Class-Level Logit Perturbation

Mengyang Li, Fengguang Su, Ou Wu, and Ji Zhang, *Senior Member, IEEE*

***Abstract*— Features, logits, and labels are the three primary data when a sample passes through a deep neural network (DNN). Feature perturbation and label perturbation receive increasing attention in recent years. They have been proven to be useful in various deep learning approaches. For example, (adversarial) feature perturbation can improve the robustness or even generalization capability of learned models. However, limited studies have explicitly explored for the perturbation of logit vectors. This work discusses several existing methods related to class-level logit perturbation. A unified viewpoint between regular/irregular data augmentation and loss variations incurred by logit perturbation is established. A theoretical analysis is provided to illuminate why class-level logit perturbation is useful. Accordingly, new methodologies are proposed to explicitly learn to perturb logits for both the single-label and multilabel classification tasks. Meta-learning is also leveraged to determine the regular or irregular augmentation for each class. Extensive experiments on benchmark image classification datasets and their long-tail versions indicated the competitive performance of our learning method. As it only perturbs on logit, it can be used as a plug-in to fuse with any existing classification algorithms. All the codes are available at https://github.com/limengyang1992/lpl.**

***Index Terms*— Adversarial training, data augmentation, long-tail classification, multilabel classification.**

I. Introduction

THERE are several main paradigms (which may overlap) among numerous deep learning studies, including new network architecture, new loss, new data perturbation scheme, and new learning strategy (e.g., weighting). Training data perturbation mainly refers to feature and label perturbations.

Many data augmentation tricks can be viewed as feature perturbation methods when the input is the raw feature (i.e., raw samples). For example, cropped or rotated images can be seen as the perturbed samples of the raw images in computer vision; sentences with modified words can also be seen as the perturbed texts in text classification. Luo et al. [1] proposed a powerful yet efficient approach to learn effective representations for dynamically weighted directed network. The linear bias vectors in their approach actually belong to feature perturbation. Another well-known feature perturbation technique is about the generation of adversarial training samples [2], which attracts great attention in various AI applications especially in computer vision [3] and natural language processing [4]. Adversarial samples are those that can fool the learned models. They can be obtained by solving the following objective function:

$$\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \arg\max_{\|\boldsymbol{\delta}\| \le \epsilon} l(f(\boldsymbol{x} + \boldsymbol{\delta}), \boldsymbol{y}) \tag{1}$$

where $\boldsymbol{x}$ is the input or the hidden feature; $\boldsymbol{\delta}$ is the perturbation term; ϵ is the perturbation bound; $\boldsymbol{y}$ is the one-hot label; and $\boldsymbol{x}_{\text{adv}}$ is the generated adversarial sample. A number of methods have been proposed to optimize (1) [2], [5]. Adversarial samples can be used to train more robust models.

In label perturbation, the labels are modified or corrected to avoid overfitting and noises. For example, a popular yet simple training trick, label smoothing [6], generates a new label for each sample according to $\boldsymbol{y}' = \boldsymbol{y} + \lambda((\boldsymbol{I}/C) - \boldsymbol{y})$, where $\boldsymbol{y}$ is the one-hot vector label; C is the number of categories; $\boldsymbol{I}$ is a vector with all the elements equaling to 1; $((\boldsymbol{I}/C) - \boldsymbol{y})$ is the perturbation term; and λ is a hyperparameter. Other methods such as boostrapping loss [7], label correction [8], [9], and meta-label corrector [10] can be seen as a type of label perturbation. Mixup [11] can be attributed to the combination of feature and label perturbation.

Logit vectors (or logits) are the outputs of the final feature encoding layer in most deep neural networks (DNNs). Although logits are nearly indispensable in the DNN data pipeline, only several learning methods in data augmentation and long-tail classification directly (without optimization) or implicitly use class-level logit perturbation. Li et al. [12] exerted a classwise Gaussian augmentation in DNN training and derived a logit-perturbation-based training loss. Based on the loss analysis of five representative methods, the loss variations incurred by logit perturbation are highly related to the purpose of regular/irregular augmentation[1] on training data. A theoretical analysis is conducted to reveal the connections among loss variations, performance improvements, and class-level logit perturbation. Accordingly, new methodologies are proposed to learn a class-level logit perturbation (LPL) for single-label and multilabel learning tasks, respectively, in this study. Meta-learning is also used to judge whether a sample should be regularly or irregularly augmented. Extensive experiments are run on benchmark datasets to show the competitiveness of our methodologies.

Parts of the results in this article were published originally in its conference version [13]. In our conference version, several classical methods are rediscussed in terms of logit perturbation and regular/irregular augmentation. A new method is proposed to learn to perturb logits which can be used in implicit data augmentation and long-tail classification contexts for the single-label classification tasks. The experimental results show that our method outperforms the existing state-of-the-art

Manuscript received 12 August 2022; revised 18 March 2023; accepted 22 April 2023. This work was supported in part by NSFC under Grant 62076178, in part by ZJF under Grant 2019KB0AB03, and in part by TJF under Grant 22ZYYYJC00020 and Grant 19ZXAZNGX00050. *(Mengyang Li and Fengguang Su contributed equally to this work.) (Corresponding author: Ou Wu.)*

Mengyang Li, Fengguang Su, and Ou Wu are with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: limengyang@tju.edu.cn; fengguangsu@tju.edu.cn; wuou@tju.edu.cn).

Ji Zhang is with the University of Southern Queensland, Toowoomba, QLD 4350, Australia (e-mail: Ji.Zhang@usq.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3273355.

Digital Object Identifier 10.1109/TNNLS.2023.3273355

[1] In this study, irregular augmentation denotes the augmentation which aims to reduce the (relative) performances of some categories. Accordingly, existing augmentation methods are regular.

methods related to logit perturbation in both the contexts. This article extends our earlier work in several important aspects.

1) We conduct a theoretical analysis for the roles of logit-perturbation-based explicit irregular and regular augmentations in learning for binary classification tasks. Two typical scenarios, namely, class imbalance and variance imbalance, are considered in our analysis.
2) We extend our LPL to the multilabel classification, which contains both class and variance imbalances, and empirically validate its effectiveness on multilabel benchmarks. Meta-learning is also used to infer the perturbation directions of each category.
3) Extensive experiments on large-scale long-tail datasets such as iNaturalist are performed. Our method LPL still achieves competitive results.

II. Related Work

A. Data Augmentation

Data augmentation is prevalent in deep learning. In its early stage, heuristic operations are used on raw samples, such as image flip, image rotation, and word replacement in sentences. Recently, advanced tricks are investigated, such as mixup [11] and cutout [14]. Hu et al. [15] introduced reinforcement learning to automatically augment data. Although these methods yield good results, the training complexity is obviously increased when new training data are involved. Two recent implicit data augmentation methods, implicitly semantic data augmentation (ISDA) [16] and MetaSAug [17], demonstrate competitive performance. They actually belong to the logit perturbation methods. The advantage of both ISDA and MetaSAug is that no additional training data are generated, and thus the training complexity is not significantly increased. Nevertheless, ISDA fails in imbalance learning, and the training complexity of MetaSAug is still high.

In addition to the conventional data augmentation techniques, researchers have attempted to apply feature augmentation in several challenging learning problems. For example, Wu et al. [18] proposed a novel strategy to augment data layer-by-layer to perform highly accurate representation for high-dimensional and sparse data processing. Luo et al. [19] conducted a pioneering study to introduce the feature augmentation into the construction of nonnegative latent factor model, which can effectively improve the model diversity.

In this study, the existing data augmentation is called regular data augmentation. Irregular data augmentation, proposed in this study, may be helpful when we intend to restrain the (relative) performance of certain categories (e.g., to keep fairness in some tasks).

B. Long-Tail Classification

Real data usually conform to a skewed or even a long-tail distribution. In long-tail classification, the proportions of tail samples are considerably small compared with those of head samples. Long-tail classification may be divided into two main strategies. The first strategy is to design new network architectures. Zhou et al. [20] designed a bilateral-branch network (BBN) to learn the representations of head and tail samples. BBN can balance the feature learning and the performance on tail categories. Nevertheless, the whole network becomes large. The second strategy is to modify the training loss. In this way, the weighting scheme [21] is the most common practice. Relatively larger weights are exerted on the losses of the tail samples. Some studies adopt regularization [22] for the training loss. Hu et al. [23] leveraged a new support vector machine with two soft margins to alleviate the negative impact of class imbalance. Furthermore, their algorithm can also deal with noisy samples effectively. Besides weighting and regularizing, some recent studies modify the logits or augment the features to change the whole loss, such as logit adjustment (LA) [24] and feature augmentation [19]. Zhao et al. [25] alleviated the negative influence of sparsity of certain samples using graph embedding. These new methods achieve higher accuracy in many benchmark data corpora compared with conventional techniques.

C. Multilabel Classification

Real data usually also contain multiple objectives. Unlike the two single-label classification tasks mentioned above, there are two main challenges in the multilabel classification tasks, namely, the co-occurrence of labels and the dominance of negative samples [26], [27]. Wei and Li [28] investigated the impact of labels on evaluation metrics for large-scale multilabel learning and proposed to restrain labels that have less impact on performance to speed up prediction and reduce model complexity. Wu et al. [26] perturbed logits to emphasize the positive samples of tail categories to prevent class-specific overfitting. Some other studies focus on applying auxiliary information to aid multilabel classification [29], [30]. For example, Yu et al. [31] elaborated a novel manner to use user click information to aid fine-grained image recognition. The advantage of leveraging this type of information is that hierarchical semantic relationship among words can be automatically and effectively constructed. Extensive experiments demonstrate the effectiveness and good extension capability of their proposed methodology. In the multilabel classification task, weighting scheme [32] is also a typically used method.

D. Adversarial Training

Adversarial training is an important way to enhance the robustness of neural networks [33], [34]. The most important step in adversarial training is to generate adversarial training examples in (1), which can be used to improve the robustness of neural networks. Madry et al. [2] propose projected gradient descent (PGD) to compute the adversarial training samples. There are also some studies that attempt to leverage regularization techniques to replace adversarial training and avoid generating real adversarial samples [35].

III. Methodology

This section first discusses several typical learning methods related to logit perturbation. Theoretical analysis is conducted for logit perturbation. New algorithms are then proposed for both single-label and multilabel learning tasks. Finally, meta-learning is used to infer the perturbation direction.

The notations and symbols are defined as follows. Let $S = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ be a corpus of N training samples, where $\boldsymbol{x}_i$ is the input feature and $\boldsymbol{y}_i$ is the label. In single-label classification, $\boldsymbol{y}_i$ is a one-hot vector. In multilabel classification, $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,C}] \in \{0, 1\}^C$. Let C be the number of categories and $\pi_c = N_c/N$ be the proportion of the samples, where N_c is the number of the samples that contain the cth

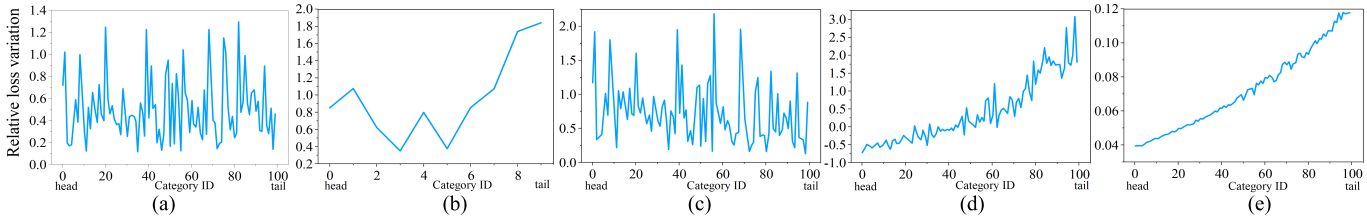


Fig. 1. Relative loss variations $((l' - l)/l)$ in the three methods on different categories on different datasets. (a) and (b) Relative loss variation in ISDA on CIFAR100 and CIFAR10, respectively. (c)–(e) Relative loss variation in ISDA, LA, and LDAM on CIFAR100-LT with imbalance ratio 100:1, respectively. (a) ISDA on 100 balanced categories. (b) ISDA on ten balanced categories. (c) ISDA on 100 imbalanced categories. (d) LA on 100 imbalanced categories. (e) LDAM on 100 imbalanced categories.

category in S. Without loss of generality, we assume that $\pi_1 > \cdots > \pi_c > \cdots > \pi_C$.

