Predictive Consistency Learning for Long-Tailed Recognition

Nan Kang[1,2]
nan.kang@vipl.ict.ac.cn
Hong Chang[1,2]
changhong@ict.ac.cn
Bingpeng Ma[2]
bpma@ucas.ac.cn
Shutao Bai[1,2]
shutao.bai@vipl.ict.ac.cn
Shiguang Shan[1,2,3]
sgshan@ict.ac.cn
Xilin Chen[1,2]
xlchen@ict.ac.cn

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS
Beijing, 100190, China

[2] University of Chinese Academy of Sciences
Beijing, 100049, China

[3] CAS Center for Excellence in Brain Science and Intelligence Technology,
Shanghai, 200031, China

Abstract

Real-world data often exhibit long-tailed distributions, on which modern deep networks often make skewed predictions. Post-hoc correction approaches tackle this problem by introducing class-dependent correction biases to adjust the posterior distribution $\hat{p}_s(y|x)$, thereby compensate the discrepancy between training distribution $p_s(y)$ and test distribution $p_t(y)$. Most works along this line focus on the design of correction bias, but little attention has been paid to the estimation of $\hat{p}_s(y|x)$ which is fairly crucial for post-hoc approaches. In this paper, we highlight the inaccurate estimation of $\hat{p}_s(y|x)$ learned through cross-entropy loss minimization, which produces poorly calibrated predictions and limits the effectiveness of post-hoc correction, particularly under large label distribution shifts. To this end, we propose Predictive Consistency Learning (PCL) for long-tailed learning that learns to maintain consistency between current predictions and the aggregation of historical predictions, which iteratively refine $\hat{p}_s(y|x)$ to improve the post-hoc correction. In large-scale dataset, the storage of historical predictions requires high space complexity. To address this issue while maintaining similar performance, we further propose the compressed PCL (ComPCL) that reduces the space complexity of storing historical predictions to linear by label compression and debias operations. Experiments demonstrate that our method achieves significant improvements on several long-tailed recognition benchmarks. Code will be made available.

1 Introduction

The recent success of neural networks is largely attributed to the availability of large amounts of training data with sufficient samples per class. However, collecting sufficient samples for each class is often challenging in real-world scenarios, where the data typically follows an extreme long-tailed distribution. Unfortunately, state-of-the-art deep models tend to produce skewed predictions in long-tailed recognition and often fail to recognize tail classes.

Early long-tailed learning approaches used data re-sampling or loss re-weighting to rebalance learning for different classes. However, these approaches tend to learn overfitted representations and generalize worse than standard training [15]. To overcome this, recent works have instead adopted two-stage approaches, including decoupled training [15, 26] and post-hoc correction [10, 12, 27] by adjusting the classifier or logits in the second stage while keeping the representation of the first stage unchanged. Specifically, post-hoc correction methods introduce class-dependent correction biases to the logits, reflecting the discrepancy between the source (training) distribution $p_s(y)$ and the target (test) distribution $p_t(y)$.

While previous post-hoc correction approaches focus on the label distribution shift between $p_s(y)$ and $p_t(y)$, little attention has been paid to the predicted class posterior probabilities $\hat{p}_s(y|x)$ from long-tailed training data. In [12, 27], the correction bias is derived assuming that predicted $\hat{p}_s(y|x)$ is the true $p_s(y|x)$. However, deep neural networks often learn poorly calibrated $\hat{p}_s(y|x)$ [9], indicating a mismatch between prediction confidence and true correctness likelihood. In this work, we demonstrate that one-hot labels for cross-entropy (CE) loss encourage models to be equally confident for both head and tail classes, leading to sharp outputs and violate the calibration principle. We verify that poorly calibrated models fail to output desirable predictions via post-hoc correction under large test label distribution shift. Moreover, one-hot supervision ignores inter-class correlations, restricting knowledge transfer between classes.

To overcome these obstacles, we propose prioritizing the learning of $\hat{p}_s(y|x)$ and adapting predictions in a post-hoc manner. As illustrated above, we want hard-to-classify samples (especially tail samples) to be less confident on $\hat{p}_s(y|x)$ in order to be consistent to its correctness. We achieve this by introducing adaptive soft labels derived from the aggregation of the model's historical predictions. Specifically, we introduce *Predictive Consistency Learning* (PCL), which enforces consistency between current prediction probabilities and adaptive soft labels to mitigate miscalibration and maintain inter-class correlations. As the learning difficulty of different classes varies in long-tailed distribution, we adopt *Class-aware Weight Adjustment* (CWA) to further refine predictions for different classes. Morover, to reduce the high memory cost of storing historical predictions in PCL, we propose *Compressed PCL* (ComPCL) with *Adaptive Label Compression* (ALC), which reduces space complexity to $\mathcal{O}(N)$, where N is the size of the training set. To counteract the associated class bias from label compression, we introduce *Equivalent Class Distribution* (ECD), which replace the original class distribution $p(y)$ with the estimated $\hat{p}_{\text{ecd}}(y)$ so as to adaptively eliminate the bias with no extra cost.

