Adjusting Logit in Gaussian Form for Long-Tailed Visual Recognition

Mengke Li, *Member, IEEE*, Yiu-ming Cheung, *Fellow, IEEE*, Yang Lu, *Member, IEEE*, Zhikai Hu, *Student Member, IEEE*, Weichao Lan, *Student Member, IEEE*, and Hui Huang, *Senior Member, IEEE*

***Abstract*—It is not uncommon that real-world data are distributed with a long tail. For such data, the learning of deep neural networks becomes challenging because it is hard to classify tail classes correctly. In the literature, several existing methods have addressed this problem by reducing classifier bias, provided that the features obtained with long-tailed data are representative enough. However, we find that training directly on long-tailed data leads to uneven embedding space. That is, the embedding space of head classes severely compresses that of tail classes, which is not conducive to subsequent classifier learning. This article therefore studies the problem of long-tailed visual recognition from the perspective of feature level. We introduce feature augmentation to balance the embedding distribution. The features of different classes are perturbed with varying amplitudes in Gaussian form. Based on these perturbed features, two novel logit adjustment methods are proposed to improve model performance at a modest computational overhead. Subsequently, the distorted embedding spaces of all classes can be calibrated. In such balanced-distributed embedding spaces, the biased classifier can be eliminated by simply retraining the classifier with class-balanced sampling data. Extensive experiments conducted on benchmark datasets demonstrate the superior performance of the proposed method over the state-of-the-art ones.**

***Impact Statement*—Long-tailed visual recognition, a burgeoning field within computer vision, holds profound significance in academic discourse. It fosters advancements in real-world applications by addressing challenges posed by imbalanced datasets, thereby facilitating improved model generalization. In this article, we propose a simple yet effective logit adjustment method, applicable across different models. Our work provides comprehensive discussions of the proposed method for long-tail learning, considering aspects of optimization and geometric interpretation. These discussions contribute to a deeper understanding of long-tail learning and a novel approach for enhancing generalization on the test set. In scholarly pursuits, long-tailed visual recognition underscores the necessity for nuanced and inclusive methodologies, which are pivotal in advancing the frontiers of research in computer vision and artificial intelligence.**

***Index Terms*—Gaussian clouded logit (GCL), imbalance learning, long-tailed classification, logit adjustment.**

I. Introduction

DEEP learning methods have achieved better-than-human performance on a variety of visual recognition tasks [1], [2], [3] by virtue of the large-scale annotated datasets. In general, the success of deep neural networks (DNNs) relies on balanced-distributed data and sufficient training samples. That is, the number of samples in each class is basically the same and large enough. Unfortunately, from the practical perspective, data collected from the real world would follow a power-law distribution [4], [5], which means that a tiny number of head classes occupy large volumes of instances while the vast majority of tail classes each have fairly few samples, showing a "long tail" in the data distribution. In fact, class importance is independent of the number of training samples. In other words, few samples cannot imply the unimportance of the tail classes [6]. Even more, misclassification of tail classes can have severe consequences, especially in critical applications such as medical diagnosis [7] or road monitoring [8]. Therefore, it is important to develop methods that can effectively address the long-tailed distribution of data and improve the recognition performance on tail classes particularly.

In the literature, many researchers have addressed the issue of long-tailed visual recognition by focusing on the classifier level. It is well-known that DNN can be decoupled into a feature extractor and a classifier [9], [10]. Recently, Zhou et al. [11] have conducted empirical studies to demonstrate that the features (also referred to as *embeddings* interchangeably hereinafter) obtained from the original long-tailed dataset are already sufficiently representative. Consequently, they shifted their focus

Manuscript received 23 December 2023; revised 24 March 2024; accepted 10 May 2024. Date of publication 15 May 2024; date of current version 15 October 2024. This work was supported in part by NSFC under Grant 62306181 and Grant 62376233; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023B1515120026 and Grant 2024A1515010163; in part by DEGP Innovation Team under Grant 2022KCXTD025; in part by the NSFC/Research Grants Council (RGC) Joint Research Scheme under Grant N_HKBU214/21; in part by the General Research Fund of RGC under Grant 12201321, Grant 12202622, and Grant 12201323; in part by the RGC Senior Research Fellow Scheme under Grant SRFS2324-2S02; in part by China Fundamental Research Funds for the Central Universities under Grant 20720230038); and in part by Xiaomi Young Talents Program. This article was recommended for publication by Associate Editor Pau-Choo Chung upon evaluation of the reviewers' comments. *(Corresponding author: Yiu-ming Cheung.)*

Mengke Li is with Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518132, China (e-mail: limengke@gml.ac.cn).

Yiu-ming Cheung, Zhikai Hu, and Weichao Lan are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR 999077, China (e-mail: ymc@comp.hkbu.edu.hk; cszkhu@comp.hkbu.edu.hk; cswclan@comp.hkbu.edu.hk).

Yang Lu is with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: luyang@xmu.edu.cn).

Hui Huang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: huihuang@szu.edu.cn).

Digital Object Identifier 10.1109/TAI.2024.3401102

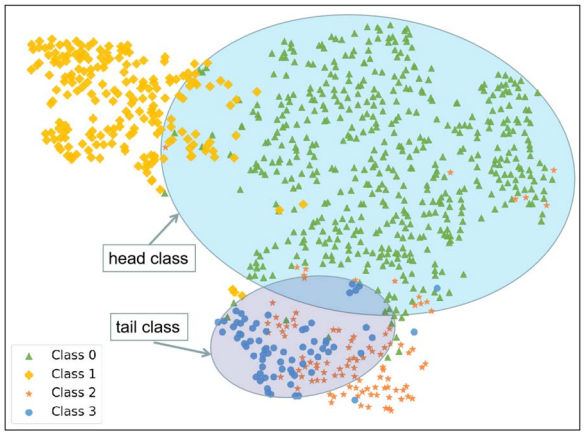

Fig. 1. T-SNE visualization of the distorted embedding space.[1] The embedding distributions of head and tail classes are shown in shaded areas. We can see that there are many overlapping regions between each class.

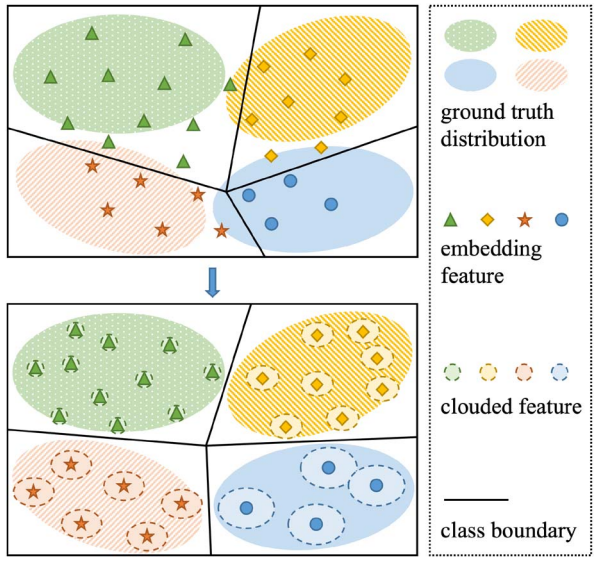

Fig. 2. Overview of the proposed method. The embedding distribution obtained by CE loss is uneven, leading to difficulty in classifying the tail class. By assigning larger cloud sizes to the tail class features, the distortion of the embedding space can be well-calibrated.

to balancing the classifier through two versions of sampling data. Also, two-stage decoupling methods [12], [13], [14], [15] have been proposed to obtain a representation in the first stage and then retrain the classifier on balanced sampling data in the second stage. These methods obtain the representation by cross-entropy (CE) loss, which, however, leads to a severely uneven distribution of the embedding space, hindering the acquisition of a better classifier. Furthermore, retraining the classifier can only alleviate the classifier bias but cannot adjust the distorted embedding space, which is not conducive to further promoting the model performance.

For the feature issue, specifically, the embedding spatial span of tail classes is drastically compressed by head classes because they have limited training samples that cannot cover the true distribution in embedding space. For ease of understanding, we use a simple experiment to demonstrate the distortion of the embedding space, as illustrated in Fig. 1, where the features are projected by t-SNE [16]. It can be observed that the tail class occupies a much smaller spatial span than the head class.

A straightforward way to calibrate the distorted embedding space is to enlarge the spatial distribution of tail classes. Analogous to human cognition, where a person is capable of inferring the extension of an entire category from a single instance [17], we treat one training sample as a set of similar samples. By augmenting the features, we can control the spatial span of the embedding. As only the orientation of the class anchors contributes to the classification, we increase the perturbation amplitude of the tail classes along the direction of the corresponding class anchors. This expands the spatial distribution of tail classes and prevents them from being overly compressed by head classes. Conversely, these amplitudes for head classes should be small. Since their samples with enough diversity already cover the actual spacial span, additional expansion is no need anymore. Eventually, as shown in Fig. 2, the tail class samples can be pushed further away from the other classes so that the distortion of the embedding space can be well calibrated. To this end, we first expand the embedding spatial span with a Gaussian form of perturbation. Based on this, we propose a novel logit adjustment method in two forms: normalized Euclidean and Angular. This method improves model performance with negligible additional computation. Since Gaussian distribution has a cloud-like shape, we name the perturbation amplitude as cloud size and the proposed method as Gaussian clouded logit (GCL). After calibrating the embedding space with GCL, the features of different classes can be more evenly distributed. It turns out that the classifier bias can be easily eliminated through class-balanced sampling data [18], [19] in such a balanced-distributed space. Extensive comparison experiments implemented on multiple commonly used long-tailed benchmarks demonstrate the superiority of the proposed GCL.