Following Menon et al. [24] and Wu et al. [26], we determine the head and tail categories based on N_c. The larger N_c means that c is the head category index, and the smaller N_c means that c is the tail category index. Following Guo and Wang [27], if $y_{i,c} = 1$, $\boldsymbol{x}_i$ is the positive sample of category c; otherwise, $\boldsymbol{x}_i$ is the negative sample of category c. Let $\boldsymbol{u}_i$ be the logit vector of $\boldsymbol{x}_i$ which can be obtained by $\boldsymbol{u}_i = f(\boldsymbol{x}_i, \boldsymbol{W})$, where $f(\cdot, \cdot)$ is the DNN with parameter $\boldsymbol{W}$. Let $\boldsymbol{\delta}_i$ be the perturbation term of $\boldsymbol{x}_i$. Let $\mathcal{L}$ be the entire training loss and ℓ_i be the loss of $\boldsymbol{x}_i$. The standard cross-entropy (CE) loss is used throughout the study.

A. Logit Perturbation in Existing Methods

To enlighten our analysis, the logit-perturbation-based loss can be written in the following form:

$$\mathcal{L} = \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i)} \ell(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_i, \boldsymbol{y}_i) \tag{2}$$

where $\tilde{\boldsymbol{\delta}}_i$ is the logit perturbation for $\boldsymbol{x}_i$. In classwise logit perturbation, $\tilde{\boldsymbol{\delta}}_i$s of samples in the same class are identical.

1) Logit Adjustment [24]: This method is designed for single-label long-tail classification [36]. The perturbation term $\boldsymbol{\delta}_i$ is as follows:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}} = \lambda[\log\pi_1, \ldots, \log\pi_c, \ldots, \log\pi_C]^T \tag{3}$$

where $\tilde{\boldsymbol{\delta}}$ is the corpus-level vector[2]; $\boldsymbol{\delta}_i$ is the sample-level vector; thus, $\boldsymbol{\delta}_i$s for all the samples are identical; and λ is a hyperparameter.

Previously, we assumed that $\pi_1 > \cdots > \pi_c > \cdots > \pi_C$; hence, the losses of the samples in the first category (head) are decreased, while those of the samples in the last category (tail) are increased. The variations in the losses of the remaining categories depend on the concrete loss of each sample.

2) Implicitly Semantic Data Augmentation [16]: ISDA is an implicit data augmentation method for single-label classification. Given a sample $\boldsymbol{x}_i$, ISDA modifies the CE loss with the following logit perturbation vector:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}}_k = \frac{\lambda}{2}\begin{bmatrix} (\boldsymbol{w}_1 - \boldsymbol{w}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{w}_1 - \boldsymbol{w}_k) \\ \vdots \\ (\boldsymbol{w}_C - \boldsymbol{w}_k)^T \boldsymbol{\Sigma}_k (\boldsymbol{w}_C - \boldsymbol{w}_k) \end{bmatrix} \tag{4}$$

where $\boldsymbol{\Sigma}_k$ is the covariance matrix for the kth category, and $\boldsymbol{w}_c$ is the network parameter for the logit vectors. Each element of $\boldsymbol{\delta}_i$ is nonnegative. Therefore, the new loss of each category is larger than the loss from the standard CE loss.

[2]Corpus level is viewed as a special kind of class level in this study.

3) LDAM [37]: This method is designed for single-label long-tail classification. Its perturbation term $\boldsymbol{\delta}_i$ is as follows:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}}_k = \lambda\left[0, \ldots, -C(\pi_k)^{-\frac{1}{4}}, \ldots, 0\right]^T \tag{5}$$

which is also a category-level vector. The losses for all the categories are increased in LDAM.

4) Negative-Tolerant Regularization [26]: In this method, a multilabel classification task is first decomposed into C independent binary classification tasks. Negative-tolerant regularization (NTR) defines the following perturbation vector $(\boldsymbol{\delta}_i)$:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}} = -\psi\left[\log\left(\frac{N}{N_1} - 1\right), \ldots, \left(\frac{N}{N_C} - 1\right)\right]^T \tag{6}$$

where ψ is nonnegative in the experiments conducted by Wu et al. [26]. $\boldsymbol{\delta}_i$ is a corpus-level term vector. If $N < 2N_c$, the loss will be reduced when $y_{i,c} = 1$, and the loss will be increased when $y_{i,c} = 0$. When $N > 2N_c$, it is opposite.

5) Logit Compensation [27]: Logit compensation (LC) assumes that logits conform to a normal distribution. For the positive samples, the perturbation term δ_i is as follows:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}} = \left[\mu_1^p, \mu_2^p, \ldots, \mu_C^p\right]. \tag{7}$$

For the negative samples, the perturbation term $\boldsymbol{\delta}_i$ is as follows:

$$\boldsymbol{\delta}_i = \tilde{\boldsymbol{\delta}} = \left[\mu_1^n, \mu_2^n, \ldots, \mu_C^n\right] \tag{8}$$

where μ_c^p and μ_c^n $(c \in \{1, \ldots, C\})$ are the mean of the positive and negative samples that can be learned, respectively. Both the perturbation items are corpus-level vectors. According to [27], the magnitude of loss reduction for tail categories' positive samples is smaller than that for head categories'.

Logit perturbations result in the loss variations. Fig. 1 shows the statistics for the relative loss variations incurred by ISDA, LA, and LDAM for each category on a balanced dataset (CIFAR100 [38]) and two long-tail sets (CIFAR10-LT [39] and CIFAR100-LT [39]) which are introduced in Section IV. The loss variations of all the categories are positive using ISDA. ISDA achieves the worst results on CIFAR100-LT [39] (shown in the experimental parts), indicating that the nontail-priority augmentation in long-tail problems is ineffective (ISDA achieves relatively better results on CIFAR10-LT [39]). Only the curves on CIFAR100-LT are shown for LA and LDAM because similar trends can be observed on CIFAR10-LT. The loss variations in head categories are negative, and those of tail are positive using LA. All the variations are positive, yet there is an obvious increasing trend using LDAM. Fig. 2 shows the statistics for the relative loss variations incurred by NTR and LC in multilabel classification. The dataset COCO-MLT [26] is used. The relative loss variations in positive samples and negative samples are counted separately.

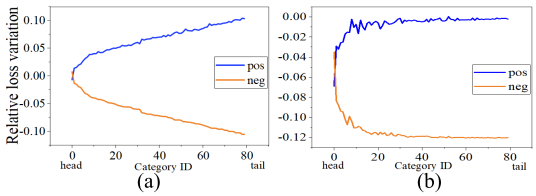


Fig. 2. Relative loss variations $((l' - l)/l)$ in the two methods on different categories on COCO-MLT. "pos" means the relative loss variations of positive samples. "neg" means the relative loss variations of negative samples. (a) NTR on COCO-MLT. (b) LC on COCO-MLT.

In NTR, for positive samples, the loss variations in head categories are less than 0, and those of tail are greater than 0. However, the situation is opposite for negative samples. LC and NTR have a similar trend of the relative loss variation, but the relative loss variation in LC is less than 0.

We propose two conjectures based on the above observations and from a unified data augmentation viewpoint.

1) If one aims to regularly augment the samples in a category, the training loss of this category should be increased after logit perturbation. The larger the loss increment, the greater the augmentation. Consequently, the performance of this category will (relatively) increase.
2) If one aims to irregularly augment the samples in a category, then the loss of this category should be reduced after logit perturbation. The larger the loss decrement, the greater the irregular augmentation. The performance of this category will (relatively) decrease.

The above two conjectures are empirically supported by the aforementioned five methods. For single-label classification, to handle a long-tail problem, LA should regularly augment tail samples and irregularly augment head ones. Hence, the losses of tail samples are increased, and those of heads are decreased. ISDA aims to regularly augment samples in all the categories; thus, the losses for all the categories are increased. LDAM aims to regularly augment tail samples more than head ones. Hence, the increments of tail categories are larger than those of the head. For the multilabel classification task, the positive and negative ones need to be considered separately. For positive samples, NTR regularly augments the tail categories and irregularly augments the head categories. For negative ones, the condition is opposite. Therefore, the losses of tails are increased, whereas those of heads are decreased. LC aims to irregularly augment all the categories. For positive samples, the reductions of head categories are larger than those of the tail. For negative ones, the situation is opposite.

B. Theoretical Analysis for Logit Perturbation

To theoretically verify the reasonableness of the two conjectures, simple binary classification tasks are used to quantitatively investigate the relationship among loss variations, performance improvement, and logit perturbation. First, a typical binary classification scenario, namely, learning on imbalance training data, is considered. A large number of studies on logit perturbation concerns imbalance learning. The binary classification setting established by Xu et al. [40] is followed. The data from each of the two classes $\mathcal{Y} = \{-1, +1\}$ follow two Gaussian distributions, which are centered on $\boldsymbol{\theta} = [\eta, \ldots, \eta]$ (d-dimensional vector and $\eta > 0$) and $-\boldsymbol{\theta}$, respectively. The data follow

$$y, \overset{u.a.r}{\sim} \{-1, +1\} \tag{9}$$

$$\boldsymbol{x}, \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma_+^2 \boldsymbol{I}\right), & \text{if } y = +1 \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma_-^2 \boldsymbol{I}\right), & \text{if } y = -1. \end{cases} \tag{10}$$

For a classifier f, the overall standard error is defined as $\mathcal{R}(f) = \Pr \cdot (f(\boldsymbol{x}) \neq y)$. We use $\mathcal{R}(f; y)$ to denote the standard error conditional on a specific class y. The class "$+1$" is harder because an optimal linear classifier will give a larger error for the class "$+1$" than that for the class "-1" when $\sigma_+^2 > \sigma_-^2$ [40]. Two types of class-level logit perturbation are considered in our theoretical analysis. Let ϵ_c be the perturbation bound. The first type of perturbation is defined as follows:

$$\tilde{\delta}_c^* = \arg \max_{||\tilde{\delta}_c|| < \epsilon_c} \mathrm{E}_{(\boldsymbol{x}, y): y = c}[\ell(u + \tilde{\delta}_c, c)]. \tag{11}$$

The second type is defined as follows:

$$\tilde{\delta}_c^* = \arg \min_{||\tilde{\delta}_c|| < \epsilon_c} \mathrm{E}_{(\boldsymbol{x}, y): y = c}[\ell(u + \tilde{\delta}_c, c)] \tag{12}$$

where $u = \boldsymbol{w}^T \boldsymbol{x} + b$. The first type implements regular augmentation, while the second type implements irregular.

Assuming that the perturbation bounds between the two classes satisfy that $\epsilon_+ = \rho_+ \cdot \epsilon$ and $\epsilon_- = \rho_- \cdot \epsilon$. Now, the variances of the data distributions in (10) for the two classes are assumed to be equal, i.e., $\sigma_+ = \sigma_-$. Nevertheless, the prior probabilities of the two classes $P(y = +1)$ (P_+) and $P(y = -1)$ (P_-) are assumed to be different. Without loss of generality, we assume $P_+ : P_- = 1 : \Gamma$ and $\Gamma > 1$. That is, class imbalance exists, and the class $+1$ and the class -1 are the tail and head classes, respectively. We have the following theorem.

Theorem 1: For the above-mentioned binary classification task, the logit perturbation bounds of classes "$+1$" and "-1" are assumed to be $\rho_+ \cdot \epsilon$ ($0 \leq \rho_+ \cdot \epsilon < \eta$) and ϵ ($\rho_- = 1$), respectively. Only the first perturbation type is used. The optimal linear classifier f_{opt} that minimizes the average classification error is

$$f_{\text{opt}} = \arg \min_f \Pr \cdot \left(\mathbb{S}\left(u + \tilde{\delta}_c^*\right) \neq y\right) \tag{13}$$

where $u = f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$; $\mathbb{S}(\cdot)$ is the signum function [if $a \geq 0$, then $\mathbb{S}(a) = 1$; else, $\mathbb{S}(a) = -1$]. It has the intraclass standard error for the two classes

$$\begin{aligned} \mathcal{R}(f_{\text{rob}}, -1) &= \Pr \cdot \left\{ \mathcal{N}(0, 1) < \frac{A}{2} + \frac{\log \Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma} \right\} \\ \mathcal{R}(f_{\text{rob}}, +1) &= \Pr \cdot \left\{ \mathcal{N}(0, 1) < \frac{A}{2} - \frac{\log \Gamma}{A} - \frac{\epsilon \rho_+}{\sqrt{d}\sigma} \right\} \end{aligned} \tag{14}$$

where $A = ((\epsilon - 2d\eta + \epsilon\rho_+)/\sqrt{d}\sigma)$.

