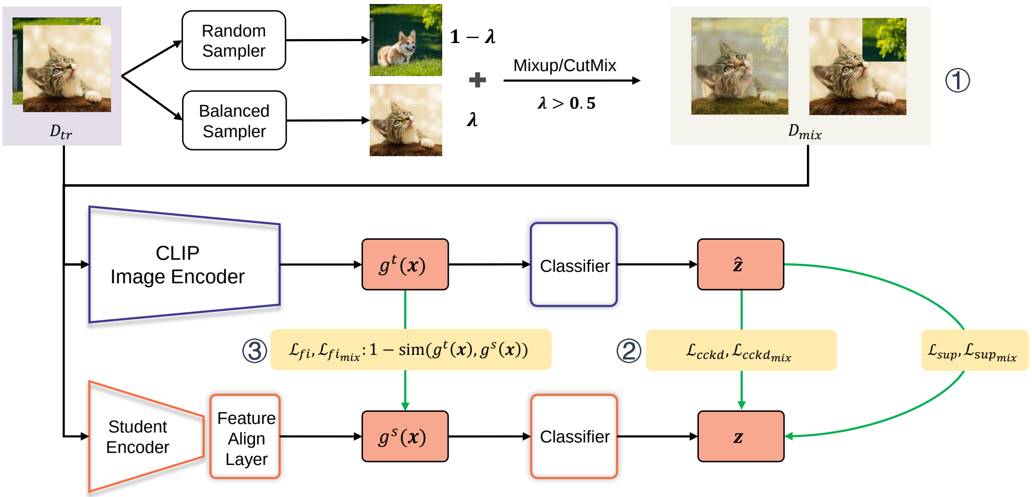


Figure 2. Our method has three main components. ① We construct a balanced dataset D_{mix} by mixing the training data. ② We employ Class-Conditional Knowledge Distillation (CCKD) on both D_{tr} and D_{mix} to learn the $p^t(\boldsymbol{x}|\boldsymbol{y})$ of teacher model. ③ We facilitate the learning of $p^t(\boldsymbol{x}|\boldsymbol{y})$ by feature imitation loss.

The detailed proof can be found in the Appendix A. It is essential to note that $p^t(\boldsymbol{y}) = [p_1^t, p_2^t, \ldots, p_C^t]$ is used to mitigate the class bias present in the pre-trained model during prediction. It should be estimated using a validation set D_{val} that shares the same distribution as the test data, i.e.,

$$p^t(\boldsymbol{y}) = \frac{1}{|D_{val}|} \sum_{\boldsymbol{x} \in D_{val}(\boldsymbol{x})} p^t(\boldsymbol{y}|\boldsymbol{x}). \tag{9}$$

When D_{val} is not available, it can be substituted by resampling the training set. $p_{te}^s(\boldsymbol{y}) = [p_{te,1}^t, p_{te,2}^t, \ldots, p_{te,C}^t]$ represents the class-conditional probability distribution $p^t(\boldsymbol{x}|\boldsymbol{y})$ learned by the student model on the test data. Since the pre-trained teacher model can learn a reliable $p^t(\boldsymbol{x}|\boldsymbol{y})$, we can approximate $p_{te}^s(\boldsymbol{y})$ as the class distribution of the test data.

3.3. Distillation on Synthesis Data

In traditional knowledge distillation, the teacher model is usually a larger model trained on the same data as the student. When the teacher model is pre-trained, and the training data is imbalanced, solely training on this imbalanced data fails to capture the task-specific information embedded in the teacher model. To address this challenge, augmenting the training data with a more extensive set of task-related samples is crucial.

Mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019) are widely adopted techniques for synthesizing task-related samples. Mixup generates novel training samples by blending two training samples linearly. Assuming λ is a randomly sampled value from the Beta distribution within the range $[0, 1]$, $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are two training samples, the synthetic sample $\boldsymbol{x}_{\text{mixup}}$ created by Mixup can be expressed as:

$$\boldsymbol{x}_{\text{mixup}} = \lambda \boldsymbol{x}_a + (1 - \lambda_1)\boldsymbol{x}_b. \tag{10}$$

CutMix is another data augmentation technique that operates by cutting and pasting patches between training images. In CutMix, a random rectangular region is cropped from sample $\boldsymbol{x}_a$ and pasted onto the corresponding position in sample $\boldsymbol{x}_b$. Assuming Mask is a binary matrix representing the cropped region from sample $\boldsymbol{x}_a$ (with values of 1) and the uncropped region (with values of 0), the size of the cropping region, i.e., the proportion of the region with values of 1 in Mask, depends on the λ randomly sampled from a beta distribution. The new sample can be represented as:

$$\boldsymbol{x}_{\text{cutmix}} = \text{Mask} \cdot \boldsymbol{x}_a + (1 - \text{Mask}) \cdot \boldsymbol{x}_b. \tag{11}$$

We employ Mixup and CutMix to construct a dataset, denoted as D_{mix}, that is both class-balanced and task-specific. Specifically, we select an instance $\boldsymbol{x}_a$ from D_{tr} using class-balanced sampling and another instance $\boldsymbol{x}_b$ through random sampling. We then randomly apply either Mixup or CutMix to create the synthetic data $\boldsymbol{x}_{mix}$. To maintain class balance within D_{mix}, we ensure that the mixing coefficient λ is greater than 0.5 during the synthesis process.

Based on Equation (6), we define the loss function for knowledge distillation when applied to data from D_{mix} as follows:

$$\mathcal{L}_{cckd_{mix}} = \text{KL}\left(q(\boldsymbol{y}|\boldsymbol{x}_{mix}) \parallel p_{tr}^s(\boldsymbol{y}|\boldsymbol{x}_{mix})\right). \tag{12}$$

In the supervised loss, we provide supervision by combining the labels of the $\boldsymbol{x}_{mix}$ with its prediction from the teacher model. Specifically, $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are the two samples used to synthesize $\boldsymbol{x}$, and they have corresponding labels y_a and y_b. $\hat{\boldsymbol{z}} = [\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_C]$ represents the logit of $\boldsymbol{x}$ predicted by the teacher model. For binary classification, the probabilities of $\boldsymbol{x}_{mix}$ belonging to these two classes are:

$$p_{y_a} = \frac{e^{\hat{z}_{y_a}}}{e^{\hat{z}_{y_a}} + e^{\hat{z}_{y_b}}}, p_{y_b} = \frac{e^{\hat{z}_{y_a}}}{e^{\hat{z}_{y_a}} + e^{\hat{z}_{y_b}}}. \tag{13}$$

We ues p_{y_a} and p_{y_b} as weights for the supervised loss corresponding to these two classes:

$$\mathcal{L}_{sup_{mix}} = -p_{y_a} \log(p^s_{tr,y_a}) - p_{y_b} \log(p^s_{tr,y_b}). \tag{14}$$

3.4. Feature Imitation

For a deep model, it comprises two components: the feature encoder and the classifier. Regarding the classifier, $p(\boldsymbol{x}|\boldsymbol{y})$ represents the distribution of features. In other words, when the two models learn similar $p(\boldsymbol{x}|\boldsymbol{y})$, the feature distributions across each class should also be similar.