Compared with our preliminary work reported in [20], the primary distinction of this article can be summarized as follows: First, this article provides a general form of perturbed logit by perturbing the logit to calibrate the distribution of embedding space. Accordingly, two specific forms based on different metrics are derived from this general form. Second, we present the analysis and explanation of the rationale of GCL in detail, based on which more general parameter selection strategies are provided. After calibrating the embedding space with GCL, the classifier bias can be mitigated by simply retraining with the balanced sampling data. Third, more experiments are conducted to demonstrate the effectiveness of the proposed method. Specifically, we add more classification baselines to show the efficacy of GCL. Furthermore, we demonstrate that GCL can enhance the performance of mixture of experts (MoE) model. Additionally, we provide in-depth theoretical and experimental analyses of the characteristics of GCL in both its normalized

[1]The embeddings are obtained by CE loss from a subset with four classes in CIFAR-10-LT. We randomly select 500, 200, 100, and 50 samples for each class to simulate the data imbalance.

Euclidean and angular forms. In summary, the main contributions of this article are threefold.

1) We propose a simple but effective GCL adjustment method derived from the Gaussian perturbed feature. Tail classes are assigned larger cloud sizes than head classes along the direction of the corresponding class anchors. Consequently, it can address the problem of the distorted embedding space caused by long-tailed data.
2) We provide in-depth discussions into GCL for long-tail learning from the perspective of optimization and geometric interpretation. They help set the sign and magnitude of the perturbation and provide a new idea for better generalization to the test set.
3) We obtain two specific forms of GCL. Both of them outperform state-of-the-art counterparts on long-tailed benchmark datasets without additional computation. Their advantages and disadvantages in different long-tailed scenarios are analyzed in detail.

The remainder of this article is organized as follows: Section II makes an overview of the recent related works. Section III details the derivation and rational analysis behind the proposed GCL. Section V presents our experimental results in comparison with the baseline methods, as well as model validation and analysis. Finally, Section VI draws a conclusion.

II. Related Works

Over the past years, a number of methods have been proposed to address long-tailed visual recognition. This section provides an overview of the most related four regimes. That is, data augmentation, two-stage method, Mixture of Experts (MoEs), and loss modification and logit adjustment (LA).

A. Data Augmentation

Input augmentation increases sample diversity in the data space. The classical augmentation methods [1] encompass operations such as flipping, rotating, cropping, padding, etc. Most recently, Wang et al. [21] proposed rare-class sample generator (RSG) that augments tail classes by utilizing encoded variation information obtained from head classes. Major-to-minor translation (M2m) [22] establishes a well-balanced dataset through the translation of samples from head classes to tail classes, facilitated by an auxiliary pre-trained classifier.

Feature augmentation serves to enhance data diversity within the feature space. Knowledge transfer is a promising technology. For instance, Yin et al. [23] exemplified knowledge transfer by leveraging the intraclass variance derived from head classes in an encoder–decoder-based network to augment the features of tail class samples. Liu et al. [24] employed the transfer of angular variance, computed from head classes, to enrich the intraclass diversity within tail classes. Moreover, recent applications in addressing long-tailed data incorporate the use of class activation maps (CAM) [25]. Chu et al. [26] utilized CAM to decompose the features into a class-generic and a class-specific component. Then, tail classes are augmented by fusing the class-specific components obtained from the tail classes with the class-generic components of the head classes. Also, Zhang et al. [27] exploited CAM to obtain the foreground in an image and then augment the obtained foreground object by flipping, rotating, jittering, etc. The augmented foreground is then covered on the unchanged background to obtain a new informative image.

Those methods mentioned above require either an increase in data size or model complexity to solve the issues in long-tailed distribution, resulting in additional computational costs.

B. Two-Stage Method

Recently, two-stage methods have been proposed and empirically demonstrated their efficacy. For example, Cao et al. [13] proposed label-distribution-aware marginwith deferred reweighting (LDAM-DRW), wherein features are learned in the initial stage, and a DRW strategy is employed to refine the classifier in the subsequent stage. While it markedly enhances long-tailed prediction accuracy, the theoretical underpinnings of the deferred DRW strategy remain unclear. Following this, Kang et al. [12] precisely identified that the learning process of representation and classifier can be decoupled into two separate stages. The first stage performs representation learning on the original long-tail data. The second stage fixes the parameters of the backbone network and retrains the classifier using class-balanced sampling data. Several studies [14], [15], [28] have further refined this strategy. For example, Zhang et al. [15] proposed an adaptive calibration function to calibrate the predicted logits of different classes, aligning them with a balanced class prior to preparation for the second stage. Zhong et al. [28] proposed class-based soft labels to address varying degrees of overconfidence in the predicted logit of each class, which can improve the classifier learning in the second stage. Another alternative approach is proposed by Zhou et al. [11], wherein the network structure is bifurcated into two branches. One branch focuses on learning the representation of head classes, while the other is tailored for tail classes. This structure incorporates feature mixup [29] into a cumulative learning strategy, yielding state-of-the-art results. Subsequently, Wang et al. [30] introduced contrastive learning into this bilateral-branch structure, further enhancing the performance of long-tailed classification.

C. Mixture of Experts

More recently, researchers have explored the use of MoEs methods to enhance performance by integrating multiple models into the learning framework. The fundamental concept behind these approaches is to introduce diversity to the data or models, which enables experts to concentrate on different portions of the data or allows experts with different structures to analyze the data. BBN [11] proposes a two-branched classifier that learns both the long-tailed and inverse distributions simultaneously, with a smooth transition of focus between them. Balanced group softmax (BAGS) [31], Learning from multiple experts (LFME) [32], and Ally complementary experts (ACE) [33] divide the long-tailed data into different subsplits and fit multiple experts on them. ResLT [34] designs residual structured classifiers that allow experts to specialize in different parts of the long-tailed data and complement each other. Routing

diverse experts (RIDE) [35] and Trustworthy long-tailed classification (TLC) [36] employ multiple experts, each trained on different augmented data, to independently learn the long-tailed distribution. The predictions of all experts are then gradually integrated to reduce overall model variance or uncertainty. Self-heterogeneous integration with knowledge excavation (SHIKE) [37] investigates the impact of feature depth on data of varying scales in long-tailed visual recognition. The authors proposed a new architecture, which incorporates features from different layers of a neural network to exploit the rich information present at different depths of a network. Nested collaborative learning (NCL) [38] adopts multiple complete networks to learn the long-tailed data individually and uses self-supervised contrastive strategy [39] to collaboratively transfer knowledge among each individual expert.

D. Loss Modification and LA

Reweighting the loss function is one of the most intuitive ways to improve the attention of DNN model on tail classes. In the literature, sample-wise reweighting [40], [41] introduces the fine-grained coefficients into the loss function to make the model pay more attention to the difficult samples. Furthermore, class-wise reweighting [18], [42], [43] assigns the standard CE loss with category-specific parameters that are inversely proportional to the class sizes. These methods can alleviate the data imbalance to a certain extent. However, when the imbalance ratio is very high, large weights may cause overfitting to the tail classes. Besides that, another side effect of assigning higher weights to difficult samples/tail classes is overly focusing on harmful samples (e.g., abnormal samples or mislabeled data) [44].

Loss function can also be modified by adjusting the logit. Menon et al. [45] proposed LA, which is consistent in minimizing the balanced error. The logit shifting in LA of different classes is based on label frequencies of training data. By contrast, label distribution disentangling (LADE) [46] postprocesses the model prediction by disentangling the training set distribution from the prediction. This method does not require the test set to be a uniform distribution. Also, DisAlign [15] adjusts the logit by calibrating the distribution of model prediction to a balanced one by minimizing the expected KL divergence. Overall speaking, these three methods can well adjust the classifier but do not take into account the distorted embedding space. Alternatively, remargining methods [13], [47], [48] address long-tailed data by leaving large relative margins for tail classes during training. For example, LDAM loss [13] utilizes Rademacher complexity to theoretically prove that the margin should be inversely proportional to a quarter power of class sizes. The hard margin on target logit helps make the samples within a class more compact but the strict margin constraints increase the risk of overfitting and cannot actually expand the tail class coverage area in embedding space.

III. Proposed Method

The basic idea of our proposed method is to perturb the features with varying magnitudes in the directions of different class anchors, thereby automatically balancing the spatial span of head and tail classes. The details of the proposed approach are presented as follows.

A. Basic Notations

This section defines the notation used throughout this article.

1) For Dataset: Suppose $\{x, y\} \in \mathcal{T}$ represents a sample $\{x, y\}$ from the training set $\mathcal{T}$, where $\mathcal{T}$ has C classes and N training samples in total, x represents the image that needs to be classified and $y \in \{1, \ldots, C\}$ is the ground truth label. The number of training samples of class $j, (j = \{1, 2, \cdots, C\})$ is n_j and $\sum_{j=1}^{C} n_j = N$.

2) For Backbone: The feature vector $\mathbf{f} \in \mathbb{R}^D$ is derived from the embedding layer, with a dimensionality of D. $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C\} \in \mathbb{R}^{D \times C}$ represents the weight matrix of the classifier, where $\mathbf{w}_j$ represents the anchor vector of class j in the classifier. The predicted logit of class j is represented by z_j, thus, $z_j = \mathbf{w}_j^T \mathbf{f}$. The subscript y indicates the target class. That is, z_y denotes the target logit and $z_j, j \neq y$ is the non-target logit.

B. Embedding Space Calibration

Suppose a feature point and a small area around it belong to the same type. It is reasonable that the adjacent points around a feature can be regarded as similar to it, and can naturally be considered as the same class.

1) General Form via Perturbing the Embedding Representation: We sample a set of points by adding perturbations following a specific distribution to a given feature. Then, a perturbed feature $\mathbf{f}^{\text{ptb}}$ of the input is represented as

$$\mathbf{f}^{\text{ptb}} \triangleq \mathbf{f} + \delta \mathbf{E} \tag{1}$$

where $\mathbf{E}$ represents the perturbation and $\delta > 0$ is the amplitude of it. To avoid misleading the final classification, the perturbation amplitude cannot be too large, thus δ should be a small number. This perturbed feature is the input of the classifier. Then, the corresponding perturbed logit z_j^{ptb} of class j is calculated as

$$\begin{aligned} z_j^{\text{ptb}} &= \mathbf{w}_j^T \mathbf{f}^{\text{ptb}} + \mathbf{b}_j \\ &= \mathbf{w}_j^T \mathbf{f} + \mathbf{b}_j + \mathbf{w}_j^T (\delta \mathbf{E}) \\ &= z_j + \delta (\mathbf{w}_j^T \mathbf{E}) \end{aligned} \tag{2}$$

where z_j^{ptb} is the original logit z_j augmented by a perturbing a perturbing item $\delta(\mathbf{w}_j^T \mathbf{E})$.