The proof is attached in the Appendix. Theorem 1 indicates that the logit perturbation parameterized by ϵ and ρ_+ influences performance of both the classes. We then show how the classification errors of the two classes change as ρ_+ increases.

Corollary 1: For the binary classification task investigated in Theorem 1, when $\Gamma < e^{(((2d-1)\eta - \epsilon)^2)/2d\sigma^2}$, as ρ_+ increases, the logit perturbations on Theorem 1 will decrease the error for class "$+1$" and increase the error for class "-1."

The proof is attached in the Appendix. Corollary 1 indicates that a larger scope of the first type of logit perturbation on a class will increase the performance of the class. Note that a larger scope of the first type of logit perturbation will result in a large loss increment, and the first conjecture is supported by Corollary 1. To better illuminate Corollary 1, we plot

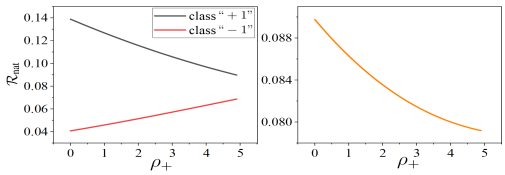


Fig. 3. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ with varied ρ_+ when the first perturbation type is used. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$.

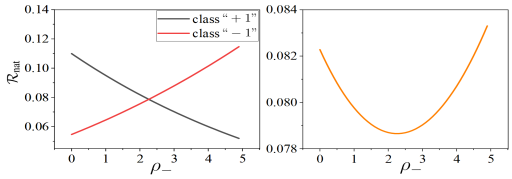


Fig. 4. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ with varied ρ_- when the two perturbation types are used. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$.

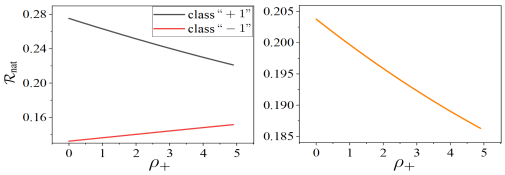


Fig. 5. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ with varied ρ_+ when the class imbalance is the primary challenge. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$.

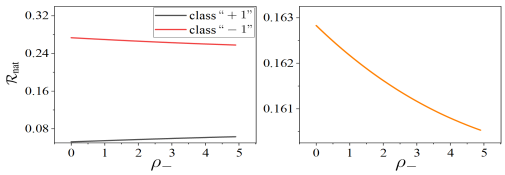


Fig. 6. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ with varied ρ_- when the variance imbalance is the primary challenge. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$.

$\mathcal{R}(f_{\text{opt}}, -1)$, $\mathcal{R}(f_{\text{opt}}, +1)$, and $\mathcal{R}(f_{\text{opt}})$ for a specific learning task. Fig. 3 shows the results when the values of Γ, d, η, ϵ, and σ are 2, 2, 1, 0.2, and 1, respectively.

Theorem 1 only considers the first type of logit perturbation. When the second type of logit perturbation is also involved, the following theorem can be obtained.

Theorem 2: For the above-mentioned binary classification task, the perturbation bounds of both the classes "+1" and "−1" are assumed to be ϵ ($\rho_+ = 1$) and $\rho_- \cdot \epsilon$ ($0 \le \rho_- \cdot \epsilon < \eta$), respectively. The first perturbation type is used for class "+1," and the second perturbation type is used for "−1." The optimal linear classifier f_{opt} that minimizes the average classification error is

$$f_{\text{opt}} = \arg\min_f \Pr\left(\mathbb{S}\left(u + \tilde{\delta}_c^*\right) \neq y\right). \tag{15}$$

It has the intraclass standard error for the two classes

$$\begin{aligned}\mathcal{R}(f_{\text{opt}}, -1) &= \Pr\cdot\left\{\mathcal{N}(0,1) < \frac{A}{2} + \frac{\log\Gamma}{A} + \frac{\epsilon\rho_-}{\sqrt{d}\sigma}\right\}\\ \mathcal{R}(f_{\text{opt}}, +1) &= \Pr\cdot\left\{\mathcal{N}(0,1) < \frac{A}{2} - \frac{\log\Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma}\right\}\end{aligned} \tag{16}$$

where $A = ((\epsilon - 2d\eta - \epsilon\rho_-)/\sqrt{d}\sigma)$.

The proof of Theorem 2 is similar to that of Theorem 1. Likewise, we have the following corollary.

Corollary 2: For the learning task investigated in Theorem 2, when $\Gamma > 1$, as ρ_- increases, the logit perturbations on Theorem 2 will increase the accuracy for class "+1" and decrease the accuracy for class "−1."

According to Corollary 2, a larger scope of the second type of logit perturbation on a class will decrease the performance of the class. Note that a larger scope of the second type of logit perturbation will result in a large loss decrement, and the second conjecture is supported. Likewise, we plot $\mathcal{R}(f_{\text{opt}}, -1)$, $\mathcal{R}(f_{\text{opt}}, +1)$, and $\mathcal{R}(f_{\text{opt}})$. Fig. 4 shows the results for the specific learning task discussed in Fig. 3. In Fig. 4, the values of Γ, d, η, ϵ, and σ are 2, 2, 1, 0.2, and 1, respectively.

Theorems 1 and 2 and Corollaries 1 and 2 concern the class-imbalance issue, i.e., $P_+ \neq P_-$. However, in many binary tasks, although the two involved classes are balanced, their corresponding performances are still unequal. To this end, a more general learning scenario is explored. The variances of the data distributions in (10) for the two classes are assumed to be unequal, i.e., $\sigma_+ \neq \sigma_-$. That is, variance imbalance exists. Without loss of generality, we assume $\sigma_+ : \sigma_- = 1 : K$, where $K > 1$. And $P_+ : P_- = 1 : \Gamma$ also holds, where $\Gamma > 1$. We have the following theorem.

Theorem 3: For the above-mentioned binary classification task, the bounds of classes "+1" and "−1" are assumed to be $\rho_+ \cdot \epsilon$ and $\rho_- \cdot \epsilon$ ($0 \le \rho_+, \rho_- < (\eta/\epsilon)$), respectively. Only the first perturbation type is used. The optimal linear classifier f_{opt} that minimizes the average classification error is

$$f_{\text{opt}} = \arg\min_f \Pr\cdot\left(\mathbb{S}\left(u + \tilde{\delta}_c^*\right) \neq y\right) \tag{17}$$

where $u = f(x) = \boldsymbol{w}^T\boldsymbol{x} + b$. It has the intraclass standard error for the two classes

$$\begin{aligned}&\mathcal{R}(f_{\text{opt}}, +1)\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < -K\sqrt{B^2 + q(K,\Gamma)} - B - \frac{\epsilon\cdot\rho_+}{\sqrt{d}\sigma}\right)\right\}\\ &\mathcal{R}(f_{\text{opt}}, -1)\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < KB + \sqrt{B^2 + q(K,\Gamma)} - \frac{\epsilon\cdot\rho_-}{K\sqrt{d}\sigma}\right\}\end{aligned} \tag{18}$$

where $B = (\epsilon\cdot\rho_+ + \epsilon\cdot\rho_- - 2d\eta)/(\sqrt{d}\sigma(K^2-1))$, and $q(K,\Gamma) = (2\log(K/\Gamma))/(K^2-1)$.

Thus, training with different logit perturbation bounds for the two classes can still influence the performance according to Theorem 3. We then show how the classification errors of the two classes change as ρ_- or ρ_+ increases.

Corollary 3: For the data distribution and logit perturbation investigated in Theorem 3, the following holds.

1) If $Ke^{(2d\eta-\epsilon)^2/(2dK^2\sigma^2)} < \Gamma < Ke^{2d\eta^2/((K^2-1)\sigma^2)}$, then $\mathcal{R}(f_{\text{opt}}, +1) > \mathcal{R}(f_{\text{opt}}, -1)$. That is, class imbalance is the primary challenge and class "+1" is harder than class "−1." Then if $\rho_- = 1$ and the first logit perturbation type is used, the error of class "+1" decreases and the error of class "−1" increases, as ρ_+ increases.
2) If $K > \Gamma$, then $\mathcal{R}(f_{\text{opt}}, +1) < \mathcal{R}(f_{\text{opt}}, -1)$. That is, variance imbalance is the primary challenge and class "−1" is harder than class "+1." If $\rho_+ = 1$ and the first logit perturbation type is used, the error of class "+1" increases and the error of class "−1" decreases, as ρ_- increases.

The first conjecture can also be justified by Corollary 3. Likewise, we plot $\mathcal{R}(f_{\text{opt}}, -1)$, $\mathcal{R}(f_{\text{opt}}, +1)$, and $\mathcal{R}(f_{\text{opt}})$. Figs. 5 and 6 show the results. As shown in Fig. 5, increasing ρ_+ can decrease the error of class "+1" and increase the error

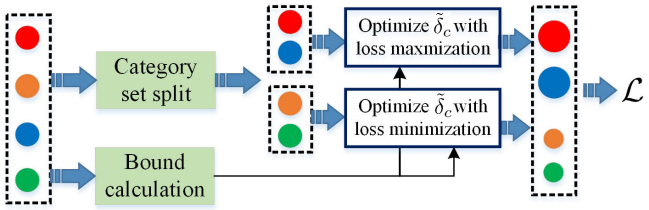


Fig. 7. Overview of the logit-perturbation-based new loss. Four solid circles denote four categories. Two categories are regularly augmented via loss maximization and the remaining two are irregularly augmented via minimization.

of class "−1." The values of K, Γ, d, η, ϵ, and σ are 3, 3.5, 2, 1, 0.1, and 1, respectively. In Fig. 6, increasing ρ_- can decrease the error of class "−1" and increase the error of class "+1." The values of K, Γ, d, η, ϵ, and σ are 2.5, 1.1, 2, 1, 0.2, and 1, respectively.

When both the types are used, we can obtain the following conclusion. When class "−1" is harder than class "+1," if the first logit perturbation type is used for class "−1" and the second logit perturbation is used for class "+1," then the error of class "−1" will decrease and the error of class "+1" will increase. Similarly, when class "+1" is harder than class "−1," if the first logit perturbation is used for class "+1" and the second logit perturbation type is used for class "−1," then the error of class "+1" will decrease and the error of class "−1" will increase. That is, the second conjecture is also justified.

C. Logit Perturbation Method (LPL) for Single-Label Learning

On the basis of our conjectures and theoretical investigation, we establish the following new training loss:

$$\mathcal{L} = \sum_{c\in\mathcal{N}_a}\sum_{\boldsymbol{x}_i\in S_c} \min_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} \ell(\text{softmax}(\boldsymbol{u}_i+\tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i) + \sum_{c\in\mathcal{P}_a}\sum_{\boldsymbol{x}_i\in S_c} \max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} \ell(\text{softmax}(\boldsymbol{u}_i+\tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i) \quad (19)$$

where ϵ_c is the perturbation bound related to the extent of augmentation; $\mathcal{N}_a$ is the index set of categories which should be irregularly augmented; $\mathcal{P}_a$ is the index set of categories which should be regularly augmented; and S_c is the set of samples in the cth category. The loss maximization for the $\mathcal{P}_a$ categories is actually the category-level adversarial learning on the logits; the loss minimization for the $\mathcal{N}_a$ categories is the opposite. Fig. 7 illustrates the calculation of the logit-perturbation-based new loss in (19).

The split of the category set (i.e., $\mathcal{N}_a$ and $\mathcal{P}_a$) and the definition (calculation) of ϵ_c are crucial for the learning with (19). Category set split determines the categories that should be regularly or irregularly augmented. Meanwhile, the value of ϵ_c determines the augmentation extent.

1) Category Set Split: The split depends on specific learning tasks. Two common cases are explored in this study. The first case splits categories according to their performances. In this case, (19) becomes the following compact form:

$$\mathcal{L} = \sum_c \left\{ \mathbb{S}(\tau-\bar{q}_c) \times \sum_{\boldsymbol{x}_i\in S_c} \max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} [\ell(\text{softmax}(\boldsymbol{u}_i+\tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i)\mathbb{S}(\tau-\bar{q}_c)] \right\} \quad (20)$$

Algorithm 1 PGD-Like Optimization

Input: The logit vectors $(\boldsymbol{u}_i)$ for the cth category in the current mini-batch, ϵ_c, and α.

1: Let $\boldsymbol{u}_i^0 = \boldsymbol{u}_i$ for the input vectors;
2: Calculate K_c by $\lfloor \frac{\epsilon_c}{\alpha} \rfloor$;
3: **for** $k = 1$ to K_c **do**
4: $\tilde{\boldsymbol{\delta}}_c^{k+1} = \frac{\alpha}{N_c}\sum_{j:y_{j,c}=1}(\text{softmax}(\boldsymbol{u}_j^k) - \boldsymbol{y}_j)$ for maximization; or $\tilde{\boldsymbol{\delta}}_c^{k+1} = -\frac{\alpha}{N_c}\sum_{j:y_{j,c}=1}(\text{softmax}(\boldsymbol{u}_j^k) - \boldsymbol{y}_j)$ for minimization;
5: $\boldsymbol{u}_i^{k+1} := \boldsymbol{u}_i^k + \tilde{\boldsymbol{\delta}}_c^{k+1}$.
6: **end for**

Output: $\tilde{\boldsymbol{\delta}}_c = \boldsymbol{u}_i^{K_c} - \boldsymbol{u}_i$

Algorithm 2 Learning to Perturb Logits

Input: S, τ, max iteration T, hyperparameters for PGD-like optimization, and other conventional training hyperparameters.

1: Randomly initialize $\boldsymbol{W}$.
2: **for** $t = 1$ to T **do**
3: Sample a mini-batch from S.
4: Update τ if it is not fixed [e.g., mean$(\bar{q}_c)$ is used] and split the category set.
5: Compute ϵ_c for each category using (24) if varied bounds are used.
6: Infer $\tilde{\boldsymbol{\delta}}_c$ for each category using a PGD-like optimization method for (20) in balanced classification, (22) in long-tail classification, or (25) in multilabel classification.
7: Update the logits for each sample and the loss.
8: Update $\boldsymbol{W}$ with SGD.
9: **end for**

Output: $\boldsymbol{W}$

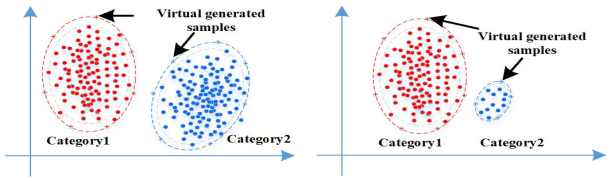


Fig. 8. Illustrative example for ISDA. Both the categories are regularly augmented (new samples are virtually generated) according to feature distributions.

where τ is a threshold, $y_{i,c} = 1$, and $\bar{q}_c$ is calculated by

$$\bar{q}_c = \frac{1}{N_c}\sum_{\boldsymbol{x}_i\in S_c} q_{i,c} = \frac{1}{N_c}\sum_{\boldsymbol{x}_i\in S_c}\frac{\exp(u_{i,c})}{\sum_{c'}\exp(u_{i,c'})}. \quad (21)$$

When $\tau = \text{mean}(\bar{q}_c) = \sum_{c=1}^{C}\bar{q}_c/C$, (20) indicates that if the performance of a category is below the mean performance, it will be regularly augmented. Meanwhile, when the performance is above the mean, it will be irregularly augmented. When $\tau > \max_c\{\bar{q}_c\}$, all the categories will be regularly augmented as in ISDA.

The second case is special for a long-tail problem, and it splits categories according to the proportion order of each

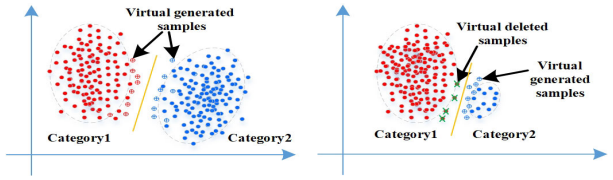


Fig. 9. Illustrative example for LPL. Samples near the classification boundary are virtually generated or deleted.

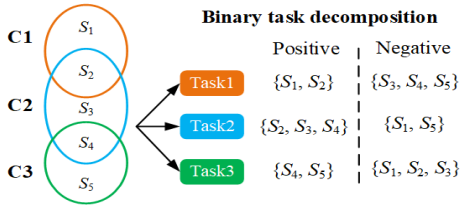


Fig. 10. Binary decomposition for a multilabel task.

category. Equation (19) becomes the following compact form:

$$\mathcal{L} = \sum_c \left\{ \mathbb{S}(c-\tau) \times \sum_{\boldsymbol{x}_i \in S_c} \max_{\|\tilde{\boldsymbol{\delta}}_c\| \le \epsilon_c} [\ell(\text{softmax}(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i)\mathbb{S}(c-\tau)] \right\} \tag{22}$$

where τ is a threshold and $y_{i,c} = 1$. In (22), the tail categories locate in $\mathcal{P}_a$ and will be regularly augmented.

Equations. (20) and (22) can be solved with an optimization approach similar to PGD [2]. We propose a more specific optimization method called PGD-like optimization based on PGD. First, we have the derivative

$$\nabla_{\tilde{\boldsymbol{\delta}}_c} \ell(\text{softmax}(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i)|_{\boldsymbol{0}} = \text{softmax}(\boldsymbol{u}_i) - \boldsymbol{y}_i. \tag{23}$$

According to the derivative of the CE loss function with respect to the logit vector in (23), our PGD-like optimization method can be implemented simply. The PGD-like optimization in Algorithm 1 involves two hyperparameters: step size and number of steps. Let α be the step size, and K_c be the number of steps (#steps) for category c. The α is searched in $\{0.01, 0.02, 0.03\}$. The PGD-like optimization is detailed in Algorithm 1.

2) Bound Calculation: The category with a relatively low/high performance should be more regularly/irregularly augmented; the category closer to the tail/head should be more regularly/irregularly augmented. We define

$$\epsilon_c = \epsilon + \Delta\epsilon\left|\tau - \bar{q}_c\right|$$
$$\text{or } \epsilon_c = \begin{cases} \epsilon + \Delta\epsilon \dfrac{\bar{q}_c}{\bar{q}_1}, & c \le \tau \\ \epsilon + \Delta\epsilon \dfrac{\bar{q}_C}{\bar{q}_c}, & c > \tau. \end{cases} \tag{24}$$

In (24), the larger the difference between the performance ($\bar{q}_c$) of the current category and the threshold τ, or the larger the ratio $\bar{q}_c/\bar{q}_1$ (and $\bar{q}_C/\bar{q}_c$), the larger the bound ϵ_c. This notion is in accordance with our previous conjecture. When $\Delta\epsilon$ in (24) equals to zero, the bound is fixed. The algorithmic steps of our LPL for single-label learning are shown in Algorithm 2.

3) Comparative Analysis: We compare the perturbations in ISDA and our LPL in terms of data augmentation.

In the ISDA's rationale, new samples are (virtually instead of really) generated based on the distribution of each category. Fig. 8 shows the (virtually) generated samples by ISDA. In the right case, the regular augmentation for head category may further hurt the performance of the tail category. ISDA fails in the long-tail problem. Li et al. [17] leverage meta-learning to adapt ISDA for the long-tail problem.

In contrast to the above-mentioned methods, our proposed LPL method conducts regular or irregular augmentation according to the directions of loss maximization and minimization. According to our Corollaries 1–3, loss maximization will force the category to move close to the decision boundary (i.e., the category is regularly augmented or virtual samples are generated for this category). In contrast, loss minimization will force the category to be far from the boundary (i.e., the category is irregularly augmented or samples are virtually deleted for this category). Fig. 9 shows an illustrative example.

D. Logit Perturbation Method (LPL) for Multilabel Learning

Multilabel learning is usually decomposed into C binary learning tasks as shown in Fig. 10. Each class is taken turns to be selected as positive and the remaining classes are negative. As a result, multilabel learning becomes a series of binary learning problems. Basically, the learning algorithms for binary tasks can be directly used for multilabel learning. However, compared with conventional single-label binary tasks, both variance imbalance and class imbalance usually exist in each of the C tasks, simultaneously. First, variance imbalance exists in each of the C tasks. The reason lies in that negative samples in each of the C tasks actually come from the remaining $C-1$ classes, whereas positive samples in each task come from only one class. Naturally, the variance of the negative samples is larger than that of the positive ones as shown in Fig. 11(b).

Theoretically, the negative samples require the first type of logit perturbation and the positive samples require the second type. Second, class imbalance may exist in each of the C tasks as shown in Fig. 11(c). However, the class-imbalance degrees for tasks in which the positive samples are from the tail categories are larger than those for tasks in which the positive samples are from the head categories. Therefore, according to Corollaries 1 and 2, the negative samples require the second type of logit perturbation, and the positive samples require the first type of logit perturbation (especially for the tasks when the positive samples belong to tail categories).

Obviously, there is a contradiction between the two cases. To deal with variance imbalance, the negative samples should perform the first type of logit perturbation. Meanwhile, to deal with class imbalance, the negative samples should perform the second type of logit perturbation. Corollary 3 demonstrates that the perturbation type depends on the primary challenge of class or variance imbalances. Consequently, we extend (22) to the following form for multilabel learning:

$$\mathcal{L} = \frac{1}{C \times N} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i)} \sum_{c=1}^{C} \mathbb{S}(c-\tau) \times \left\{ \max_{|\tilde{\delta}_c| \le \epsilon_c} y_{i,c} \log(1 + e^{-u_{i,c} + \tilde{\delta}_c}) \times \mathbb{S}(c-\tau) + \min_{|\tilde{\delta}_c| \le \epsilon_c} (1 - y_{i,c}) \log(1 + e^{u_{i,c} - \tilde{\delta}_c}) \times \mathbb{S}(c-\tau) \right\} \tag{25}$$

where $\tilde{\delta}_c$ is a scalar, and τ is a hyperparameter (threshold) for the category split. This new loss can effectively tune the cooperation of the two types of logit perturbation by setting

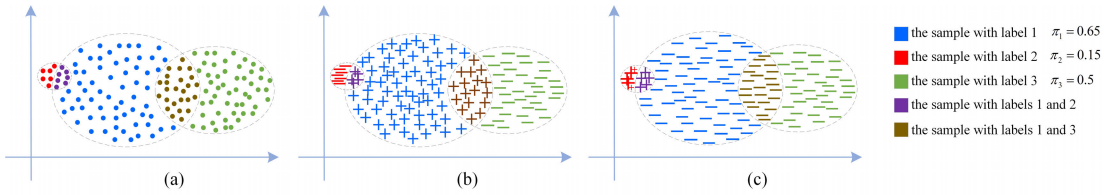


Fig. 11. Illustrative example for the variance imbalance and class imbalance in multilabel learning. "+" means the positive samples. "−" means the negative samples. (a) Multilabel learning task ($C = 3$). Different colors mean those samples with one or more labels. (b) Case of variance imbalance. (c) Case of class imbalance.

an appropriate value of τ. There are three typical settings for τ, namely, $\tau = 0$, $1 < \tau < C$, and $\tau = C + 1$. The first setting only considers class imbalance and the third setting only considers variance imbalance. Our theoretical analysis (Theorem 3 and Corollary 3) indicates that both the class and variance imbalance should be considered in multilabel learning. Therefore, the second setting is adopted in this study.

If $1 < \tau < C$, then $\mathbb{S}(c - \tau) \equiv 1$ when $c > \tau$ and $\mathbb{S}(c - \tau) \equiv -1$ when $c < \tau$. When $c < \tau$, the positive samples perform the second type of logit perturbation and the negative samples perform the first type of logit perturbation. This is reasonable because the cth class belongs to the head categories, and thus variance imbalance rather than the class imbalance is the primary concern. When $c > \tau$, the positive samples perform the first type of logit perturbation and the negative samples perform the second type of logit perturbation. This is reasonable because the cth class belongs to the tail categories, and class imbalance rather than the variance imbalance becomes the primary concern in learning.

Similarly, we can perform PGD-like maximization and minimization as Algorithm 1. The algorithmic steps of our LPL for multilabel learning are also in Algorithm 2.

E. Meta-Learning-Based Logit Perturbation Method

In our LPL, the category set split for $\mathcal{N}_a$ and $\mathcal{P}_a$ depends on heuristic rules. In (20), the two sets are determined by comparing the performance of a category with the mean performance. In (22), the two sets are determined according to the proportion order of each category. This section introduces a meta-learning-based strategy to split categories. First, the objective function is reformulated as follows:

$$\begin{aligned}\mathcal{L} = \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i)} \big\{ & \alpha_c \times \ell\big(\text{softmax}\big(\boldsymbol{u}_i + \boldsymbol{\delta}_c^{\max}\big), \boldsymbol{y}_i\big) \\ & + \beta_c \times \ell\big(\text{softmax}(\boldsymbol{u}_i + \boldsymbol{\delta}_c^{\min}, \boldsymbol{y}_i)\big)\big\} \\ \text{s.t. } & \alpha_c + \beta_c = 1 \text{ and } \alpha_c, \beta_c \in \{0, 1\} \end{aligned} \tag{26}$$

where c satisfies $y_{i,c} = 1$; $\boldsymbol{\delta}_c^{\max}$ and $\boldsymbol{\delta}_c^{\min}$ are calculated by Algorithm 1, respectively; α_c and β_c are the combination weights for the category c. When $\alpha_c = 1$, the category c belongs to $\mathcal{P}_a$.

In our proposed heuristic rules, $\mathcal{N}_a$ and $\mathcal{P}_a$ are determined by the performance or the proportion of each category as used in (20) and (22). Both performance and proportion are the characteristics of a category. In meta-learning-based logit perturbation method (MLPL), they are determined according to more characteristics of each category, including training dynamics (e.g., average loss, average performance, and average margin) and category proportion. Specifically, we assume that α_c and β_c depend on the training dynamics and proportion of the category c and are produced by a weighting network $g_{\boldsymbol{\theta}}(\cdot)$ parameterized by $\boldsymbol{\theta}$. The input of $g_{\boldsymbol{\theta}}(\cdot)$ consists of four quantities including average loss, average prediction, average margin, and proportion. Indeed, our category split rules used in (20) and (22) can be viewed as a heuristic definition for $g_{\boldsymbol{\theta}}(\cdot)$.

TABLE I

MEAN VALUES AND STANDARD DEVIATIONS OF THE TEST TOP-1 ERRORS FOR ALL THE INVOLVED METHODS ON CIFAR10

Method	WRN-28-10	ResNet-110
Basic	3.82 ± 0.15%	6.76 ± 0.34%
Large Margin	3.69 ± 0.10%	6.46 ± 0.20%
Disturb Label	3.91 ± 0.10%	6.61 ± 0.04%
Focal Loss	3.62 ± 0.07%	6.68 ± 0.22%
Center Loss	3.76 ± 0.05%	6.38 ± 0.20%
Lq Loss	3.78 ± 0.08%	6.69 ± 0.07%
CGAN	3.84 ± 0.07%	6.56 ± 0.14%
ACGAN	3.81 ± 0.11%	6.32 ± 0.12%
infoGAN	3.81 ± 0.05%	6.59 ± 0.12%
ISDA	3.60 ± 0.23%	6.33 ± 0.19%
ISDA + Dropout	3.58 ± 0.15%	5.98 ± 0.20%
LPL (mean + fixed ϵ_c)	3.39 ± 0.04%	5.83 ± 0.21%
LPL (mean + varied ϵ_c)	3.37 ± 0.04%	5.72 ± 0.05%
MLPL (fixed ϵ_c)	2.82 ± 0.09%	5.36 ± 0.11%
MLPL (varied ϵ_c)	**2.43 ± 0.05%**	**4.91 ± 0.09%**

MLPL learns $g_{\boldsymbol{\theta}}(\cdot)$ following the training manner used in Meta-weight-net [41] on the basis of a meta dataset $\mathcal{S}^m = \{(\boldsymbol{x}_i^m, \boldsymbol{y}_i^m)\}_{i=1}^M$. The updating of the backbone network and the weighting network is as follows:

$$\hat{\boldsymbol{W}}^t(\boldsymbol{\theta}) = \boldsymbol{W}^{t-1} - \eta_1 \frac{1}{N} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}\Big|_{\boldsymbol{W}^{t-1}} \tag{27}$$

where η_1 is the learning rate. $\boldsymbol{\theta}$ is updated after receiving feedback from the backbone network as follows:

$$\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \eta_2 \frac{1}{M} \frac{\partial \mathcal{L}_m}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^{t-1}} \tag{28}$$

where η_2 is the learning rate for meta-learning; $\mathcal{L}_m = \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i)} \ell(\text{softmax}(\boldsymbol{u}_i^m), \boldsymbol{y}_i^m)$ is the loss on S^m and $\boldsymbol{u}_i^m = f(\boldsymbol{x}_i^m, \hat{\boldsymbol{W}}^t(\boldsymbol{\theta}))$. Finally, the backbone network is updated with new α_c and β_c in (26) based on $\boldsymbol{\theta}^t$ as follows:

$$\boldsymbol{W}^t(\boldsymbol{\theta}^t) = \boldsymbol{W}^{t-1} - \eta_1 \frac{1}{N} \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}\Big|_{\boldsymbol{W}^{t-1}}. \tag{29}$$

IV. EXPERIMENTS

Our proposed LPL and MLPL are first evaluated on tasks that involve data augmentation, long-tail classification, and multilabel classification. We then perform additional experiments to analyze the properties of our method. These experiments are conducted on a Linux platform equipped with four RTX3090 graphics cards, each with a capacity of 24 GB.

A. Experiments on Data Augmentation

1) Datasets and Competing Methods: In this section, two benchmark image classification datasets, namely,

TABLE II
MEAN VALUES AND STANDARD DEVIATIONS OF THE TEST TOP-1 ERRORS FOR ALL THE INVOLVED METHODS ON CIFAR100

Method	WRN-28-10	ResNet-110
Basic	18.53 ± 0.07%	28.67 ± 0.44%
Large Margin	18.48 ± 0.05%	28.00 ± 0.09%
Disturb Label	18.56 ± 0.22%	28.46 ± 0.32%
Focal Loss	18.22 ± 0.08%	28.28 ± 0.32%
Center Loss	18.50 ± 0.25%	27.85 ± 0.10%
Lq Loss	18.43 ± 0.37%	28.78 ± 0.35%
CGAN	18.79 ± 0.08%	28.25 ± 0.36%
ACGAN	18.54 ± 0.05%	28.48 ± 0.44%
infoGAN	18.44 ± 0.10%	27.64 ± 0.14%
ISDA	18.12 ± 0.20%	27.57 ± 0.46%
ISDA + Dropout	17.98 ± 0.15%	26.35 ± 0.30%
LPL (mean + fixed ϵ_c)	18.19 ± 0.07%	26.09 ± 0.16%
LPL (mean + varied ϵ_c)	17.61 ± 0.30%	25.42 ± 0.07%
MLPL (fixed ϵ_c)	17.27 ± 0.10%	25.05 ± 0.13%
MLPL (varied ϵ_c)	**16.87 ± 0.06%**	**24.76 ± 0.11%**

CIFAR10 [38] and CIFAR100 [38], are used. Both the data consist of 32×32 natural images in ten classes for CIFAR10 and 100 classes for CIFAR100. There are 50 000 images for training and 10 000 images for testing. The training and testing configurations used in [16] are followed. Several classical and state-of-the-art robust loss functions and (semantic) data augmentation methods are compared: Large-margin loss [42], Disturb label [43], Focal Loss [32], Center loss [44], Lq loss [45], CGAN [46], ACGAN [47], infoGAN [48], ISDA, and ISDA + Dropout.

The Wide-ResNet-28-10 (WRN-28-10) [49] and ResNet-110 [50] are used as the base neural networks. Considering that the training/testing configuration is fixed for both the sets, the results of the above competing methods reported in the ISDA paper [16] are directly presented (some are from their original papers). The training settings for the above base neural networks also follow the instructions of ISDA paper and its released codes. Our LPL algorithm has two variants.

1) *LPL (Mean + Fixed Bound):* In this version, the optimization in (20) is used. Mean denotes that the threshold is mean($\bar{q}_c$). Fixed bound means that the value of ϵ_c is fixed and identical for all the categories during optimization. It is searched in {0.1, 0.2, 0.3, 0.4}.
2) *LPL (Mean+Varied Bound):* In this version, the optimization in (20) is used. Theoretically, varied bound means that the value of ϵ_c is varied according to (24). However, the varied bounds in the same batch make the implementation more difficult and increase the training complexity. In our implementation, we choose to set a varied number of updating steps for each category in our PGD-like optimization. $\Delta\epsilon$ is searched in {0.1, 0.2}.

Correspondingly, MLPL also has two variants, MLPL (fixed bound) and MLPL (varied bound). MLPL no longer needs to split categories, and its bound calculation method is consistent with that of LPL. The hyperparameter settings in the meta-learning part follow those of Shu et al. [41]. Likewise, we created a meta dataset by randomly selecting ten images per class from the training set.

The Top-1 error is used as the evaluation metric. The performances of the base neural networks with the standard CE loss are rerun before running our methods to conduct a fair comparison. Almost identical results are obtained compared with the published results in the ISDA [16].

2) Results: Tables I and II present the results of all the competing methods on CIFAR10 and CIFAR100, respectively. Our LPL (two versions) performs almost better than all the comparative methods on both the backbone networks. In addition, when using meta-learning strategies, our MLPL shows further improvement. ISDA performs second best. Only in the case of WRN-28-10 on CIFAR100, LPL (mean + fixed ϵ_c) is inferior to ISDA, indicating that when the backbone network is powerful in feature learning, the improvement brought by logit perturbation is limited or even ineffective under fixed perturbation bound.

Both the LPL and MLPL produce better results when using varied bounds compared with using fixed bounds. This comparison indicates the rationality of our motivation that the category with relatively low (high) performance should be more regularly (irregularly) augmented. In the final part of this section, more analyses will be conducted to compare ISDA and our method. Naturally, the varied threshold will further improve the performances.

B. Experiments on Long-Tail Classification

1) Datasets and Competing Methods: In comparison to the conference version of this article, we supplement the experiments with real-world datasets. In the synthetic dataset experiment, the long-tail versions of CIFAR10 and CIFAR100 compiled by Cui et al. [39] are used and called CIFAR10-LT and CIFAR100-LT, respectively. The training and testing configurations used in [24] are followed. In the real-world dataset experiment, large-scale datasets iNaturalist 2017 (iNat2017) [51] and iNaturalist 2018 (iNat2018) [52] with extremely imbalanced class distributions are used. iNat2017 contains 579 184 training images categorized into 5089 classes, with an imbalance factor of 3919/9. iNat2018 consists of 435 713 images distributed among 8142 classes, with an imbalance factor of 1000/2. Several classical and state-of-the-art robust loss functions and semantic data augmentation methods are compared: Class-balanced CE loss [16], Class-balanced fine-tuning [53], Meta-weight net [41], Focal Loss [32], Class-balanced focal loss [39], LDAM [37], LDAM-DRW [37], ISDA + Dropout, and LA.

In the synthetic dataset experiment, Menon et al. [24] released the training data when the imbalance ratio (i.e., π_1/π_{100}) is 100:1; hence, their data and reported results for the above competing methods are directly presented. When the ratio is 10:1, the results of ISDA + Dropout and LA are obtained by running their released codes. The results of the rest methods are from the study conducted by Li et al. [17]. The hyperparameter λ in LA is searched in {0.5, 1, 1.5, 2, 2.5} according to the suggestion in [24]. Similar to the experiments in [24], ResNet-32 [50] is used as the backbone network. The results presented are the average of five repeated runs.

In the real-world dataset experiment, the results of the above competing methods reported in [24] are directly presented. The results of LA on iNat2018 are from the original paper [24]. The other results, such as ISDA + dropout and LA on iNat2017, are obtained by running their released codes. Likewise, the hyperparameter λ in LA is searched in {0.5, 1, 1.5, 2, 2.5}. Similar to the experiments in [26], ResNet-50 [50] is used as the backbone network. All the results are the average of five repeated runs.

LPL have two variants: LPL (varied threshold + fixed bound) and LPL (varied threshold + varied bound). The

TABLE III
TEST TOP-1 ERRORS ON CIFAR100-LT (RESNET-32)

Ratio	100:1	10:1
Class-balanced cross-entropy loss	61.23%	42.43%
Class-balanced fine-tuning	58.50%	42.43%
Meta-weight net	58.39%	41.09%
Focal Loss	61.59%	44.22%
Class-balanced focal loss	60.40%	42.01%
LDAM	59.40%	42.71%
LDAM-DRW	57.11%	41.22%
ISDA + Dropout	62.60%	44.49%
LA	56.11%	41.66%
LPL (varied τ + fixed ϵ_c)	58.03%	41.86%
LPL (varied τ + varied ϵ_c)	55.75%	39.03%
MLPL (fixed ϵ_c)	55.34%	38.62%
MLPL (varied ϵ_c)	**54.85%**	**38.37%**

TABLE IV
TEST TOP-1 ERRORS ON CIFAR10-LT (RESNET-32)

Ratio	100:1	10:1
Class-balanced cross-entropy loss	27.32%	13.10%
Class-balanced fine-tuning	28.66%	16.83%
Meta-weight net	26.43%	12.45%
Focal Loss	29.62%	13.34%
Class-balanced focal loss	25.43%	12.52%
LDAM	26.45%	12.68%
LDAM-DRW	25.88%	11.63%
ISDA + Dropout	26.45%	12.98%
LA	22.33%	11.07%
LPL (varied τ + fixed ϵ_c)	23.97%	11.09%
LPL (varied τ + varied ϵ_c)	22.05%	10.59%
MLPL (fixed ϵ_c)	21.79%	10.27%
MLPL (varied ϵ_c)	**21.43%**	**9.78%**

threshold τ is searched in $\{0.4C, 0.5C, 0.6C\}$. In the fixed bound version, the value of $\Delta\varepsilon$ is set to 0, and ϵ is searched in $\{1.5, 2.5, 5\}$. In the varied bound version, the value of ϵ is set to 0, and $\Delta\varepsilon$ is searched in $\{1.0, 2.0, 3.0\}$. Similarly, MLPL has two variants: MLPL (fixed bound) and MLPL (varied bound). The category set split is determined by α_c and β_c with a threshold of 0.5, slightly different from the bound calculation of LPL. The hyperparameter settings in the meta-learning part follow those of Shu et al. [41]. Following Li et al. [17], for our metadata, we chose five images per class from the iNat2017 training set and two images per class from the iNat2018 training set. We mainly aim to compare methods that only modify the training loss. Other methods, such as BBN [20], which focus on the new network structure are also not included in the comparisons.

2) Results: The Top-1 error is also used. Table III shows the results of all the methods on the CIFAR100-LT data. On the ratios 100:1 and 10:1, MLPL (varied ϵ_c) yields the lowest Top-1 errors. It exceeds the best competing method LA by 1.26% and 3.29% on the ratios 100:1 and 10:1, respectively. Table IV shows the results of all the methods on the CIFAR10-LT data. MLPL (varied ϵ_c) continues to achieve the lowest Top-1 error rates on both the ratios. Table V shows the results of all the methods on iNat2017 and iNat2018. For real-world long-tail datasets, it still exceeds LA 1.63% and 1.80%, respectively. On all the comparisons, ISDA obtains poor results. On CIFAR100-LT, ISDA achieves the worst performances on both the ratios. This result is expected because ISDA aims to regularly augment all the categories equally and does not favor tail categories, which may lead to tail categories suffering from this regular augmentation. Nevertheless, ISDA has a better performance on CIFAR10-LT than on CIFAR100-LT. In Fig. 1(b), the loss increments of tail categories are larger than those of head. That is, larger augmentations are exerted on tail categories.

TABLE V
TEST TOP-1 ERRORS ON REAL-WORLD DATASETS (RESNET-50)

Method	iNat2017	iNat2018
Class-balanced cross-entropy loss	42.02%	33.57%
Class-balanced fine-tuning	41.77%	34.16%
Meta-weight net	37.48%	32.50%
Focal Loss	38.98%	72.69%
Class-balanced focal loss	41.92%	38.88%
LDAM	39.15%	34.13%
LDAM-DRW	37.84%	32.12%
ISDA + Dropout	43.37%	39.92%
LA	36.75%	31.56%
LPL (varied τ + fixed ϵ_c)	38.47%	32.06%
LPL (varied τ + varied ϵ_c)	35.86%	30.59%
MLPL (fixed ϵ_c)	35.47%	30.17%
MLPL (varied ϵ_c)	**35.12%**	**29.76%**

TABLE VI
ERROR REDUCTION OF LPL (VARIED τ + VARIED ϵ) OVER LA ON THE TWO DATASETS

Ratio	100:1		10:1	
LA	56.11%	22.33%	41.66%	11.07%
LPL	55.75% (-0.36%)	22.05% (-0.28%)	39.03% (-2.63%)	10.59% (-0.48%)

We listed the Top-1 errors of LA and LPL (varied τ + varied ϵ_c) in Table VI to better present the comparison. When the ratio is smaller, the improvements (error reductions) are relatively larger. This result is reasonable because when the ratio becomes small, the effectiveness of LA will be subsequently weakened. When the imbalance ratio is 1, indicating that there is no imbalance, LA will lose effect; however, our LPL can still augment the training data effectively.

C. Experiments on Multilabel Classification

1) Datasets and Competing Methods: In this part, the long-tail multilabel versions of VOC [54] and MS-COCO [55] compiled by Wu et al. [26] are used and called VOC-MLT and COCO-MLT, respectively. The training and test configurations used in [26] are followed. The training set of VOC-MLT is sampled from the train-val set of VOC2012, containing 1142 images from 20 categories, with a maximum of 775 images per category and a minimum of four images per category. A total of 4952 images from the VOC2007 test set are used for evaluation. COCO-MLT is sampled from the MS COCO-2017 dataset, containing 1909 images from 80 categories, with a maximum of 1128 images per category and a minimum of six images per category. In all, 5000 images from the MS COCO-2017 test set are used for evaluation.

We mainly compare NTR and LC that perturb logit. The code of LC is not open-sourced. To keep the consistency of the experimental setup, we conduct both the comparison experiments on the basis of R-BCE [26]. Several classical and state-of-the-art robust loss functions and multilabel methods are compared: empirical risk minimization (ERM), Reweighting, Focal Loss [32], Resampling [56], ML-GCN [57], OLTR [58], LDAM [37], CB-Focal [39], R-BCE [26], R-BCE-Focal [26], R-BCE + NTR [26], R-BCE-Focal + NTR [26], R-BCE + LC [27], and R-BCE-Focal + LC [27].

Wu et al. [26] released the training data and code. Hence, their data and reported results for the above competing meth-

TABLE VII
RESULTS OF MAP BY OUR METHODS AND OTHER COMPARING APPROACHES ON VOC-MLT AND COCO-MLT

Datasets	VOC-MLT				COCO-MLT			
Method	total	head	medium	tail	total	head	medium	tail
ERM	70.86%	68.91%	80.20%	65.31%	41.27%	48.48%	49.06%	24.25%
Re-Weighting	74.70%	67.58%	82.81%	73.96%	42.27%	48.62%	45.80%	32.02%
Focal Loss	73.88%	69.41%	81.43%	71.56%	49.46%	49.80%	54.77%	42.14%
Re-Sampling	75.38%	70.95%	82.94%	73.05%	46.97%	47.58%	50.55%	41.70%
RS-Focal	76.45%	72.05%	83.42%	74.52%	51.14%	48.90%	54.79%	48.30%
ML-GCN	68.92%	70.14%	76.41%	62.39%	44.24%	44.04%	48.36%	38.96%
OLTR	71.02%	70.31%	79.80%	64.95%	45.83%	47.45%	50.63%	38.05%
LDAM	70.73%	68.73%	80.38%	69.09%	40.53%	48.77%	48.38%	22.92%
CB-Focal	75.24%	70.30%	83.53%	72.74%	49.06%	47.91%	53.01%	44.85%
R-BCE	76.34%	71.40%	82.76%	75.22%	49.43%	48.77%	53.00%	45.33%
R-BCE-Focal	77.39%	72.44%	83.16%	76.77%	52.75%	50.20%	56.52%	50.02%
R-BCE+NTR	78.65%	73.16%	84.11%	78.66%	52.53%	50.25%	56.33%	49.54%
R-BCE-Focal+NTR	78.94%	73.22%	84.18%	79.30%	53.55%	51.13%	57.05%	51.06%
R-BCE+LC	78.08%	73.10%	83.49%	77.75%	53.68%	50.58%	57.10%	51.90%
R-BCE-Focal+LC	78.66%	72.74%	83.45%	79.52%	53.94%	50.99%	57.47%	51.88%
R-BCE+LPL (varied τ + fixed ϵ_c)	78.64%	73.00%	82.81%	79.74%	53.97%	50.23%	57.36%	52.79%
R-BCE+LPL (varied τ + varied ϵ_c)	79.02%	72.39%	82.14%	81.64%	54.35%	51.48%	57.72%	52.42%
R-BCE-Focal+LPL (varied τ + fixed ϵ_c)	79.17%	73.33%	83.56%	80.27%	54.37%	51.14%	57.68%	52.85%
R-BCE-Focal+LPL (varied τ + varied ϵ_c)	**79.57**%	73.47%	83.95%	80.87%	**54.76**%	50.78%	58.12%	53.81%

ods are directly presented. The experimental results of LC are reimplemented from the original paper's formula. Similar to the experiments in [26], ResNet-50 [50] is used.

Our methods have two variants: LPL (varied threshold + fixed bound) and LPL (varied threshold + varied bound). The threshold τ is searched in $\{0.4C, 0.5C, 0.6C\}$. In the fixed bound version, the value of $\Delta\varepsilon$ is set to 0, and ϵ is searched in $\{0.05, 0.1, 0.15\}$. In the varied bound version, the value of ϵ is set to 0, and $\Delta\varepsilon$ is searched in $\{0.1, 0.2, 0.3\}$. Other experimental setups such as training epochs follow NTR.

2) Results: The evaluation metric mAP is used. Table VII shows the results of all the methods on VOC-MLT and COCO-MLT. Our method achieves competitive or better results. R-BCE-Focal + LPL (varied τ + varied ϵ_c) achieves the best results on VOC-MLT and COCO-MLT. R-BCE-Focal + LPL (varied τ + varied ϵ_c) outperforms R-BCE-Focal + NTR by 0.63% and 1.21%, respectively, and outperforms R-BCE-Focal + LC by 0.91% and 0.82%, respectively. In the comparison experiment, R-BCE-Focal + LPL (varied τ + varied ϵ_c) exceeds R-BCE-Focal by 2.18% on VOC-MLT and by 2.01% on COCO-MLT, respectively. Similarly, when our method is added to the baseline R-BCE, our method can further improve the performance. The effectiveness of LPL is well-proven.

D. More Analysis and Discussion for Our Method

1) Improvements on Existing Methods: Our LPL method seeks the perturbation via an optimization scheme. In ISDA and LA, the perturbations are directly calculated rather than optimization. A natural question arises, that is, whether the perturbations in existing methods further improved via our method. Therefore, we propose a combination method with the following loss in imbalance image classification:

$$\sum_{c\in\mathcal{N}_a}\sum_{\boldsymbol{x}_i\in\boldsymbol{S}_c}\min_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} l(\text{softmax}(\boldsymbol{u}_i+\lambda\log\boldsymbol{\pi}+\tilde{\boldsymbol{\delta}}_c),\boldsymbol{y}_i) + \sum_{c\in\mathcal{P}_a}\sum_{\boldsymbol{x}_i\in\boldsymbol{S}_c}\max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} l(\text{softmax}(\boldsymbol{u}_i+\lambda\log\boldsymbol{\pi}+\tilde{\boldsymbol{\delta}}_c),\boldsymbol{y}_i)$$

where $\log\boldsymbol{\pi} = [\log\pi_1,\ldots,\log\pi_C]$. When all ϵ_cs are zero, the above-mentioned loss becomes the loss of LA; when λ is zero, the above loss becomes our LPL (with fixed bound). We conducted experiments on CIFAR10-LT100 and CIFAR100-LT100. The results are shown in Table VIII.

TABLE VIII
TEST TOP-1 ERRORS OF THREE METHODS ON TWO DATASETS

Method	CIFAR10-LT100	CIFAR100-LT100
LA	22.33%	56.11%
LPL	22.05%	55.75%
LA+LPL	21.46%	53.89%

ResNet-32 is used as the basic model. The value of λ is searched in $\{0.5, 1, 1.5, 2, 2.5\}$. The threshold τ is set as 4 and 40 on CIFAR10 and CIFAR100, respectively. Other parameters follow the setting in the previous experiments.

The combination of LA + LPL achieves the lowest errors in both the comparisons, indicating that LPL can further improve the performance of the existing SOTA methods. Similarly, ISDA can also be improved in the same manner.

2) More Comparisons With ISDA: ISDA claims that it does not increase the number of parameters compared with the direct learning with the basic DNN models. Our method also does not increase the number of model parameters. The reason lies in that the perturbation terms are no longer used in the final prediction.

Table IX shows the comparisons between ISDA and LPL (two variants) on three additional base DNN models, namely, SE-ResNet110 [59], Wide-ResNet-16-8 (WRN-16-8) [49], and ResNet-32. The numbers of parameters are equal for ISDA and LPL. Nevertheless, the two variants of our method LPL outperform ISDA on both the datasets under all the five base models. Nevertheless, the increment becomes smaller when more powerful base neural networks are used.

3) Loss Variations in LPL During Training: For single-label classification, we plot the loss variations in LPL on two balanced and two long-tail datasets to assess whether our method LPL is in accordance with the two conjectures. The curves are shown in Fig. 12(a) and (b). On the balanced data, the relative loss variations are similar to those of ISDA; on the long-tail data, the losses of head categories are reduced, whereas those of tail ones are increased, which is similar to those of LA. For the multilabel classification, Fig. 12(c) shows the results. In comparison to NTR and LC, our method LPL

TABLE IX
NUMBER OF PARAMETERS AND TEST TOP-1 ERRORS OF ISDA AND LPL WITH DIFFERENT BASE NETWORKS

Method	#Params	CIFAR10	CIFAR100
ResNet-32+ISDA	0.5M	7.09 ± 0.12%	30.27 ± 0.34%
ResNet-32+LPL (mean + fixed ϵ_c)	0.5M	7.01 ± 0.16%	29.59 ± 0.27%
ResNet-32+LPL (mean + varied ϵ_c)	0.5M	**6.66 ± 0.09%**	**28.53 ± 0.16%**
SE-Resnet110+ISDA	1.7M	5.96 ± 0.21%	26.63 ± 0.21%
SE-Resnet110+LPL (mean + fixed ϵ_c)	1.7M	5.87 ± 0.17%	26.12 ± 0.24%
SE-Resnet110+LPL (mean + varied ϵ_c)	1.7M	**5.39 ± 0.10%**	**25.70 ± 0.07%**
WRN-16-8+ISDA	11.0M	4.04 ± 0.29%	19.91 ± 0.21%
WRN-16-8+LPL (mean + fixed ϵ_c)	11.0M	3.97 ± 0.09%	19.87 ± 0.02%
WRN-16-8+LPL (mean + varied ϵ_c)	11.0M	**3.93 ± 0.10%**	**19.83 ± 0.09%**

TABLE X
RESULTS OF MAP BY OUR METHODS AND OTHER COMPARING APPROACHES ON MS-COCO

Method	MS-COCO
R-BCE+NTR	83.65%
R-BCE+LC	84.51%
R-BCE+LPL (varied τ + varied ϵ_c)	**85.43%**

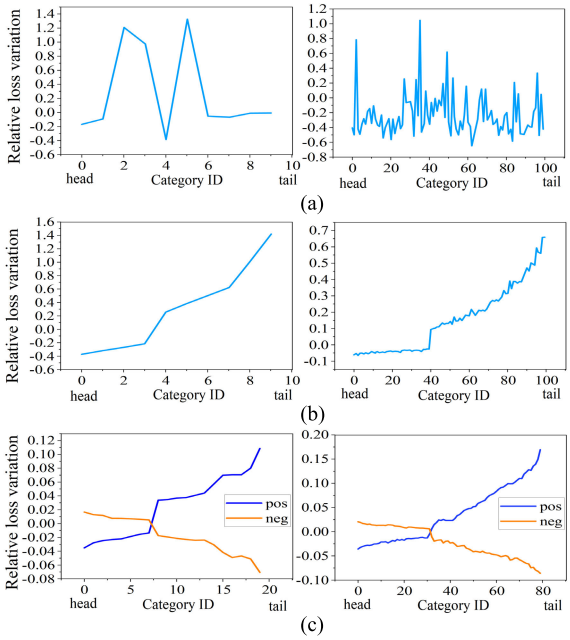


Fig. 12. Relative loss variations in our LPL on two balanced datasets, two long-tail datasets, and two multilabel datasets. "pos" means the relative loss variations of positive samples, while "neg" means those of negative samples. (a) Balanced datasets on CIFAR10 (left) and CIFAR100 (right). (b) Long-tail datasets on CIFAR10-LT (left) and CIFAR100-LT (right). (c) Multilabel datasets on VOC-MLT (left) and COCO-MLT (right).

focuses more on the tail categories according to the trends of relative loss reduction.

4) More Comparisons With NTR and LC: We also compare our method with NTR and LC on the original multilabel dataset MS-COCO. MS-COCO contains 122 218 images with 80 different labels, which is divided into a training set with 82 081 images and a test set with 40 137 images. In this part, ResNet-110 is used as the base neural network and the input size is 448 × 448. Other setups follow Section IV-C. Table X shows the results. The evaluation metric mAP is used. Once again, our method achieves the competitive results. R-BCE + LPL (varied τ + varied ϵ_c) outperforms R-BCE + NTR and R-BCE + LC by 1.78% and 0.86%, respectively.

5) Possible Extensions for Our Method: In our future work, two extensions will be considered. The first is the application of logit perturbation into important areas such as bioinformatics [60], [61]. Hu et al. [62] presented a comprehensive survey for computational models in bioinformatics and pointed out that serious class imbalance exists in protein–protein interactions' prediction. The second is to apply logit perturbation for high challenging issues such as learning for high-dimensional and incomplete data [63], [64].

V. CONCLUSION

This study investigates the class-level logit perturbation in deep learning. Two conjectures for the relationship between (logit perturbation-incurred) loss increment/decrement and regular/irregular data augmentation are proposed. To support the two conjectures, theoretical investigation is performed in the presence of class imbalance and variance imbalance. On the basis of the two conjectures and our theoretical findings, new methodologies are introduced to learn to perturb logits during DNN training for both the single-label and multilabel learning tasks. Two key components of LPL, namely, category-set split and boundary calculation, are investigated. Meta-learning is also leveraged to determine the perturbation direction. Extensive experiments on data augmentation (for balanced classification), long-tail classification, and multilabel classification are conducted. LPL achieves the best performances in both the situations under different basic networks. If metadata are available, LPL achieves better performance with meta-learning. The existing methods with logit perturbation (e.g., LA) can also be improved using our method.

APPENDIX

A. Proof for Theorem 1

Proof: Xu et al. [40] proved that $\boldsymbol{w} = \mathbf{1}$ when the data distribution in (10) is given (see [40, Lemma 1]). According to [40, Lemma 1], we can easily prove that when $P_+ : P_- = 1 : \Gamma$ and $\Gamma > 1$, $\boldsymbol{w} = \mathbf{1}$ holds. Thus, $f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i + b$. Then (13) can be written as follows:

$$b^* = \arg\min_b \Pr \cdot \left(\mathbb{S}\left(\sum_{i=1}^{d} x_i + b + \tilde{\delta}_c^* \right) \neq y \right). \tag{30}$$

Now, we can calculate the optimal b^* when the logit perturbation is used. Then, the optimal linear classifier is

$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i + b^*$. We use $\mathcal{R}_{\text{lp}}(f)$ to denote the error after logit perturbation

$$\begin{aligned}\mathcal{R}_{\text{lp}}(f) \propto\ & \Gamma \cdot \Pr\cdot(\exists\|\tilde{\delta}_-\| \le \epsilon, \quad \mathbb{S}(u+\tilde{\delta}_-) \neq -1 \mid y=-1)\\ & + \Pr\cdot(\exists\|\tilde{\delta}_+\| \le \epsilon\cdot\rho_+, \quad \mathbb{S}(u+\tilde{\delta}_+) \neq +1 \mid y=+1)\\ =\ & \Gamma\cdot\Pr\cdot(\mathbb{S}(u+\epsilon)\neq -1 \mid y=-1)\\ & + \Pr\cdot(\mathbb{S}(u-\epsilon\cdot\rho_+)\neq +1 \mid y=+1)\\ =\ & \Gamma\cdot\Pr\cdot\left\{\sum_{i=1}^{d} x_i + b + \epsilon > 0 \mid y=-1\right\}\\ & + \Pr\cdot\left\{\sum_{i=1}^{d} x_i + b - \epsilon\cdot\rho_+ < 0 \mid y=+1\right\}. \end{aligned} \tag{31}$$

According to (31), we have

$$\begin{aligned}\mathcal{R}_{\text{lp}}(f) \propto\ & \Gamma\cdot\Pr\cdot\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b+\epsilon}{\sqrt{d}\sigma}\right\}\\ & + \Pr\cdot\left\{\mathcal{N}(0,1) < -\left(\frac{\sqrt{d}\eta}{\sigma} + \frac{b-\epsilon\cdot\rho_+}{\sqrt{d}\sigma}\right)\right\}. \end{aligned} \tag{32}$$

The optimal b^* to minimize $\mathcal{R}_{\text{lp}}(f)$ is achieved at the point that $((\partial R_{\text{lp}}(f))/\partial b) = 0$. Then we can get the optimal b^*

$$b^* = \frac{1}{2}\epsilon(\rho-1) + \frac{d\sigma^2\log\Gamma}{\epsilon - 2d\eta + \epsilon\cdot\rho_+}. \tag{33}$$

By taking b^* into $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$, we can get the theorem

$$\begin{aligned}\mathcal{R}\big(f_{\text{opt}}, -1\big) &= \Pr\cdot\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\}\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < \frac{A}{2} + \frac{\log\Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma}\right\}\\ \mathcal{R}\big(f_{\text{opt}}, +1\big) &= \Pr\cdot\left\{\mathcal{N}(0,1) < -\left(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right)\right\}\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < \frac{A}{2} - \frac{\log\Gamma}{A} - \frac{\epsilon\cdot\rho_+}{\sqrt{d}\sigma}\right\} \end{aligned} \tag{34}$$

where $A = ((\epsilon\cdot\rho_+ - 2d\eta + \epsilon)/\sqrt{d}\sigma)$. □

B. Corollary 1

Proof: According to (34), we compute the partial derivatives of b^*_{rob} with respect to ρ to proof the corollary

$$\frac{\partial b^*}{\partial \rho_+} = \frac{\epsilon}{2} - \frac{d\epsilon\sigma^2\log\Gamma}{(\epsilon - 2d\eta + \epsilon\cdot\rho_+)^2}. \tag{35}$$

When $(\partial b^*/\partial\rho_+) > 0$, b^* increases as ρ_+ increases. We reorganize $(\partial b^*/\partial\rho) > 0$ to get the following equation:

$$\log\Gamma < \frac{(\epsilon + \epsilon\cdot\rho_+ - 2d\eta)^2}{2d\sigma^2}. \tag{36}$$

The minimum value of the right-hand term of inequality (36) is taken at $\rho_+ = ((2d\eta - \epsilon)/\epsilon)$. But obviously, we have $((2d\eta-\epsilon)/\epsilon) > (\eta/\epsilon)$. So we bring $\rho_+ = (\eta/\epsilon)$ into the right-hand side of inequality (36), and we get the following inequality:

$$\Gamma < e^{\frac{((2d-1)\eta-\epsilon)^2}{2d\sigma^2}}. \tag{37}$$

When (37) holds, b^* is a monotonically increasing function of ρ_+. According to (34), the corollary holds. □

C. Proof for Theorem 3

Proof: Like the proof in Theorem 1, we can get the following equations:

$$\begin{aligned}\mathcal{R}_{\text{lp}}(f) \propto\ & \Pr\cdot(\exists\|\tilde{\delta}_+\| \le \epsilon\cdot\rho_+, \mathbb{S}(u+\tilde{\delta}_+)\neq +1 \mid y=+1)\\ & + \Gamma\cdot\Pr\cdot(\exists\|\tilde{\delta}_-\| \le \epsilon\cdot\rho_-, \mathbb{S}(u+\tilde{\delta}_-)\neq -1 \mid y=-1)\\ =\ & \Gamma\cdot\max_{\|\tilde{\delta}_-\|\le\epsilon\cdot\rho_-}\Pr\cdot(\mathbb{S}(u+\tilde{\delta}_-)\neq -1 \mid y=-1)\\ & + \max_{\|\tilde{\delta}_+\|\le\epsilon\cdot\rho_+}\Pr\cdot(\mathbb{S}(u+\tilde{\delta}_+)\neq +1 \mid y=+1)\\ =\ & \Gamma\cdot\Pr\cdot\left\{\sum_{i=1}^{d} x_i + b + \epsilon\cdot\rho_+ > 0 \mid y=-1\right\}\\ & + \Pr\cdot\left\{\sum_{i=1}^{d} x_i + b - \epsilon\cdot\rho_+ < 0 \mid y=+1\right\}. \end{aligned} \tag{38}$$

According to (38), we have

$$\begin{aligned}\mathcal{R}_{\text{lp}}(f) \propto=\ & \Gamma\cdot\Pr\cdot\left\{\mathcal{N}(0,1) < \frac{1}{K}\left(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b+\epsilon\cdot\rho_-}{\sqrt{d}\sigma}\right)\right\}\\ & + \Pr\cdot\left\{\mathcal{N}(0,1) < -\left(\frac{\sqrt{d}\eta}{\sigma} + \frac{b-\epsilon\cdot\rho_+}{\sqrt{d}\sigma}\right)\right\}. \end{aligned} \tag{39}$$

The optimal b^* to minimize $\mathcal{R}_{\text{lp}}(f)$ is achieved at the point that $((\partial R_{\text{lp}}(f))/\partial b) = 0$. Then we can get the optimal b^*

$$\begin{aligned}b^* = \frac{1}{K^2-1}\Big(&\epsilon(\rho_- + K^2\rho_+) - d\eta(K^2+1)\\ &+ K\sqrt{(\epsilon\rho_- + \epsilon\rho_+ - 2d\eta)^2 + 2d(K^2-1)\sigma^2\log\left(\frac{K}{\Gamma}\right)}\Big). \end{aligned} \tag{40}$$

Therefore, the optimal standard error rates for the two classes can be obtained, respectively,

$$\begin{aligned}&\mathcal{R}\big(f_{\text{opt}}, +1\big)\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < -\left(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right)\right\}\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < -K\sqrt{B^2 + q(K,\Gamma)} - B - \frac{\epsilon\rho_+}{\sqrt{d}\sigma}\right)\right\}\\ &\mathcal{R}\big(f_{\text{opt}}, -1\big)\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < \frac{1}{K}\left(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right)\right\}\\ &= \Pr\cdot\left\{\mathcal{N}(0,1) < KB + \sqrt{B^2 + q(K,\Gamma)} - \frac{\epsilon\rho_-}{K\sqrt{d}\sigma}\right\} \end{aligned} \tag{41}$$

where $B = (\epsilon\rho_- + \epsilon\rho_+ - 2d\eta)/(\sqrt{d}\sigma(K^2-1))$ and $q(K,\Gamma) = (2\log(K/\Gamma))/(K^2-1)$. □

D. Corollary 3

Proof: When $\rho_- = 0$ and $\rho_+ = 0$, we have

$$\begin{aligned}b^* = \frac{1}{K^2-1}\Big(&- d\eta(K^2+1)\\ &+ K\sqrt{4d^2\eta^2 + 2d(K^2-1)\sigma^2\log\left(\frac{K}{\Gamma}\right)}\Big). \end{aligned} \tag{42}$$

Let U_+ and U_- be as follows:

$$U_+ = -\left(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right); \quad U_- = \frac{1}{K}\left(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right). \tag{43}$$

It is easy to verify that when $Ke^{(2d\eta-\epsilon)^2/(2dK^2\sigma^2)} < \Gamma < Ke^{2d\eta^2/((K^2-1)\sigma^2)}$, $U_+ > U_-$ holds. Therefore we have $\mathcal{R}(f_{\text{opt}}, +1) > \mathcal{R}(f_{\text{opt}}, -1)$, that is, class "+1" is harder than class "−1."

When $Ke^{(2d\eta-\epsilon)^2/(2dK^2\sigma^2)} < \Gamma < Ke^{2d\eta^2/((K^2-1)\sigma^2)}$, (44) holds

$$\frac{\partial b^*}{\partial t} = \frac{K^2\epsilon + \frac{K^2\epsilon(\epsilon+\epsilon\rho_+-2d\eta)}{K\sqrt{(\epsilon+\epsilon\rho_+-2d\eta)^2+2d(K^2-1)\sigma^2\log\left(\frac{K}{\Gamma}\right)}}}{K^2-1} \leq 0. \tag{44}$$

When $(\partial b^*/\partial\rho_+) <= 0$, the error of class "+1" decreases and the error of class "−1" increases as ρ_+ increases. Similarly, we can also prove other cases. □

REFERENCES

[1] X. Luo, H. Wu, Z. Wang, J. Wang, and D. Meng, "A novel approach to large-scale dynamically weighted directed network representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9756–9773, Dec. 2022.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. ICLR*, 2018, pp. 1–9.

[3] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 819–828.

[4] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," in *Proc. AAAI*, 2020, pp. 8018–8025.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[7] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Proc. ICLR*, 2015, pp. 1–9.

[8] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1944–1952.

[9] X. Wang, Y. Hua, E. Kodirov, D. A. Clifton, and N. M. Robertson, "ProSelfLC: Progressive self label correction for training robust deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 752–761.

[10] Y. Wu, J. Shu, Q. Xie, Q. Zhao, and D. Meng, "Learning to purify noisy labels via meta soft label corrector," in *Proc. AAAI*, 2021, pp. 10388–10396.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018, pp. 1–9.

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

[13] M. Li, F. Su, O. Wu, and J. Zhang, "Logit perturbation," in *Proc. AAAI*, 2022, pp. 10388–10396.

[14] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[15] Z. Hu, B. Tan, R. Salakhutdinov, T. Mitchell, and E. P. Xing, "Learning data manipulation for augmentation and weighting," in *Proc. NeurIPS*, 2019, pp. 15738–15749.

[16] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, and C. Wu, "Implicit semantic data augmentation for deep networks," in *Proc. NeurIPS*, 2019, pp. 12635–12644.

[17] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "MetaSAug: Meta semantic augmentation for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5212–5221.

[18] D. Wu, X. Luo, Y. He, and M. Zhou, "A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 9, 2022, doi: 10.1109/TNNLS.2022.3200009.

[19] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.

[20] 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. (CVPR)*, Jun. 2020, pp. 9719–9728.

[21] Y. Fan, S. Lyu, Y. Ying, and B.-G. Hu, "Learning with average top-k loss," in *Proc. NeurIPS*, 2017, pp. 497–505.

[22] M. Chen, C. He, and X. Luo, "MNL: A highly-efficient model for large-scale dynamic weighted directed network representation," *IEEE Trans. Big Data*, early access, Oct. 31, 2022, doi: 10.1109/TBDATA.2022.3218064.

[23] L. Hu, Z. Li, Z. Tang, C. Zhao, X. Zhou, and P. Hu, "Effectively predicting HIV-1 protease cleavage sites by using an ensemble learning approach," *BMC Bioinf.*, vol. 23, no. 1, p. 447, Oct. 2022.

[24] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *Proc. ICLR*, 2021, pp. 1–9.

[25] B.-W. Zhao, L. Hu, Z.-H. You, L. Wang, and X.-R. Su, "HINGRL: Predicting drug–disease associations with graph representation learning on heterogeneous information networks," *Briefings Bioinf.*, vol. 23, no. 1, Jan. 2022, Art. no. bbab515.

[26] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *Proc. ECCV*, 2020, pp. 162–178.

[27] H. Guo and S. Wang, "Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15089–15098.

[28] T. Wei and Y. Li, "Does tail label help for large-scale multi-label learning?" *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2315–2324, Jul. 2020.

[29] T. Han, S. Zhao, X. Sun, and J. Yu, "Modeling long-term video semantic distribution for temporal action proposal generation," *Neurocomputing*, vol. 490, pp. 217–225, Jun. 2022.

[30] M. Tan, J. Yu, H. Zhang, Y. Rui, and D. Tao, "Image recognition by predicted user click feature with multidomain multitask transfer deep network," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6047–6062, Dec. 2019.

[31] J. Yu, M. Tan, H. Zhang, Y. Rui, and D. Tao, "Hierarchical deep click feature prediction for fine-grained image recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 563–578, Feb. 2022.

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[33] Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, and J. Y. Zou, "Adversarial training helps transfer learning via better representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 25179–25191.

[34] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15721–15730.

[35] A. Shafahi, A. Ghiasi, F. Huang, and T. Goldstein, "Label smoothing and logit squeezing: A replacement for adversarial training?" 2019, *arXiv:1910.11585*.

[36] T. Wu, Z. Liu, Q. Huang, Y. Wang, and D. Lin, "Adversarial robustness under long-tailed distribution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8659–8668.

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

[38] A. Krizhevsky. (2009). *Learning Multiple Layers of Features From Tiny Images*. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

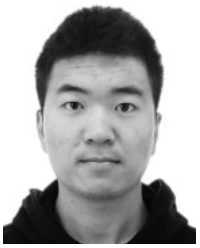
[39] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.

[40] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *Proc. ICML*, 2021, pp. 11492–11501.

[41] J. Shu et al., "Meta-weight-net: Learning an explicit mapping for sample weighting," in *Proc. NeurIPS*, 2019, pp. 1917–1928.
[42] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, pp. 507–516.
[43] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the loss layer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4753–4762.
[44] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*, 2016, pp. 499–515.
[45] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. NeurIPS*, 2018, pp. 8778–8788.
[46] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[47] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017, pp. 2642–2651.
[48] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. NeurIPS*, 2016, pp. 2180–2188.
[49] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, p. 87.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[51] G. Van Horn et al., "The iNaturalist species classification and detection dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8769–8778.
[52] (2018). *iNaturalist 2018 Competition Dataset*. [Online]. Available: https://github.com/visipedia/inat_comp
[53] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4109–4118.
[54] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[55] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[56] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *Proc. ECCV*, 2016, pp. 467–482.
[57] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5177–5186.
[58] 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. (CVPR)*, Jun. 2019, pp. 2537–2546.
[59] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[60] X. Wang et al., "PPISB: A novel network-based algorithm of predicting protein-protein interactions with mixed membership stochastic block-model," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 1–8, Mar. 2022.
[61] L. Hu, P. Hu, X. Luo, X. Yuan, and Z. You, "Incorporating the coevolving information of substrates in predicting HIV-1 protease cleavage sites," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 6, pp. 2017–2028, Nov. 2020.
[62] L. Hu, X. Wang, Y.-A. Huang, P. Hu, and Z.-H. You, "A survey on computational models for predicting protein–protein interactions," *Briefings Bioinf.*, vol. 22, no. 5, pp. 1–10, Sep. 2021.
[63] Y. Yuan, X. Luo, M. Shang, and Z. Wang, "A Kalman-filter-incorporated latent factor analysis model for temporally dynamic sparse data," *IEEE Trans. Cybern.*, early access, Jul. 25, 2022, doi: 10.1109/TCYB.2022.3185117.
[64] X. Luo, H. Wu, and Z. Li, "NeuLFT: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6148–6166, Jun. 2022.

Mengyang Li received the B.Eng. degree from the Zhengzhou University of Aeronautics, Zhengzhou, China, in 2015, and the M.Eng. degree from the Civil Aviation University of China, Tianjin, China, in 2019. He is currently pursuing the Ph.D. degree with Tianjin University, Tianjin, under the supervision of Professor Ou Wu.

His research interests include data augmentation and imbalanced learning.



Fengguang Su received the B.Sc. degree in computational mathematics from Qingdao University, Qingdao, China, in 2020. He is currently pursuing the M.Sc. degree in applied mathematics with Tianjin University, Tianjin, China, advised by Prof. Ou Wu.

His research interests include meta-learning and adversarial attacks.

Ou Wu received the B.Sc. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003, and the M.Sc. and Ph.D. degrees in computer science from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2006 and 2012, respectively.

In 2007, he joined NLPR as an Assistant Professor. In 2017, he became a Full Professor with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include data mining and machine learning.

Ji Zhang (Senior Member, IEEE) is currently a Full Professor with the University of Southern Queensland (USQ), Toowoomba, QLD, Australia, and a Visiting Professor with the Zhejiang Laboratory, Hangzhou, China; Tsukuba University, Tsukuba, Japan; Nanyang Technological University (NTU), Singapore; and Michigan State University (MSU), East Lansing, MI, USA. He has authored more than 230 papers, many appearing in top-tier international journals and conferences. His research interests include data science, big data analytics, data mining, and health informatics.

Dr. Zhang is an IET Fellow, an Australian Endeavor Fellow, and a Queensland International Fellow, Australia.