To further improve the learning of $p^t(\boldsymbol{x}|\boldsymbol{y})$ during distillation, we introduce a feature imitation loss to encourage similarity in feature distributions between the teacher and student models. Specifically, for a sample $(\boldsymbol{x}, y)$ in D_{tr} and a sample $\boldsymbol{x}_{mix}$ in D_{mix}, we achieve this by maximizing the cosine similarity between the features obtained from the teacher and student models through:

$$\begin{aligned} \mathcal{L}_{fi} &= \frac{1}{|D_y|}(1 - \mathrm{sim}(g^t(\boldsymbol{x}), g^s(\boldsymbol{x}))), \\ \mathcal{L}_{fi_{mix}} &= 1 - \mathrm{sim}(g^t(\boldsymbol{x}_{mix}), g^s(\boldsymbol{x}_{mix})), \end{aligned} \tag{15}$$

where sim represents cosine similarity. Additionally, due to the different feature dimensions of the teacher and student models, a linear layer is added at the end of the student model's feature encoder to align with the feature dimension of the teacher model.

In summary, the overall loss function is $\mathcal{L}_{acckd} = \mathcal{L}_{tr} + \mathcal{L}_{mix}$, where $\mathcal{L}_{tr}$ and $\mathcal{L}_{mix}$ denoting the losses on the training data D_{tr} and the synthetic data D_{mix}, respectively:

$$\mathcal{L}_{tr} = \mathcal{L}_{sup} + \alpha_1 \mathcal{L}_{cckd} + \alpha_2 \mathcal{L}_{fi}, \tag{16}$$

$$\mathcal{L}_{mix} = \mathcal{L}_{sup_{mix}} + \alpha_1 \mathcal{L}_{cckd_{mix}} + \alpha_2 \mathcal{L}_{fi_{mix}}. \tag{17}$$

$\mathcal{L}_{sup}$ is the logit adjustment loss, which is widely used in CIL. α_1 and α_2 control the ratio between these sub-losses. The pipeline of our method is presented in Figure 2, and the pseudo code can be found in Appendix B.

4. Experience

4.1. Experimental Setup

Dataset. We performed extensive experiments on three widely used imbalanced datasets: CIFAR-100-LT (Cui et al., 2019), ImageNet-LT (Russakovsky et al., 2015), and iNaturalist 2018 (Horn et al., 2018). For CIFAR-100, following Cui et al. (2019), we created class imbalance by the imbalance ratio of $N_{max}/N_{min} = 100$. ImageNet-LT consists of 115.8K images across 1000 classes, with a maximum of 1280 images and a minimum of 5 images per class. iNaturalist 2018 is a naturally imbalanced dataset composed of 437.5K images distributed among 8142 species.

Evaluation protocol. After training on the imbalanced dataset, we evaluated the model on the corresponding balanced test dataset. We reported the commonly used top-1 accuracy, denoted as All, across all classes. To better examine the performance across classes with varying numbers of examples seen during training, we adhered to the evaluation protocol introduced by Liu et al. (2019) to report accuracy on three splits of the set of classes: Many-shot (more than 100 images), Medium-shot (20∼100 images) and Few-shot (less than 20 images). Accuracy is reported as a percentage.

Prepare for teacher model. We employ the image encoder of CLIP (ViT-B/16) (Radford et al., 2021) as the teacher model and consider three methods to leverage this model. **(a) Zero-shot(ZS).** Zero-shot predicts by computing the cosine similarity between image features and each class center as the model output. We generate hand-crafted textual prompts (e.g., "a photo of a [CLASS].") and use the text encoder to compute their features as class centers. This prediction method is entirely unrelated to the class distribution of training samples. **(b) Nearest Class Mean(NCM)** (Kang et al., 2020). In contrast to Zero-shot predictions, NCM primarily differs by using the mean of the features of training samples for each class as the class center. Due to poor class center estimation on minority classes, this method is somewhat influenced by the class distribution. **(c) Adaptformer+Logit Adjustment(AF+LA)** (Chen et al., 2022; Menon et al., 2021). Due to the massive parameter size of the image encoder, which is challenging for fine-tuning, we employ Adaptformer, a parameter-efficient fine-tuning method, to adapt the teacher model on training data. Given the imbalanced distribution in the training data, we use logit adjustment loss as the loss function.

Comparative methods. In our experiments, we compare multiple approaches. Firstly, we evaluate the performance of student models trained solely on the training set using either cross entropy (CE) or logit adjustment (LA) loss, without the assistance of a teacher model. This set of experiments serves as the baseline, providing a comparison point for the student model's performance before incorporating various teacher models and corresponding knowledge distillation methods.

We then categorize the experiments into three groups based on different teacher models: zero-shot, NCM, and AF+LA. For each group, we compare the results of different distillation methods. Initially, we present the performance of the teacher models. It is important to note that since the teacher models are trained on large-scale datasets and possess a large number of parameters, the student models, despite receiving guidance from the teacher models, cannot be fairly compared due to their limited and imbalanced data and smaller parameter sizes. Therefore, the performance of the teacher models is provided as a reference, rather than

Table 1. Top-1 accuracy (%) on CIFAR-100-LT and ImageNet-LT. CCKD and ACCKD are proposed method.The first two rows show results from training the student model alone, while the subsequent rows present results for three teacher models and the student model leveraging these teacher models through different distillation methods.

Methods	CIFAR100-LT					ImageNet-LT				
	Param Num	Many	Medium	Few	All	Param Num	Many	Medium	Few	All
Student (CE)	0.47M	68.3	39.1	7.2	39.8	12.3M	60.5	33.8	12.0	41.1
Student (LA)	0.47M	60.8	45.7	27.2	45.5	12.3M	55.7	40.9	26.4	44.6
Teacher (ZS)	89.2M	66.9	65.9	67.3	66.7	89.2M	68.2	66.4	66.6	67.1
w / KD	0.47M	65.5	50.7	23.9	47.9	12.3M	58.6	47.3	33.5	49.8
w / WKD	0.47M	63.5	47.4	25.1	46.4	12.3M	57.5	45.3	32.3	48.3
w / DiVE	0.47M	69.5	49.6	7.2	43.8	12.3M	61.0	45.1	24.6	48.4
w / CCKD	0.47M	63.2	54.5	23.2	48.2	12.3M	58.9	50.2	41.3	52.3
w / ACCKD	0.47M	65.1	53.1	29.0	**50.1**	12.3M	60.8	52.8	42.9	**54.5**
Teacher (NCM)	89.2M	71.0	68.4	53.8	64.9	89.2M	69.1	63.5	51.1	64.0
w / KD	0.47M	68.1	50.0	10.5	44.5	12.3M	58.9	45.8	28.4	48.5
w / WKD	0.47M	68.8	47.1	10.6	43.8	12.3M	61.9	44.1	21.2	47.8
w / DiVE	0.47M	68.6	46.1	1.6	40.6	12.3M	60.3	45.8	25.4	48.6
w / CCKD	0.47M	60.7	46.2	28.0	45.8	12.3M	59.1	46.4	31.9	49.3
w / ACCKD	0.47M	64.3	48.4	25.5	**47.1**	12.3M	60.0	47.1	34.1	**50.3**
Teacher (AF+LA)	89.3M	85.1	76.4	64.6	75.9	89.9M	80.6	75.6	68.6	76.6
w / KD	0.47M	65.1	47.0	23.7	46.3	12.3M	60.4	46.3	32.6	49.9
w / WKD	0.47M	59.4	45.7	22.1	43.4	12.3M	57.6	45.9	33.0	48.6
w / DiVE	0.47M	68.6	51.3	10.8	45.2	12.3M	60.2	45.0	25.0	48.2
w / CCKD	0.47M	55.5	47.8	34.6	46.6	12.3M	57.5	51.3	44.8	52.8
w / ACCKD	0.47M	64.6	50.7	30.2	**49.1**	12.3M	61.1	53.4	46.5	**55.4**

an expectation for the student models to surpass the teacher models. Specifically, the performance of the AF+LA teacher model can be considered as an upper bound for the student model's performance.