2) Normalized Euclidean Form: It should be noted that the perturbing item has different degrees of influence on the final predicted results based on different predicted logits. The impact on z_j^{ptb} is relatively minor when the original logit z_j is large. Conversely, it becomes more pronounced for z_j^{ptb} when z_j is small. Consequently, it is imperative to normalize the effects induced by varying predicted logits while preserving the consistency of the perturbing item's influence. We achieve this by employing cosine distance through the normalization of the perturbed logits. Here, s_e and s_a represent the norms of the embedding and the class anchor, respectively, that is

$s_e = \|\mathbf{f}\|$ and $s_a = \|\mathbf{w}_j\|$. The normalized perturbed logit $\tilde{z}_j^{\text{ptb}}$ is expressed as

$$\begin{aligned}\tilde{z}_j^{\text{ptb}} &= \frac{s_a \mathbf{w}_j^T \cdot s_e \mathbf{f}^{\text{ptb}}}{\|\mathbf{w}_j^T\| \|\mathbf{f}^{\text{ptb}}\|} \\ &= \tilde{s} \cdot \left(\frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta \mathbf{E}\|} + \delta \frac{\mathbf{w}_j^T \mathbf{E}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta \mathbf{E}\|} \right) \end{aligned} \tag{3}$$

where $\tilde{s} = s_a \cdot s_e$. $\|\mathbf{f} + \delta \mathbf{E}\|$ is approximate to $\|\mathbf{f}\|$ because δ is a small number. For the second term, we use $\mathbf{I}_j$ to represent the unit vector that has the same direction as $\mathbf{w}_j^T$, namely $\mathbf{I}_j = (\mathbf{w}_j^T / \|\mathbf{w}_j^T\|)$. Equation (3) is simplified as

$$\begin{aligned}\tilde{z}_j^{\text{ptb}} &\approx \tilde{s} \cdot \left(\frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f}\|} + \delta \mathbf{I}_j \frac{\mathbf{E}}{s_e} \right) \\ &= \tilde{s} \cdot \left(\cos\theta_j + \frac{\delta}{s_e} \mathbf{I}_j \mathbf{E} \right) \end{aligned} \tag{4}$$

where θ_j is the angle between $\mathbf{f}$ and $\mathbf{w}_j$. Inspired by [49], the predictions can be made solely based on the angle between the feature and the class anchor. Therefore, following [2], [50], we can utilize a fixed norm of individual class anchor to substitute s_a. Without loss of generality, we employ $s_a = 1$. Additionally, following [49], [51], [52], the norm of the embedding feature can also be replaced with a constant s, that is, set $s_e = s$. Consequently, the logit is calculated using features distributed on a hypersphere of radius s. As for the perturbation, we set it to Gaussian distribution, i.e. $\mathbf{E} \sim \mathcal{N}(\mathbf{M}, \Sigma)$ where $\mathbf{M} \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D \times D}$. The rationale behind this choice lies in the widespread adoption of additive Gaussian noise in machine learning [53] attributed to the simplicity and universality [54], [55] of Gaussian distribution. Moreover, we specifically set $\Sigma = \sigma \mathbf{I}$ where $\mathbf{I} \in \mathbb{R}^{D \times D}$ is the identity matrix. Then $\mathbf{I}_j \mathbf{E}$ is the projection of the perturbation on the direction of the anchor vector of class j. We directly use ε_j to represent this value, which can be interpreted as the amplitude of the projection. By substituting the aforementioned norms and perturbation into Equation (4) and uniformly shifting the class-related variable to the predefined perturbation amplitude δ for simplicity, we derive a more concise expression for $\tilde{z}_j^{\text{ptb}}$

$$\begin{aligned}\tilde{z}_j^{\text{ptb}} &= s \cdot \left(\cos\theta_j + \frac{\delta}{s} \varepsilon_j \right) \\ &\Leftrightarrow s \cdot (\cos\theta_j + \delta_j \varepsilon). \end{aligned} \tag{5}$$

Since ε is also distributed in Gaussian form, it has a cloud-like shape. δ_j is the class-based perturbation amplitude that depends on label frequencies. We name δ_j cloud size because it controls the amplitude of ε. To broaden the embedding space for the tail classes, the cloud size for tail classes is required to be larger than that of the head classes. Therefore, δ_j is negatively correlated with n_j. In addition, given that $\cos\theta_j \in [-1, 1]$, the consistency of the influence of the perturbing item can be maintained.

As ε makes the logit has a cloud-like shape, we name the perturbed logit as GCL. We delve into Equation (5). If $\varepsilon > 0$, $\tilde{z}_j^{\text{ptb}}$ corresponds to the points that are closer to the anchor vector of class j. The correct classification of proximal points does not guarantee the accurate classification of distant points within the same class. Therefore, $\varepsilon > 0$ will not be helpful for classification. On the contrary, a reduced logit corresponds to the points that are relatively far from the class anchor. If the relatively distant points can be predicted correctly, the closer one will definitely be able to assign the right label. The points in the same class that are relatively far from the class anchor should be focused on. ε therefore should always be negative. We name this logit as GCL in normalized Euclidean form (GCL-E for short) because it is derived from normalized Euclidean distance metric. We modify the perturbed logit and use $\tilde{z}_j^{\text{GCL-E}}$ to represent it, which is expressed as

$$\tilde{z}_j^{\text{GCL-E}} = s \cdot \left(\cos\theta_j - \delta_j^E \|\varepsilon\| \right) \tag{6}$$

where δ_j^E is the cloud size for GCL-E.

3) Angular Form: The final logit of GCL in normalized Euclidean form is equivalent to adding a class-based perturbation on cosine logit. From another perspective, namely metric learning, Equation (6) corresponds to adding a Gaussian form margin with class-based variance to the cosine logit (Section IV-B provides a detailed analysis). Inspired by Deng et al. [49], this Gaussian form margin can also be introduced into the angular distance metric. For the sake of distinguishing from GCL-E, this version of GCL is named GCL in Angular form (GCL-A for short). Using $\tilde{z}_j^{\text{GCL-A}}$ to represent. These two forms can be unified into a single expression

$$z_j^\star = s \cdot \left[\cos\left(\theta_j + \nu^A \delta_j^A \|\varepsilon\| \right) - \nu^E \delta_j^E \|\varepsilon\| \right], \tag{7}$$

where $\nu^A \in \{0, 1\}$ and $\nu^E \in \{0, 1\}$ are the switch parameters.

1) When $\nu^A = 1$ and $\nu^E = 0$, we obtain the Angular form, expressed as follows:

$$\tilde{z}_j^{\text{GCL-A}} = s \cdot \cos\left(\theta_j + \delta_j^A \|\varepsilon\| \right). \tag{8}$$

2) When $\nu^A = 0$ and $\nu^E = 1$, we obtain the normalized Euclidean form, denoted as $\tilde{z}_j^{\text{GCL-E}}$, as expressed in Equation (6).

By taking the GCL into the original softmax, we obtain the final loss function of GCL

$$\mathcal{L}_{\text{GCL}}^* = -\frac{1}{N} \sum_i \log \frac{e^{\tilde{z}_{y_i}^{\text{GCL}}}}{\sum_j e^{\tilde{z}_j^{\text{GCL}}}} \tag{9}$$

where $\tilde{z}_j^{\text{GCL}}$ can be the logit of GCL-E ($\tilde{z}_j^{\text{GCL-E}}$) or GCL-A ($\tilde{z}_j^{\text{GCL-A}}$). $\mathcal{L}_{\text{GCL-E}}$ is utilized to represent the loss function of GCL-E and $\mathcal{L}_{\text{GCL-A}}$ denotes that of GCL-A.

C. Classifier Rebalance

Although both GCL-E and GCL-A calibrate the distorted embedding space well, the problem of classifier bias still remains to be addressed.

In the following, we analyze the reasons for the biased classifier. Equation (13) implies that the sample of the target class y punishes the classifier weights $\mathbf{w}_j$ of nontarget class $j, j \neq y$ w.r.t. p_j. In general, the number of training instances in head classes is enormously greater than in tail classes. Therefore, the classifier weights of tail classes receive much more penalty than

positive signals during training. Consequently, the classifier will be biased toward the head classes and the predicted logits of the tail classes will be seriously suppressed, resulting in low classification accuracy of the tail classes [43], [56], [57]. We call this problem of the CE loss function in long-tailed learning *negative gradient over-suppression*. A straightforward approach to cope with it is to make the sample numbers of each class equal [58] to balance the negative gradients. To achieve this goal, we can make the tail classes over-sampling and then retrain the classifier. The sampling rate of each class is $(1/C)$. Then, the class-balanced sampling rate q_j^{cb} of each sample x from class j is calculated as

$$q_j^{cb} = \frac{1}{C \cdot n_j}. \tag{10}$$

This strategy is called classifier retraining (cRT) [12]. It can also be combined with the *effective number* [18]. We can replace the actual sample number n_j of class j with the so-called *effective number* n_j^{en}, the effective sampling rate q_j^{en} of each sample from class j is given as

$$q_j^{en} = \frac{1}{C \cdot n_j^{en}} \tag{11}$$

where n_j^{en} is calculated as

$$n_j^{en} = \frac{1-\beta^{n_j}}{1-\beta} N \tag{12}$$

with hyperparameter $\beta \in [0, 1)$. Algorithm 1 summarizes the overall training procedure of the proposed method.