We evaluate our method on several long-tailed recognition benchmarks, demonstrating significant improvements over different post-hoc correction methods and achieving state-of-the-art performance. Furthermore, our method incurs minimal computational overhead in terms of training time and introduces only an additional linear space complexity. Therefore, it can serve as a plug-and-play module to enhance various existing methods.

2 Related Work

Long-Tailed Learning. Data *resampling* and loss *reweighting* are two common approaches for mitigating class imbalance. However, they usually learn overfitted representations than standard learning [15]. There are also many other approaches proposed for long-tailed learning with different strategies including decoupled training [15, 26], meta-learning [24, 28], contrastive learning [5, 14, 16, 17, 30], knowledge distillation [11, 22] and ensemble [31, 38]. Recently, some logits manipulation approaches have been proposed, including *loss modification* [27, 28, 34] which modifies the logits during training, and *post-hoc correction* [10, 12, 23] which post-processes the model predictions during evaluation. Although implemented differently, the former is equivalent to the latter if the learning objectives are convex [27]. In practice, post-hoc correction is more flexible since it does not require retraining the model when the test distribution changes. Prior post-hoc correction approaches have primarily focused on the bias term associated with $p_s(y)$ and $p_t(y)$, while our method emphasizes learning a more calibrated $\hat{p}_s(y|x)$ to better align with the post-hoc correction theory and remedy the flaw of post-hoc correction approaches.

Confidence Calibration. In addition to classification performance, calibration is an important property for the reliability and interpretability of machine learning algorithms. It refers to the alignment between model's predictive confidence and the true correctness likelihood. [9] discussed that modern neural networks are often poorly calibrated. Many strategies have been proposed to improve calibration, including temperature scaling [36], histogram binning [35], ensemble methods [21, 32], and mixup [29, 36, 37]. In this work, we investigate the relationship between calibration and post-hoc correction of label distribution shift.

3 Method

3.1 Preliminaries

Notations. Let $\{N_1, N_2, \cdots, N_K\}$ and $\{M_1, M_2, \cdots, M_K\}$ denote the number of samples per class for the training (source) set D_s and test (target) sets D_t, respectively. Without loss of generality, we assume that $N_1 \geq N_2 \geq \cdots \geq N_K$. In long-tailed recognition, the training set is highly imbalanced with a high imbalance ratio $r = N_1/N_K$, and the tail classes have very few training samples. Early long-tailed recognition approaches assume a balanced test distribution $p_t(y)$ with $M_i = M_j$ for all i, j, while recent works [12, 38] allow for arbitrary test distributions.

Revisit of Post-hoc Correction. Under the label distribution shift between $p_s(y)$ and $p_t(y)$, previous works introduce correction biases to minimize the average classification error. These biases can be applied either through loss modification during training or post-hoc correction during evaluation. For example, [27, 28] introduce an additive bias to the training logits: $f_\theta(x)[c] + \tau \cdot \log p_s(c)$, where τ is a hyper-parameter to control the strength. The post-hoc version of the bias is given by reversing the sign of the bias term at test time:

$$\arg\max_c f_\theta(x)[c] - \tau \cdot \log p_s(c). \tag{1}$$

Assuming the class-conditional data distribution remains unchanged, *i.e.* $p_s(x|y) = p_t(x|y)$, this bias can be derived by noting that $p(y|x) \propto p(x|y) \cdot p(y)$. In theory, τ should be equal to 1. However, in practice, the optimal τ^* may be different, which reflect the degree of bias of model predictions. In this case, tuning τ to maximize accuracy is close to calibrating the logits using temperature scaling [9].