Within each group of teacher model experiments, we compare three different distillation methods: (a) Knowledge Distillation (KD) involves directly using the knowledge distillation loss by minimizing the KL divergence between the outputs of the teacher and student models. (b) Weighted KD (WKD) extends the knowledge distillation loss by weighting the loss of different class samples according to the inverse of their frequency. (c) DiVE (He et al., 2021) is designed for distilling on class-imbalanced data. It adjusts the logits of the teacher model by introducing a temperature parameter, making the teacher model's output distribution flatter. CCKD and Augmented CCKD (ACCKD) are the methods we propose. Compared to other methods, CCKD replaces the distillation loss with class-conditional knowledge distillation loss. ACCKD extends CCKD by incorporating additional losses on synthesis data and feature imitation losses. To ensure a fair comparison, each method employs logit adjustment loss for the supervised component.

Implementation Details. Following Cao et al. (2019), we use ResNet-32 as the student model on CIFAR-100-LT and train for 200 epochs. For ImageNet-LT and iNaturalist 2018, we utilize ResNet-18 as the backbone and train for 200 and 90 epochs, respectively. All experiments are run three times and the average results are reported. More details on the implementation are presented in Appendix C.

4.2. Main results

Results on Cifar100-LT and ImageNet-LT. Table 1 presents the performance of different methods on CIFAR-100-LT and ImageNet-LT. Experimental results show that with CCKD loss, our proposed approach outperforms all comparative methods in various settings. With the addition of feature imitation loss and assistance from synthesized mixed data, the performance further improves. Compared to models without distillation losses (LA), it achieves an average improvement of 3.8% on CIFAR100-LT and 8.8% on ImageNet-LT. Moreover, different from methods that enhance the performance of few-shot and medium-shot classes

Table 2. Top-1 accuracy (%) on iNaturalist 2018(Resnet-18). CCKD and ACCKD are proposed method. The first two rows show results from training the student model alone, while the subsequent rows present results for three teacher models and the student model leveraging these teacher models through different distillation methods.

Methods	Many	Medium	Few	All
Student (CE)	47.7	63.3	54.0	52.4
Student (LA)	59.2	52.2	57.5	57.6
Teacher (ZS)	7.4	3.9	3.4	4.1
w / KD	49.3	54.9	54.1	52.3
w / WKD	59.3	51.5	40.3	47.9
w / DiVE	55.3	56.8	57.5	56.6
w / CCKD	51.2	58.3	57.1	54.9
w / ACCKD	58.2	58.0	59.7	**58.9**
Teacher (NCM)	35.1	42.6	46.5	43.4
w / KD	55.7	51.3	55.7	55.2
w / WKD	61.4	55.7	45.5	52.2
w / DiVE	55.0	55.3	56.9	56.0
w / CCKD	54.9	58.7	57.6	56.6
w / ACCKD	60.3	56.9	60.6	**60.1**
Teacher (AF+LA)	71.9	78.1	80.1	78.2
w / KD	61.5	58.4	61.5	61.1
w / WKD	46.4	57.3	59.3	56.9
w / DiVE	60.1	57.5	59.8	59.7
w / CCKD	67.0	54.4	63.2	63.8
w / ACCKD	54.8	66.1	70.5	**66.7**

at the expense of reducing the performance of many-shot classes, our approach consistently improves the performance of classes with different frequencies.

Results on iNaturalist 2018. The experimental results on iNaturalist are shown in Table 2. Due to the poor performance of the zero-shot teacher model on iNaturalist data, comparative methods exhibit a decline in performance when utilizing this teacher model. Only ACCKD can effectively leverage the teacher model to enhance performance. When NCM and AF+LA serve as teachers, both CCKD and ACCKD outperform comparative methods, with CCKD and ACCKD achieving average performance improvements of 2.6% and 5.8%, respectively. These experimental results strongly highlight the effectiveness of our approach.

4.3. Discussion

Do out method learn more generalizable features for minority classes? The performance of the model is influenced by both the feature encoder and the classifier. To better evaluate the generalization of the features, we first determine the feature center for each class on a balanced validation set. Then, we extract features from the test set and classify the test samples using the nearest class mean approach. This method uses only the feature encoder, excluding the influence of the classifier. The results are shown in Figure 3. It is evident that our method achieves the most competitive results across different frequencies and overall accuracy. This demonstrates that, compared to other methods, our method can significantly improve the generalization of the learned features with the help of the teacher model.

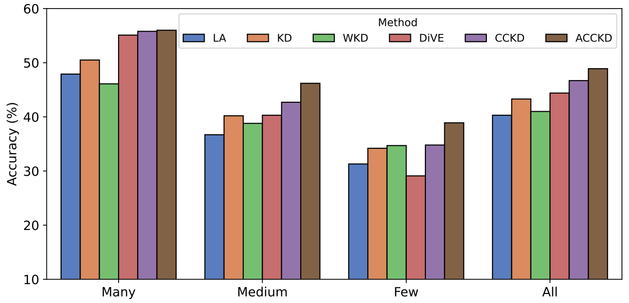


Figure 3. Accuracy of different models using the nearest class mean classification on the test set, with class centers obtained from a balanced validation set.

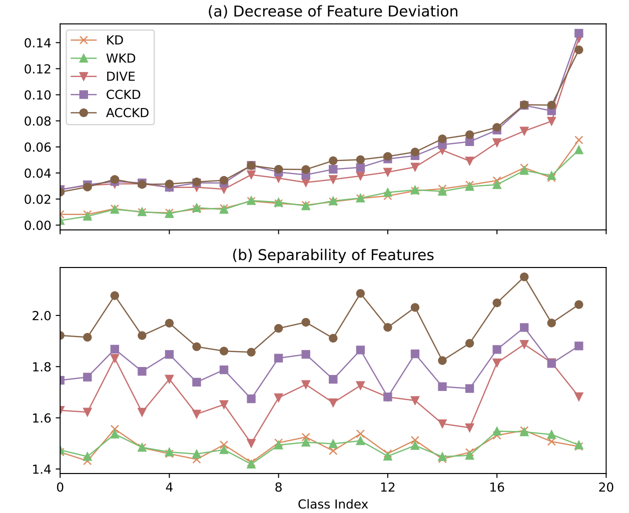


Figure 4. (a) Reduction in feature deviation across different classes. (b) Feature separability across different classes. The class index is sorted in descending order by frequency, and the displayed values are averaged over every 50 classes.