Algorithm 1: GCL with cRT

Input: Training dataset $\mathcal{T}$;
Output: Predicted labels;
Initialize the model parameters ω of the backbone network $\phi((x, y); \omega)$ randomly ;
for $iteration = 1$ *to* I_0 **do**
  Sample a batch of data $\mathcal{B}$ from the original long-tailed dataset $\mathcal{T}$ with a batch size of b;
  Calculate the logit cloud size δ_j by Equation (16) (or Equation (17)):
  $\delta_j \leftarrow n_{max} \cdot n_j^{-k}$ (or $\delta_j \leftarrow \log n_{max} - \log n_j$);
  Calculate the loss by Equation (19):
  $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x,y)\in\mathcal{B}} \mathcal{L}^*_{GCL}(x, y)$;
  Update model parameters:
  $\omega = \omega - \alpha \nabla_\omega \mathcal{L}((x, y); \omega)$.
end
for $iteration = I_0 + 1$ *to* $I_0 + I_1$ **do**
  Calculate sampling rate by Equation (10) (or Equation (11)):
  $q_j \leftarrow n_j / \sum n_j$ (or $q_j \leftarrow n_j^{en} / \sum n_j^{en}$);
  Sample a batch of data $\mathcal{B}'$ with the sampling rate q_j and the batch size b;
  Calculate the loss using Equation (9):
  $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x,y)\in\mathcal{B}'} \mathcal{L}_{GCL}(x, y)$;
  Update the classifier parameters ω_{cls} while keeping the representation parameters frozen:
  $\omega_{cls} = \omega_{cls} - \alpha \nabla_{\omega_{cls}} \mathcal{L}((x, y); \omega_{cls})$.
end

IV. Rationale Analysis

This section provides a detailed rationale analysis of how Equations (7) and (8) balance the embedding space from two perspectives, considering both model optimization and metric learning perspectives, following with a time-complexity analysis.

A. The Perspective of Model Optimization

In backward propagation, the gradients on logit z_i are calculated as

$$\frac{\partial \mathcal{L}}{\partial z_i} = \begin{cases} p_i - 1, & i = y \\ p_i, & i \neq y \end{cases} \tag{13}$$

where $p_i = (e^{z_i} / \sum_{j=1}^{C} e^{z_j})$. We take the binary case to illustrate without loss of generality. Suppose the input image is from class 1. The gradient on z_1 is calculated as

$$\frac{\partial \mathcal{L}}{\partial z_1} = -\frac{1}{1 + e^{z_1 - z_2}}. \tag{14}$$

It indicates that the gradient of the target class rapidly approaches zero with the increase of the target logit. This phenomenon is called softmax saturation [59], [60]. This inopportune early gradient vanishing weakens the validity of training samples and impedes model training. Therefore, softmax can only slightly separate various classes, and lacks the impetus to evenly distribute each class in the embedded space. We can also observe that there are many overlapping areas among each class in Fig. 2. Especially under the circumstances of long-tailed classification, the tail class features are insufficient to cover the ground truth distribution in embedding space. The early gradient vanish caused by soft saturation exacerbates the squeezing of the embedding distribution in tail class.

Different from the original softmax loss function, the logit difference (Δ_{y-j}) obtained by GCL of Equation (6) between the target and nontarget classes is calculated as

$$\begin{aligned} \Delta_{y-j} &= \tilde{z}_y^{\text{GCL-E}} - \tilde{z}_j^{\text{GCL-E}} \\ &= s \cdot \left(\cos\theta_y - \cos\theta_j - (\delta_y^E - \delta_j^E) \|\varepsilon\| \right). \end{aligned} \tag{15}$$

In case the target class is a tail class, $\delta_y - \delta_j > 0$, which decreases the softmax saturation and thereby helps increase the validity of tail class samples. Equation (8) has the same effect. Thus, Equations (6) and (8) can automatically balance the sample validity of different classes and provide incentives for the model to make each class more separable. They achieve the aim of calibrating the distorted embedding space.

B. The Perspective of Metric Learning

Compared with the prior work that enlarges the interclass separability via the "hard margin", e.g., see [13], [49], [60], Equations (6) and (8) are equivalent to adding a "soft" margin. That is, the farther away from the class anchor, the lower the probability that the point belongs to this class. Fig. 3 schematically shows the comparison of the prior hard margin and the

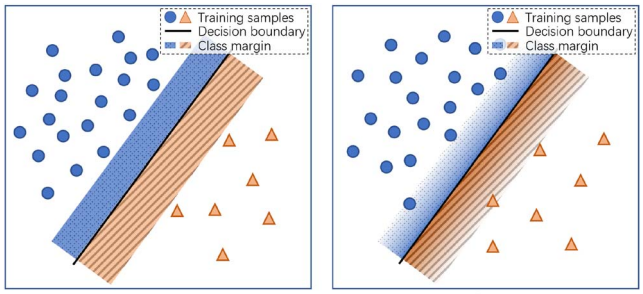

Fig. 3. Schematic comparison of hard margin and soft margin. The blue dots and pink triangles represent the head and tail classes, respectively. (Color for the best view.) (a) Hard margin strictly restricts samples from appearing in the corresponding region. (b) Soft margin allows outliers to appear in the region with a lower probability, which increases generalization.

proposed soft margin. Hard margins will cause the samples to shrink toward the class anchor if the margin is too large. In addition to this, hard margins can lead to overfitting because they prohibit outliers, which can impair the robustness ability of the model. The proposed soft margin provides a smooth transition area, allowing the outliers to appear near the target class with a lower probability. This is both intuitively and theoretically more reasonable.

The cloud size δ_j^* may also take different expression forms, where the superscript $*$ indicates the adopted specific form. Cao et al. [13] obtained the optimal tradeoff of the hard margin (m_i) and the class size via Rademacher complexity. They have proved that $m_i \propto n_i^{-1/4}$. The exponent should be $-1/3$ derived from Wei and Ma [61]. Inspired by these works, we can set the cloud size in power function form as follows:

$$\delta_j^{\text{pow}} = n_{\max} \cdot n_j^{-k} \tag{16}$$

where $n_{\max}$ is the sample number of the most frequent class. k can be $1/3$ or $1/4$. Menon et al. [45] used the Fisher consistency with respect to the balanced error and obtained that $m_i \propto \log(1/n_j)$. Therefore, we can also set the cloud size in logarithmic form as follows:

$$\delta_j^{\log} = \log n_{\max} - \log n_j. \tag{17}$$

We also experimentally demonstrate the effectiveness of the cloud size in different expression forms in Section V-D.

In short, GCL in the form of either normalized Euclidean distance or angular distance can achieve the following three advantages: 1) reduce the softmax saturation and thereby increase the sample validity of tail classes; 2) avoid overfitting and improve robustness through randomly sampling the values in Gaussian distribution; and 3) enlarge the margin of class boundary for tail classes and thus calibrate the distortion of the embedding space. The slight disparity between the two forms lies in the procedural approach: GCL-E incorporates class-based perturbance onto features prior to logit calculation, whereas GCL-A is equivalent to sampling disturbed feature points subsequent to determining their distance from the class anchor. In addition, we systematically illustrate two versions of GCL and their distinctions from previous methods, exemplified by CE and LDAM [13], as shown in Fig. 4.

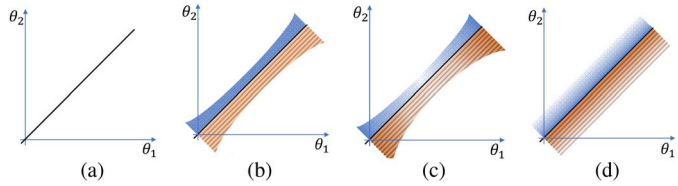

Fig. 4. Geometric illustration of class margins of various loss functions. (Color for the best view.) (a) CE; (b) LDAM; (c) GCL-E; and (d) GCL-A.

C. Time-Complexity Analysis

The softmax has a time complexity of $\mathcal{O}(C)$, which is linear with the dimension of logit. It is the same as CE loss $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{GCL}}$ in both forms. The main difference in time complexity comes from the calculation of logit. For the original normalized logit (which is denoted as $\tilde{z}_j = s \cdot \cos\theta_j$), its main computational cost is vector multiplication. It contains $D \cdot C$ multiplications and $(D-1) \cdot C$ additions. Thus, the time complexity of computing $\tilde{z}_j$ is $\mathcal{O}(DC)$. Equation (6) shows that GCL-E only adds C scalar additions to $\tilde{z}_j$. As a result, computing $\tilde{z}_j^{\text{GCL-E}}$ has $\mathcal{O}(DC)$ time-complexity. For GCL-A, we first expand Equation (8) to $\tilde{z}_j^{\text{GCL-A}} = s \cdot (\cos\theta_j \cos\delta_j \|\varepsilon\| - \sin\theta_j \sin\delta_j \|\varepsilon\|)$. The sine value can be obtained from the corresponding cosine value. Compared with $\tilde{z}_j$, GCL-A adds an additional $2C$ multiplications and C subtractions. Computing $\tilde{z}_j^{\text{GCL-A}}$ also has $\mathcal{O}(DC)$ time-complexity. It is obvious that GCL in both forms imposes a negligible additional burden on the training process.

V. Experiments

This section first introduces five long-tailed datasets used in our experiments in Section V-A. Then, the detailed implementation settings of the experiments are presented in Section V-B. To demonstrate the effectiveness of GCL, we compare the proposed two forms of GCL with state-of-the-art methods based on a single model structure. The classification accuracy is compared in Section V-C. Moreover, Section V-E validates that GCL can also enhance the performance of MoE model. Finally, the model validation experiments and ablation studies are conducted to show the properties of our proposed method in Section V-E.

A. Benchmark Datasets

We use five benchmarks: CIFAR-10-LT and CIFAR-100-LT, ImageNet-LT, iNaturalist 2018, and Places-LT.

