SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Deep Learning for Computer Vision

June 2022, Vol. 65 160105:1–160105:15
https://doi.org/10.1007/s11432-021-3489-1

Prototype-based classifier learning for long-tailed visual recognition

Xiu-Shen WEI[1,2,3], Shu-Lin XU[1,3], Hao CHEN[1], Liang XIAO[1*] & Yuxin PENG[4*]

[1]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing* 210094*, China;*
[2]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an* 710071*, China;*
[3]*Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, Nanjing* 210094*, China;*
[4]*Wangxuan Institute of Computer Technology, Peking University, Beijing* 100871*, China*

Received 14 August 2021/Revised 23 March 2022/Accepted 22 April 2022/Published online 16 May 2022

Abstract In this paper, we tackle the long-tailed visual recognition problem from the categorical prototype perspective by proposing a prototype-based classifier learning (PCL) method. Specifically, thanks to the generalization ability and robustness, categorical prototypes reveal their advantages of representing the category semantics. Coupled with their class-balance characteristic, categorical prototypes also show potential for handling data imbalance. In our PCL, we propose to generate the categorical classifiers based on the prototypes by performing a learnable mapping function. To further alleviate the impact of imbalance on classifier generation, two kinds of classifier calibration approaches are designed from both prototype-level and example-level aspects. Extensive experiments on five benchmark datasets, including the large-scale iNaturalist, Places-LT, and ImageNet-LT, justify that the proposed PCL can outperform state-of-the-arts. Furthermore, validation experiments can demonstrate the effectiveness of tailored designs in PCL for long-tailed problems.

Keywords long-tailed distribution, categorical prototype, classifier generation, classifier calibration, class imbalance

Citation Wei X-S, Xu S-L, Chen H, et al. Prototype-based classifier learning for long-tailed visual recognition. Sci China Inf Sci, 2022, 65(6): 160105, https://doi.org/10.1007/s11432-021-3489-1

1 Introduction

With the advent of research on deep convolutional neural networks (CNNs), more and more datasets that reflect real-world challenges are proposed and further accepted by the computer vision community. One of the fundamental and challenging problems is that real-world datasets always have skewed distributions with a long tail [1, 2], i.e., a few classes (also known as (a.k.a.) head classes) compose most of the data, while most classes (a.k.a. tail classes) have rarely few samples. Such long-tailed distributions can be observed in diverse computer vision tasks, e.g., fine-grained classification [3], instance segmentation [4], and object detection [5]. When dealing with these tasks, deep learning methods are not feasible to achieve outstanding recognition accuracy due to both the data-hungry limitation of deep models and the extreme class imbalance trouble of long-tailed data distributions.

In the literature, existing methods for long-tailed distributions attempt from various aspects, including class re-balancing methods [6–9], decoupled learning [10, 11], knowledge transfer [12–14], and loss margin modification [15–17]. Different from those previous studies, we investigate how categorical prototypes can help the long-tailed recognition task. The so-called prototype refers to a vector that can accurately represent the semantics of the category (i.e., genuine class representation) in the visual space, which is usually realized by the feature centroid of a specific class [18]. Our motivations hereby are two-fold. The first one is that as the categorical prototypes are statistical representations with respect to (w.r.t.) their categories, they have good generalization ability to describe the semantical meaning of categories, as well as associating with good adversarial robustness [19, 20]. The second one is that, in general, each category

* Corresponding author (email: xiaoliang@mail.njust.edu.cn, pengyuxin@pku.edu.cn)

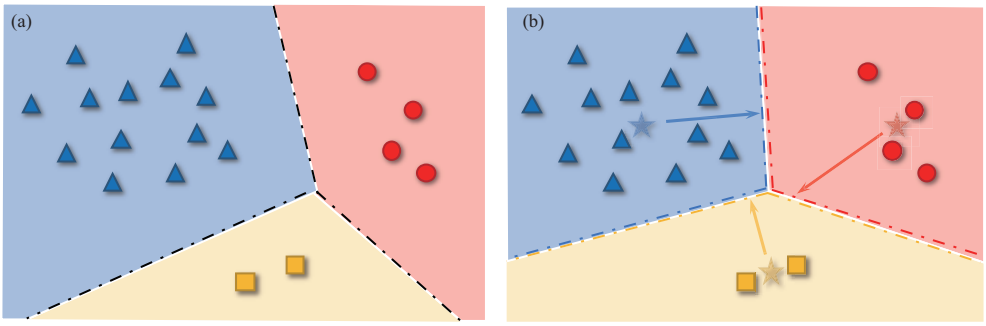

Figure 1 (Color online) Key idea of our prototype-based classifier learning method. Different marks represent examples of different classes, and the star marks are the categorical prototypes. (a) is the conventional classifier boundaries for long-tailed distribution data, where cross-entropy loss learns skewed features and results in biased classifiers. (b) is our proposal to map the categorical prototype towards the corresponding classifier. Since the prototype representations of each class have good generalization ability and more importantly they are class-balanced, the generated classifiers based on prototypes have the potential to handle long-tailed recognition, without being affected by individual examples of long-tailed distribution, especially for the head data.

has only one prototype, which is balanced across different categories. Therefore, if the class-balanced and representative categorical prototypes can be used for generating categorical classifiers, it is potential to alleviate the impact of long-tailed imbalance; see Figure 1.

Motivated by this, in this paper, we propose a prototype-based classifier learning (PCL) method for long-tailed recognition. Specifically, our PCL method consists of two main components, including (1) prototype-based classifier generation and (2) classifier calibration. As the term suggests, prototype-based classifier generation aims to generate categorical classifiers $\boldsymbol{f}_c$ of class c based on the corresponding categorical prototype $\bar{\boldsymbol{x}}_c$. In concretely, we realize this process by performing a learnable mapping function from $\bar{\boldsymbol{x}}_c$ to $\boldsymbol{f}_c$. Then, the learned classifiers $\boldsymbol{f}_c$ are employed for recognition and the parameters of our PCL model can be updated by minimizing the losses upon $\boldsymbol{f}_c$ during training. However, by recalling the under-presented issue of tail data, their prototypes might not be "accurate" as those of the head classes. Meanwhile, since $\boldsymbol{f}_c$ is generated directly based on the prototypes, the quality of representation ability of $\bar{\boldsymbol{x}}_c$ is quite crucial to some extent. Therefore, we introduce two kinds of classifier calibration approaches from both prototype-level and example-level to reduce the discrepancy of the learned $\boldsymbol{f}_c$ for alleviating data imbalance, particularly for the classifiers of tail data. More specifically, as the head classes have adequate examples, their prototypes should be relatively accurate. Thus, we calibrate the classifiers of tail classes by aggregating prototypical statistics from their similar head classes. On the other side, "representative examples" of each class (especially for tail data) also have valuable information to modify the learned classifiers, where these so-called representative examples hereby are those data either with high confidence but misclassified or with low confidence but correctly classified. After selecting representative examples, they are employed to generate a modification variable w.r.t. $\boldsymbol{f}_c$ in a meta-learning fashion. Finally, the classifiers after calibrations can be obtained and applied for long-tailed recognition in the inference phase. Experiments are conducted on five long-tailed benchmark datasets, including long-tailed CIFAR [21], iNaturalist 2018 [3], Places-LT [22], and ImageNet-LT [23]. Empirical results and ablation studies can validate the effectiveness of our proposed PCL method and our proposals in PCL.

The main contributions of this paper are as follows.

- We investigate how categorical prototypes can benefit long-tailed recognition, and propose a novel method of learning the corresponding categorical classifiers based on these prototypes.
- We develop two kinds of tailored classifier calibration approaches to modify the learned categorical classifiers derived from the prototypes for further improving the accuracy of long-tailed recognition.
- We conduct experiments on five long-tailed recognition benchmark datasets, which demonstrates superior results compared to the state-of-the-art methods, including the large-scale iNaturalist, Places-LT, and ImageNet-LT datasets.

The rest of this paper is organized as follows. Section 2 reviews the related work of long-tailed recognition and prototypes in computer vision. Section 3 elaborately presents the proposed PCL method, as well as the training algorithm. In Section 4, we report the empirical settings and experimental results for evaluating the effectiveness of our PCL. Finally, Section 5 gives the conclusion and promising future work.

2 Related work

We briefly review the previous studies in this paper from two related aspects, including long-tailed recognition and prototypes in computer vision tasks.

2.1 Long-tailed recognition

Long-tailed recognition is a fundamental research topic in machine learning, where the key is to overcome the data imbalance challenge [24–27]. With the advent of research on deep neural networks, increasing attention is being shifted to deal with long-tailed recognition by developing deep learning based methods. Broadly, existing long-tailed recognition approaches can be organized into the following paradigms.

Class re-balancing strategies. Re-balancing strategies, e.g., data re-sampling [6, 7] and loss re-weighting [8, 9], are conventional solutions for dealing with imbalance data or long-tailed distribution. Re-sampling methods as one of the most important class re-balancing strategies could be divided into two types: (1) Over-sampling by simply repeating data of minority classes [6, 28, 29] and (2) under-sampling by abandoning data of dominant classes [7, 28, 30]. Recently, OLTR [23] was proposed to firstly learn representations with instance-balanced sampling and then fine-tune these representations with class-balanced sampling with a memory module. But sometimes, with re-sampling, duplicated tailed samples might lead to over-fitting upon minority classes [31, 32], while discarding precious data will certainly impair the generalization ability of deep networks.

On the other side, re-weighting methods belong to another series of prominent class re-balancing strategies, which usually allocate large weights for training samples of tail classes in loss functions [9, 33]. However, re-weighting is not capable of handling the large-scale, real-world scenarios of long-tailed data and tends to cause optimization difficulty [34]. Consequently, Cui et al. [32] proposed to adopt the effective number of samples instead of proportional frequency. Shu et al. [35] proposed an explicit weighting function that is adaptively learned from the data. Salman et al. [36] demonstrated an uncertainty based class imbalance learning framework, and then learned robust features, as well as generalizable classifiers.

Decoupled learning. Decoupled learning is a recent trend towards effective long-tailed recognition, which aims to improve long-tailed recognition by decoupling the learning of representation and classifier. Specifically, Kang et al. [11] decoupled the deep models of long-tailed recognition into two separate stages, i.e., representation learning and classifier learning. They used the cross-entropy loss as the loss function for both of these two stages and concluded that uniform sampling could benefit representation learning and class-balance sampling favored classifier learning. Parallel to this, Zhou et al. [10] proposed a bilateral-branch network to dynamically combine uniform sampling and class-balance sampling as a unified framework. In [10], each branch performed its own duty of representation learning and classifier learning separately, and the whole model equipped with a cumulative learning strategy can first learn the universal patterns and then pay attention to the tail data gradually. Also, the work in this research line reveals that choosing proper data sampling strategies is crucial for different learning tasks in deep models, as well as a common conclusion of "better features, better models" [37].

Knowledge transfer. Due to the under-represented issue, tail data of long-tailed distribution is always shot of representative during reference. Transfer learning based approaches aim to transfer useful knowledge/information from majority classes (a.k.a. head classes) to example-scarce minority classes (a.k.a. tail classes) [12, 13, 23, 38, 39]. In particular, Liu et al. [40] proposed to estimate the head data by the Gaussian distribution and then used it to enrich the representation of tail data. Kim et al. [12] developed a major-to-minor translation method to augment tail classes via translating samples from head classes. Wang et al. [33] attempted to transfer the meta-knowledge of parameters evolution by designing a meta network. Besides, some other methods, e.g., [13, 39], defined the concepts of class-specific features and class-generic features by class activation maps [41] and mixed these two kinds of features in the training phase for augmenting the feature space of tail data.

Margin modification. Another research line of long-tailed recognition is to modify the loss during network training to encourage the model to have the optimal trade-off between per-class margins. In the previous work [29], it was revealed that the effect of re-weighting can diminish when the data is separable, which inspires to shift the classification boundary to move closer to a head (dominant) class. Specifically, Menon et al. [15] developed a large relative margin between logits of rare positive versus dominant negative labels. Cao et al. [16] proposed to integrate per-class margin into the traditional cross-entropy loss, where margins are inversely proportional to the prior probability of a class and can

enforce larger margins between a tail class and the others. While in [17], the authors tried to suppress the negative gradients resulting from head data for each tail data.

2.2 Prototypes in computer vision

In neural science, the infero-temporal cortex has a kind of prototypical representation via neurons tuned to respond to different categories [18]. Inspired by the observations, prototype based methods are developed for handling various computer vision and machine learning tasks. For instance, in traditional machine learning, nearest prototype classifier is a popular classification model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation, which is applied in diverse applications, e.g., relevance feedback [42] and classification of tumors [43]. Additionally, in few-shot learning [44], prototypical networks [45] were proposed as related to the neural statistician [46] from the generative modeling literature, which extended the variational autoencoder to learn generative models of datasets rather than individual points. In particular, the statistic network in the neural statistician summarizes a set of data points into a statistic vector to perform the rest processing of nearest neighbors. Later, deriving from this work, there were also Gaussian prototypical networks [47], which predicted a covariance radius for each prototype, yielding some insight into the discriminating force of each one of them. Beyond that, the prototypical sampling [48] on the data space, as performed by self-organizing maps, allows the representation of what is known about a data distribution, using methods quoted in prototypical networks and known as optimizing statistical criteria [49]. Compared with previous work, we investigate how the categorical prototypes as the category-level knowledge directly generate the corresponding categorical classifiers for dealing with the class imbalance issue.

3 Methodology

In this section, we elaborate the proposed PCL method in the following three aspects, i.e., prototype-based classifier generation, classifier calibration, and its overall algorithm process.

3.1 Prototype-based classifier generation

The key idea of our PCL method is to learn the categorical classifiers based on the corresponding categorical prototypes by performing a learnable mapping function, i.e., $m(\cdot)$. The generated classifiers are then employed for the final recognition. There are two advantages beneath the prototype-based learning fashion. (1) The one is that the prototypical features have good generalization ability to represent their semantical categories, as well as associating with good adversarial robustness [19,20]. (2) The other one is that prototypes of different classes are balanced, since each class has only one prototype. Therefore, categorical classifiers derived from their corresponding prototypes can not only achieve strong generalization ability for recognition, but also be beneficial to alleviate the impact of raw data imbalance.

More specifically, given the training image $\mathcal{I}$ associated with its corresponding label $y \in \{1, 2, \ldots, C\}$ (where C is the number of classes), we can use any CNN base models to get the holistic features of an image and define it as $\boldsymbol{x}$. Regarding the categorical prototype of class c, it is obtained by

$$\bar{\boldsymbol{x}}_c = \frac{1}{|\Omega_c|} \sum_{i \in \Omega_c} \boldsymbol{x}_i, \tag{1}$$

where $\Omega_c = \{j | y_j = c\}$ represents the set of indices of all examples belonging to class c. Then, the categorical classifier $\boldsymbol{f}_c$ of class c is produced via the aforementioned mapping as

$$\boldsymbol{f}_c = m(\bar{\boldsymbol{x}}_c; \phi), \tag{2}$$

where ϕ is the parameter in such a mapping function. The generated categorical classifiers $\{\boldsymbol{f}_c\}$ of all classes can be employed for the final recognition. In concretely, given a training example $\boldsymbol{x}'$ with label $y' = c$, we compute its prediction distribution via softmax as

$$p_m(y' = c | \boldsymbol{x}') = \frac{\exp(\boldsymbol{f}_c \cdot \boldsymbol{x}')}{\sum_{c'} \exp(\boldsymbol{f}_{c'} \cdot \boldsymbol{x}')}. \tag{3}$$

The model parameters are trained via minimizing the negative log-likelihood $\mathcal{J}(\boldsymbol{x}', y') = -\log(p_m(c|\boldsymbol{x}'))$.

Categories		Hornbill	House finch	Macaw	Damselfly	Pickup truck	Brain coral	Odometer
	Similarities (cf. Eq. (4))	91.875	87.625	83.938	76.875	63.625	58.688	53.406
	Exampled images							

Figure 2 (Color online) Similarity of feature means of different categories in the balanced test set from ImageNet-LT [23]. Given the category of Goldfinch (left), we calculate its prototype (i.e., feature mean) and also compute its similarity between other categories and itself.

3.2 Classifier calibration

As the classifier is learned based on the categorical prototypes, it is particularly important that these prototypes can accurately represent the corresponding categories without significant bias. For the head data of the long-tailed distribution, its prototype should be relatively "accurate". However, due to the under-represented tail data, the prototypes of these tail classes might deviate from the oracle representations, which in turn leads to poor classifier learning. Therefore, we introduce two kinds of classifier calibration approaches to reduce the discrepancy of these learned classifiers for alleviating data imbalance, especially for the tail data's classifiers.

3.2.1 *Prototype calibration*

The first calibration approach focuses on reducing the feature distribution discrepancy of prototypes, which aims to alleviate overfitting on tail classes and obtain more accurate prototypical representations (especially for tail data) for further generating classifiers with better generalization.

As a preliminary experiment shown in Figure 2, we report the similarity of feature mean (a.k.a. prototypes) on the balanced test set of ImageNet-LT [23]. It can be observed that, for a specific category, e.g., Goldfinch, similar categories have similar feature means (i.e., prototypes) w.r.t. their visual representations. Meanwhile, head classes could have more accurate feature distribution statistics thanks to their sufficient training data. Inspired by this, the prototypes of tail classes can be calibrated to reduce the feature discrepancy by transferring statistics from the prototypes of head classes.

In concretely, regarding the prototype of class c in (1), the similarity between the other head class (taking q for an example) and itself is obtained by

$$s_{c,q} = \frac{\bar{\boldsymbol{x}}_c^{\mathrm{T}} \cdot \bar{\boldsymbol{x}}_q}{\|\bar{\boldsymbol{x}}_c\| \|\bar{\boldsymbol{x}}_q\|}. \tag{4}$$

Then, a set of the top-k most similar categories w.r.t. class c can be formed as

$$\Gamma_c = \arg\operatorname*{top}_q k\{s_{c,q}\} \tag{5}$$

by ranking the similarity scores in descending order. Thus, the prototype of class c is calibrated by the statistics of its top-k similar head classes by performing

$$\bar{\boldsymbol{x}}'_c = (1-\lambda_c) \cdot \bar{\boldsymbol{x}}_c + \lambda_c \cdot \frac{\sum_{j \in \Gamma_c} \bar{\boldsymbol{x}}_j}{k}, \tag{6}$$

where λ_c is a tradeoff parameter to adjust the importance of the prototypes of class c and other similar head classes. In details, λ_c is set according to the number of examples by following the original long-tailed distribution, which is formulated as $\lambda_c = t \cdot \frac{N_{\max} - N_c}{N_{\max} - N_{\min}}$. Among the equation, t is a temperature scalar. $N_{\max}$, $N_{\min}$ and N_c represent the number of examples of the class having the maximum examples, the class having the minimum examples and class c, respectively.

In consequently, when class c is a head class, its calibrated prototype will be almost dominated by itself. While, if class c is a tail class, the prototype will be calibrated by aggregating statistics from its similar head classes. After that, following (2), the updated categorical classifier of class c is generated based on its calibrated prototype:

$$\boldsymbol{f}'_c = m(\bar{\boldsymbol{x}}'_c; \phi). \tag{7}$$

3.2.2 *Representative example calibration*

Beyond the prototype calibration, we also consider to incorporate representative examples of each class (particularly for tail data) to modify the learned classifier $\boldsymbol{f}'_c$. The so-called representative examples hereby are those data either with high confidence but misclassified or with low confidence but correctly classified. Both kinds of examples are representative and beneficial for modeling powerful categorical classification boundaries for long-tailed recognition.

Representative example selection. Specifically, we leverage the expected accuracy and average confidence of reliability diagrams [50] to realize the metrics to distinguish these representative examples. By formulations, to estimate the expected accuracy from finite examples, we group predictions into H interval bins where each of bins has a size of $1/H$. Let Λ_h denote the set of indices of examples whose prediction confidence falls into the interval $I_h = \left(\frac{h-1}{H}, \frac{h}{H}\right]$. Thus, the accuracy of Λ_h can be calculated as

$$\mathrm{acc}(\Lambda_h) = \frac{1}{|\Lambda_h|} \sum_{i \in \Lambda_h} \mathbf{1}(\hat{y}_i = y_i), \tag{8}$$

where $\hat{y}_i$ and y_i are the predicted and ground truth labels w.r.t. the inputs $\mathcal{I}_i$. Besides, the average confidence within bin Λ_h can be defined as

$$\mathrm{conf}(\Lambda_h) = \frac{1}{|\Lambda_h|} \sum_{i \in \Lambda_h} \hat{p}_i, \tag{9}$$

where $\hat{p}_i$ is the confidence for example $\boldsymbol{x}_i$. Therefore, regarding the aforementioned two kinds of representative examples within a bin Λ_h, we use

$$\{\boldsymbol{x}_i | \mathrm{conf}(\Lambda_h) - \mathrm{acc}(\Lambda_h) > \delta \ \cap \ \hat{y}_i \neq y_i \ \cap \ i \in \Lambda_h\} \tag{10}$$

to present the examples with high confidence but misclassified, while

$$\{\boldsymbol{x}_i | \mathrm{acc}(\Lambda_h) - \mathrm{conf}(\Lambda_h) > \delta \ \cap \ \hat{y}_i = y_i \ \cap \ i \in \Lambda_h\} \tag{11}$$

is used for presenting the examples with low confidence but correctly classified. By combining (10) and (11), we obtain a unified formulation to select the representative examples in Λ_h as

$$\{\boldsymbol{x}_i | |\mathrm{acc}(\Lambda_h) - \mathrm{conf}(\Lambda_h)| > \delta \ \cap \ i \in \Lambda_h\}, \tag{12}$$

where δ is a scalar as the threshold. For all H bins, based on (12), we obtain the corresponding representative examples and then separate them according to their classes, which is denoted by a set $\mathcal{X}_c^{\mathrm{repres}}$ w.r.t. class c.

Intuitive observations are shown in Figure 3. For the long-tailed CIFAR-100 dataset with different imbalance ratios, we can find that especially for the tail data, the mismatch between the expected accuracy and average confidence is quite large. In the following, we propose to learn a classifier calibration based on these representative examples in a meta-learning fashion.

Meta-learning classifier calibrations based on representative examples. The main idea of meta-learning classifier calibrations is to firstly hold out a balanced development set $\mathcal{S}_d$ from the training set $\mathcal{S}_{\mathrm{tr}}$ for obtaining representative examples $\mathcal{X}_c^{\mathrm{repres}}$, and then use $\mathcal{X}_c^{\mathrm{repres}}$ to generate a modification variable $\Delta \boldsymbol{f}_c$ to adjust the categorical classifier $\boldsymbol{f}'_c$ in (7).

More specifically, we obtain the representative examples $\mathcal{X}_c^{\mathrm{repres}}$ from $\mathcal{S}_d$ by following (12). Then, $\Delta \boldsymbol{f}_c$ is derived from another mapping function $m_{\phi_\Delta}(\cdot)$, which is as follows:

$$\Delta \boldsymbol{f}_c = m_{\phi_\Delta}(\bar{\boldsymbol{x}}_c^{\mathrm{repres}}; \phi_\Delta), \tag{13}$$

where ϕ_Δ is the parameter in $m_{\phi_\Delta}(\cdot)$, and $\bar{\boldsymbol{x}}_c^{\mathrm{repres}}$ is the feature mean of these representative examples of class c, i.e., $\bar{\boldsymbol{x}}_c^{\mathrm{repres}} = \frac{1}{|\mathcal{X}_c^{\mathrm{repres}}|} \sum_{\boldsymbol{x}_i \in \mathcal{X}_c^{\mathrm{repres}}} \boldsymbol{x}_i$. In particular, for the classes with no selected representative examples, we set $\bar{\boldsymbol{x}}_c^{\mathrm{repres}} = \mathbf{0}$, and thus $\Delta \boldsymbol{f}_c$ equals $\mathbf{0}$.

At last, the final learned categorical classifier after both prototype calibration and representative example calibration is obtained by

$$\boldsymbol{f}''_c = \boldsymbol{f}'_c + \alpha \Delta \boldsymbol{f}_c, \tag{14}$$

where α is the optimization step size. Then, the parameters of our PCL method are updated by minimizing the negative log-likelihood on the balanced development set $\mathcal{S}_d$:

$$-\log\left(\frac{\exp(\boldsymbol{f}''_c \cdot \boldsymbol{x}')}{\sum_{c'} \exp(\boldsymbol{f}''_{c'} \cdot \boldsymbol{x}')}\right), \quad \forall (\boldsymbol{x}', y') \in \mathcal{S}_d. \tag{15}$$

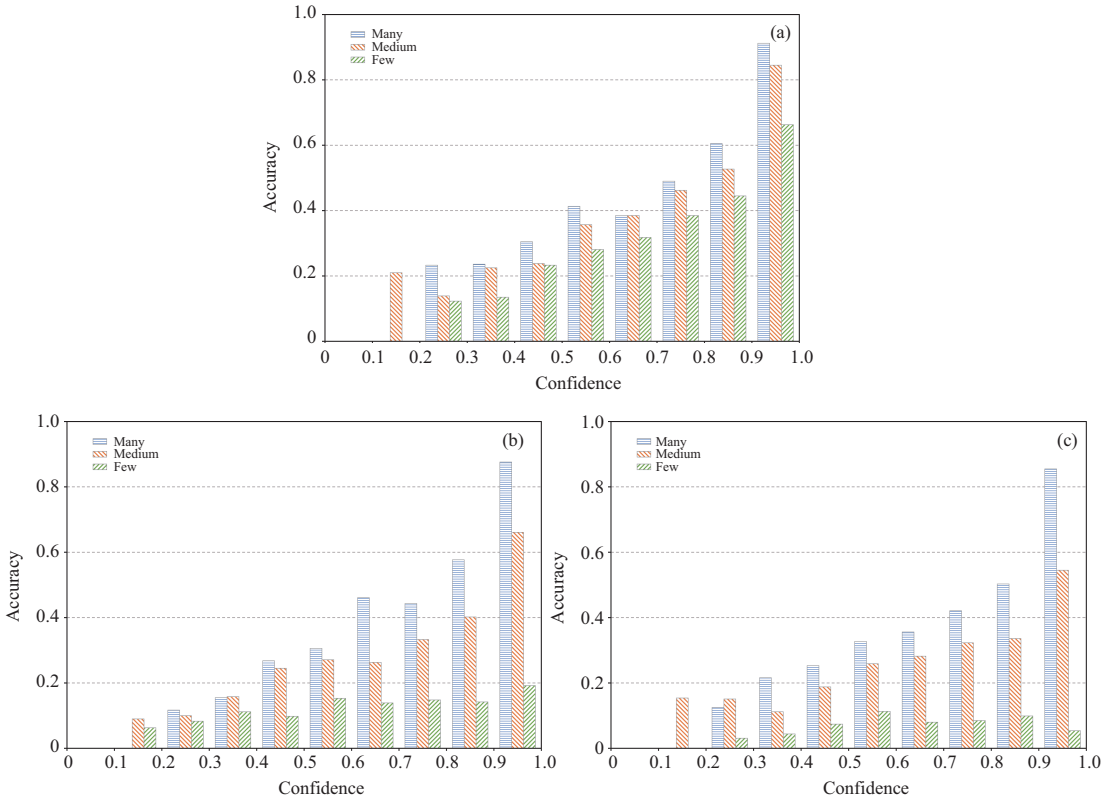

Figure 3 (Color online) Reliability diagrams for a ResNet-32 model on the long-tailed CIFAR-100 dataset. In each subfigure, "IR" represents the imbalance ratio. "Many", "Medium", and "Few" stand for the many/medium/few-shot classes [23], respectively. (a) IR = 10; (b) IR = 50; (c) IR = 100.

3.3 Overall algorithm

As aforementioned, our proposed PCL consists of two stages for learning satisfactory categorical classifiers $\{\boldsymbol{f}_c''\}$, i.e., the prototype-based classifier generation stage and the classifier calibration stage. Algorithm 1 illustrates the training process of our PCL in more details.

Algorithm 1 The proposed prototype-based classifier learning (PCL) method

Require: $\Theta_{\text{cnn}}(\cdot)$ denotes a backbone CNN model to extract deep representation $\boldsymbol{x}$ from raw image data; $\mathcal{S}_{\text{tr}} = \{(\boldsymbol{x}, y)\}$ is the training set from original long-tailed distributions; H and δ denote the width of interval bins and a threshold for selecting representative examples by reliability diagrams; α denotes the optimization step size in meta-learning classifier calibrations.

for c in $\{1, 2, \ldots, C\}$ **do**
  Compute the categorical prototype $\bar{\boldsymbol{x}}_c$ of class c by the following (1);
  Calibrate the categorical prototype as $\bar{\boldsymbol{x}}_c'$ by incorporating the statistics of head classes as presented in (6);
  Generate the categorical classifier $\boldsymbol{f}_c'$ upon the prototype calibration $\bar{\boldsymbol{x}}_c'$ by the mapping function $m_\phi(\cdot)$, cf. (7);
end for
Update the parameters of $\Theta_{\text{cnn}}(\cdot)$ and $m_\phi(\cdot)$ by minimizing $\mathcal{J}(\boldsymbol{x}', y')$ $(\forall(\boldsymbol{x}', y') \in \mathcal{S}_{\text{tr}})$ in (3) with $\boldsymbol{f}_c'$ until convergence;
while until model convergency **do**
  Randomly hold out a balanced development set $\mathcal{S}_d$ from $\mathcal{S}_{\text{tr}}$;
  Select the representative examples as $\mathcal{X}_c^{\text{repres}}$ from $\mathcal{S}_d$ by the following (12);
  Generate the classifier modification variable $\Delta\boldsymbol{f}_c$ from $\mathcal{X}_c^{\text{repres}}$ by another mapping function $m_{\phi_\Delta}(\cdot)$, cf. (13);
  Obtain the final learned categorical classifier $\boldsymbol{f}_c''$ based on both $\boldsymbol{f}_c'$ and $\Delta\boldsymbol{f}_c$, cf. (14);
  Update $\Theta_{\text{cnn}}(\cdot)$, $m_\phi(\cdot)$, and $m_{\phi_\Delta}(\cdot)$ by minimizing the negative log-likelihood on $\mathcal{S}_d$ by the following (15);
end while

4 Experiments

In this section, we first introduce the datasets and experimental settings, as well as the implementation details. Then, we report the main results on these long-tailed recognition datasets. Ablation studies are followed for further discussion.

4.1 Datasets and empirical settings

We conduct experiments on five long-tailed benchmark datasets for accuracy evaluations, including long-tailed CIFAR-10 [21], long-tailed CIFAR-100 [21], Places-LT [22], ImageNet-LT [23], and iNaturalist 2018 [3].

Long-tailed CIFAR-10 and CIFAR-100. Both CIFAR-10 and CIFAR-100 contain 60000 images, 50000 for training, and 10000 for validation with category numbers of 10 and 100, respectively. For fair comparisons, we use the long-tailed versions of CIFAR datasets as the same as those used in [16] with controllable degrees of data imbalance. We use an imbalance factor β to describe the severity of the long tail problem with the number of training samples for the most frequent class and the least frequent class, e.g., $\beta = \frac{N_{\max}}{N_{\min}}$. Imbalance factors we use in experiments are 10, 50 and 100.

iNaturalist 2018. The iNaturalist species classification datasets are large-scale real-world datasets that suffer from extremely imbalanced label distributions. The 2018 version is composed of 437513 images from 8142 categories. Note that, besides the extreme imbalance, the iNaturalist datasets also face the fine-grained problem [51–54]. In this paper, the official splits of training and validation images are utilized for fair comparisons.

Places-LT. The Places-LT dataset is also a large-scale long-tailed dataset artificially created from the balanced Places-2 [22] dataset. It contains 184.5k images from 365 diverse scene categories. The distribution of labels in the training set is also extremely long-tailed, where the sample number of each class ranges from 5 to 4980. The difference between the sample numbers of head or tail classes is larger than that of CIFAR and iNaturalist datasets.

ImageNet-LT. The ImageNet-LT dataset is a long-tailed version of the original ImageNet-2012 [55], which is constructed by sampling a subset following the Pareto distribution with the power value $\alpha = 6$. Overall, it has 115.8k images from 1000 categories, with maximally 1280 images per class and minimally 5 images per class.

4.2 Implementation details

For different benchmark datasets, we employ ResNet-32 [56], ResNet-50 [56], and ResNet-152 [56] as the backbone networks by the following previous studies [10, 23] to conduct experiments for fair comparisons. Regarding data augmentation, for long-tailed CIFAR-10 and CIFAR-100 datasets, we follow the data augmentation strategies proposed in [56]: randomly crop a 32×32 patch from the original image or its horizontal flip with 4 pixels padded on each side. For iNaturalist, we firstly resize the image by setting the shorter side to 256 pixels and then take a 224×224 crop from it or its horizontal flip. While for Places-LT, its all images are resized to 256×256 and randomly cropped to 224×224. Data augmentations include random horizontal flip with probability of 0.5 and random color jitter on brightness, contrast and saturation with jitter factor of 0.4. For ImageNet-LT, we follow the empirical setting and implementation of [57] to conduct experiments. Additionally, a supervised contrastive loss [58] is used to pre-train the backbones for better representation learning on each dataset. Regarding the hyper-parameters in our PCL method, we set k in (5) as 2, t in (6) as 0.5, H in Subsection 3.2.2 as 20, δ in (12) as 0.15, and α in (14) as 0.1, respectively. For these two mapping functions, i.e., $m_\phi(\cdot)$ and $m_{\phi_\Delta}(\cdot)$, they are realized by a three-layer and a two-layer multilayer perceptron with ELU [59] as the activation functions, respectively. The number of their hidden units per layer is 2048. All experiments are conducted on four NVIDIA V100 GPUs.

4.3 Main results

In experiments, we compare our PCL with three groups of methods.

• Baseline methods. We employ plain training with cross-entropy loss and focal loss [60] as our baselines for comparisons.

• Two-stage fine-tuning strategies. We also compare with the two-stage fine-tuning strategies proposed in previous state-of-the-art [16]. We train networks with cross-entropy (CE) on imbalanced data in the first stage, and then conduct class re-balancing training in the second stage. "CE-DRW" and "CE-DRS" refer to the two-stage baselines using re-weighting and re-sampling at the second stage.

• State-of-the-art methods. For state-of-the-art methods, we compare with the previously proposed methods which achieve good classification accuracy on these long-tailed benchmark datasets, including

Table 1 Top-1 error rates of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100[a)]

Method	Published in	Top-1 error rate (%)					
		Long-tailed CIFAR-10			Long-tailed CIFAR-100		
		IR=100	IR=50	IR=10	IR=100	IR=50	IR=10
CE	–	29.64	25.19	13.61	61.68	56.15	44.29
Focal loss [60]	ICCV 2017	29.62	23.28	13.34	61.59	55.68	44.22
CB-Focal [32]	CVPR 2019	25.43	20.73	12.90	60.40	54.83	42.01
CE-DRW [16]	NeurIPS 2019	23.66	20.03	12.44	58.49	54.71	41.88
CE-DRS [16]	NeurIPS 2019	24.39	20.19	12.62	58.39	54.52	41.89
LDAM-DRW [16]	NeurIPS 2019	22.97	18.97	11.84	57.96	53.38	41.29
CB-DA [61]	CVPR 2020	20.00	17.77	12.60	55.92	50.84	42.00
M2m [12]	CVPR 2020	20.90	–	12.50	56.50	–	42.40
BBN [10]	CVPR 2020	20.18	17.82	11.68	57.44	52.98	40.88
Causal model [62]	NeurIPS 2020	19.40	16.40	11.50	55.90	49.70	40.40
Hybrid-SC [63]	CVPR 2021	18.60	14.64	8.88	53.28	48.13	36.95
Our PCL	This paper	**17.66**	**13.32**	**8.30**	**52.33**	**47.11**	**35.87**

a) Best results are marked in bold.

Table 2 Top-1 error rates of ResNet-50 on large-scale long-tailed dataset iNaturalist 2018[a)]

Method	Published in	Top-1 error rate (%)
CE	–	42.84
CB-Focal [32]	CVPR 2019	38.88
CE-DRW [16]	NeurIPS 2019	36.27
CE-DRS [16]	NeurIPS 2019	36.44
LDAM-DRW [16]	NeurIPS 2019	32.00
CB-DA [61]	CVPR 2020	32.45
FeatAug [13]	ECCV 2020	34.09
Decoupling [11]	ICLR 2020	34.80
BBN [10]	CVPR 2020	33.71
DisAlign [57]	CVPR 2021	30.50
Hybrid-SC [63]	CVPR 2021	33.26
Our PCL	This paper	**28.88**

a) Best results are marked in bold.

CB-DA [61], M2m [12], BBN [10], FeatAug [13], Decoupling [11], Causal model [62], DisAlign [57], and Hybrid-SC [63].

The comparisons between the proposed PCL method and existing approaches on long-tailed CIFAR datasets are presented in Table 1. We conduct extensive experiments on long-tailed CIFAR datasets with three different imbalanced ratios: 10, 50, and 100. The compared methods cover various categories of ideas for imbalance classification, e.g., loss re-weighting [60], margin modification [16], transfer learning [12], data augmentation [13], decoupling of classifier and representation learning [10, 11], and causal inference [62]. As validated in Table 1, our PCL method consistently achieves the best results among these compared methods on all the settings of different imbalance ratios. In particular, compared with the recent work Hybrid-SC [63], our PCL outperforms it by about 1% under extreme imbalance settings (i.e., imbalance ratio of 100).

Table 2 shows the results on the large-scale fine-grained and long-tailed dataset, i.e., iNaturalist 2018. As shown in Table 2, when compared with other methods, our PCL still outperforms competing approaches and baselines on iNaturalist. Compared with the state-of-the-art methods, e.g., DisAlign [57] and Hybrid-SC [63], the proposed method obtains 2% and 4% improvements over these approaches, respectively. Similar observations can be found in Table 3 [64–66]. In Table 3, we report the comparison results on the Places-LT dataset for evaluating the performance of scene-centric data. Our PCL can still achieve about 2% improvements over these compared methods. Furthermore, the results on ImageNet-LT are presented in Table 4, which also shows the superiority of the proposed method.

Table 3 Top-1 error rates of ResNet-152 on the Places-LT dataset[a)]

Method	Published in	Top-1 error rate (%)
Focal loss [60]	ICCV 2017	63.40
Range loss [64]	CVPR 2017	64.90
FSLwF [65]	CVPR 2018	65.10
Lifted loss [66]	CVPR 2019	64.80
OLTR [23]	CVPR 2019	64.10
Decoupling [11]	ICLR 2020	62.10
BBN [10]	CVPR 2020	63.10
DisAlign [57]	CVPR 2021	60.70
Our PCL	This paper	**59.19**

a) Best results are marked in bold.

Table 4 Top-1 error rates of ResNet-50 on ImageNet-LT[a)]

Method	Published in	Top-1 error rate (%)
LDAM-DRW [16]	NeurIPS 2019	57.00
M2m [12]	CVPR 2020	56.30
Decoupling [11]*	ICLR 2020	50.50
Causal model [62]*	NeurIPS 2020	48.20
DisAlign [57]	CVPR 2021	47.10
Our PCL	This paper	**45.56**

a) Best results are marked in bold. * presents that the method's backbone is ResNeXt-50.

Table 5 Ablation studies on long-tailed CIFAR-10 and CIFAR-100[a)]

Method	Setting			Top-1 error rate (%)					
				Long-tailed CIFAR-10			Long-tailed CIFAR-100		
	Subsection 3.1	Subsection 3.2.1	Subsection 3.2.2	IR=100	IR=50	IR=10	IR=100	IR=50	IR=10
Vanilla backbone	–	–	–	29.64	25.19	13.61	61.68	56.15	44.29
BBN [10]	–	–	–	20.18	17.82	11.68	57.44	52.98	40.88
Our PCL (♯1)	✓			27.81	23.15	11.65	61.05	55.70	42.43
Our PCL (♯2)	✓	✓		20.27	17.75	10.84	58.35	52.47	39.90
Our PCL (♯3)			✓	29.01	24.23	12.59	61.31	55.98	43.78
Our PCL (♯4)	✓	✓	✓[b)]	18.37	15.88	9.04	56.60	49.83	37.27
Our PCL (♯5)	✓	✓	✓[c)]	20.57	17.41	10.78	58.62	52.87	39.48
Our PCL (♯6)	✓	✓	✓	**17.66**	**13.32**	**8.30**	**52.33**	**47.11**	**35.87**

a) Best results are marked in bold.
b) Performing representative example calibration by only using data with high confidence but misclassified.
c) Performing representative example calibration by only using data with low confidence but correctly classified.

4.4 Ablation studies and discussion

In this subsection, we conduct ablation studies on long-tailed CIFAR to characterize the proposed PCL method, especially for its three main components and these two kinds of classifier calibration.

Effects of components of PCL. We firstly investigate the effects of main components of our PCL, i.e., prototype-based classifier generation (Subsection 3.1), prototype calibration (Subsection 3.2.1), and representative example calibration (Subsection 3.2.2). In Table 5, we report the results by performing various empirical settings, i.e., ♯1 to ♯6. Specifically, ♯1, ♯2, and ♯6 are the results by stacking these three components in PCL one by one. It is observed that the recognition accuracies on long-tailed CIFAR are steadily improved. The observations justify the effectiveness of our proposed components in PCL. Moreover, we also compare the results with the vanilla backbone and BBN [10] as baselines for in-depth discussions. When only equipped with the prototype-based classifier generation, our PCL even obtains (slightly) better accuracy than the vanilla backbone. It shows our proposal of prototype-based classifier mapping is effective, i.e., categorical classifiers derived from the corresponding prototypes not only achieve strong generalization recognition power, but also can alleviate the impact of raw data imbalance. Besides, our designs of classifier calibration (i.e., prototype calibration and representative example calibration) also work well. With only prototype calibration, our method can achieve comparable accuracy with BBN. By further combining representative example calibration, our PCL gets new state-of-

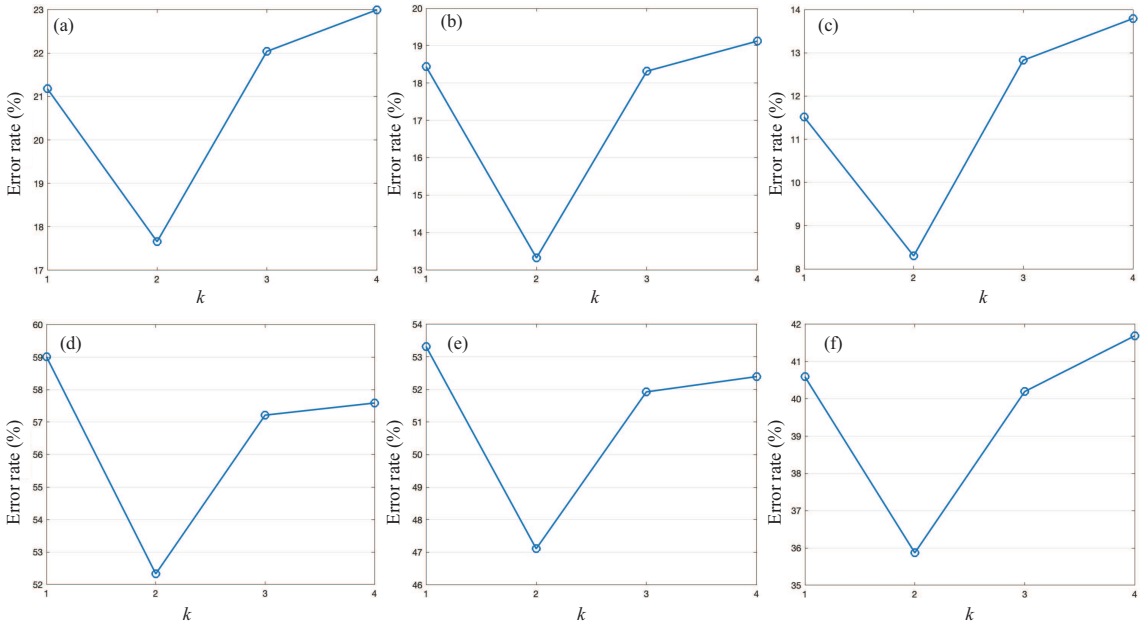

Figure 4 (Color online) Error rate comparisons with different numbers of prototype calibration, i.e., k in (6). The lower, the better. (a) Long-tailed CIFAR-10 (IR = 100); (b) long-tailed CIFAR-10 (IR = 50); (c) long-tailed CIFAR-10 (IR = 10); (d) long-tailed CIFAR-100 (IR = 100); (e) long-tailed CIFAR-100 (IR = 50); (f) long-tailed CIFAR-100 (IR = 10).

the-art results. Furthermore, for in-depth analyses about the representative example calibration of PCL, we also perform ♯3, ♯4, and ♯5. ♯3 only uses the prototypes of representative examples to generate the final categorical classifiers, which do not have the base categorical classifiers and cause data bias. It is no surprise that ♯3 achieves unsatisfactory performance. For ♯4 and ♯5, as shown in Figure 3, there are more examples with high confidence but misclassified than examples with low confidence but correctly classified in long-tailed distributions. Thus, ♯4 obtains better recognition results than ♯5. Besides, in these examples with high confidence but misclassified, the tail data is mostly, which allows ♯4 to achieve good results.

Effects of different numbers of prototype calibration. To explore the effects of different numbers of prototype calibration, i.e., k in (6), on long-tailed recognition accuracy, we change the values of k in a set of $\{1, 2, 3, 4\}$, as depicted in Figure 4. Generally, this figure shows that the prototype calibration should be conducted within a small number of similar categories. Otherwise, it might be overly affected by irrelative categorical prototypes. In experiments, we choose the optimal value of k by performing a validation set. We can see that when $k = 2$, it can achieve the best recognition accuracy on long-tailed CIFAR. If k is small, prototype calibration will not play a significant role; while if k is large (e.g., larger than 2), side effects will gradually appear.

Effects of different α for representative example calibration. We vary the values of the tradeoff parameter α in (14) for representative example calibration, and show the results in Figure 5. As presented, the optimal value of α is 0.2. Particularly, when $\alpha = 0$, it is equivalent to no example-level calibration. Additionally, as the value of α increasing, the final classifiers are gradually affected by more and more representative example calibration, resulting in bias or discrepancy. Especially when $\alpha = 1$, the impact is the most serious.

Effects of two mapping functions. There are two mapping functions in the PCL method, i.e., $m_\phi(\cdot)$ and $m_{\phi_\Delta}(\cdot)$. By considering the computational efficiency, it is desirable to investigate: Would it be possible to just learn a single mapping function from the categorical prototypes and the prototypes of representative examples to the classifier parameters? Therefore, we perform such a single mapping function in PCL and report the results in Figure 6. As shown, it is apparent to observe that using a single mapping function gets worse results than the results of our proposal. The reason could be that the inputs and outputs of $m_\phi(\cdot)$ and $m_{\phi_\Delta}(\cdot)$ are all different; thus, we should have two mapping functions to sufficiently model these two different processes. In concretely, for $m_\phi(\cdot)$, its inputs and outputs are the categorical prototypes and the categorical (base) classifiers, respectively (cf. (7)). While, for $m_{\phi_\Delta}(\cdot)$, its inputs and outputs are representative examples and a modification variable of the base classifier,

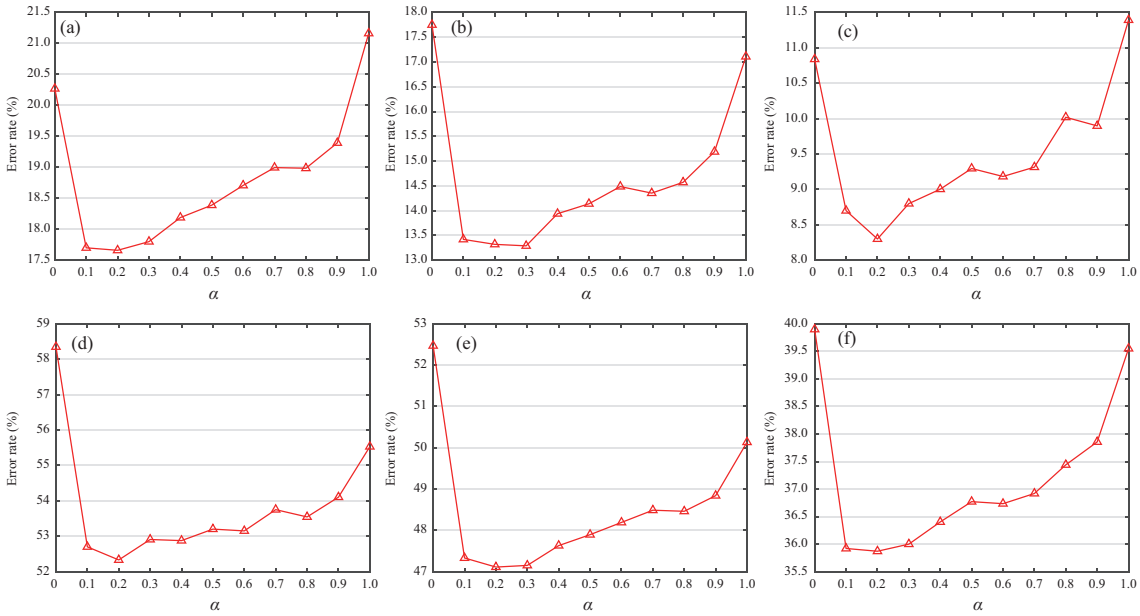

Figure 5 (Color online) Error rate comparisons with different values of α in (14). The lower, the better. (a) Long-tailed CIFAR-10 (IR = 100); (b) long-tailed CIFAR-10 (IR = 50); (c) long-tailed CIFAR-10 (IR = 10); (d) long-tailed CIFAR-100 (IR = 100); (e) long-tailed CIFAR-100 (IR = 50); (f) long-tailed CIFAR-100 (IR = 10).

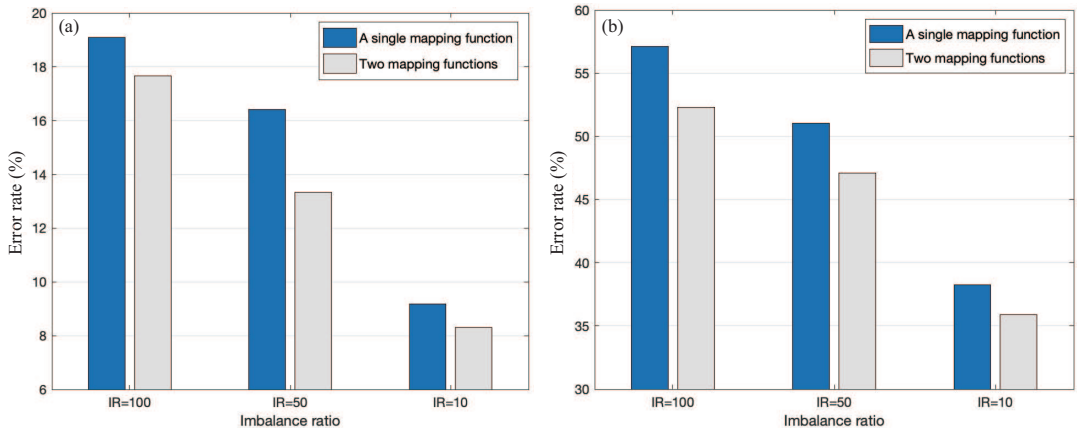

Figure 6 (Color online) Error rate comparisons with a single mapping function vs. two mapping functions in PCL. The lower, the better. (a) Long-tailed CIFAR-10; (b) long-tailed CIFAR-100.

cf. (13).

Effects of the number of layers of mapping functions. In this subsection, we investigate how the number of layers of mapping functions, i.e., $m_\phi(\cdot)$ and $m_{\phi_\Delta}(\cdot)$, affects the final results. In concretely, we fix $m_{\phi_\Delta}(\cdot)$ and change the number of layers of $m_\phi(\cdot)$ from 1 to 4; and then we fix $m_\phi(\cdot)$ and change the number of layers of $m_{\phi_\Delta}(\cdot)$ from 1 to 3. On long-tailed CIFAR-10 and CIFAR-100, the results of this ablation study are presented in Figures 7 and 8. It is clear to see that when the number of layers equals 1 (which means it is a linear mapping), its error rates are high. While, if the number of the layer increases, error rates firstly significantly decrease and then reach the optimal, and later there is a slight increase (perhaps due to overfitting).

5 Conclusion

In this paper, we proposed a PCL method for dealing with long-tailed recognition. Specifically, motivated by the advantages of categorical prototypes, we developed a learnable mapping function upon these prototypes for generating the corresponding categorical classifiers. Then, in order to explicitly alleviate

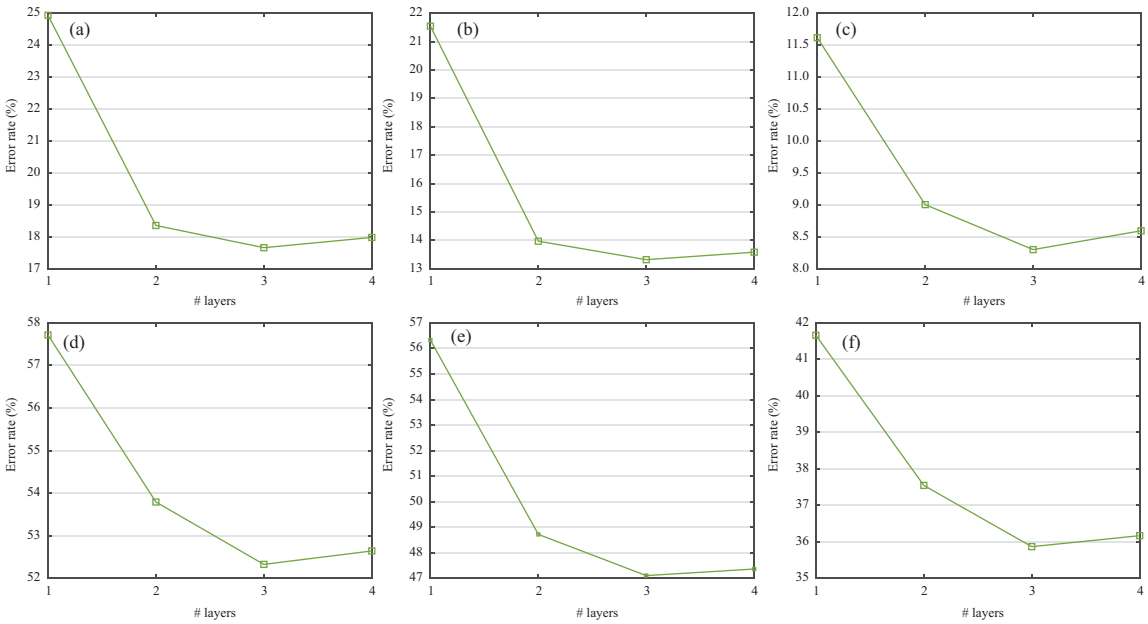

Figure 7 (Color online) Error rate comparisons with different number of layers of $m_{\phi}(\cdot)$ when fixing $m_{\phi_{\Delta}}(\cdot)$. The lower, the better. (a) Long-tailed CIFAR-10 (IR = 100); (b) long-tailed CIFAR-10 (IR = 50); (c) long-tailed CIFAR-10 (IR = 10); (d) long-tailed CIFAR-100 (IR = 100); (e) long-tailed CIFAR-100 (IR = 50); (f) long-tailed CIFAR-100 (IR = 10).

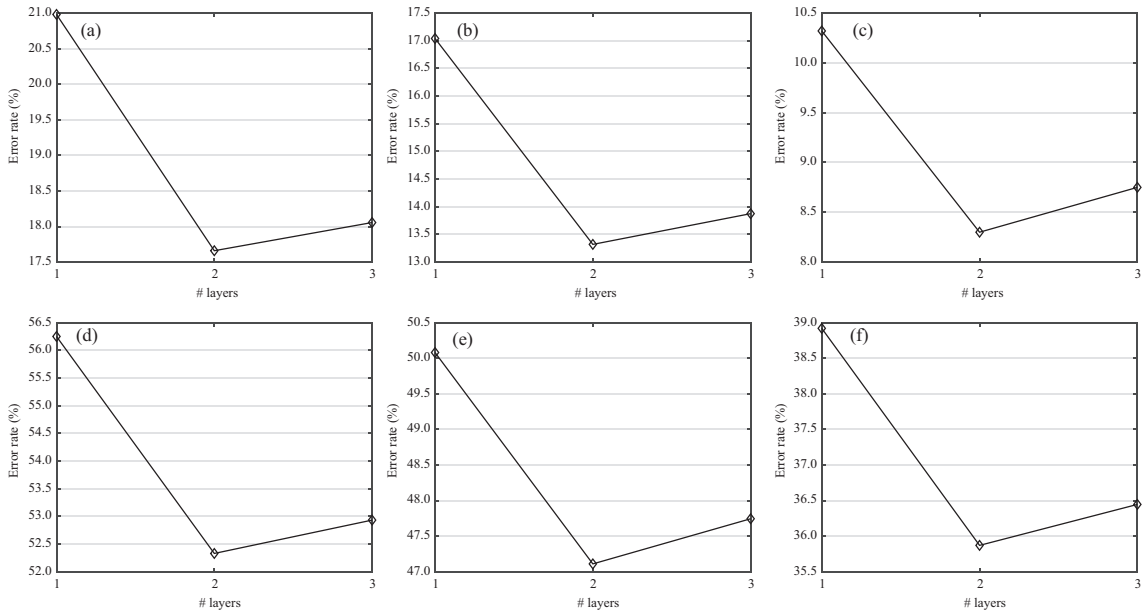

Figure 8 Error rate comparisons with different number of layers of $m_{\phi_{\Delta}}(\cdot)$ when fixing $m_{\phi}(\cdot)$. The lower, the better. (a) Long-tailed CIFAR-10 (IR = 100); (b) long-tailed CIFAR-10 (IR = 50); (c) long-tailed CIFAR-10 (IR = 10); (d) long-tailed CIFAR-100 (IR = 100); (e) long-tailed CIFAR-100 (IR = 50); (f) long-tailed CIFAR-100 (IR = 10).

the imbalance issue, two kinds of classifier calibration approaches, i.e., prototype calibration and representative example calibration, were designed for modifying the generated classifiers in the previous stage. After calibration, the classifiers were employed for the final recognition and the losses can be utilized for training the PCL model in an end-to-end manner. By conducting extensive experiments, we proved that our PCL could achieve the best results on long-tailed benchmarks, including the large-scale datasets of iNaturalist, Places-LT, and ImageNet-LT. In the future, we attempt to tackle the long-tailed detection problem with our PCL method.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2021YFA1001100), National Natural Science Foundation of China (Grant Nos. 61925201, 62132001, U21B2025, 61871226), Natural Science Foundation of Jiangsu Province of China (Grant No. BK20210340), Fundamental Research Funds for the Central Universities (Grant No.

30920041111), CAAI-Huawei MindSpore Open Fund, and Beijing Academy of Artificial Intelligence (BAAI). We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks), and Ascend AI Processor used for this research.

References

1 Kendall M G, Stuart A, Ord J K, et al. Kendall's Advanced Theory of Statistics. Volume 1. Distribution Theory. 5th ed. New York: Oxford University Press, 1987
2 van Horn G, Perona P. The devil is in the tails: fine-grained classification in the wild. 2017. ArXiv:1709.01450
3 van Horn G, Mac Aodha O, Song Y, et al. The iNaturalist species classification and detection dataset. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 8769–8778
4 Gupta A, Dollár P, Girshick R. LVIS: a dataset for large vocabulary instance segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 5356–5364
5 Wei X S, Cui Q, Yang L, et al. RPC: a large-scale retail product checkout dataset. 2019. ArXiv:1901.07249
6 Shen L, Lin Z, Huang Q. Relay backpropagation for effective learning of deep convolutional neural networks. In: Proceedings of European Conference on Computer Vision, 2016. 467–482
7 Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intell Data Anal, 2002, 6: 429–449
8 Liu X Y, Zhou Z H. The influence of class imbalance on cost-sensitive learning: an empirical study. In: Proceedings of IEEE International Conference on Data Mining, 2006. 970–974
9 Huang C, Li Y, Loy C C, et al. Learning deep representation for imbalanced classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 5375–5384
10 Zhou B, Cui Q, Wei X S, et al. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 9719–9728
11 Kang B, Xie S, Rohrbach M, et al. Decoupling representation and classifier for long-tailed recognition. In: Proceedings of International Conference on Learning Representations, 2020. 1–16
12 Kim J, Jeong J, Shin J. M2m: imbalanced classification via major-to-minor translation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 13896–13905
13 Chu P, Bian X, Liu S, et al. Feature space augmentation for long-tailed data. In: Proceedings of European Conference on Computer Vision, 2020. 694–710
14 He Y Y, Wu J, Wei X S. Distilling virtual examples for long-tailed recognition. In: Proceedings of IEEE International Conference on Computer Vision, 2021. 235–244
15 Menon A K, Jayasumana S, Rawat A S, et al. Long-tail learning via logit adjustment. In: Proceedings of International Conference on Learning Representations, 2020. 1–13
16 Cao K, Wei C, Gaidon A, et al. Learning imbalanced datasets with label-distribution-aware margin loss. In: Proceedings of International Conference on Neural Information Processing Systems, 2019. 1–18
17 Tan J, Wang C, Li B, et al. Equalization loss for long-tailed object recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 11662–11671
18 Viéville T, Crahay S. Using an Hebbian learning rule for multi-class SVM classifiers. J Comput Neurosci, 2004, 17: 271–287
19 Yang H M, Zhang X Y, Yin F, et al. Robust classification with convolutional prototype learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3474–3482
20 Xiao M, Kortylewski A, Wu R, et al. TDMPNet: prototype network with recurrent top-down modulation for robust object classification under partial occlusion. In: Proceedings of European Conference on Computer Vision, 2020. 447–463
21 Krizhevsky A, Hinton G. Learning Multiple Layers of Features From Tiny Images. Technical Report, TR-2009-1618. 2009
22 Zhou B, Lapedriza A, Khosla A, et al. Places: a 10 million image database for scene recognition. IEEE Trans Pattern Anal Mach Intell, 2017, 40: 1452–1464
23 Liu Z, Miao Z, Zhan X, et al. Large-scale long-tailed recognition in an open world. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2537–2546
24 Zhou Z-H, Liu X-Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng, 2006, 18: 63–77
25 Liu X-Y, Wu J X, Zhou Z-H. Exploratory undersampling for class-imbalance learning. IEEE Trans Syst Man Cybern B, 2009, 39: 539–550
26 Zhang M L, Li Y K, Liu X Y. Towards class-imbalance aware multi-label learning. In: Proceedings of International Joint Conference on Artificial Intelligence, 2017. 4041–4047
27 Zhang J, Liu L, Wang P, et al. Exploring the auxiliary learning for long-tailed visual recognition. Neurocomputing, 2021, 449: 303–314
28 Buda M, Maki A, Mazurowski M A. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 2018, 106: 249–259
29 Byrd J, Lipton Z. What is the effect of importance weighting in deep learning? In: Proceedings of International Conference on Machine Learning, 2019. 872–881
30 He H B, Garcia E A. Learning from imbalanced data. IEEE Trans Knowl Data Eng, 2009, 21: 1263–1284
31 Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res, 2002, 16: 321–357
32 Cui Y, Jia M, Lin T Y, et al. Class-balanced loss based on effective number of samples. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 9268–9277
33 Wang Y X, Ramanan D, Hebert M. Learning to model the tail. In: Proceedings of International Conference on Neural Information Processing Systems, 2017. 7029–7039
34 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of International Conference on Neural Information Processing Systems, 2013. 3111–3119
35 Shu J, Xie Q, Yi L, et al. Meta-weight-net: learning an explicit mapping for sample weighting. In: Proceedings of International Conference on Neural Information Processing Systems, 2019. 1917–1928
36 Salman K, Munawar H, Waqas Z S, et al. Striking the right balance with uncertainty. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 103–112
37 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
38 Zhu L, Yang Y. Inflated episodic memory with region self-attention for long-tailed visual recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 4343–4352

39 Zhang Y, Wei X S, Zhou B, et al. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In: Proceedings of American Association for Artificial Intelligence, 2021. 3447–3455
40 Liu J, Sun Y, Han C, et al. Deep representation learning on long-tailed data: a learnable embedding augmentation perspective. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 2970–2979
41 Zhou B, Khosla A, lapedriza A, et al. Learning deep features for discriminative localization. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2921–2929
42 Manning C, Raghavan P, Schutze H. Vector space classification. Cambridge: Cambridge University Press, 2008
43 Tibshirani R, Hastie T. Diagnosis of multiple cancer types by shrunken centroids of gene expression. In: Proceedings of the National Academy of Sciences, 2002. 6567–6572
44 Wang P, Liu L, Shen C, et al. Multi-attention network for one shot learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2721–2729
45 Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. In: Proceedings of International Conference on Neural Information Processing Systems, 2017. 1–11
46 Edwards H, Storkey A. Towards a neural statistician. In: Proceedings of International Conference on Learning Representations, 2017. 1–14
47 Fort S. Gaussian prototypical networks for few-shot learning on Omniglot. 2017. ArXiv:1708.02735
48 Hecht T, Gepperth A. Computational Advantages of Deep Prototype-Based Learning. Technical Report, hal-01418135. 2016
49 Banerjee A, Merugu S, Dhillon I S, et al. Clustering with Bregman divergences. J Mach Learn Res, 2005, 61: 1705–1749
50 Niculescu-Mizil A, Caruana R. Predicting good probabilities with supervised learning. In: Proceedings of International Conference on Machine Learning, 2005. 625–632
51 Wei X S, Song Y Z, Aodha O M, et al. Fine-grained image analysis with deep learning: a survey. IEEE Trans Pattern Anal Mach Intell, 2021. doi: 10.1109/TPAMI.2021.3126648
52 Wei X S, Luo J H, Wu J, et al. Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Trans Image Process, 2017, 26: 2868–2881
53 Wei X S, Wang P, Liu L, et al. Piecewise classifier mappings: learning fine-grained learners for novel categories with few examples. IEEE Trans Image Process, 2019, 28: 6116–6125
54 Wei X S, Shen Y, Sun X, et al. A^2-Net: learning attribute-aware hash codes for large-scale fine-grained image retrieval. In: Proceedings of International Conference on Neural Information Processing Systems, 2021. 5720–5730
55 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
56 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
57 Zhang S, Li Z, Yan S, et al. Distribution alignment: a unified framework for long-tail visual recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2021. 2361–2370
58 Khosla P, Teterwak P, Wang C, et al. Supervised contrastive learning. In: Proceedings of International Conference on Neural Information Processing Systems, 2020. 18661–18673
59 Clevert D A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). In: Proceedings of International Conference on Learning Representations, 2015. 1–14
60 Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. In: Proceedings of IEEE International Conference on Computer Vision, 2017. 2980–2988
61 Jamal M A, Brown M, Yang M H, et al. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 7610–7619
62 Tang K, Huang J, Zhang H. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In: Proceedings of International Conference on Neural Information Processing Systems, 2020. 1513–1524
63 Wang P, Han K, Wei X S, et al. Contrastive learning based hybrid networks for long-tailed image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2021. 943–952
64 Zhang X, Fang Z, Wen Y, et al. Range loss for deep face recognition with long-tailed training data. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5409–5418
65 Gidaris S, Komodakis N. Dynamic few-shot visual learning without forgetting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 4367–4375
66 Liu Z, Miao Z, Zhan X, et al. Deep metric learning via lifted structured feature embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2537–2546