How do our methods learn more generalizable features? To investigate how our method learns more generalizable features, we first evaluate the feature deviation of different models. Feature deviation is defined as the distance between the mean feature vectors of each class in the training and test sets. Ye et al. (2020) highlight that one reason for the poor generalization of minority classes in class-imbalanced learning is the large feature deviation observed in these classes. We calculate the feature deviation for different models on the ImageNet-LT dataset using cosine distance as the distance metric. In Figure 4(a), we show the reduction in

Table 3. Ablation study on ImageNet-LT.

Loss on training data						
$\mathcal{L}_{cckd}$	✓	✓	✓	✓	✓	✓
$\mathcal{L}_{fi}$					✓	✓
Loss on synthesis data						
$\mathcal{L}_{sup_{mix}}$		✓		✓	✓	✓
$\mathcal{L}_{cckd_{mix}}$			✓	✓	✓	✓
$\mathcal{L}_{fi_{mix}}$						✓
Accuracy	52.8	53.4	54.4	54.7	55.0	55.4

Table 4. Performance comparison of student models with different sizes on ImageNet-LT.

Student Model	Param Num	CE	LA	KD	ACCKD
Resnet-10	6.07M	38.3	42.4	46.6	**51.5**
Resnet-18	12.3M	41.1	44.6	48.2	**55.4**
Resnet-34	22.4M	42.8	46.8	51.7	**58.0**
Resnet-50	25.8M	45.4	48.6	55.3	**61.0**
Average	/	41.9	45.6	50.4	**56.5**

feature deviation for various classes when using different distillation methods compared to models trained solely with the logit adjustment loss. All methods can reduce feature deviation to some extent, but ours significantly reduce feature deviation, especially for minority classes.

To further compare the features learned by different methods, we first calculate the distance from each sample in the test set to the center of its own class. Then, we calculate the average distance from the sample to the centers of other classes. The ratio of these two distances serves as a metric for the separability of the features. Figure 4(b) shows the average value of this metric for each class. As shown in Figure 4(b), although the feature deviation of DiVE and CCKD is similar to that of ACCKD, ACCKD learns more separable features, thereby improving feature generalization.

Does each component of ACCKD contribute effectively? o verify the effectiveness of each component in ACCKD, we perform ablation experiments on ImageNet-LT using the teacher model fine-tuned with Adaptformer. Gradually incorporating each component of our proposed method, we report the accuracy on the test data. As depicted in Table 3, each component in ACCKD positively contributes to the final performance.

Can student models of different sizes benefit from our approach? We perform a series of experiments on Imagenet-LT using ResNet-$\{10, 18, 34, 50\}$ as backbones, as detailed in Table 4. ACCKD consistently outperforms KD across various backbones, achieving an average performance improvement of 6.1%. Compare to LA, ACCKD demonstrates performance gains of 9.1%, 10.8%, 11.2%, and 12.4% on different backbones. This indicates that our method can achieve higher gains on larger student models.

Table 5. Performance of teacher models obtained through different fine-tuning methods and the corresponding student models obtained through ACCKD on the ImageNet-LT dataset.

FT Method (Params Num)	Model	Many	Med	Few	All
VPT-shallow (0.08M)	Tea.	77.9	73.2	62.3	73.5
	Stu.	60.2	52.7	42.8	54.2
VPT-deep (0.17M)	Tea.	79.6	75.2	68.0	75.9
	Stu.	60.6	53.6	46.3	55.3
LoRA (1.26M)	Tea.	79.1	74.6	67.1	75.3
	Stu.	60.2	53.2	45.7	54.9
Adapter (0.69M)	Tea.	80.6	75.4	67.6	76.4
	Stu.	60.6	53.3	47.2	55.3
AdaptFormer (0.69M)	Tea.	80.6	75.6	68.6	76.6
	Stu.	61.1	53.4	46.5	55.4

Does the method of fine-tuning the pre-trained model impact the learning of the student model? We fine-tune the pre-trained image encoder using various methods, such as VPT-shallow, VPT-deep (Jia et al., 2022), LoRA (Hu et al., 2022), Adapter (Houlsby et al., 2019), and Adaptformer (Chen et al., 2022). The fine-tuned models are employed as teacher models, and ACCKD is applied to assist in training student models on ImageNet-LT. As shown in Table 5, there is a positive correlation between the performance of student models and teacher models. In the case of fine-tuned teacher models, higher-performing teacher models result in better-performing student models. Considering the balance between fine-tuning costs and final performance, Adaptformer emerges as the most suitable fine-tuning method.

5. Conclusion

To address the challenge of learning generalizable features for minority classes with limited samples in class-imbalanced learning, we propose leveraging a powerful pre-trained model to supplement missing information. The introduction of Class-Conditional Knowledge Distillation (CCKD) loss captures the class-conditional probability distribution of the pre-trained model on imbalanced data. Additionally, we present Augmented CCKD (ACCKD) to further enhance the learning of these probabilities. Extensive experiments on various datasets validate the effectiveness of our approach, consistently outperforming comparative methods.

Acknowledgements

This work is partially supported by National Science and Technology Major Project (2022ZD0114805), NSFC (62376118, 62006112, 62250069, 61921006), Postgraduate Research & Practice Innovation Program of Jiangsu Province, Collaborative Innovation Center of Novel Software Technology and Industrialization.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Afonin, A. and Karimireddy, S. P. Towards model agnostic federated learning using knowledge distillation. In *The Tenth International Conference on Learning Representations*, 2022.

Ahn, S., Ko, J., and Yun, S. CUDA: curriculum of data augmentation for long-tailed recognition. In *The Eleventh International Conference on Learning Representations*, 2023.

Alshammari, S., Wang, Y., Ramanan, D., and Kong, S. Long- tailed recognition via weight balancing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Ando, S. and Huang, C. Y. Deep over-sampling framework for classifying imbalanced data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017*. Springer, 2017.

Cao, K., Wei, C., Gaidon, A., Aréchiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, 2019.

Chao, W.-L., Ye, H.-J., Zhan, D.-C., Campbell, M., and Weinberger, K. Q. Revisiting meta-learning as supervised learning. *ArXiv preprint*, 2020.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 2002.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 2022.

Cui, J., Zhong, Z., Liu, S., Yu, B., and Jia, J. Parametric contrastive learning. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Cui, Y., Jia, M., Lin, T., Song, Y., and Belongie, S. J. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Dong, B., Zhou, P., Yan, S., and Zuo, W. Lpt: long-tailed prompt tuning for image classification. In *The Eleventh International Conference on Learning Representations*, 2022.

Drummond, C., Holte, R. C., et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, 2003.

Han, H., Wang, W.-Y., and Mao, B.-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*. Springer, 2005.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, (9), 2009.

He, Y., Wu, J., and Wei, X. Distilling virtual examples for long-tailed recognition. In *2021 IEEE/CVF International Conference on Computer Vision*, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *ArXiv preprint*, 2015.

Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., and Chang, B. Disentangling label distribution for long-tailed visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Horn, G. V., Aodha, O. M., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. J. The inaturalist species classification and detection dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

Huang, C., Li, Y., Loy, C. C., and Tang, X. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jamal, M. A., Brown, M., Yang, M., Wang, L., and Gong, B. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*. Springer, 2022.

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations*, 2020.

Kang, B., Li, Y., Xie, S., Yuan, Z., and Feng, J. Exploring balanced feature spaces for representation learning. In *9th International Conference on Learning Representations*, 2021.

Kim, J., Jeong, J., and Shin, J. M2m: Imbalanced classification via major-to-minor translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

Li, L., Tao, B., Han, L., Zhan, D.-c., and Ye, H.-j. Twice class bias correction for imbalanced semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/AAAI.V38I12.29260.

Liu, H., HaoChen, J. Z., Gaidon, A., and Ma, T. Self-supervised learning is more robust to dataset imbalance. In *The Tenth International Conference on Learning Representations*, 2022.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *9th International Conference on Learning Representations*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2021.

Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., and Li, H. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015.

Shi, J.-X., Wei, T., Xiang, Y., and Li, Y.-F. How re-sampling helps for long-tail learning? *Advances in Neural Information Processing Systems*, 2023a.

Shi, J.-X., Wei, T., Zhou, Z., Han, X.-Y., Shao, J.-J., and Li, Y.-F. Parameter-efficient long-tailed recognition. *ArXiv preprint*, 2023b.

Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. X. Long-tailed recognition by routing diverse distribution-aware experts. In *9th International Conference on Learning Representations*, 2021.

Wang, Y., Ramanan, D., and Hebert, M. Learning to model the tail. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017.

Wang, Y., Yu, Z., Wang, J., Heng, Q., Chen, H., Ye, W., Xie, R., Xie, X., and Zhang, S. Exploring vision-language models for imbalanced learning. *International Journal of Computer Vision*, 132(1), 2024.

Xiang, L., Ding, G., and Han, J. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Computer Vision–ECCV 2020: 16th European Conference*. Springer, 2020.

Ye, H.-J., Chen, H.-Y., Zhan, D.-C., and Chao, W.-L. Identifying and compensating for feature deviation in imbalanced deep learning. *ArXiv preprint*, 2020.

Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., and Choe, J. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision*, 2019.

Zhang, H., Cissé, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations*, 2018.

Zhang, S., Chen, C., Hu, X., and Peng, S. Balanced knowledge distillation for long-tailed learning. *Neurocomputing*, 2023.

Zhong, Z., Cui, J., Liu, S., and Jia, J. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Zhou, B., Cui, Q., Wei, X., and Chen, Z. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Zhou, Z.-H. Learnability with time-sharing computational resource concerns. *ArXiv preprint*, 2023.

Zhu, J., Wang, Z., Chen, J., Chen, Y. P., and Jiang, Y. Balanced contrastive learning for long-tailed visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

A. Proof of Proposition 3.1

Proof. According to Equation (6), the class-conditional knowledge distillation loss is:

$$\begin{aligned}\mathcal{L}_{CCKD} &= \mathrm{KL}\left(q(\boldsymbol{y}|\boldsymbol{x}) \parallel \frac{p^s_{te}(\boldsymbol{y}|\boldsymbol{x})p^s_{tr}(\boldsymbol{y})}{p^s_{te}(\boldsymbol{y})}\right) \\ &= -\sum_{i=1}^{C} q_i(\boldsymbol{y}|\boldsymbol{x})\log(r_i(\boldsymbol{y}|\boldsymbol{x})) + q_i(\boldsymbol{y}|\boldsymbol{x})\log(q_i(\boldsymbol{y}|\boldsymbol{x})),\end{aligned} \tag{18}$$

where $q_i(\boldsymbol{y}|\boldsymbol{x})$ and $r_i(\boldsymbol{y}|\boldsymbol{x})$ represent the i-th elements of $q(\boldsymbol{y}|\boldsymbol{x})$ and $\mathrm{norm}(\frac{p^s_{te}(\boldsymbol{y}|\boldsymbol{x})p^s_{tr}(\boldsymbol{y})}{p^s_{te}(\boldsymbol{y})})$, respectively. $q_i(\boldsymbol{y}|\boldsymbol{x})\log(q_i(\boldsymbol{y}|\boldsymbol{x}))$ is typically ignored due to the absence of gradients.

Let's ignore normalization and examine $q(\boldsymbol{y}|\boldsymbol{x}) = \frac{p^t(\boldsymbol{y}|\boldsymbol{x})}{p^t(\boldsymbol{y})}$, where $p^t(\boldsymbol{y}|\boldsymbol{x}) = [p^t_1(\boldsymbol{y}|\boldsymbol{x}), p^t_2(\boldsymbol{y}|\boldsymbol{x}), \ldots, p^t_C(\boldsymbol{y}|\boldsymbol{x})], p^t(\boldsymbol{y}) = [p^t_1, p^t_2, \ldots, p^t_C]$. We can express it as

$$\begin{aligned}q(\boldsymbol{y}|\boldsymbol{x}) &= \frac{p^t(\boldsymbol{y}|\boldsymbol{x})}{p^t(\boldsymbol{y})} \\ &= \left[\frac{p^t_1(\boldsymbol{y}|\boldsymbol{x})}{p^t_1}, \frac{p^t_2(\boldsymbol{y}|\boldsymbol{x})}{p^t_2}, \ldots, \frac{p^t_C(\boldsymbol{y}|\boldsymbol{x})}{p^t_C}\right].\end{aligned} \tag{19}$$

Given the fact that the sum of all elements in the normalized distribution $q(\boldsymbol{y}|\boldsymbol{x})$ is 1:

$$\sum_{j=1}^{C} \frac{p^t_j(\boldsymbol{y}|\boldsymbol{x})}{p^t_j} = 1, \tag{20}$$

we can obtain the normalized $q(\boldsymbol{y}|\boldsymbol{x})$:

$$q(\boldsymbol{y}|\boldsymbol{x}) = \frac{q(\boldsymbol{y}|\boldsymbol{x})}{1} = \left[\frac{\frac{p^t_1(\boldsymbol{y}|\boldsymbol{x})}{p^t_1}}{\sum_{j=1}^{C}\frac{p^t_j(\boldsymbol{y}|\boldsymbol{x})}{p^t_j}}, \frac{\frac{p^t_2(\boldsymbol{y}|\boldsymbol{x})}{p^t_2}}{\sum_{j=1}^{C}\frac{p^t_j(\boldsymbol{y}|\boldsymbol{x})}{p^t_j}}, \ldots, \frac{\frac{p^t_C(\boldsymbol{y}|\boldsymbol{x})}{p^t_C}}{\sum_{j=1}^{C}\frac{p^t_i(\boldsymbol{y}|\boldsymbol{x})}{p^t_i}}\right]. \tag{21}$$

According to $p^t_1(\boldsymbol{y}|\boldsymbol{x})$ is obtained by applying the softmax function to the logits $\hat{\boldsymbol{z}}$ (The role of τ is to serve as the temperature parameter during distillation, adjusting the smoothness of the probability distribution, which we ignore in our proof):

$$p^t_j(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(\hat{z}_j)}{\sum_{k=1}^{C}\exp(\hat{z}_k)}. \tag{22}$$

We can establish the relationship between $q(\boldsymbol{y}|\boldsymbol{x})$ and $\hat{\boldsymbol{z}}$:

$$\begin{aligned}q_i(\boldsymbol{y}|\boldsymbol{x}) = \frac{\frac{p^t_i(\boldsymbol{y}|\boldsymbol{x})}{p^t_i}}{\sum_{j=1}^{C}\frac{p^t_j(\boldsymbol{y}|\boldsymbol{x})}{p^t_j}} &= \frac{1}{p^t_i}\frac{\exp(\hat{z}_i)}{\sum_{k=1}^{C}\exp(\hat{z}_k)} \cdot \sum_{j=1}^{C}\left(p^t_j \frac{\sum_{k=1}^{C}\exp(\hat{z}_k)}{\exp(\hat{z}_j)}\right) \\ &= \frac{\exp(\hat{z}_i)}{p^t_i} \cdot \sum_{j=1}^{C}\left(\frac{p^t_j}{\exp(\hat{z}_j)}\right) \\ &= \frac{\exp\left(\hat{z}_i - \log(p^t_i)\right)}{\sum_{j=1}^{C}\exp\left(\hat{z}_j - \log(p^t_j)\right)}.\end{aligned} \tag{23}$$

Similarly, it can be shown that

$$r_i(\boldsymbol{y}|\boldsymbol{x}) = \frac{\frac{p^s_{tr,i}}{p^s_{te,i}}p^s_{te}(\boldsymbol{y}|\boldsymbol{x})}{\sum_{j=1}^{C}\frac{p^s_{tr,j}}{p^s_{te,j}}p^t_j(\boldsymbol{y}|\boldsymbol{x})} = \frac{\exp(z_i + \log(p^s_{tr,i}) - \log(p^s_{te,i}))}{\sum_{j=1}^{C}\exp\left(z_j\log(p^s_{tr,j}) - \log(p^s_{te,j})\right)}. \tag{24}$$

By substituting Equation (23) and Equation (24) into Equation (18), the proof is obtained. □

B. Pseudo Code of ACCKD

Algorithm 1 Augmented Class-Conditional Knowledge Distillation (ACCKD)

```
Input: Training data $D_{tr}$, teacher model $f^t = g^t \cdot h^t$, student model $f^s = g^s \cdot h^s$, parameters $\alpha_1, \alpha_2$
for $epoch = 1$ to $T$ do
  for $batch = 1$ to $B$ do
    Draw a mini-batch $D_b$ by a random sampler from $D_{tr}$. $(\boldsymbol{x}, \boldsymbol{y})$ is a sample in $D_b$
    Draw a mini-batch $D_{b'}$ by a balanced sampler from $D_{tr}$. $(\boldsymbol{x}', \boldsymbol{y}')$ is a sample in $D_{b'}$
    Draw $\lambda$ from beta distribution
    if $\lambda < 0.5$ then
      $\lambda = 1 - \lambda$
    end if
    Random sample $\gamma$ between $(0, 1)$
    if $\gamma > 0.5$ then
      $\boldsymbol{x}_{mix} = \lambda \boldsymbol{x}' + (1 - \lambda)\boldsymbol{x}$
    else
      $\boldsymbol{x}_{mix} = \text{Mask} \cdot \boldsymbol{x}' + (1 - \text{Mask}) \cdot \boldsymbol{x}$
    end if                                  // Generated a batch of synthesis data by mixup or cutmix
    Acquiring the teacher's feature vectors $g^t(\boldsymbol{x})$ and $g^t(\boldsymbol{x}_{mix})$, as well as their logits $f^t(\boldsymbol{x})$ and $f^t(\boldsymbol{x}_{mix})$.
    Acquiring the student's feature vectors $g^s(\boldsymbol{x})$ and $g^s(\boldsymbol{x}_{mix})$, as well as their logits $f^s(\boldsymbol{x})$ and $f^s(\boldsymbol{x}_{mix})$.
    For training data $\boldsymbol{x}$, let $\boldsymbol{z} = f^s(\boldsymbol{x})$ and $\hat{\boldsymbol{z}} = f^t(\boldsymbol{x})$
    Compute $\mathcal{L}_{sup}$ by logit adjustment loss
    Compute $\mathcal{L}_{cckd}$ by by Equation (8)          // Class class-conditional knowledge distillation loss
    For synthesis data $\boldsymbol{x}_{mix}$, let $\boldsymbol{z} = f^s(\boldsymbol{x}_{mix})$ and $\hat{\boldsymbol{z}} = f^t(\boldsymbol{x}_{mix})$
    Compute $\mathcal{L}_{sup_{mix}}$ by Equation (14)
    Compute $\mathcal{L}_{cckd_{mix}}$ by Equation (12)                    // Loss on synthesis data
    Compute $\mathcal{L}_{fi}, \mathcal{L}_{fi_{mix}}$ by Equation (16) and Equation (17)    // Feature imitation loss
    $\mathcal{L}_{acckd} = \mathcal{L}_{sup} + \mathcal{L}_{sup_{mix}} + \alpha_1(\mathcal{L}_{cckd} + \mathcal{L}_{cckd_{mix}}) + \alpha_2(\mathcal{L}_{fi} + \mathcal{L}_{fi_{mix}})$
    Update student model by $\mathcal{L}_{acckd}$
  end for
end for
```

C. Implementation Detail

Following Cao et al. (2019), for the CIFAR-100-LT (Cui et al., 2019) dataset, we use ResNet-32 as the backbone network. We set the batch size to 128 and train for 200 epochs, using SGD as the optimizer. The initial learning rate of the optimizer is set to 0.1, with a momentum of 0.9 and a weight decay of 5×10^{-4}. During the 200 epochs of training, the learning rate of SGD decays by 0.01 at the 160th and 180th epochs. In the ACCKD experiments on CIFAR-100, due to the significant difference in feature dimensions between the student model and the teacher model, the feature align layer and feature imitation loss were not used.

For ImageNet-LT (Russakovsky et al., 2015) and iNaturalist 2018 (Horn et al., 2018), we use ResNet-18 as the backbone network. The batch size is set to 256, and the models are trained for 200 and 90 epochs, respectively. The optimizer is SGD, with an initial learning rate of 0.2, momentum parameter of 0.9, and weight decay coefficient of 5×10^{-4}. The learning rate is decayed using a cosine annealing schedule during the training process. In the fine-tuning of the teacher model as well as in the student model, we use the cosine classifier.

In terms of hyperparameters, we set the balancing parameters for the loss terms, α_1 and α_2, to 1 and 2, respectively. The temperature parameter for distillation, τ, is set to 2. When constructing synthetic data, the parameters for the beta distribution are set to 0.5. For comparative methods, the distillation temperature parameter τ is set to 1. In the DiVE (He et al., 2021), an additional temperature parameter of 0.5 is applied to the teacher model, following its default settings. Our code is available at https://github.com/Lain810/CCKD.