1) CIFAR-10/100-LT: The original versions of CIFAR-10 and CIFAR-100 [62] are uniformly distributed datasets, which consist of 10 and 100 classes, respectively. They both contain 60K images with a size of 32×32. The training set contains 50 000 samples and the test set has 10 000 samples. Following the experimental settings in [18] and [13], we down-sampling training images per class with the exponential function $n_i = n_i^o \times \lambda^i$, where i is the class index (0-indexed), n_i^o is the label

frequency in the original balanced version and $\lambda \in (0, 1)$. The test sets are kept unchanged. The imbalance ratio r is defined as the ratio of the maximum and minimum label frequencies, i.e., $r = \max(n_i)/\min(n_i), i = 1, 2, ..., C$. In the comparative experiments, we employ the three most widely used imbalance ratios, namely $r = 50, 100$, and 200.

2) ImageNet-LT and Places-LT: The original versions of ImageNet [63] and Places [64] are artificially balanced, large-scale real-world datasets for classification and localization. Following Liu et al.'s [65], we construct long-tailed versions of these datasets by truncating a subset using the Pareto distribution with a power value $\alpha = 6$ from the balanced versions. The original validation sets are employed for testing. In summary, ImageNet-LT comprises 115.8K training images from 1K categories with $r = 1280/5$. Places-LT consists of 62.5K training images spanning 365 categories with $r = 4980/5$.

3) iNaturalist 2018: iNaturalist 2018 [66] is a real-world fine-grained dataset for classification and detection, exhibiting a naturally long-tailed distribution. It contains different species of plants and animals collected from the real world in a wide variety of situations. This dataset contains over 437.5K training samples and more than 24.4K validation images from 8142 categories. The official validation set is utilized for testing in the experiments. The imbalance ratio of iNaturalist 2018 is $r = 1000/2$.

B. Basic Setting

The parameters that need to be preset are the Gaussian distribution parameters (μ, σ^2). For GCL-E, the maximum cloud size cannot exceed 1 because $\cos\theta_i \in [-1, 1]$. Gaussian distribution has a probability of 99.7% falling in $[\mu - 3\sigma, \mu + 3\sigma]$, we therefore set $\mu = 0$ and $\sigma = (1/3)$. We further clamp ε to $[-1, 1]$ to prevent the cloud size from exceeding 1. For GCL-A, we first constrain the range of ε to $[-1, 1]$ in the same way as the cosine form GCL. Then, we multiply ε with a constant $\frac{\pi}{2}$ to limit the cloud size in angular form to $[-(\pi/2), (\pi/2)]$ based on the lemma[2] proposed by Ranjan et al. [51]. Moreover, we normalize δ_i by $\delta_i \triangleq \delta_i/\max(\delta_i), i = 1, 2, ..., C$ to ensure that maximum value of δ_i does not exceed 1. For data augmentation techniques, we follow Zhong et al. [28], except for basic augmentation such as image flip, rotation, and random crop, only mixup [67] is adopted in all experiments to ensure fair comparisons.

PyTorch [68] is utilized to implement the backbone network training. We adopt the SGD optimizer with a momentum of 0.9, coupled with a multistep learning rate schedule. All models are trained from scratch, except for ResNet-152, which is pretrained on the original balanced version of ImageNet-1K. For the first stage, we select ResNet-32 as the backbone network and follow the experimental settings in Cao et al. [13] for CIFAR-10/100-LT. For the experiments conducted on large-scale datasets, namely, ImageNet-LT, iNatralist 2018, and Places-LT, we mainly follow Kang et al.'s settings [12] except for the learning rate schedule. For the second stage, i.e., rebalancing the classifier, we follow Kang et al.'s setting [12] for all datasets.

[2]**Lemma**: The classes can be distributed on a hyper-sphere of dimension D such that any two class centers (namely, class anchors in this paper) are at least $\pi/2$ apart if the number of classes C is less than twice the feature dimension D.

C. Main Comparison Results

1) Competing Methods: The competing methods can be categorized into the following two groups:

a) Baseline methods: Vanilla training with CE loss serves as one of our baseline methods. Previous studies in visual recognition [13], [73], [74], [75] have demonstrated the effectiveness of cosine similarity in mitigating the impact of imbalanced feature bias within imbalanced data distributions. Therefore, we also include CosFace [50] and ArcFace [49] as additional baseline methods.

b) State-of-the-art methods: We compare with the most recently proposed state-of-the-art methods, including targeted supervised contrastive learning (TSC) [70], memory-based Jitter (MBJ) [71], feature-balanced loss (FBL) [72], and two-stage methods including LDAM-DRW [13] and MisLAS [28]. These methods have demonstrated notable classification accuracy across the aforementioned long-tailed datasets. For a fair comparison, we implement the experiment of the two-stage strategy, i.e., adding mixup [67] to decoupling [12] on all datasets. For CIFAR-10/100-LT datasets, we make a comparison with the LA method (deconfound-TDE [69]). BBN [11] and contrastive learning [30] are also included in the competing methods. For large-scale datasets, the representation learning method (open long-tailed recognition [OLTR] [65]), and LA method (LA [45]) are included. The two-stage methods including decoupling [12], and DisAlign [15] are also compared.

2) Comparison Results: Extensive comparative experiments are conducted to illustrate the efficacy of our proposed GCL in two forms (GCL-E and GCL-A). The evaluation metric for assessing performance is top-1 accuracy on the test/validation sets. For comparison methods that have not released official code or relevant hyperparameters, we quote the results directly from the original papers

a) Results on CIFAR-10/100-LT: The proposed GCL-E and GCL-A both outperform the previous methods by notable margins with all imbalanced ratios. Especially for the largest r, i.e., 200, the proposed approach has obvious improvement. For example, GCL-E gets 79.03% and 44.84% in top-1 classification accuracy for CIFAR-10-LT and CIFAR-100-LT with $r = 200$, which surpasses the second-best method, i.e., FBL [72] (on CIFAR-10-LT) and MisLAS [28] (on CIFAR-100-LT) by a significant margin of 0.93% and 2.51%, respectively. GCL-A further improves the performance compared with cosine form except on CIFAR-10-LT with $r = 100$ (82.72% top-1 accuracy, which is still higher than the existing methods). For example, it increases the top-1 accuracy from 44.84% to 46.53% for CIFAR-100-LT with $r = 200$ compared with the cosine form. The margin is more than 3% compared with MisLAS. Interestingly, we can observe that CosFace [50] and ArcFace [49] perform well compared with CE loss, illustrating the efficacy of angular distance metric in long-tail learning. In comparison to LDAM-DRW [13] which is also based on angular distance

TABLE I
COMPARISON RESULTS ON CIFAR-10/100-LT W.R.T. TOP-1 ACCURACY (IN PERCENT)

Dataset	CIFAR-10-LT			CIFAR-100-LT		
Imbalance ratio	200	100	50	200	100	50
CE loss	65.68	70.70	74.81	34.84	38.43	43.9
CosFace [50]	66.22	72.08	77.40	35.36	39.21	43.11
ArcFace [49]	66.50	73.76	78.19	36.64	39.06	43.40
LDAM-DRW [13]	73.52	77.03	81.03	38.91	42.04	47.62
De-confound-TDE*[69]	-	80.60	83.60	-	44.15	50.31
Decoupling [12]	73.06	79.15	84.21	41.73	45.12	50.86
BBN [11]	73.47	79.82	81.18	37.21	42.56	47.02
Contrastive learning [30]	-	81.40	85.36	-	46.72	51.87
MisLAS [28]	77.31	82.06	85.16	42.33	47.50	52.62
TSC*[70]	-	79.70	82.90	-	43.80	47.40
MBJ [71]	77.06	81.10	**85.45**	41.92	46.05	52.43
FBL [72]	78.10	82.46	84.30	40.67	45.22	50.65
GCL-E (ours)	**79.03**	*82.73*	85.43	**44.84**	**48.69**	**53.51**
GCL-A (ours)	*79.31*	**82.72**	*85.58*	*46.53*	*49.97*	*54.75*

Note: *denotes that the results are quoted from the corresponding papers. Other results are obtained by reimplementing with the official codes. The best and the second-best results are shown in italic and bold, respectively.

TABLE II
COMPARISON RESULTS ON IMAGENET-LT, INATURALIST 2018, AND PLACES-LT W.R.T. TOP-1 ACCURACY (IN PERCENT)

Dataset	img-LT	iNat	Pla-LT
Backbone	ResNet-50	ResNet-50	ResNet-152
CE loss	44.51	63.80	27.13
CosFace [50]	44.95	72.08	27.19
ArcFace [49]	44.54	66.72	27.63
LDAM-DRW [13]	49.96	68.15	37.73
OLTR* [65]	-	-	35.90
Decoupling [12]	51.68	70.16	38.51
LA*[45]	51.11	66.36	-
DisAlign*[15]	52.91	70.06	39.30
MisLAS [28]	52.71	**71.57**	**40.36**
TSC*[70]	52.40	69.70	-
MBJ*[71]	52.10	70.00	38.10
FBL [72]	50.70	69.90	38.66
GCL-E (Ours)	**54.84**	*72.01*	*40.62*
GCL-A (Ours)	*55.12*	71.14	39.22

Note: img-LT, iNat and Pla-LT short for ImageNet-LT, iNaturalist 2018 and Places-LT, respectively. Others are the same as Table I.