Analysis. The post-hoc correction biases reflect the discrepancy between the source distribution $p_s(y)$ and the target distribution $p_t(y)$, and can be explicitly calculated and uniquely

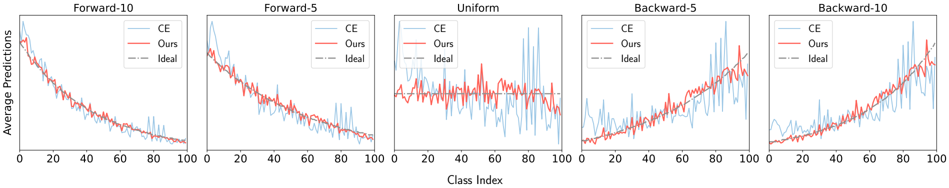


Figure 1: The average prediction $\mathbb{E}_x[\hat{p}_t(y|x;\theta)]$ of CIFAR-100-LT under different test distributions. The training distribution is Forward-100. As the distribution shift becomes large (from left to right), the average prediction no longer matches $p_t(y)$.

determined. However, they typically assume that the estimated posterior probabilities $\hat{p}_s(y|x)$ are accurate approximations of the true probabilities $p_s(y|x)$, which may not hold, particularly for tail classes.

In Fig. 1, we illustrate that poorly calibrated models fail to adapt to the target distribution by visualizing the average prediction probability $\mathbb{E}_x[\hat{p}_t(y|x;\theta)]$. From a statistical perspective, a well-calibrated model should have an average prediction probability $\mathbb{E}_x[\hat{p}_t(y|x;\theta)]$ that closely matches the label frequency $p_t(y)$. When $p_t(y)$ is imbalanced, models are expected to be more confident in sample-rich categories, which is a reasonable solution for maximizing the overall accuracy. However, in Fig. 1, model trained using CE is unable to well match the ideal $p_t(y)$ when the distribution shift becomes large.

The reasons can be summarized into the following two points. Firstly, minimizing the cross-entropy loss with one-hot labels enforces each class to be equally confident, which violates the calibration principle when dealing with long-tailed distributions. To reflect the true correctness likelihood, hard-to-classify samples, especially tail samples, should output smoother predictions on long-tailed dsitribution with more relaxed targets, rather than being uniformly pushed towards 1.0 using one-hot labels. Moreover, one-hot labels provide limited supervision for minority classes, as the inter-class correlations are ignored. This hampers the transfer of knowledge from the majority to the minority classes, which is essential for post-hoc correction, where the correction is applied in a class-wise manner as shown in Eq. (1).

3.2 Predictive Consistency Learning

The observations discussed above motivate us to estimate the model's posterior probabilities $p(y|x)$ more accurately, which would enhance the effectiveness of post-hoc correction. Ideally, we want hard-to-classify samples to produce less confident predictions and also maintain the inter-class correlations. It can be achieved by introducing a smoothing function $\mathcal{T}(x) \in \mathbb{R}^K$ for each sample and obtaining the soft target cross-entropy loss $\mathcal{L}(x,\mathcal{T})$ as:

$$\mathcal{L}(x,\mathcal{T}) = -\sum_{j=1}^{K} \mathcal{T}_j(x) \cdot \log\left[\hat{p}_s(y=j|x)\right], \tag{2}$$

where $\mathcal{T}_j(x)$ is the j-th element of $\mathcal{T}(x)$. It can be proved that for soft target cross-entropy, the optimal solution of Eq. (2) is $p_s^*(y=j|x) = \mathcal{T}_j(x)/\sum_c \mathcal{T}_c(x)$. Thus, the prediction $\hat{p}_s(y|x)$ would align with $\mathcal{T}(x)$, meaning that lower values of $\mathcal{T}_j(x)$ would result in lower predictions for that class and vice versa.

To obtain an accurate estimation of $\hat{p}_s(y|x)$, $\mathcal{T}(x)$ can be optionally formulated as the output of a teacher model or an ensemble model. However, this would bring extra training costs. Besides, the output of another model may not well capture the highly predictive uncertainty of tail classes, leading to $\mathcal{T}(x)$ being too far from the current prediction and producing large gradients during training. Motivated by [18, 20], we instead propose using the exponential moving average of historical predictions $\mathcal{H}(x)$ for $\mathcal{T}(x)$. By doing so, Eq. (2) learns to maintain consistency for each sample and constrain the prediction probability in an adaptive way.

To achieve this goal, we propose *Predictive Consistent Learning* (PCL), which optimizes the posterior prediction $\hat{p}_s(y|x)$ to enforce prediction consistency of training samples with historical predictions. PCL can be viewed as an iterative learning method of the $\hat{p}_s(y|x)$, similar to the expectation-maximization (EM) algorithm. In the E-step, it updates the expected prediction $\mathcal{T}(x)$ based on historical predictions $\mathcal{H}(x)$. In the M-step, it uses soft target cross-entropy to encourage the model's prediction to be consistent with $\mathcal{T}(x)$.

Given an input x and ground-truth label i, $\mathcal{T}(x)$ in epoch e for the j-th class is defined as:

$$\mathcal{T}_j^e(x) = (1-\alpha_{e,i}) \cdot \delta_{i,j} + \alpha_{e,i} \cdot \bar{\mathcal{H}}_j^e(x), \tag{3}$$

$$\bar{\mathcal{H}}_j^e(x) = (1-\beta) \cdot \mathcal{H}_j^{e-1}(x) + \beta \cdot \bar{\mathcal{H}}_j^{e-1}(x), \tag{4}$$

$$\mathcal{H}_j^{e-1}(x) = \hat{p}_s(y=j|x;\theta^{e-1}), \tag{5}$$

where β is the EMA factor and $\delta_{i,j}$ is the Kronecker Delta function, representing the one-hot target. $\alpha_{e,i}$ is a parameter controlling the relative strength of one-hot and $\bar{\mathcal{H}}(x)$, which will be introduced next.

During the early stages of training, the historical predictions may not be well learned and, consequently, may be less trustworthy in representing the hardness of the samples. Therefore, it is necessary to progressively increase the strength of $\bar{\mathcal{H}}_j^e(x)$. PSKD [18] propose a simple solution to set the $\alpha_e = e/E$, where E represent the total number of epochs. However, due to the lack of sufficient gradient descent updates, the predictions for tail classes may be arbitrarily random and, therefore, less trustworthy than those for head classes. To tackle this, we propose *Class-aware Weight Adjustment* (CWA) for α_e. Specifically, we introduce a class-dependent factor:

$$\alpha_{e,i} = \alpha \cdot \left(\frac{e}{E}\right)^{\lambda \cdot (1-q_i)}, \tag{6}$$

where $q_i = N_i/N_1$ is correlated with class frequencies. The hyper-parameter λ controls the exponential term. By using $1-q_i$, the exponential term is designed to be *negatively* correlated with the number of samples per class.

PCL improves $\hat{p}_s(y|x)$ by regularizing the predictions of hard samples. To illustrate this, consider a toy example of binary classification. Let us assume that the one-hot target for input x is $[1,0]$, and the predicted probability is $[p,1-p]$. According to Eq. (3), the weighted $\mathcal{T}(x)$ would be $[1-\alpha(1-p), \alpha(1-p)]$. When x is a hard sample, meaning that $1-p$ is high, $\mathcal{T}(x)$ is flatter than the one-hot target, thereby enforcing a flat prediction of x to match its true likelihood. Conversely, when x is an easy sample, meaning that $1-p$ is low, $\mathcal{T}(x)$ is close to the one-hot target, and the effect of PCL on them is much smaller.

3.3 Compressed PCL

Although effective, storing $\mathcal{H}(x)$ requires $\mathcal{O}(N \cdot K)$ additional memory cost, which can be expensive when the dataset is large. To address this issue, we propose *Compressed PCL*

(ComPCL), which compresses the storage of $\mathcal{H}$ via a compression function $\mathcal{C}$ and then performs a debias operation to alleviate the associate class bias caused by the class imbalance.

Compression. To compress the size of $\mathcal{T}(x)$, the simplest solution is to set $\beta = 0$ and store only the top-k highest prediction scores, which reduces the memory cost to $\mathcal{O}(N \cdot k)$. However, this solution cannot ultimately maintain the information of the complete $\mathcal{H}(x)$, since hard samples require a larger k to approximate the original one. Therefore, we propose *Adaptive Label Compression* (ALC) via the adaptive prediction set (APS) [1] on conformal predictions to adaptively compress $\mathcal{H}(x)$.

Denote by $o(\cdot)$ the permutation function such that $\mathcal{H}_{o(1)}(x) \geq \mathcal{H}_{o(2)}(x) \geq \cdots \geq \mathcal{H}_{o(K)}(x)$. Then, the size of the adaptive prediction set $S(x;\mathcal{H},\gamma)$ at the coverage level γ is defined as:

$$S(x;\mathcal{H},\gamma) = \min\{c \in [1,K] : \sum_{j=1}^{c} \mathcal{H}_{o(j)}(x) \geq \gamma\}. \tag{7}$$

Then, the adaptive prediction set is defined as

$$C_\gamma(x) = \{c : o(c) \leq S(x;\mathcal{H},\gamma)\}. \tag{8}$$

Based on $C_\gamma(x)$, we keep the value of $\mathcal{T}_j(x)$ if $j \in C_\gamma(x)$ and set it to zero otherwise. In practice, $\gamma = 0.95$ is sufficient to compress the average size of CIFAR-100-LT to be less than 5, while maintaining 95% prediction information.

Debias. After compression, as the model prediction more prefers majority classes, the minority classes are more likely to be excluded by C_γ. Thus, the compressed $\mathcal{H}(x)$ would exhibit more bias towards the majority, leading to a higher τ^* in Eq. (1). To tackle this issue, one option is solving the optimal transport (OT) [10] on the predictions with the Sinkhorn-Knopp algorithm [6], which is an effective debias operation when the balanced validation or test data is available [10, 38]. To avoid the computation of solving OT, following the distribution criterion of Fig. 1, a simplified alternative is to tune τ to maximizing the entropy of the average class distribution (EntMax) to make it closer to uniform distribution. It is also an effective operation that can find nearly optimal values of τ. However, both approaches require an additional balanced set for tuning and can be computationally inefficient.

To make the debias operation more efficient and eliminate the dependence of an additional balanced set, we propose to compute the *Equivalent Class Number* of the training set, which is defined as the average prediction at the last epoch: $\hat{N}_{\text{ecn}}(c) = \sum_x \hat{p}_s(y = c|x;\theta^E)$,$\forall c \in [1,K]$. Then we replace $p_s(y)$ in Eq. (1) with the *Equivalent Class Distribution* (ECD), which is defined as $\hat{p}_{\text{ecd}}(y) = \hat{N}_{\text{ecn}}(y)/\sum_c \hat{N}_{\text{ecn}}(c)$. This adaptively eliminates the bias based solely on training logits. the introduction of $\hat{p}_{\text{ecd}}$ decouples the class bias from the value of τ. Thus, for any strength of class bias, τ only needs to be set to the default value of 1. Besides, The computation of $\hat{N}_{\text{ecd}}$ can be performed online, incurring minimal computational overhead.

4 Experiments

4.1 Datasets

CIFAR-LT. CIFAR-10-LT and CIFAR-100-LT are long-tailed versions of CIFAR-10 and CIFAR-100 datasets [19] with different imbalance ratios $r = N_1/N_K$, where $r \in \{10, 50, 100\}$.

Unless otherwise specified, we use 100 as the default ratio. To obtain more reliable results, we report the average and standard deviation of 5 different runs. We use ResNet-32 as the backbone following [2], which contains 0.47M parameters.

ImageNet-LT. ImageNet-LT is a long-tailed subset of ImageNet [7] with an imbalance ratio of 256. We train the ResNet-50 from scratch with 200 epochs with a cosine classifier.

Places-LT. Places-LT is a long-tailed subset of Places [40] with an imbalance ratio of 996. Following previous works, we fine-tune an ImageNet pre-trained ResNet-152 on Places-LT.

4.2 Comparisons to Prior Methods

We compare our method with prior methods in Tab. 1 and Tab. 2. Existing methods usually adopt two kinds of settings: *weak setting*, which involves using standard data augmentation and training techniques; and *strong setting*, which employs advanced techniques like stronger data augmentation or contrastive learning. Therefore, in order to have a fair comparison, we present the table in two segments: the upper segment corresponds to the weak setting, while the lower segment corresponds to the strong setting. In the strong settings, we use AutoAugment [3] for CIFAR-10-LT and CIFAR-100-LT, and use the the sharpness-aware optimization (SAM) [8] and RandAugment [4] for ImageNet-LT and Places-LT.

(1) CIFAR-LT. The results for CIFAR-LT are presented in Tab. 1. Compared with the PC-Softmax baseline, our method shows impressive improvements for different imbalance ratios. For example, when the imbalance ratio is 100, our method achieves up to 4.4% and 3.9% improvements for CIFAR-10-LT and CIFAR-100-LT, respectively. **(2) ImageNet-LT.** The results for ImageNet-LT are presented in Tab. 2. Our method achieves 55.8% top-1 accuracy *without* any additional modifications like strong data augmentation, mixup, or ensemble, which is a superior performance compared to previous methods. In the strong setting, we further incorporate the SAM and RandAugment into our method. By increasing the training epochs to 400 following [5], we achieve top-1 accuracy of 60.0%, as shown in the table. **(3) Places-LT.** We evaluate our method on Places-LT, as presented in Tab. 2, and again achieve significant improvements over previous methods. Specifically, our methods produces more balanced predictions, where the accuracy for few-shot split is only around 5% lower than the many-shot split, showing the effective of PCL for dealing with long-tailed problem.

Table 1: Top-1 accuracy (%) on CIFAR-10-LT and CIFAR-100-LT.

Dataset	CIFAR-10-LT			CIFAR-100-LT		
Imbalance Ratio	100	50	10	100	50	10
Softmax	70.4	74.8	86.4	38.4	43.9	55.8
LDAM-DRW [2]	77.1	81.1	88.4	42.1	46.7	58.8
MiSLAS [39]	82.1	85.7	90.0	47.0	52.3	**63.2**
TSC [25]	79.7	82.9	88.7	43.8	47.4	59.0
MetaSAug [24]	80.7	84.3	89.7	48.0	52.3	61.3
PC-Softmax [12]	79.4 ±0.5	82.8 ±0.3	88.4 ±0.4	45.5 ±0.7	50.3 ±0.4	60.0 ±0.3
PCL	**83.8** ±0.42	**86.1** ±0.21	**90.1** ±0.10	**49.4** ±0.37	**54.0** ±0.20	62.9 ±0.15
BALMS [28]	81.5 ±0.0	-	91.3 ±0.1	50.8 ±0.0	-	63.0 ±0.1
PaCo [5]	-	-	-	52.0	56.0	64.2
CC-SAM [41]	83.9	86.2	-	50.8	53.9	-
DCRNets [13]	85.0	-	-	51.4	-	-
PCL + AA	**85.5** ±0.34	**87.5** ±0.21	**91.3** ±0.22	**52.1** ±0.16	**57.0** ±0.24	**65.0** ±0.20

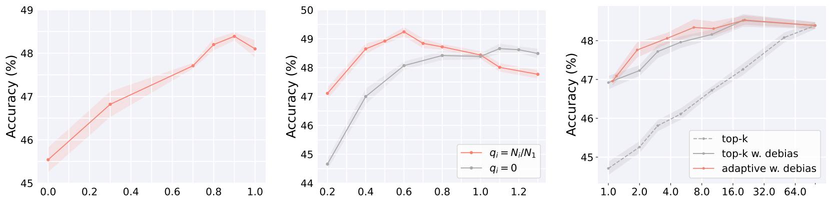
Table 2: Top-1 accuracy (%) on ImageNet-LT and Places-LT.

Dataset	ImageNet-LT				Places-LT			
Method	Many	Med.	Few	All	Many	Med.	Few	All
Softmax	64.0	33.8	5.8	41.6	**45.9**	22.4	0.4	27.2
cRT [15]	58.8	44.0	26.1	47.3	42.0	37.6	24.9	36.7
OLTM [10]	-	-	-	52.4	-	-	-	-
TSC [25]	63.5	49.7	30.4	52.4	-	-	-	-
MiSLAS [39]	61.7	51.3	35.8	52.7	39.6	43.3	36.1	40.4
PC-Softmax [12]	64.1	48.4	32.4	52.2	43.1	39.7	33.9	39.8
PCL	**66.2**	**53.0**	**36.1**	**55.8**	43.5	**42.6**	**38.0**	**42.0**
CC-SAM [41]	61.4	49.5	37.1	52.4	41.2	42.1	36.4	40.6
PaCo [5]	65.0	55.7	38.2	57.0	36.1	**47.9**	35.3	41.2
PaCo + DLSA [33]	64.6	54.9	41.8	56.9	**44.4**	44.6	32.3	42.1
PaCo + DCRNets [13]	-	-	-	58.0	-	-	-	41.7
PCL + SAM + RA	**67.3**	**58.8**	**43.5**	**60.0**	43.5	44.0	**39.9**	**43.0**

4.3 Ablation Study

The value of α. In Fig. 2(a), we investigate the impact of α by varying its value from 0 to 1 on CIFAR-100-LT. When $\alpha = 0$, Eq. (2) reduces to CE loss. As α increases, we observe a consistent improvement in performance across different values. The optimal value of α varies depending on the dataset complexity and imbalance ratio. Specifically, we use α^* is 0.9 for CIFAR-10-LT and CIFAR-100-LT, 0.7 for ImageNet-LT, and 0.4 for Places-LT.

The value of λ. In Fig. 2(b), we study the influence of λ with different computations for q_i in the CWA. When $q_i = 0$, tuning λ is equivalent to tuning a class-agnostic exponential term in Eq. (6). As shown, the class-agnostic way *cannot* achieve as high a performance improvement as the class-aware way, where $q_i = N_i/N_1$. It verifies the effectiveness of class-aware weight adjustment, which allows for composing better $\mathcal{T}(x)$ that depend on the class distribution.



(a) α (b) λ (c) Compression and Debias

Figure 2: (**a**) Ablation study of the value of α on CIFAR-100-LT. (**b**) Ablation study of the value of λ with different computations for class-aware q_i on CIFAR-100-LT. (**c**) Comparision of different compression and debias operations when varying the size of $\mathcal{T}(x)$.

Each component. In Tab. 3, we evaluate the influence of each component. As shown, each component contributes to its respective improvement on performance or efficiency. Compared with cross-entropy, PCL achieves significant improvement with all other settings being the same. By introducing the label compression and debias operation, ComPCL achieves similar performance with only much lower space complexity.

Compression and Debias. In Fig. 2(c), we examine the impact of compression and debias

Table 3: Ablation studies of the influence of each component on CIFAR-100-LT.

Method	α	CWA	ALC	ECD	Memory Cost ($\times N$)	Acc. (+PC)
CE					0	45.5
PCL	✓				100	48.4 +2.9
	✓	✓			100	49.2 +3.7
ComPCL	✓		top-5		5	46.1 +0.6
	✓		top-5	✓	5	48.2 +2.7
	✓		adaptive	✓	3.7	48.3 +2.8
	✓	✓	adaptive	✓	3.7	48.6 +3.2

operations on ComPCL with different compression size. As depicted, the ALC outperforms the *top-k* compression when the size of $\mathcal{H}(x)$ is small. Besides, we observe a significant decline in performance when compression is naively applied to $\mathcal{H}(x)$ without debias. However, by introducing the *equivalence class distribution* for debias, the performance decline caused by label compression is greatly mitigated, with only slightly lower (sometimes even better) than that of the complete PCL. This indicates that ALC and ECD are inseparable.

In Tab. 4, we further compare different debias operations. For a better comparison, in addition to the operations discussed in , we also implemented OT-Train, which involves calculating bias solely from the training set using OT, and then compose an additional debias factor for debiasing. As shown, all operations yield positive effects on ComPCL and matched performance on the complete PCL. Notably, the EntMax and OT achieve the highest accuracy, which however utilize extra data thus perform better than OT-Train. In comparison, the proposed ECD achieves similarly effective debiasing without the need for extra data and incurs minimal computational overhead.

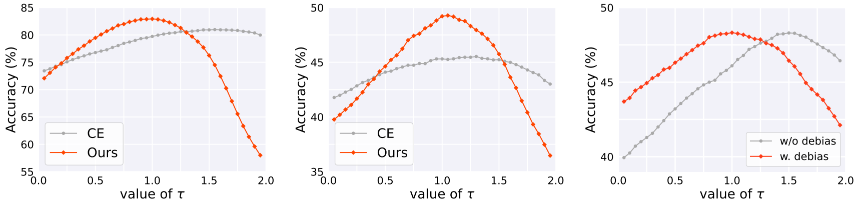
Table 4: Comparision of different debias operations.

Debias Operation	extra data	compuation cost	ComPCL	PCL
-		low	46.4	49.2
EntMax	✓	high	48.8 +2.4	49.4 +0.2
OT	✓	high	49.2 +2.8	49.4 +0.2
OT-Train		high	48.7 +2.3	49.0 -0.2
Equiv.		low	48.6 +2.2	49.2 +0.0
Oracle τ	-	-	48.9 +2.5	49.4 +0.2

4.4 Further Analysis

Optimal τ. For the PC-Softmax, $\tau = 1$ has been proved to be optimal in theory [27]. However, the optimal τ for the target distribution is often larger than 1, which indicates skewed predictions. In Fig. 3, we show that the τ^* of PCL is much closer to 1 compared to CE, which means our method learns a better $\hat{p}_s(y|x)$ that is more consistent with theory. Besides, we visualize the difference of introducing the ECD for ComPCL in Fig. 3(c). The results again indicate that relying solely on label compression can cause obvious τ value shifts, but the debias operation effectively remove the bias without re-adjusting τ.

Results on various test label distribution shifts. To further verify the effectiveness of



(a) CIFAR-10-LT (b) CIFAR-100-LT (c) ComPCL

Figure 3: (a) & (b) Accuracy on CIFAR-LT with varying τ. (c) Comparision of the debias operation for ComPCL on CIFAR-100-LT with varying τ.

Table 5: Comparison of recognition accuracy on test time shifted CIFAR-100-LT.

Dataset	Forward						Uniform	Backward					
Imbalance ratio	100	50	25	10	5	2	1	2	5	10	25	50	100
CE	66.8	64.3	61.2	56.5	52.3	**46.5**	**41.5**	**36.6**	**30.4**	**26.3**	**21.6**	**18.6**	**16.2**
PCL	**69.1**	**66.2**	**62.7**	**57.3**	**52.6**	45.4	39.8	34.3	27.3	22.6	17.5	14.2	11.6
CE + PC	66.8	63.9	60.8	56.4	53.0	48.7	45.5	43.1	40.4	39.7	39.4	39.7	40.7
PCL+ PC	**69.1**	**66.3**	**63.3**	**59.3**	**56.1**	**52.1**	**49.2**	**47.0**	**44.8**	**44.2**	**44.0**	**44.6**	**45.6**

our method under various test distributions with PC-Softmax in Tab. 5. Without the post-hoc correction, PCL actually performs slightly worse than CE, which is expected since PCL implicitly down-weights the tail classes to ensure calibration. However, we can observe that (1) before correction, PCL still performs better than CE when there is no distribution shift (Forward-100), (2) after correction, PCL consistently outperforms CE, especially under large distribution shifts (Backward regions). This indicates that PCL is beneficial on different settings.

5 Conclusion

In this work, we study the effect of $p_s(y|x)$ for post-hoc correction in long-tailed recognition. We show that $\hat{p}_s(y|x)$ learned from one-hot target cross-entropy loss fail to generalize to target distributions via post-hoc correction. To improve the learning of $\hat{p}(y|x)$, we propose Predictive Consistency Learning (PCL) to iteratively refine $\hat{p}_s(y|x)$. We also extend PCL to ComPCL that further reduces the computation cost. Our method achieves remarkable improvements on several long-tailed recognition benchmarks. We also conduct comprehensive experimental studies for further understanding our method. We hope that our method can contribute to a strong baseline and motivate more works for future research.

6 Acknowledgement

This work is partially supported by National Natural Science Foundation of China (NSFC): 61976203 and 62276246.

References

[1] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Adv. Neural Inform. Process. Syst.*, pages 1567–1578, 2019.

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 113–123, 2019.

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 702–703, 2020.

[5] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Int. Conf. Comput. Vis.*, pages 715–724, 2021.

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Adv. Neural Inform. Process. Syst.*, volume 26, 2013.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009.

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Int. Conf. Mach. Learn.*, pages 1321–1330, 2017.

[10] Dandan Guo, Zhuo Li, Meixi Zheng, He Zhao, Mingyuan Zhou, and Hongyuan Zha. Learning to re-weight examples with optimal transport for imbalanced classification. *arXiv preprint arXiv:2208.02951*, 2022.

[11] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Int. Conf. Comput. Vis.*, pages 235–244, 2021.

[12] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6626–6636, 2021.

[13] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Dual compensation residual networks for class imbalanced learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[14] Ziyu Jiang, Tianlong Chen, Bobak Mortazavi, and Zhangyang Wang. Self-damaging contrastive learning. *arXiv preprint arXiv:2106.02990*, 2021.

[15] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Int. Conf. Learn. Represent.*, 2020.

[16] Bingyi Kang, Yu Li, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *Int. Conf. Learn. Represent.*, 2021.

[17] Nan Kang, Hong Chang, Bingpeng Ma, and Shiguang Shan. A comprehensive framework for long-tailed learning via pretraining and normalization. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[18] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Int. Conf. Comput. Vis.*, pages 6567–6576, 2021.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Adv. Neural Inform. Process. Syst.*, volume 30, 2017.

[22] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6949–6958, 2022.

[23] Mengke Li, Yiu-ming Cheung, and Yang Lu. Long-tailed visual recognition via gaussian clouded logit adjustment. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6929–6938, 2022.

[24] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5212–5221, 2021.

[25] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6918–6928, 2022.

[26] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10991–11000, 2020.

[27] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[28] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020.

[29] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Adv. Neural Inform. Process. Syst.*, volume 32, 2019.

[30] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 943–952, 2021.

[31] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020.

[32] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

[33] Yue Xu, Yong-Lu Li, Jiefeng Li, and Cewu Lu. Constructing balance from imbalance for long-tailed image recognition. In *Eur. Conf. Comput. Vis.*, pages 38–56, 2022.

[34] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020.

[35] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Int. Conf. Mach. Learn.*, volume 1, pages 609–616, 2001.

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[37] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. In *Int. Conf. Mach. Learn.*, pages 26135–26160, 2022.

[38] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. In *Adv. Neural Inform. Process. Syst.*, volume 3, 2022.

[39] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16489–16498, 2021.

[40] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2017.

[41] Zhipeng Zhou, Lanqing Li, Peilin Zhao, Pheng-Ann Heng, and Wei Gong. Class-conditional sharpness-aware minimization for deep long-tailed recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3499–3509, 2023.