D. Supplemental Experimental Results and Analysis

To provide more information about the experimental results of ACCKD, we use models trained with only logit adjustment loss as the baseline. We then compare the accuracy improvement for each class with ACCKD. Figure 5 visualizes the results on ImageNet-LT and iNaturalist 2018, using different colors to represent many-shot, medium-shot, and few-shot classes. The figure shows that the accuracy gain from ACCKD is mainly concentrated in medium and minority classes. In the ImageNet-LT dataset, there is also a noticeable improvement in head classes.

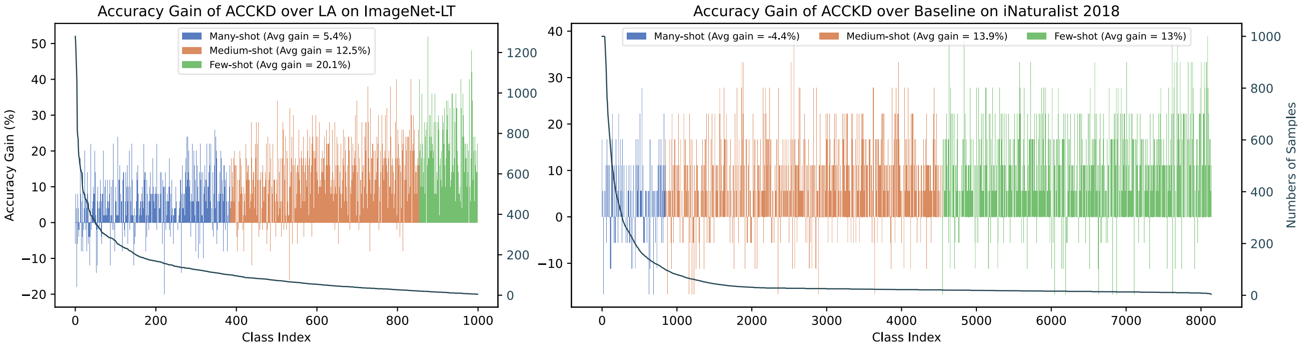


Figure 5. Accuracy gain of ACCKD over student models trained with only LA loss for each class on ImageNet-LT and iNaturalist 2018. The teacher model is fine-tuned using AF+LA. The line plot shows the number of samples for each corresponding class.

Additionally, we conduct experiments to further evaluate the quality of the features learned by different methods. We use various models to extract features from the test set, then train a linear classifier on these features and assess its performance on the test data. This result demonstrates the upper bound of the classification ability of the features extracted by the model. As shown in Figure 6, we evaluate the models trained with different methods on the ImageNet-LT dataset. Due to the strong representational capacity of pre-trained models, when CLIP is used as a feature encoder, it exhibits strong performance even without fine-tuning. Our methods outperform baseline methods across classes of varying frequencies. These results indicate that, with the help of our methods, student models learn better features compared to other methods.

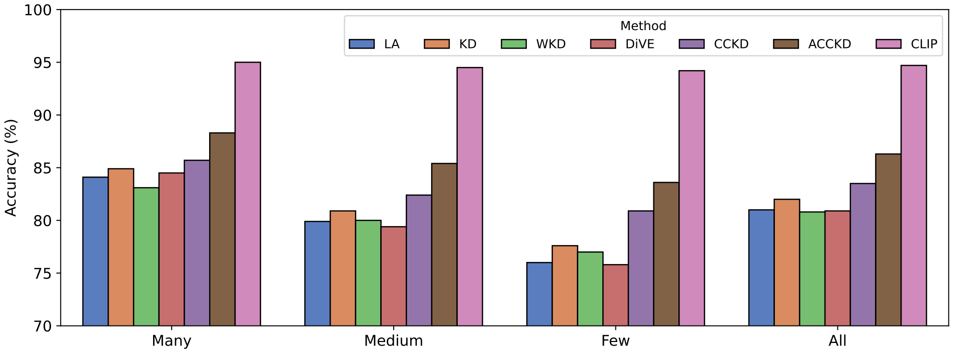


Figure 6. Performance of linear classifiers trained using features extracted from test data, which reflects the performance upper bound of the feature encoder.

E. Ablation Study on Different Sampling Strategies

To explore the role of various sampling methods in constructing synthetic data, we conduct comparative experiments on the ImageNet-LT dataset. We employ three different sampling methods to sample the two training set examples for data mixing: using two random samplers, using two balanced samplers, and using one random sampler and one balanced sampler (the method used by ACCKD). In Figure 7, we present the accuracy of student models across different classes when trained with different sampling methods under three distinct teacher models.

The results show that using two random samplers tends to favor the majority classes, leading to poor performance on minority classes. Conversely, using two balanced samplers, while improving performance on minority classes, significantly reduces performance on majority classes. Compared to these two sampling methods, our method achieves a more balanced performance across different classes, resulting in the best overall average accuracy.

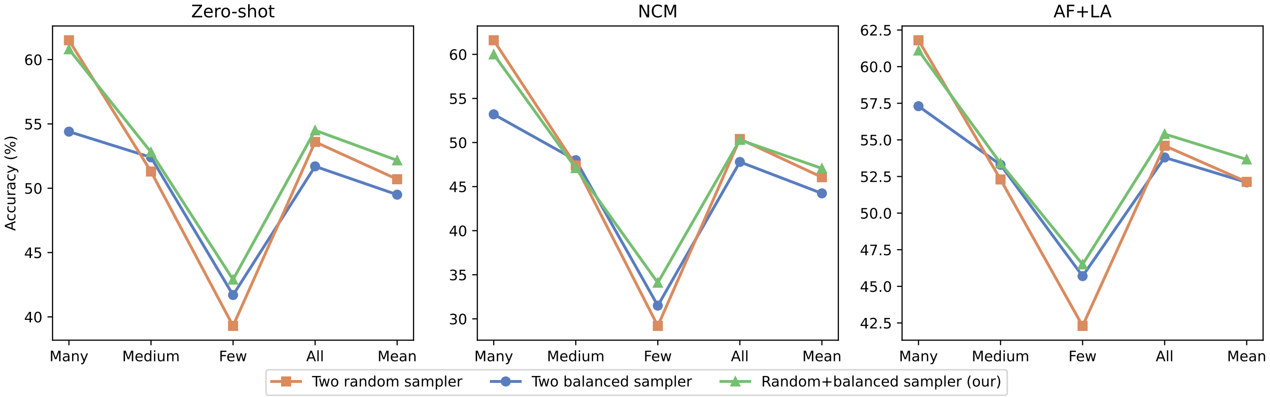


Figure 7. Comparative results of different sampling strategies for mixing samples. The "Mean" represents the average accuracy across many-shot, medium-shot, and few-shot classes.

F. Ablation Study on Temperature Parameter τ

To investigate the impact of the temperature parameter τ in ACCKD, we conducted ablation experiments on the ImageNet-LT dataset by setting τ to [1, 2, 4, 8, 10]. As shown in Figure 8, we found that τ has a minimal impact on the performance of majority class samples but significantly affects the performance of minority and medium class samples. Specifically, in most cases, as τ increases, the performance of minority and medium class samples deteriorates. Therefore, considering the overall performance across all classes in our experiments, we set $\tau = 2$.

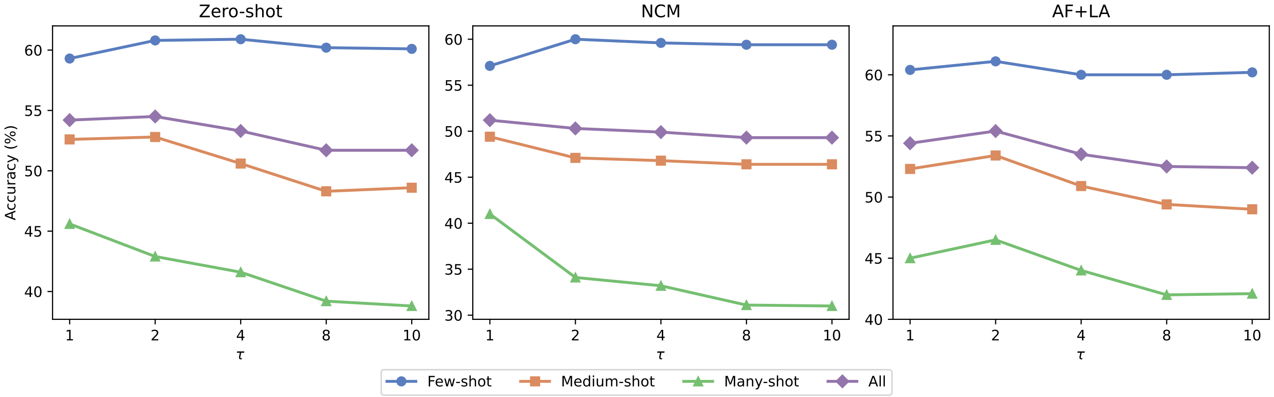


Figure 8. Comparative results of different sampling strategies for mixing samples. The "Mean" represents the average accuracy across many-shot, medium-shot, and few-shot classes.

G. Discussion on the Role of D_{mix}

To explore the impact of the constructed D_{mix}, we further compared the characteristics of the features learned by models trained on only the training set ($\mathcal{L}_{sup} + \mathcal{L}_{cckd}$), only the synthetic dataset ($\mathcal{L}_{sup_{mix}} + \mathcal{L}_{cckd_{mix}}$), and both the training set and the synthetic dataset (ACCKD). As shown in Figure 9, we present the results on ImageNet-LT using AF+LA as the teacher model. Compared to models trained solely on the synthetic dataset, those trained only on the training set can significantly reduce feature deviation in minority classes. We speculate that this is because the minority class samples in the synthetic dataset are generated from a small number of real samples, leading to a distribution shift compared to the actual distribution. However, models trained solely on the synthetic dataset can generate a richer variety of samples in the input space, resulting in better feature separation. ACCKD leverages the advantages of both approaches, achieving superior performance.

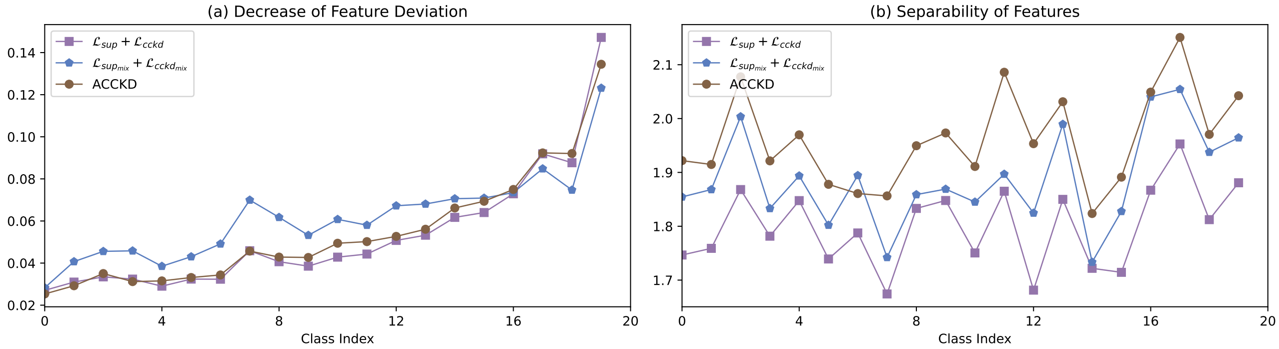


Figure 9. (a) Reduction in feature deviation across different classes. (b) Feature separability across different classes. The class index is sorted in descending order by frequency, and the displayed values are averaged over every 50 classes.

H. Adaptability to Other Class Imbalance Learning Strategies

We selected the logit adjustment loss as the supervised loss function because of its widespread use and significant effectiveness in addressing class imbalance issues. The simplicity and solid theoretical foundation of this method make it an ideal choice for our research. However, aside from logit adjustment loss, there are other methods that can effectively improve model performance on class-imbalanced data. To explore the adaptability of our method to other class imbalance learning strategies, we replaced the supervised loss with CDT (Ye et al., 2020) and LADE (Hong et al., 2021), two commonly used class imbalance learning methods, and conducted experiments on the ImageNet-LT dataset. Table 6 presents the results under different learning strategies. These results demonstrate that both CCKD and ACCKD are adaptable to other class imbalance learning methods and can effectively enhance model performance.

Table 6. Performance comparison on the ImageNet-LT when replacing LA with CDT and LADE as class imbalanced learning strategies.

		KD	WDK	DiVE	CCKD	ACCKD
CDT	Zero-Shot	50.0	48.3	44.8	51.8	54.3
	NCM	46.5	45.2	42.6	46.9	50.5
	AF+LA	51.6	51.0	45.1	52.8	55.3
LADE	Zero-Shot	45.8	49.2	48.2	52.0	54.7
	NCM	46.0	47.4	46.5	48.7	50.8
	AF+LA	46.5	49.7	48.2	53.4	55.2

I. Discussion of Three Different Teacher Models

The three methods we employ for leveraging pre-trained models—zero-shot, NCM, and AF+LA—each offer distinct advantages and are suited to different scenarios, requiring a careful balance between performance and computational cost. Notably, ACCKD consistently enhances the model's performance across various settings, with the AF+LA method yielding the best-performing teacher and student models in the majority of cases. Therefore, when computational resources are abundant, employing parameter-efficient fine-tuning methods like AF+LA is the optimal choice for leveraging the teacher model.

In contrast, zero-shot and NCM involve lower computational costs relative to AF+LA. Zero-shot predictions, which are independent of training samples, exhibit minimal class biases, particularly evident as a smaller accuracy disparity between majority and minority classes. Conversely, NCM requires estimation of class centers, which may be less accurate due to the limited number of minority class samples, leading to class biases in predictions. Thus, in datasets where zero-shot can achieve high performance, such as CIFAR-100 and ImageNet-LT, it is the preferred method for utilizing the teacher model. However, for datasets with highly fine-grained categories like iNaturalist, where zero-shot performance is markedly poor, NCM emerges as the superior alternative.