TABLE III
ABLATION EXPERIMENT OF DIFFERENT EXPRESSION FORMS OF CLOUD SIZE (δ_j^*) ON CIFAR-10-LT WITH $r = 100$

δ_j^*	Exp.	Expression	Acc (in Percent)
cos.	-	$\cos\left(n_j/n_{\max} \cdot \pi/2\right)$	79.21
power	1/3	$n_{\max} \cdot n_j^{-1/3}$	80.80
power	1/4	$n_{\max} \cdot n_j^{-1/4}$	82.31
log.	-	$\log n_{\max} - \log n_j$	**82.73**

Note: Bold indicates the best results.

metric, our proposed solution is still the clear winner. The performance gain is obtained by the smooth margin that can avoid overfitting and improve robustness. The clear performance gain compared with decoupling [12] demonstrates that calibrating the feature space via GCL is beneficial to the subsequent classifier learning. The results on CIFAR-10/100-LT datasets are summarized in Table I.

b) Results on large-scale datasets: The results on large-scale long-tailed datasets including ImageNet-LT, iNaturalist 2018, and Places-LT are reported in Table II. We observe that GCL-E is superior to the prior arts on all datasets. On ImageNet-LT, it achieves 54.84% top-1 accuracy, surpassing DisAlign [15] by a notable margin of 1.97% and MisLAS [28] by 2.77%. For iNaturalist 2018, the proposed GCL-E achieves a top-1 accuracy of 72.01%, outperforming the second-best method by 0.44%. On Place-LT, our proposed method achieves 40.62% top-1 classification accuracy. Although the performance gain compared with MisLAS on iNaturalist 2018 and Place-LT is not as high as other datasets, our method does not require hyperparameters searching for different datasets and thus is relatively easy to implement. GCL-A largely improves the performance on ImageNet-LT from 54.84% to 55.12%, but it slightly decreases the accuracy on iNaturalist 2018 and Places-LT. GCL-A achieves 71.14% top-1 classification accuracy on iNaturalist 2018, which is lower than MisLAS but still outperforms the other baseline methods by notable margins, showing the effectiveness of angular perturbation to balance the embedding space distribution. On Places-LT, it has a lower accuracy than MisLAS and DisAlign.

D. Ablation Study

1) Expression of Cloud Size: We explore several different cloud size adjustment strategies, including power form with different exponents (1/3 and 1/4), and logarithmic form. For a fair comparison, we use GCL-E, and the sampler and re-training strategy are selected as class-balanced sampling and cRT, respectively. The results are presented in Table III. The logarithmic form has the best performance and the power form with the exponent of 1/4 is also competitive.

2) Strategies for Class Rebalancing: We implement different strategies of data resampling and classifier retraining (RT) technique to better analyze our proposed method.

TABLE IV
ABLATION EXPERIMENT OF DIFFERENT RESAMPLING AND RETRAINING STRATEGIES ON CIFAR-10-LT WITH $r = 100$

Sampler	RT Tech.	Acc. (in Percent)
IBS	cRT	80.52
SRS	cRT	81.74
ENS	cRT	82.45
-	w.o. RT	80.55
CBS	LWS	82.25
CBS	τ-NC	82.16
CBS	cRT	**82.73**

Note: Bold indicates the best results.

Table IV shows the results. The resampling strategy (sampler) includes instance-balanced sampler (IBS) [12], square-root sampler (SRS) [76], effective number sampler (ENS) [18] and class balanced-sampler (CBS) [12]. The form of GCL is GCL-E and the RT techniques for all samplers are cRT. IBS decreases the performance slightly (from 80.55% to 80.52%), which indicates that training the classifier with IBS leads to classifier overfitting. SRS improves the model performance because it increases the sampling probability of tail classes. ENS and CBS have better performance because they can address the problem of negative gradient over suppression by balancing the amount of data in each class. We use CBS in the comparison experiments because it achieves the best results among these samplers. For the selection of RT technique, we first train the backbone without any RT technology using GCL-E. Then we froze the representation and rebalance the classifier with learnable weight scaling (LWS), τ-normalized classifier (τ-NC), and cRT, respectively. We can observe that even without any RT technique, our approach (the top-1 classification accuracy is 80.55%) can still beat most state-of-the-art including two-stage methods (for example, LDAM-DRW and BBN achieve 77.03% and 79.82%, respectively). All RT techniques significantly improve model performance, which demonstrates that good representation can improve classification accuracy by simply rebalancing the classifier. cRT outperforms best among the classifier retraining techniques, which improves the accuracy by 2.18% compared with no RT. Thus, we use cRT in the comparison experiments.

E. Further Analysis

We conduct a series of experiments to further analyze the proposed method.

1) Effectiveness on MoE Model: We select RIDE [35] as a representative of MoE Models. The reproduction of RIDE in our experiment follows the original settings, which utilize LDAM loss and DRW strategy. We employed three experts in our MoE model and adopted the mixup technique to ensure a fair comparison. MoE models have been shown to outperform single models, albeit at the expense of increasing model size. For instance, RIDE with GCL-E achieved an accuracy of 81.32% on CIFAR-10-LT with an imbalance ratio of 200, which is an obvious improvement from the 79.03% achieved by a single ResNet-32 model with GCL-E. However, the model size of RIDE is 5.38 Mb, whereas the single model had a size of only 1.84 Mb. Tables V and VI demonstrate the improvement in performance achieved by GCL on RIDE. Both versions of GCL can be observed to improve RIDE's performance significantly on all datasets. The improvement of GCL-A ranges from 0.90% to 2.62%, while that of GCL-E ranges from 0.82% to 2.64%.

2) GCL-E Versus GCL-A: Combining Tables I and II, it can be observed that GCL-A does not always have inferior performance compared with GCL-E, and vice versa. The reason is that iNaturalist 2018 and Places-LT have much large imbalance ratios ($r = 500$ and 996, respectively) than the other datasets (ImageNet-LT has the largest r which is 256 among these datasets). We draw the logit curve of different forms of GCL, which is shown in Fig. 5. In our setting, the large class has a small δ. The smaller the class size, the larger its corresponding δ. As the distance θ increases, the logit of GCL-A decreases faster than GCL-E. It is more noticeable for the larger δ, as shown in Fig. 5(b). A small distance will have a more obvious logit difference for GCL-A compared with GCL-E. Therefore, in the case of high imbalance ratio, GCL-E can make the separability of minority classes stronger so that the logit difference is more significant.

Another rationale arises from the discrepancy in logits restrictions caused by varying imbalance ratios. Excessively strict logit constraints may lead the model astray. Without loss of generality, we use the most frequent class (denoted by subscript "head") and the least frequent class (denoted by subscript "tail") to analyze. For an input image that is tail class, GCL-A necessitates

$$\cos\left(\theta_{\text{tail}} + \delta \cdot \frac{\pi}{2} \cdot \|\varepsilon\|\right) > \cos\theta_{\text{head}} \Rightarrow \theta_{\text{tail}} < \theta_{\text{head}} - \delta \cdot \frac{\pi}{2} \cdot \|\varepsilon\|. \tag{18}$$

Considering $\delta = 0.5$ as an example, when $\theta_{\text{head}} < (\pi/2)$, θ_{tail} being negative satisfies the requirements of the loss function, which could mislead the model training. The requirement that the angle between nontarget classes and the target weight be greater than $(\pi/2)$ is overly stringent. For highly imbalanced datasets, namely iNaturalist 2018 and Places-LT, the discrepancies in perturbations between tail and head classes are more pronounced, which contributes to this phenomenon. In datasets with a smaller imbalance ratio, the disparities in perturbations are comparatively smaller, making this restriction relatively weaker. The majority of classes can adhere to their respective soft margin restrictions. However, opting for a smaller δ might result in the added perturbation being less conspicuous, thereby leading to less differentiation between classes. For GCL-E, an input image belonging to the tail class should satisfy the following inequality:

$$\cos\theta_{\text{tail}} - \delta \cdot \|\varepsilon\| > \cos\theta_{\text{head}} \Rightarrow \cos\theta_{\text{tail}} > \cos\theta_{\text{head}} + \delta \cdot \|\varepsilon\|. \tag{19}$$

When $\delta = 0.5$, $\theta_{\text{head}} > (\pi/3)$ will cause θ_{tail} to be negative. In contrast, the constraints imposed by GCL-E are more lenient, resulting in a slight decrease in performance on datasets characterized by a low imbalance ratio compared with GCL-A. Nonetheless, this relaxation does not predispose the model to erroneous interpretations stemming from excessively stringent restrictions.

TABLE V
VALIDATION OF THE EFFECT ON MOE MODEL ON CIFAR-10/100-LT

Dataset	CIFAR-10-LT			CIFAR-100-LT		
Backbone	ResNet-32					
Imbalance ratio	200	100	50	200	100	50
RIDE [35]	80.42	83.39	85.34	47.80	50.91	54.87
RIDE w. GCL-E	81.32 (↑ 0.90)	84.32 (↑ 0.93)	87.03 (↑ 1.69)	48.96 (↑ 1.16)	52.57 (↑ 1.66)	57.49 (↑ 2.62)
RIDE w. GCL-A	82.08 (↑ 1.66)	84.73 (↑ 1.34)	86.95 (↑ 1.61)	48.62 (↑ 0.82)	52.38 (↑ 1.47)	57.51 (↑ 2.64)

TABLE VI
VALIDATION OF THE EFFECT ON MOE MODEL ON LARGE-SCALE DATASET

Dataset	ImageNet-LT	iNaturalist 2018	Places-LT
Backbone	ResNet-50	ResNet-50	ResNet-152
RIDE [35]	55.55	72.17	39.91
RIDE w. GCL-E	57.01 (↑ 1.46)	74.27 (↑ 2.10)	41.06 (↑ 1.15)
RIDE w. GCL-A	57.25 (↑ 1.70)	73.56 (↑ 1.39)	41.50 (↑ 1.59)

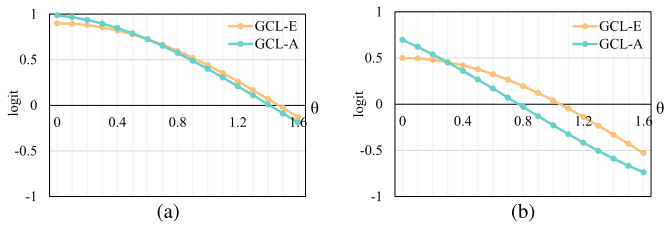

Fig. 5. Logit curve of GCL in different forms. For ease of visualization, the scale parameter s is omitted. (a) $\delta = 0.1$ and (b) $\delta = 0.5$.[3]

Moreover, from another perspective, the selection of the perturbation magnitude δ holds a pivotal role for GCL-A. Additionally, cloud size selection should extend beyond mere class size considerations, with each variant of GCL potentially requiring its optimal strategy for cloud size selection. It is conceivable that the logarithmic form of cloud size utilized for GCL-A does not constitute the optimal choice. We leave these as our future study.

3) The Effect of Gaussian Cloud: To obtain additional insight, we visualize the embedding distribution using t-SNE projection. Since CE loss is selected as the loss function for several methods [11], [12], [65], especially MisLAS performs the second-best in most cases, we visualize the embedding distribution obtained by CE loss for comparison. LDAM [13] is an angular distance metric-based method but utilizes the hard margin, we also show its embedding distribution. The embeddings are calculated from the samples in CIFAR-10-LT with $r = 100$. Fig. 6 shows the results. From Fig. 6(a), it can be seen that the embeddings of each class obtained via CE loss are clustered together and are relatively difficult to separate. The obscure region of CE loss embedding is larger than that of other approaches. LDAM and GCL in both forms are all angular distance metric based methods, thus their embeddings are basically radial. Fig. 6(b) shows that the LDAM embedding of each class is more slender. This is caused by the hard margin that strictly restricts the class region, resulting in overfitting the training set. Thus, LDAM does not generalize well on the test set compared with our proposed GCL. In Fig. 6(c) and 6(d), on training set, the embeddings for each class obtained via GCL in both forms have more obvious margins compared with CE and also are more scattered compared with LDAM. The results of the test set verify the efficacy of our proposed approach. GCL-E and GCL-A have better generalization performance, and it can be found that the misclassified classes are mainly in the edge regions of each class. For better illustration, we additionally compare the embedding distribution of the most (class 0) and least (class 9) frequent classes, along with their respective decision boundaries derived from various loss functions in Fig. 7. Concerning the acquired features, within the training set, the overlap between the features of the head and tail classes by LDAM and GCL is reduced compared with those obtained by CE loss, with a pronounced disparity observed in GCL-A. In addition, it presents more clearly that compared with our proposed GCL, the LDAM embeddings appear to perform better on the training set, but cannot be well generalized to the unseen test samples. In Fig. 7(b), there are more points of class 9 appearing inside the class 0 area on the test set. By contrast, as shown in Fig. 7(c) and 7(d), the misclassified points of class 9 are mainly in the edge area of class 0 on test set. Regarding the decision boundary, CE loss exhibits a tendency to predominantly ensure accurate classification of head classes while often disregarding tail classes. In contrast, due to the presence of margins or perturbations beneficial to the tail class, both LDAM and GCL adopt a holistic approach to class performance. However, this approach comes at the expense of head class performance to some extent. The decision boundary delineates specific head class samples into the tail class.

4) Performance on Classes With Different Scales: To investigate the impact of GCL, we report the accuracy of various scale classes on ImageNet-LT. The results are presented in Table VII. The classification accuracy of baseline methods drops a lot in the middle and tail classes. LDAM-DRW increases the accuracy of middle and tail classes but decreases that of head classes a lot. GCL-E outperforms the other state-of-the-art methods on middle and tail classes with large margins. Meanwhile, the accuracy of the head class decreases the least. By contrast, GCL-A has more improvement in middle and tail classes, but the damage to head classes is slightly higher than GCL-E and decoupling. In general, GCL-E performs well in all class scales. GCL-A has the highest overall classification accuracy. Significantly improving the accuracy of tail classes while preventing that of the head classes from diminishing illustrates the superiority of our approach.

[3]Specifically, the logit curves show $\tilde{z}^{GCL-E} = \cos(\theta) - \delta \cdot \|\varepsilon\|$, and $\tilde{z}^{GCL-A} = \cos(\theta + \delta \cdot (\pi/2) \cdot \|\varepsilon\|)$, namely $s = 1$.

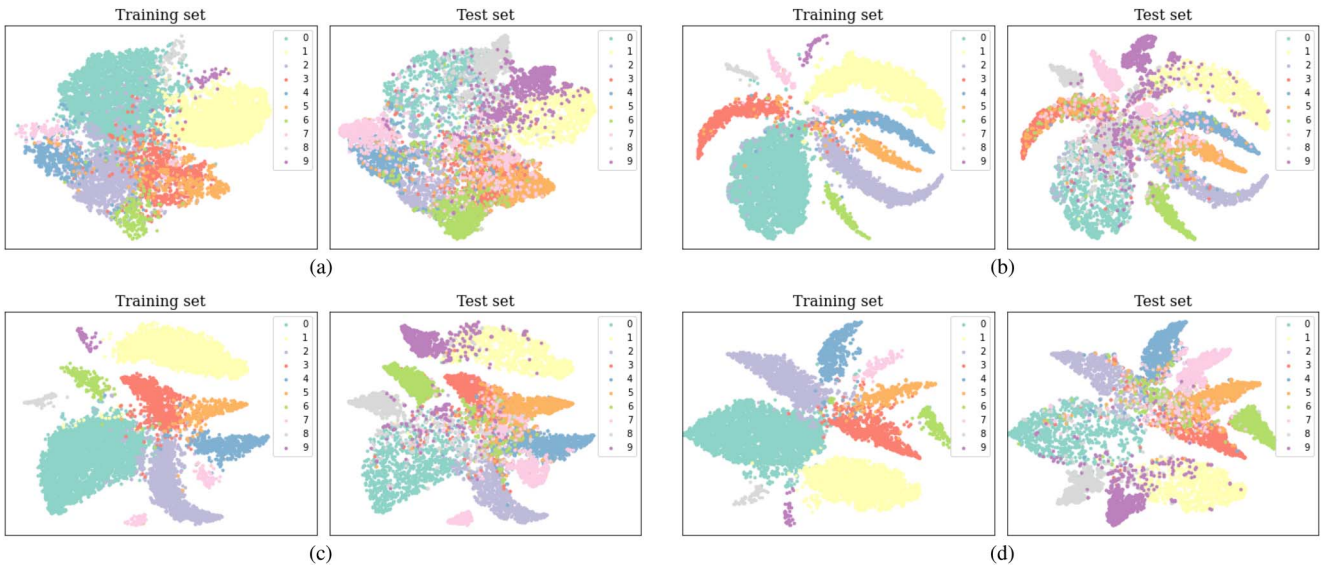

Fig. 6. Visualization of the embedding distribution obtained by different methods. t-SNE projection is utilized. The dataset is CIFAR-10-LT with $r = 100$. ResNet-32 is used as the backbone. (Color for the best view.) (a) CE loss; (b) LDAM loss; (c) GCL-E; and (d) GCL-A.

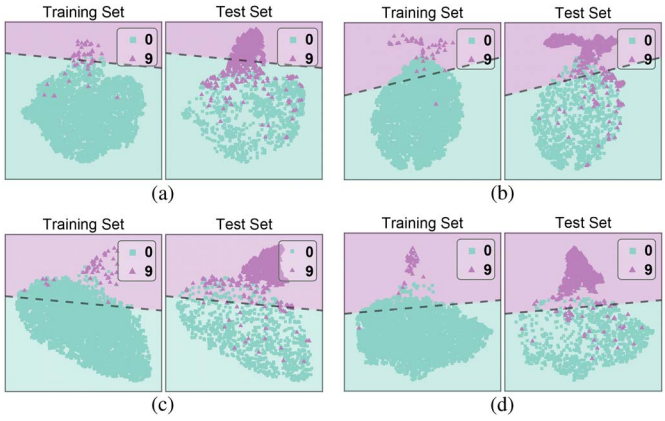

Fig. 7. T-SNE visualization of decision boundary (dashed line) between head (class 0) and tail (class 9) classes. The dataset is CIFAR-10-LT with $r = 100$ and the backbone network is ResNet-32. (Color for the best view.) (a) CE loss; (b) LDAM loss; (c) GCL-E; and (d) GCL-A.

TABLE VII
TOP-1 CLASSIFICATION ACCURACY (in Percent) OF THE THREE SPLITS ON IMAGENET-LT

Method	Head	Middle	Tail	Overall
Class size	$n_j > 100$	$20 < n_j \leq 100$	$n_j \leq 20$	-
CE	*64.91*	38.10	11.28	44.51
CosFace	64.48	39.26	11.55	44.95
ArcFace	**64.86**	38.07	11.75	44.54
LDAM-DRW	58.63	48.95	30.37	49.96
OLTR	61.93	44.68	19.98	47.72
Decoupling	63.71	43.01	20.55	47.70
MisLAS	62.43	49.31	33.89	52.11
GCL-E	63.78	**52.62**	**38.70**	**54.84**
GCL-A	62.72	*53.26*	*40.95*	*55.12*

VI. CONCLUSION

In this article, we have proposed to use Gaussian form perturbance to augment the features for long-tailed classification. Eventually, we have derived two GCL forms, which are simple but effective. Both of these two forms make tail classes have larger perturbance amplitudes on their corresponding class anchors, which can expand the spatial distribution of tail class embeddings. Furthermore, we have analyzed the rationale of the proposed method from different perspectives, which provides insights into how to obtain a representative and balanced-distributed embedding. After obtaining a balanced distributed embedding space, the classifier bias can be effectively addressed by simply retraining it with class-balanced sampling. Comprehensive experiments on various benchmark datasets have demonstrated that the proposed GCL in both forms achieves significant performance gains compared with the state-of-the-art methods. In addition, we have also validated the properties of the proposed GCL by t-SNE visualization and the performance on different scales of classes.

DATA AVAILABILITY STATEMENT

Source code is available at https://github.com/Keke921/GCLLoss.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[2] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "NormFace: L_2 hypersphere embedding for face verification," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 1041–1049.
[3] J. Tian et al., "Synergetic focal loss for imbalanced classification in federated XGBoost," *IEEE Trans. Artif. Intell.*, vol. 5, no. 2, pp. 647–660, Feb. 2024.
[4] M. G. Kendall et al., *The Advanced Theory of Statistics*. London, U.K.: Charles Griffin and Co., Ltd., 1948.
[5] Y. Zhang, C. Shi, X. Li, Z. Zhang, and X. Hu, "Multi-component similarity graphs for cross-network node classification," *IEEE Trans. Artif. Intell.*, vol. 5, no. 3, pp. 1411–1424, Mar. 2024.
[6] S. Das, S. S. Mullick, and I. Zelinka, "On supervised class-imbalanced learning: An updated perspective and some key challenges," *IEEE Trans. Artif. Intell.*, vol. 3, no. 6, pp. 973–993, Dec. 2022.
[7] A. Basu, S. Das, S. S. Mullick, and S. Das, "Do preprocessing and class imbalance matter to the deep image classifiers for COVID-19 detection?

An explainable analysis," *IEEE Trans. Artif. Intell.*, vol. 4, no. 2, pp. 229–241, Apr. 2023.

[8] K. Li et al., "CODA: A real-world road corner case dataset for object detection in autonomous driving," in *Proc. Eur. Conf. Comput. Vis.*, New York, NY, USA: Springer-Verlag, 2022, pp. 406–423.

[9] D.-W. Li and H. Huang, "Few-shot class-incremental learning via compact and separable features for fine-grained vehicle recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21418–21429, Nov. 2022.

[10] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, "Knowledge distillation with the reused teacher classifier," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11933–11942.

[11] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9719–9728.

[12] B. Kang et al., "Decoupling representation and classifier for long-tailed recognition," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16.

[13] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1567–1578.

[14] T. Wang et al., "The devil is in classification: A simple framework for long-tail instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 728–744.

[15] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun, "Distribution alignment: A unified framework for long-tail visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2361–2370.

[16] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[17] L. B. Smith and L. K. Slone, "A developmental approach to machine learning?" *Frontiers Psychol.*, vol. 8, 2017, Art. no. 2124.

[18] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9268–9277.

[19] Y. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 7029–7039.

[20] M. Li, Y. Cheung, and Y. Lu, "Long-tailed visual recognition via Gaussian clouded logit adjustment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6929–6938.

[21] J. Wang, T. Lukasiewicz, X. Hu, J. Cai, and X. Zhenghua, "RSG: A simple but effective module for learning imbalanced datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3784–3793.

[22] J. Kim, J. Jeong, and J. Shin, "M2m: Imbalanced classification via major-to-minor translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13896–13905.

[23] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5704–5713.

[24] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, "Deep representation learning on long-tailed data: A learnable embedding augmentation perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2967–2976.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[26] P. Chu, X. Bian, S. Liu, and H. Ling, "Feature space augmentation for long-tailed data," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12374, 2020, pp. 694–710.

[27] Y. Zhang, X.-S. Wei, B. Zhou, and J. Wu, "Bag of tricks for long-tailed visual recognition with deep convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3447–3455.

[28] Z. Zhong, J. Cui, S. Liu, and J. Jia, "Improving calibration for long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16489–16498.

[29] V. Verma et al., "Manifold mixup: Better representations by interpolating hidden states," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.

[30] P. Wang, K. Han, X. Wei, L. Zhang, and L. Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 943–952.

[31] Y. Li et al., "Overcoming classifier imbalance for long-tail object detection with balanced group softmax," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10991–11000.

[32] L. Xiang, G. Ding, and J. Han, "Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 247–263.

[33] J. Cai, Y. Wang, and J.-N. Hwang, "ACE: Ally complementary experts for solving long-tailed recognition in one-shot," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 112–121.

[34] J. Cui, S. Liu, Z. Tian, Z. Zhong, and J. Jia, "ResLT: Residual learning for long-tailed recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3695–3706, Mar. 2023.

[35] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. X. Yu, "Long-tailed recognition by routing diverse distribution-aware experts," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–16.

[36] B. Li, Z. Han, H. Li, H. Fu, and C. Zhang, "Trustworthy long-tailed classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6970–6979.

[37] Y. Jin, M. Li, Y. Lu, Y.-m. Cheung, and H. Wang, "Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23695–23704.

[38] J. Li, Z. Tan, J. Wan, Z. Lei, and G. Guo, "Nested collaborative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6949–6958.

[39] J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia, "Parametric contrastive learning," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 715–724.

[40] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4331–4340.

[41] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[42] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[43] J. Tan et al., "Equalization loss for long-tailed object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11659–11668.

[44] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1885–1894.

[45] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–27.

[46] Y. Hong, S. Han, K. Choi, S. Seo, B. Kim, and B. Chang, "Disentangling label distribution for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6626–6636.

[47] D. Cao, X. Zhu, X. Huang, J. Guo, and Z. Lei, "Domain balancing: Face recognition on long-tailed domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5670–5678.

[48] M. Li, Y.-m. Cheung, and Z. Hu, "Key point sensitive loss for long-tailed visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4812–4825, Apr. 2023.

[49] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.

[50] H. Wang et al., "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5265–5274.

[51] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," 2017, Art. no. 1703.09507.

[52] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *Statist. Probability Lett.*, vol. 25, no. 7, pp. 926–930, 2018.

[53] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise. Unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Trans. Cryptographic Hardware Embedded Syst.*, vol. 2019, no. 3, pp. 148–179, 2019.

[54] C. Rasmussen, "The infinite Gaussian mixture model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 554–560.

[55] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to Probability and Statistics*, 14th ed. Boston, MA, USA: Cengage Learning, Richard Stratton, 2013.

[56] J. Wang et al., "Seesaw loss for long-tailed instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Virtual Event, 2021, pp. 9695–9704.

[57] T. Wang, Y. Zhu, C. Zhao, W. Zeng, J. Wang, and M. Tang, "Adaptive class suppression loss for long-tail object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Virtual Event, 2021, pp. 3103–3112.

[58] M. Ochal, M. Patacchiola, J. Vazquez, A. Storkey, and S. Wang, "Few-shot learning with class imbalance," *IEEE Trans. Artif. Intell.*, vol. 4, no. 5, pp. 1348–1358, Oct. 2023.
[59] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4021–4030.
[60] W. Zhang, Y. Chen, W. Yang, G. Wang, J.-H. Xue, and Q. Liao, "Class-variant margin normalized softmax loss for deep face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4742–4747, 2021.
[61] C. Wei and T. Ma, "Improved sample complexities for deep neural networks and robust classification via an all-layer margin," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–37.
[62] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," University of Tront, *Tech. Rep*., 2009. [Online]. Available: https://web.cs.toronto.edu/
[63] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[64] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[65] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2537–2546.
[66] G. Van Horn et al., "The iNaturalist species classification and detection dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8769–8778.
[67] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, pp. 1–13, 2018.
[68] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.
[69] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1513–1524.
[70] T. Li et al., "Targeted supervised contrastive learning for long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6918–6928.
[71] J. Liu, W. Li, and Y. Sun, "Memory-based Jitter: Improving visual recognition on long-tailed data with diversity in memory," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1720–1728.
[72] M. Li, Y.-M. Cheung, and J. Jiang, "Feature-balanced loss for long-tailed visual recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2022, pp. 1–6.
[73] Y.-J. Li, F.-E. Yang, Y.-C. Liu, Y.-Y. Yeh, X. Du, and Y.-C. F. Wang, "Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshop*, 2018, pp. 172–178.
[74] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "VRSTC: Occlusion-free video person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7183–7192.
[75] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021.
[76] D. Mahajan et al., "Exploring the limits of weakly supervised pretraining," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11206, 2018, pp. 185–201.

Mengke Li (Member, IEEE) received the B.S. degree in communication engineering from Southwest University, Chongqing, China, in 2015, the M.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2018, and the Ph.D. degree in computer science from the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China, under the supervision of Prof. Yiu-ming Cheung, in 2022.

Currently, she is an Associate Researcher with Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Guangdong, China. Her current research interests include imbalanced data learning, long-tail learning, and pattern recognition.

Yiu-ming Cheung (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering from The Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently a Chair Professor in artificial intelligence with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China. His research interests include machine learning, visual computing, data science, pattern recognition, multiobjective optimization, and information security.

Dr. Cheung is currently the Editor-in-Chief of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. Also, he serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (2014–2020), *Pattern Recognition*, *Pattern Recognition Letters*, and *Neurocomputing*, to name a few. He is a Fellow of AAAS, IET, BCS, and AAIA. For more information, see https://www.comp.hkbu.edu.hk/ymc.

Yang Lu (Member, IEEE) received the B.Sc. and M.Sc. degrees in software engineering from the University of Macau, Macau, China, in 2012 and 2014, respectively, and the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, China, in 2019.

He is currently an Assistant Professor with the Department of Computer Science, School of Informatics, Xiamen University, Xiamen, China. His current research interests include deep learning, federated learning, long-tail learning, and meta-learning.

Zhikai Hu (Student Member, IEEE) received the B.S. degree in computer science from China Jiliang University, Hangzhou, China, in 2015, and the M.S. degree in computer science from Huaqiao University, Xiamen, China, in 2019. He is currently working toward the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China, under the supervision of Prof. Yiu-ming Cheung.

His current research interests include information retrieval, pattern recognition, and data mining.

Weichao Lan (Student Member, IEEE) received the B.S. degree in electronics and information engineering from Sichuan University, Chengdu, China, in 2019, and the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong SAR, China, under the supervision of Prof. Yiu-ming Cheung, in 2024.

Her current research interests include network compression and acceleration, and lightweight models.

Hui Huang (Senior Member, IEEE) received the Ph.D. degree in applied mathematics from The University of British Columbia, British Columbia, Canada, in 2008.

She is a Distinguished Professor with Shenzhen University, serving as the Dean of College of Computer Science and Software Engineering while also directing the Visual Computing Research Center. Her current research interests include computer graphics, computer vision, and visual analytics, focusing on geometry, points, shapes and images. She is currently on the editorial board of ACM TOG and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS.