Class-Level Logit Perturbation

Mengyang Li*, Fengguang Su*, Ou Wu, Ji Zhang

***Abstract*—Features, logits, and labels are the three primary data when a sample passes through a deep neural network. 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 positive/negative 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 single-label and multi-label classification tasks. Extensive experiments on benchmark image classification data sets 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*—Data Augmentation, Long-tail Classification, Multi-label Classification, Adversarial Training.**

I. Introduction

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

In feature perturbation, 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. Another well-known feature perturbation technique is about the generation of adversarial training samples [1], which attracts great attention in various AI applications especially in computer vision [2] and natural language processing [3]. Adversarial samples are those that can fool the learned models. They can be obtained by solving the following objective function:

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

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 $\boldsymbol{x}_{adv}$ is the generated adversarial sample. A number of methods have been proposed to optimize Eq. (1) [1], [4]. 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 [5], generates a new label for each sample according to $\boldsymbol{y}' = \boldsymbol{y} + \lambda(\frac{\boldsymbol{I}}{C} - \boldsymbol{y})$, where $\boldsymbol{y}$ is the one-hot vector label; C is the number of categories; $\boldsymbol{I}$ is a vector with all elements equaling to 1; $(\frac{\boldsymbol{I}}{C} - \boldsymbol{y})$ is the perturbation term; and λ is a hyper-parameter. Other methods such as Boostrapping loss [6], label correction [7], [8], and Meta label corrector [9] can be seen as a type of label perturbation. Mixup [10] 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 almost all deep neural networks (DNNs). Although logits are important in the DNN data pipeline, only several learning methods in data augmentation and long-tail classification directly (without optimization) or implicitly employ class-level logit perturbation. Based on the loss analysis of these methods, the loss variations incurred by logit perturbation are highly related to the purpose of positive/negative 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 multi-label learning tasks, respectively, in this study. Extensive experiments are run on benchmark data sets for single-label classification and multi-label classification tasks. The results show the competitiveness of our methodologies.

Parts of the results in this paper were published originally in its conference version [11]. In our conference version, several classical methods are rediscussed in terms of logit perturbation and positive/negative 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 single-label classification tasks. Experimental results show that our method outperforms existing state-of-the-art methods related to logit perturbation in both contexts. This paper extends our earlier work in several important aspects:

- We conduct a theoretical analysis for the roles of logit perturbation-based explicit negative and positive augmentations in learning for binary classification tasks. Two typical scenarios, namely, class imbalance and variance imbalance, are considered in our analysis.

Manuscript received August, 2022. * denotes the equal contribution. Corresponding author: Ou Wu.

M. Li, F. Su and O. Wu are with National Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: limengyang@tju.edu.cn; fengguangsu@tju.edu.cn; wuou@tju.edu.cn).

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

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

- We extend our LPL algorithm to the multi-label classification, which contains both class and variance imbalances, and empirically validate its effectiveness on multi-label classification benchmarks.
- Extensive experiments on large-scale long-tail data sets such as iNaturalist are performed. Our method LPL still achieves competitive results.

II. Related Work

A. Data Augmentation

Data augmentation is prevailed in almost all deep learning approaches. In its early stage, heuristic operations on raw samples are utilized, such as image flip, image rotation, and word replacing in sentences. Recently, advanced tricks are investigated, such as mixup [10], semantic data augmentation [12], and meta semantic augmentation [13]. In mixup, given a sample $\{\boldsymbol{x}_1, \boldsymbol{y}_1\}$, its perturbation term is $\{\lambda(\boldsymbol{x}_2 - \boldsymbol{x}_1), \lambda(\boldsymbol{y}_2 - \boldsymbol{y}_1)\}$, where λ is a random parameter (not a hyper-parameter), and $\{\boldsymbol{x}_2, \boldsymbol{y}_2\}$ is another randomly selected sample. Hu et al. [14] introduce reinforcement learning to automatically augment data.

In this study, existing data augmentation is called positive data augmentation. Negative data augmentation, which is proposed in this study, may be helpful when we aim 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. [15] design a bilateral-branch network to learn the representations of head and tail samples. The second strategy is to modify the training loss. In this way, the weighting scheme [16] is the most common practice. Relatively larger weights are exerted on the losses of the tail samples. Besides weighting, some recent studies modify the logits to change the whole loss, such as logit adjustment (LA) [17]. This new path achieves higher accuracies in benchmark data corpora compared with weighting [18].

C. Multi-label Classification

Real data usually also contain multiple objectives. Unlike the two single-label classification tasks mentioned above, there are two main challenges in multi-label classification tasks, namely, the co-occurrence of labels and the dominance of negative samples [19], [20]. Li et al. [21] introduce a novel and effective deep metric learning method, which explores the relationship of images and labels by learning a two-way deep distance metric over two embedding spaces. Wei et al. [22] investigate the impact of labels on evaluation metrics for large-scale multi-label learning and propose to restrain labels that have less impact on performance to speed up prediction and reduce model complexity. Wu et al. [19] perturb logits to emphasize the positive samples of tail categories to prevent class-specific overfitting. In multi-label classification task, weighting scheme [23] is also a typically used method.

D. Adversarial Training

Adversarial training is an important way to enhance the robustness of neural networks [24], [25]. The most important step in adversarial training is to generate adversarial training examples in Eq. (1), which can be used to improve the robustness of neural networks. Numerous works have been proposed to generate adversarial examples [1], [4], [26]. Gradient-based attack methods are commonly used [27]. Goodfellow et al. [4] propose to quickly compute adversarial training examples by using the gradient sign. Madry et al. [1] propose projected gradient descent (PGD) to compute the adversarial training samples. PGD executes an iterative computation that performs multiple gradient descent updates with small steps within the perturbation bound ϵ to update the adversarial training samples.

III. Methodology

This section first discusses several typical learning methods related to logit perturbation.

A. Logit Perturbation in Existing Methods

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 multi-label classification, $\boldsymbol{y}_i = [y_{i,1}, y_{i,2}, \cdots, y_{i,C}] \in \{0,1\}^C$. 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 category in S. Without loss of generality, we assume that $\pi_1 > \cdots > \pi_c > \cdots > \pi_C$. Following Menon et al. [17] and Wu et al. [19], we determine the head and tail categories by 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 et al. [20], 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 deep neural network 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.

Logit adjustment (LA) [17]. This method is designed for single-label long-tail classification and achieves competitive performance in benchmark data sets [18]. The employed loss in LA is defined as follows:

$$\mathcal{L} = -\sum\nolimits_i \log \frac{\exp(u_{i,k} + \lambda \log \pi_k)}{\sum_c \exp(u_{i,c} + \lambda \log \pi_c)}, \tag{2}$$

where $u_{i,k}$ is kth element of $\boldsymbol{u}_i$; $y_{i,k}$ is kth element of $\boldsymbol{y}_i$; c and k are the category index; and k satisfies $y_{i,k} = 1$. In Eq. (2), the perturbation term $\boldsymbol{\delta}_i$ is as follows:

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

where $\tilde{\boldsymbol{\delta}}$ is corpus-level vector[2]; $\boldsymbol{\delta}_i$ is sample-level vector; thus $\boldsymbol{\delta}_i$ for all the samples in the corpus $\boldsymbol{S}$ are identical. Eq. (2) can be re-written as follows:

$$\mathcal{L}=-\sum\nolimits_i \log \frac{\exp(u_{i,k})}{\sum_c \exp(u_{i,c}+\lambda(\log \pi_c - \log \pi_k))}. \quad (4)$$

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 of the losses of the rest categories depend on the concrete loss of each sample.

Implicitly semantic data augmentation (ISDA) [12]. ISDA is an explicit data augmentation method for single-label classification. Given a sample $\boldsymbol{x}_i$, ISDA assumes that each (virtual) new sample can be sampled from a distribution $\mathcal{N}(\boldsymbol{x}_i, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\Sigma}_k$ is the co-variance matrix for the kth category. With the M (virtual) new samples for each sample, the loss becomes

$$\mathcal{L}=-\frac{1}{N\cdot M}\sum\nolimits_{i=1}^{N}\sum\nolimits_{m=1}^{M} l(f(\boldsymbol{x}_{i,m},\boldsymbol{W}),\boldsymbol{y}_i), \quad (5)$$

where $\boldsymbol{x}_{i,m}$ is the mth (virtual) new sample for $\boldsymbol{x}_i$. When $M \to +\infty$, the upper bound of the loss in Eq. (5) becomes

$$\mathcal{L}=-\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(u_{i,k})}{\sum\limits_{c=1}^{C}\exp(u_{i,c}+\frac{\lambda}{2}(\boldsymbol{w}_c-\boldsymbol{w}_k)^T\boldsymbol{\Sigma}_k(\boldsymbol{w}_c-\boldsymbol{w}_k))}, \quad (6)$$

where c and k are the category index and k satisfies $\boldsymbol{y}_{i,k}=1$; $\boldsymbol{w}_c$ is the network parameter for the logit vectors and $u_{i,c}={\boldsymbol{w}_c}^T\tilde{\boldsymbol{x}}_{\boldsymbol{i}}+b_c$; $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$ is the output of the last feature encoding layer. In contrast with previous data augmentation methods, ISDA does not generate new samples or features. In Eq. (6), there is an implicit perturbation term $\boldsymbol{\delta_i}$ defined as follows:

$$\boldsymbol{\delta_i}=\tilde{\boldsymbol{\delta}}_k=\frac{\lambda}{2}\begin{bmatrix}(\boldsymbol{w}_1-\boldsymbol{w}_k)^T\boldsymbol{\Sigma}_k(\boldsymbol{w}_1-\boldsymbol{w}_k)\\ \vdots \\ (\boldsymbol{w}_C-\boldsymbol{w}_k)^T\boldsymbol{\Sigma}_k(\boldsymbol{w}_C-\boldsymbol{w}_k)\end{bmatrix}, \quad (7)$$

where $\tilde{\boldsymbol{\delta}}_k$ is class-level vector; thus $\boldsymbol{\delta}_i$ is the same for each class of samples. Each element of $\boldsymbol{\delta}_i$ is non-negative. Therefore, the new loss of each category from Eq. (6) is larger than the loss from the standard CE loss.

LDAM [28]. This method is designed for single-label long-tail classification. It's new loss is defined as

$$\mathcal{L}=-\sum_{i=1}^{N}\log\frac{\exp(u_{i,k}-C(\pi_k)^{-1/4})}{\exp(u_{i,k}-C(\pi_k)^{-\frac{1}{4}})+\sum_{c\neq k}\exp(u_{i,c})}, \quad (8)$$

where k satisfies $y_{i,k}=1$. The perturbation term $\boldsymbol{\delta}_i$ is as follows:

$$\boldsymbol{\delta}_i=\tilde{\boldsymbol{\delta}}_k=\lambda[0,\cdots,-C(\pi_k)^{-\frac{1}{4}},\cdots,0]^T, \quad (9)$$

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

Negative-tolerant Regularization (NTR) [19]. In this method, a multi-label classification task is first decomposed into C independent binary classification tasks. NTR defines the following negative-tolerant binary loss:

$$\begin{aligned}\mathcal{L}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{C}\sum_{c=1}^{C} & y_{i,c}\log(1+\exp(-u_{i,c}+v_c)) \\ & +\frac{1}{\lambda}(1-y_{i,c})\log(1+\exp(\lambda(u_{i,c}-v_c)))\end{aligned}, \quad (10)$$

where $v_c=\psi\log(\frac{N}{N_c}-1)$; λ and ψ are hyper-parameters. In Eq. (10) , an implicit logit perturbation term ($\boldsymbol{\delta}_i$) can also be observed as follows:

$$\boldsymbol{\delta_i}=\tilde{\boldsymbol{\delta}}=-\psi[\log(\frac{N}{N_1}-1),\cdots,(\frac{N}{N_C}-1)]^T. \quad (11)$$

The perturbation is a corpus-level term vector. ψ is non-negative in the experiments conducted by Wu et al. [19]. Therefore, if $N<2N_c$, then samples with label c are dominant and v_c in Eq. (10) is smaller than zero. When $y_{i,c}$=1, the loss will be reduced, and if $y_{i,c}$=0, the loss will be increased. When $N>2N_c$, it is opposite.

Logit Compensation (LC) [20]. LC assumes that logits conform to a normal distribution. The loss of logit compensation is defined as follows:

$$\begin{aligned}\mathcal{L}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{C}\sum_{c=1}^{C} & y_{i,c}\log(1+\exp(-(\sigma_c^p\cdot u_{i,c}+\mu_c^p))) \\ & +(1-y_{i,c})\log(1+\exp(\sigma_c^n\cdot u_{i,c}+\mu_c^n))\end{aligned}, \quad (12)$$

where μ_c^p, σ_c^p, μ_c^n, and σ_c^n ($c\in\{1,\cdots,C\}$) are the mean and variance of the positive and negative samples that can be learned. For the positive samples, the perturbation term δ_i is as follows:

$$\boldsymbol{\delta_i}=\tilde{\boldsymbol{\delta}}=[\mu_1^p,\mu_2^p,\cdots,\mu_C^p]. \quad (13)$$

For the negative samples, the perturbation term δ_i is as follows:

$$\boldsymbol{\delta_i}=\tilde{\boldsymbol{\delta}}=[\mu_1^n,\mu_2^n,\cdots,\mu_C^n]. \quad (14)$$

Both the two perturbation items are corpus-level vectors. In addition, the logit is weighted in accordance with the variance simultaneously. According to the analysis in [20], LC mainly (relatively) increases the loss for positive samples and emphasizes the tail categories.

B. Theoretical Analysis for Logit Perturbation

The losses of the five example methods analyzed in the previous subsection can be written as follows:

$$\mathcal{L}=\sum\nolimits_i l(\boldsymbol{u_i}+\tilde{\boldsymbol{\delta}}_i,\boldsymbol{y_i}). \quad (15)$$

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 data set (CIFAR100 [29]) and two long-tail sets (CIFAR10-LT [30] and CIFAR100-LT [30]) which are introduced in the experimental section. The loss variations of all categories are positive using ISDA. ISDA achieves the worst results on CIFAR100-LT [30] (shown in the experimental parts), indicating that the non-tail-priority augmentation in long-tail problems is ineffective (ISDA achieves relatively better results on CIFAR10-LT [30].).

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

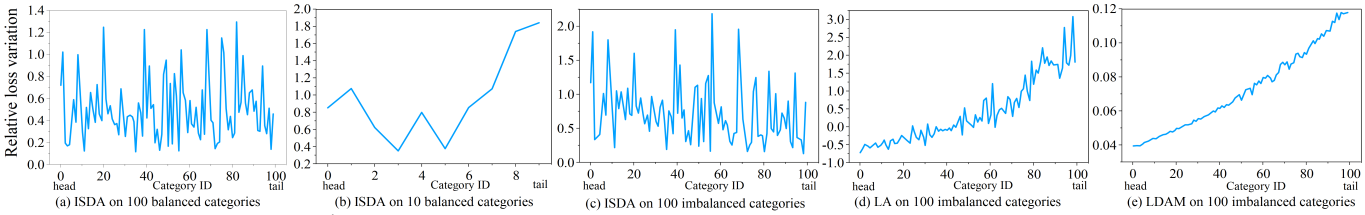


Fig. 1. The relative loss variations ($\frac{l'-l}{l}$) of the three methods on different categories on different data sets. (a) and (b) show the relative loss variation of ISDA on CIAFR100 and CIFAR10 respectively. (c), (d) and (e) show the relative loss variation of ISDA, LA and LDAM on CIAFR100-LT with imbalance ratio 100:1, respectively.

Only the curves on CIFAR100-LT are shown for LA and LDAM because similar trends can be observed on CIFAR10-LT. The loss variations of 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 multi-label classification. The data set COCO-MLT [19] is used. The relative loss variations of positive samples and negative samples are counted separately. In NTR, for positive samples, the loss variations of head categories are less than 0, and those of tail categories are greater than 0. However, for negative samples, the situation is opposite. LC and NTR have the similar trend of the relative loss variation, but the relative loss variation of LC is less than 0.

We propose two conjectures based on the above observations and from a unified logit-perturbation data augmentation viewpoint:

- If one aims to positively augment the samples in a category, the training loss of this category should be increased after logit perturbation. The larger the loss increment is, the greater the augmentation will be. Consequently, the performance of this category will (relatively) increase.
- If one aims to negatively augment the samples in a category, then the training loss of this category should be reduced after logit perturbation. The larger the loss decrement is, the greater the negative augmentation will be. 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 positively augment tail samples and negatively augment head samples. Hence, the losses of tail samples are increased, and those of heads are decreased. ISDA aims to positively augment samples in all categories; thus, the losses for all categories are increased. LDAM aims to positively augment tail samples more than head samples. Hence, the increments of tail categories are larger than those of the head. For multi-label classification task, positive samples and negative samples need to be considered separately. For positive samples, NTR positively augments the tail categories and negatively augments the head categories. For negative samples, the condition is opposite. Therefore, the losses of tails are increased, whereas those of heads are decreased. LC aims to negatively augment all categories. For positive samples, the reductions of head categories are larger than those of the tail. For negative samples, the situation is opposite.

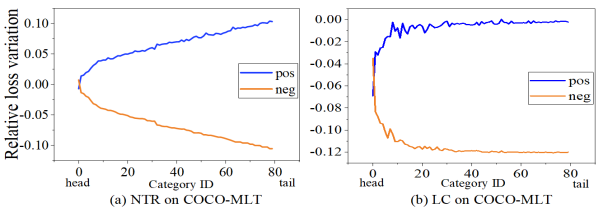


Fig. 2. The relative loss variations ($\frac{l'-l}{l}$) of the two methods on different categories on COCO-MLT. "pos" means the relative loss variations of positive samples. "neg" means the relative loss variations of negative samples.

To theoretically support the two conjectures, a simple binary classification task is employed to quantitatively investigate the relationship among loss variations, performance improvement, and logit perturbation. The binary classification setting established by Xu et al. [31] is followed. The data from each of the two classes $\mathcal{Y} = \{-1, +1\}$ follow two Gaussian distributions, which are centered on $\boldsymbol{\theta} = [\eta, \cdots, \eta]$ (d-dimensional vector and $\eta > 0$) and $-\boldsymbol{\theta}$, respectively. The data follow

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \tag{16}$$

$$\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma_+^2 \boldsymbol{I}\right) & \text{if } y = +1 \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma_-^2 \boldsymbol{I}\right) & \text{if } y = -1 \end{cases}. \tag{17}$$

For a classifier f, the overall standard error is defined as $\mathcal{R}(f) = \Pr.(f(\boldsymbol{x}) \neq y)$. 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 $\sigma_+^2 > \sigma_-^2$ [31]. 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:

$$\tilde{\delta}_c^* = arg \max_{||\tilde{\delta}_c|| < \epsilon_c} \mathrm{E}_{(\boldsymbol{x},y):y=c}[l(u + \tilde{\delta}_c, c)]. \tag{18}$$

The second type is defined as follows:

$$\tilde{\delta}_c^* = arg \min_{||\tilde{\delta}_c|| < \epsilon_c} \mathrm{E}_{(\boldsymbol{x},y):y=c}[l(u + \tilde{\delta}_c, c)], \tag{19}$$

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

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 Eq. (17) for the two

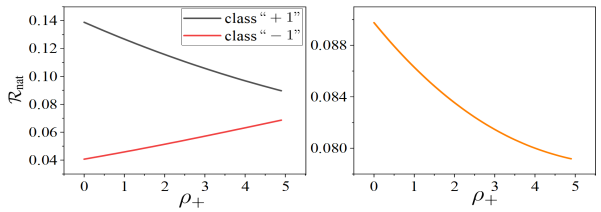


Fig. 3. Left: Natural errors $\mathcal{R}\left(f_{\text{opt}}, -1\right)$ and $\mathcal{R}\left(f_{\text{opt}}, +1\right)$ for the two classes with varied ρ_+. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$ with varied ρ_+.

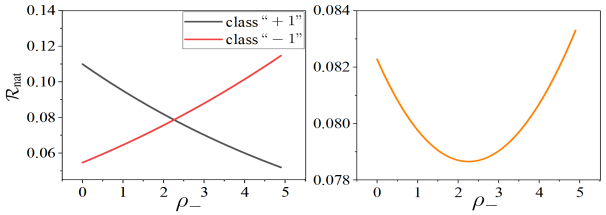


Fig. 4. Left: Natural errors $\mathcal{R}\left(f_{\text{opt}}, -1\right)$ and $\mathcal{R}\left(f_{\text{opt}}, +1\right)$ for the two classes with varied ρ_-. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$ with varied ρ_-.

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 minority and the majority classes, respectively. We have the following theorem:

Theorem 1. *For the abovementioned 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 utilized. The optimal linear classifier f_{opt} that minimizes the average classification error is*

$$f_{opt} = \arg\min_f \Pr.(\mathbb{S}(u + \tilde{\delta}_c^*) \neq y), \tag{20}$$

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 intra-class standard error for the two classes:

$$\begin{aligned}
\mathcal{R}\left(f_{rob}, -1\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} + \frac{log\Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma}\right\}, \\
\mathcal{R}\left(f_{rob}, +1\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} - \frac{log\Gamma}{A} - \frac{\epsilon\rho_+}{\sqrt{d}\sigma}\right\},
\end{aligned} \tag{21}$$

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

The proof is attached in the appendix. Theorem 1 indicates that the logit perturbation parameterized by ϵ and ρ_+ influences performance of both 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 $\Gamma < e^{\frac{((2d-1)\eta-\epsilon)^2}{2d\sigma^2}}$, 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 $\mathcal{R}\left(f_{\text{opt}}, -1\right)$, $\mathcal{R}\left(f_{\text{opt}}, +1\right)$, 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 abovementioned binary classification task, that the perturbation bounds of both classes "+1" and "−1" are assumed to be ϵ $(\rho_+ = 1)$ and $\rho_- \cdot \epsilon$ $(0 \leq \rho_- \cdot \epsilon < \eta)$, respectively. The first perturbation type is utilized for class "+1", and the second perturbation type is utilized for "−1". The optimal linear classifier f_{opt} that minimizes the average classification error is*

$$f_{opt} = \arg\min_f \Pr(\mathbb{S}(u + \tilde{\delta}_c^*) \neq y). \tag{22}$$

It has the intra-class standard error for the two classes:

$$\begin{aligned}
\mathcal{R}\left(f_{opt}, -1\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} + \frac{log\Gamma}{A} + \frac{\epsilon\rho_-}{\sqrt{d}\sigma}\right\}, \\
\mathcal{R}\left(f_{opt}, +1\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} - \frac{log\Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma}\right\},
\end{aligned} \tag{23}$$

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

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 $\mathcal{R}\left(f_{\text{opt}}, -1\right)$, $\mathcal{R}\left(f_{\text{opt}}, +1\right)$, and $\mathcal{R}(f_{\text{opt}})$. Fig. 4 shows the results for the specific learning task discussed in Fig. 3. In this figue, the values of Γ, d, η, ϵ, and σ are 2, 2, 1, 0.2, and 1, respectively.

Theorems 1-2 and Corollaries 1-2 concern the class imbalance issue, i.e., $P_+ \neq P_-$. In addition, another learning scenario is also explored. The variances of the data distributions in Eq. (17) 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 abovementioned binary classification task, the bounds of classes "+1" and "−1" are assumed to be $\rho_+ \cdot \epsilon$ and $\rho_- \cdot \epsilon$ $(0 \leq \rho_+, \rho_- < \frac{\eta}{\epsilon})$, respectively. Only the first perturbation type is utilized. The optimal linear classifier f_{opt} that minimizes the average classification error is*

$$f_{opt} = \arg\min_f \Pr.(\mathbb{S}(u + \tilde{\delta}_c^*) \neq y), \tag{24}$$

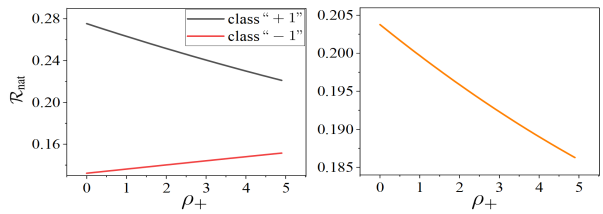


Fig. 5. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ for the two classes with varied ρ_+. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$ with varied ρ_+.

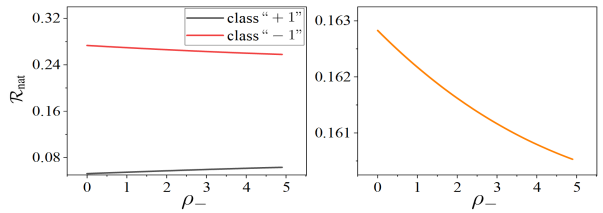


Fig. 6. Left: Natural errors $\mathcal{R}(f_{\text{opt}}, -1)$ and $\mathcal{R}(f_{\text{opt}}, +1)$ for the two classes with varied ρ_-. Right: Total natural error $\mathcal{R}(f_{\text{opt}})$ with varied ρ_-.

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

$$\begin{aligned}
&\mathcal{R}(f_{opt}, +1)\\
&= \Pr.\left\{\mathcal{N}(0,1) < -K\sqrt{B^2 + q(K,\Gamma)} - B - \frac{\epsilon\cdot\rho_+}{\sqrt{d}\sigma})\right\},\\
&\mathcal{R}(f_{opt}, -1)\\
&= \Pr.\left\{\mathcal{N}(0,1) < KB + \sqrt{B^2 + q(K,\Gamma)} - \frac{\epsilon\cdot\rho_-}{K\sqrt{d}\sigma}\right\},
\end{aligned} \tag{25}$$

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

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,*

- *if* $Ke^{\frac{(2d\eta-\epsilon)^2}{2dK^2\sigma^2}} < \Gamma < Ke^{\frac{2d\eta^2}{(K^2-1)\sigma^2}}$, *then* $\mathcal{R}(f_{opt}, +1) > \mathcal{R}(f_{opt}, -1)$. *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;*
- *if* $K > \Gamma$, *then* $\mathcal{R}(f_{opt}, +1) < \mathcal{R}(f_{opt}, -1)$. *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 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 decreases 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.

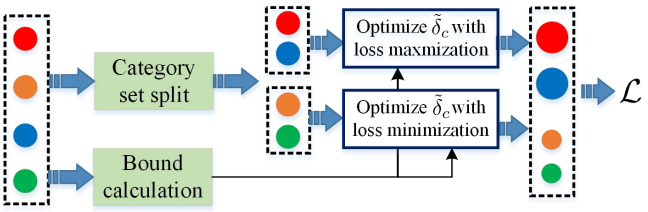


Fig. 7. Overview of the logit perturbation-based new loss. Four solid circles denote four categories. Two categories are positively augmented via loss maximization and the rest two are negatively augmented via minimization.

When both types are utilized, 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 with logit perturbation:

$$\begin{aligned}
\mathcal{L} = &\sum_{c\in\mathcal{N}_a}\sum_{\boldsymbol{x}_i\in S_c}\min_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} l(\text{softmax}(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i)\\
&+ \sum_{c\in\mathcal{P}_a}\sum_{\boldsymbol{x}_i\in S_c}\max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c} l(\text{softmax}(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i),
\end{aligned} \tag{26}$$

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 negatively augmented; $\mathcal{P}_a$ is the index set of categories which should be positively 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 Eq. (26).

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 Eq. (26). Category set split determines the categories that should be positively or negatively augmented. Meanwhile, the value of ϵ_c determines the augmentation extent.

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, Eq. (26) becomes the following compact form:

$$\begin{aligned}
\mathcal{L} = &\sum\nolimits_c \{\mathbb{S}(\tau - \bar{q}_c)\times\\
&\sum_{\boldsymbol{x}_i\in S_c}\max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c}[l(\text{softmax}(\boldsymbol{u}_i + \tilde{\boldsymbol{\delta}}_c), \boldsymbol{y}_i)\mathbb{S}(\tau - \bar{q}_c)]\},
\end{aligned} \tag{27}$$

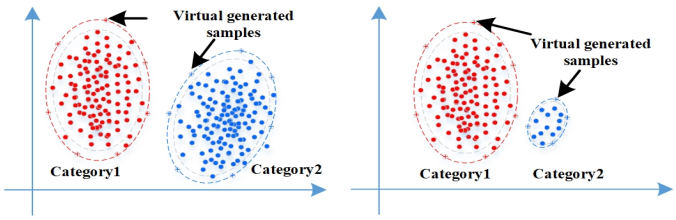


Fig. 8. Illustrative example for ISDA. Both categories are positively augmented (new samples are virtually generated) according to feature distributions.

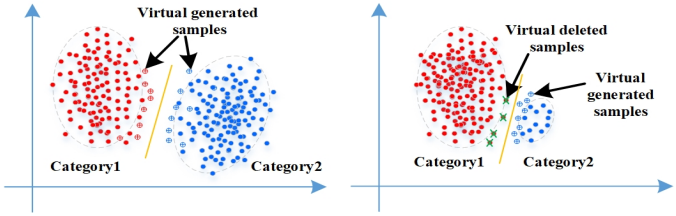


Fig. 9. Illustrative example for LPL. Samples near the classification boundary are virtually generated or deleted.

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

$$\bar{q}_c=\frac{1}{N_c}\sum_{\boldsymbol{x}_i\in S_c} q_{i,c}=\frac{1}{N_c}\sum_{\boldsymbol{x}_i\in S_c}\frac{\exp(u_{i,c})}{\sum_{c'}\exp(u_{i,c'})}. \tag{28}$$

When $\tau=\text{mean}(\bar{q}_c)=\sum_{c=1}^{C}\bar{q}_c/C$, Eq. (27) indicates that if the performance of a category is below the mean performance, it will be positively augmented. Meanwhile, when the performance is above the mean, it will be negatively augmented. When $\tau>\max_c\{\bar{q}_c\}$, all the categories will be positively 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 category. Eq. (26) becomes the following compact form:

$$\mathcal{L}=\sum_c\{\mathbb{S}(c-\tau)\times \sum_{\boldsymbol{x}_i\in S_c}\max_{\|\tilde{\boldsymbol{\delta}}_c\|\le\epsilon_c}[l(\text{softmax}(\boldsymbol{u}_i+\tilde{\boldsymbol{\delta}}_c),\boldsymbol{y}_i)\mathbb{S}(c-\tau)]\}, \tag{29}$$

where τ is a threshold for the category index and $y_{i,c}=1$. In Eq. (29), the tail categories locate in $\mathcal{P}_a$ and will be positively augmented.

Eqs. (27) and (29) can be solved with an optimization approach similar to PGD [1]. We propose a more specific optimization method called PGD-like optimization based on PGD. According to the derivative of the cross-entropy loss function with respect to logit vector, our PGD-like optimization method can be implemented simply. First, we have

$$\left.\frac{\partial l(\text{softmax}(\boldsymbol{u}_i+\tilde{\boldsymbol{\delta}}_c),\boldsymbol{y}_i)}{\partial\tilde{\boldsymbol{\delta}}_c}\right|_{\mathbf{0}}=\text{softmax}(\boldsymbol{u}_i)-\boldsymbol{y}_i. \tag{30}$$

In the maximization of Eqs. (27) and (29), $\tilde{\boldsymbol{\delta}}_c$ is updated by

$$\tilde{\boldsymbol{\delta}}_c=\frac{\alpha}{N_c}\sum_{j:y_{j,c}=1}(\text{softmax}(\boldsymbol{u}_j)-\boldsymbol{y}_j), \tag{31}$$

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: Calculate $\left.\frac{\partial l(\text{softmax}(\boldsymbol{u}_i^k+\tilde{\boldsymbol{\delta}}_c),\boldsymbol{y}_i)}{\partial\tilde{\boldsymbol{\delta}}_c}\right|_{\mathbf{0}}=\text{softmax}(\boldsymbol{u}_i^k)-\boldsymbol{y}_i$.
5: Calculate $\tilde{\boldsymbol{\delta}}_c^{k+1}$ according to Eq. (31) for maximization and Eq. (32) for minimization;
6: $\boldsymbol{u}_i^{k+1}:=\boldsymbol{u}_i^k+\tilde{\boldsymbol{\delta}}_c^{k+1}$.
7: **end for**

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

where α is the hyper-parameter. In the minimization part, $\tilde{\boldsymbol{\delta}}_c$ is updated by

$$\tilde{\boldsymbol{\delta}}_c=-\frac{\alpha}{N_c}\sum_{j:y_{j,c}=1}(\text{softmax}(\boldsymbol{u}_j)-\boldsymbol{y}_j). \tag{32}$$

The PGD-like optimization in Algorithm 1 contains two hyper-parameters, namely, step size and #steps. Let α be the step size, and K_c be the number of steps(#steps) for category c. On the balanced classification, the α is searched in $\{0.01, 0.02, 0.03\}$. With step size, the PGD-like optimization is detailed in Algorithm 1.

Bound calculation. The category with a relatively low/high performance should be more positively/negatively augmented; the category closer to the tail/head should be more positively/negatively augmented. We define

$$\epsilon_c=\epsilon+\Delta\epsilon\,|\tau-\bar{q}_c|\,, \quad \text{or } \epsilon_c=\begin{cases}\epsilon+\Delta\epsilon\frac{\bar{q}_c}{\bar{q}_1} & c\le\tau\\ \epsilon+\Delta\epsilon\frac{\bar{q}_C}{\bar{q}_c} & c>\tau\end{cases}. \tag{33}$$

In Eq. (33), 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 Eq. (33) equals to zero, the bound is fixed. The algorithmic steps of our LPL for single-label learning are in Algorithm 2.

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 positive augmentation for head category may further hurt the performance of the tail category. ISDA fails in the long-tail problem. Li et al. [13] leverage meta learning to adapt ISDA for the long-tail problem.

In contrast with the above-mentioned methods, our proposed LPL method conducts positive or negative 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 positively augmented or virtual samples are generated for this category). By contrast, loss minimization will force the category to be far from the boundary (i.e.,

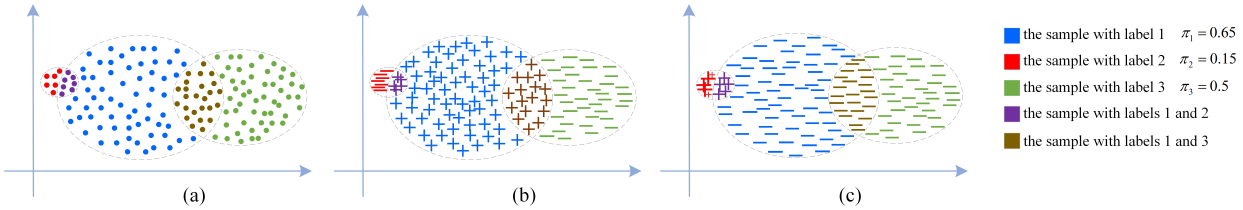


Fig. 10. An illustrative example for the variance imbalance and class imbalance in multi-label learning. "+" means the positive samples. "−" means the negative samples. (a) shows a multi-label learning task ($C = 3$). Different colors mean those samples with one label or more labels. (b) shows the case of variance imbalance. (c) shows the case of class imbalance.

the category is negatively augmented or samples are virtually deleted for this category). Fig. 9 shows an illustrative example.

D. Logit Perturbation Method (LPL) for Multi-label Learning

Multi-label learning is usually decomposed into C binary learning tasks. Compared with single-label learning, 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 will be larger than that of the positive samples as shown in Fig. 10 (b). Theoretically, the negative samples require the first type of logit perturbation and the positive samples require the second type of logit perturbation. Second, class imbalance may exist in each of the C tasks as shown in Fig. 10 (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 contradiction for the two cases discussed above. 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 is dependent of the primary challenge on the class or variance imbalances. Consequently, we extend Eq. (29) into the following form for multi-label learning.

$$\begin{aligned}\mathcal{L} = \frac{1}{C\times N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)}\sum_{c=1}^{C}\mathbb{S}(c-\tau)\times \\ \{\max_{|\tilde{\delta}_c|\le\epsilon_c} y_{i,c}\log(1+e^{-u_{i,c}+\tilde{\delta}_c})\times\mathbb{S}(c-\tau) \\ +\min_{|\tilde{\delta}_c|\le\epsilon_c}(1-y_{i,c})\log(1+e^{u_{i,c}-\tilde{\delta}_c})\times\mathbb{S}(c-\tau)\},\end{aligned} \tag{34}$$

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 an appropriate value of τ. There are three typical settings for τ as shown as follows:

- If τ is set as zero, $\mathbb{S}(c-\tau)\equiv 1$. Eq. (34) becomes

$$\begin{aligned}\mathcal{L} = \frac{1}{C\times N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)}\sum_{c=1}^{C}\{\max_{|\tilde{\delta}_c|\le\epsilon_c} y_{i,c}\log(1+e^{-u_{i,c}+\tilde{\delta}_c}) \\ +\min_{|\tilde{\delta}_c|\le\epsilon_c}(1-y_{i,c})\log(1+e^{u_{i,c}-\tilde{\delta}_c})\}.\end{aligned} \tag{35}$$

  In this situation, the positive samples of all the C tasks perform the first type of logit perturbation, indicating that class imbalance is the primary concern in all tasks.
- If τ is set as $C+1$, then $\mathbb{S}(c-\tau)\equiv -1$. Eq. (34) becomes

$$\begin{aligned}\mathcal{L} = \frac{1}{C\times N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)}\sum_{c=1}^{C}\{\min_{|\tilde{\delta}_c|\le\epsilon_c} y_{i,c}\log(1+e^{-u_{i,c}+\tilde{\delta}_c}) \\ +\max_{|\tilde{\delta}_c|\le\epsilon_c}(1-y_{i,c})\log(1+e^{u_{i,c}-\tilde{\delta}_c})\},\end{aligned} \tag{36}$$

  In this situation, the negative samples of all the C binary tasks perform the second type of logit perturbation, indicating that variance imbalance is the primary concern in all tasks.
- 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 optimization for the cth binary task becomes

$$\begin{aligned}\mathcal{L}_c = \frac{1}{N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)}\{\min_{|\tilde{\delta}_c|\le\epsilon_c} y_{i,c}\log(1+e^{-u_{i,c}+\tilde{\delta}_c}) \\ +\max_{|\tilde{\delta}_c|\le\epsilon_c}(1-y_{i,c})\log(1+e^{u_{i,c}-\tilde{\delta}_c})\},\end{aligned} \tag{37}$$

  which indicates that 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 optimization for the cth binary task becomes

$$\begin{aligned}\mathcal{L}_c = \frac{1}{N}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)}\{\max_{|\tilde{\delta}_c|\le\epsilon_c} y_{i,c}\log(1+e^{-u_{i,c}+\tilde{\delta}_c}) \\ +\min_{|\tilde{\delta}_c|\le\epsilon_c}(1-y_{i,c})\log(1+e^{u_{i,c}-\tilde{\delta}_c})\},\end{aligned} \tag{38}$$

  which presents that 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.

Algorithm 2 Learning to Perturb Logits (LPL)

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

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 Eq. (33) if varied bounds are used.
6: Infer $\tilde{\boldsymbol{\delta}}_c$ for each category using a PGD-like optimization method for Eq. (27) in balanced classification, Eq. (29) in long-tail classification, or Eq. (34) in multi-label classification.
7: Update the logits for each sample and the loss.
8: Update $\boldsymbol{W}$ with SGD.
9: **end for**

Output: $\boldsymbol{W}$

The third setting is adopted in our experiments. Similarly, we can perform PGD-like maximization and minimization as Algorithm 1. According to Eq. (34), for positive samples, the derivative of the loss with respect to $\tilde{\delta}_c$ is as follows.

$$\left.\frac{\partial \log(1+e^{-u_{i,c}+\tilde{\delta}_c})}{\partial \tilde{\delta}_c}\right|_{\mathbf{0}} = 1 - \text{sigmoid}(-u_{i,c}). \tag{39}$$

For negative sample, the derivative of the loss with respect to $\tilde{\delta}_c$ is as follows.

$$\left.\frac{\partial \log(1+e^{u_{i,c}-\tilde{\delta}_c})}{\partial \tilde{\delta}_c}\right|_{\mathbf{0}} = \text{sigmoid}(u_{i,c}) - 1. \tag{40}$$

According to Eq. (34), we use Eqs. (39) and (40) to calculate $\tilde{\delta}_c$ as follows.

$$\begin{aligned}\tilde{\delta}_c = \frac{\alpha}{C \times N} \sum_{i=1}^{N} \{ & y_{i,c}(\text{sigmoid}(-u_{i,c}) - 1) \\ & + (1 - y_{i,c})(\text{sigmoid}(u_{i,c}) - 1)\} \times \mathbb{S}(c - \tau)\}.\end{aligned} \tag{41}$$

where α is the step size. Since each image is treated as C binary classification tasks, we can further simplify Eq. (41). Positive and negative samples for each of C tasks share the same $\tilde{\delta}_c$ for the class c. Obviously, according to Eqs. (39) and (40), $1-\text{sigmoid}(-u_{i,c}) \geq 0$ and $\text{sigmoid}(u_{i,c})-1 \leq 0$ holds. The term $\tilde{\delta}_c$ is a scalar. Therefore, when the perturbation bound ϵ_c is given by Eq. (33), we can easily obtain $\tilde{\delta}_c = \epsilon_c$. Then the logit perturbation for multi-label learning can be easily calculated.

The algorithmic steps of our LPL for multi-label learning are also in Algorithm 2.

IV. Experiments

Our proposed LPL is first evaluated on data augmentation, long-tail classification and multi-label classification tasks. The properties of LPL are then analyzed with more experiments. A Linux platform with four RTX3090 graphics cards is used, and each graphics card has a capacity of 24 GB.

TABLE I
Mean values and standard deviations of the test Top-1 errors for all the involved methods on CIFAR10.

Method	Wide-ResNet-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	3.58 ± 0.15%	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%**

TABLE II
Mean values and standard deviations of the test Top-1 errors for all the involved methods on CIFAR100.

Method	Wide-ResNet-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	17.98 ± 0.15%	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%**	**25.87 ± 0.07%**

A. Experiments on Data Augmentation

Datasets and competing methods. In this subsection, two benchmark image classification data sets, namely, CIFAR10 [29] and CIFAR100 [29], are used. Both data consist of 32×32 natural images in 10 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 [12] are followed. Several classical and state-of-the-art robust loss functions and (semantic) data augmentation methods are compared: Large-margin loss [32], Disturb label [33], Focal Loss [23], Center loss [34], Lq loss [35], CGAN [36], ACGAN [37], infoGAN [38], ISDA, and ISDA + Dropout.

The Wide-ResNet-28 [39] and ResNet-110 [40] are used as the base neural networks. Considering that the training/testing configuration is fixed for both sets, the results of the above competing methods reported in the ISDA paper [12] 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 methods have two variants.

- LPL (mean+fixed bound). In this version, the optimization in Eq. (27) 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 categories during optimization. It is searched in $\{0.1, 0.2, 0.3, 0.4\}$.

- LPL (mean+varied bound). In this version, the optimization in Eq. (27) is used. Theoretically, varied bound means that the value of ϵ_c is varied according to Eq. (33). 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. The value of $\Delta\epsilon$ is searched in $\{0.1, 0.2\}$.

The Top-1 error is used as the evaluation metric. The performances of the base neural networks with the standard cross-entropy loss are re-run before running our methods to conduct a fair comparison. Almost identical results are obtained compared with the published results in the ISDA paper.

Results. Tables I and II present the results of all competing methods on the CIFAR10 and CIFAR100, respectively. Our LPL method (two versions) achieves the best performance almost under both the two base neural networks. ISDA achieves the second-best performance. Only in the case of Wide-ResNet-28-10 on CIFAR100, LPL (mean+fixed ϵ_c) is inferior to ISDA. However, the former still achieves the fourth lowest error.

The results of LPL with varied bounds are better than those of LPL with fixed bounds. This comparison indicates the rationality of our motivation that the category with relatively low (high) performance should be more positively (negatively) 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

Datasets and competing methods. In comparison with the conference version of the paper, we supplement the experiments with real-world data sets. In the synthetic data set experiment, the long-tail versions of CIFAR10 and CIFAR100 compiled by Cui et al. [30] are used and called CIFAR10-LT and CIFAR100-LT, respectively, The training and testing configurations used in [17] are followed. In the real-world data set experiment, large-scale data sets iNaturalist 2017 (iNat2017) [41] and iNaturalist 2018 (iNat2018) [42] with extremely imbalanced class distributions are used. iNat2017 includes 579,184 training images in 5,089 classes with an imbalance factor of 3919/9, while iNat2018 is composed of 435,713 images from 8,142 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 [12], Class-balanced fine-tuning [43], Meta-weight net [44], Focal loss [23], Class-balanced focal loss [30], LDAM [28], LDAM-DAR [28], ISDA, and LA.

In the synthetic data set experiment, Menon et al. [17] 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. [13]. The hyper-parameter λ in LA is searched in $\{0.5, 1, 1.5, 2, 2.5\}$ according to the suggestion in [17]. Similar to the experiments in [17], ResNet-32 [40] is used as the base network. The results of ISDA, LA, and LPL are the average of five repeated runs.

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

Ratio	100:1	10:1
Class-balanced CE 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%**

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

Ratio	100:1	10:1
Class-balanced CE 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%**

In the real-world data set experiment, the results of the above competing methods reported in [17] are directly presented. The results of LA on iNat2018 are from the original paper [17]. The other results, such as ISDA+dropout and LA on iNat2017, are obtained by running their released codes. Likewise, the hyper-parameter λ in LA is searched in $\{0.5, 1, 1.5, 2, 2.5\}$. Similar to the experiments in [19], ResNet-50 [40] is used as the base network. All results are the average of five repeated runs.

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 $\{1.5, 2.5, 5\}$. In the varied bound version, the value of ϵ is set to 0, and $\Delta\varepsilon$ is searched in $\{1.0, 2.0, 3.0\}$. Only one meta-based method, Meta-weight net, is involved, because we mainly aim to compare methods that only modify the training loss. In addition, meta-based methods require an auxiliary high-quality validation set [13]. Other methods, such as BBN [15], which focus on the new network structure are also not included in the comparisons.

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, LPL (varied τ + varied ϵ_c) yields the lowest Top-1 errors. It exceeds the best competing method LA by 0.36% and 2.63% on the ratios 100:1 and 10:1, respectively. Table IV shows the results of all the methods on the CIFAR10-LT data. LPL (varied τ + varied ϵ_c) still

TABLE V
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%
RW	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%
RS	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%

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

Method	iNat2017	iNat2018
Class-balanced CE loss	42.02%	33.57%
Class-balanced fine-tuning	–	–
Meta-weight net	–	–
Focal Loss	–	–
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**%

obtains the lowest Top-1 errors on both ratios. Table VI shows the results of all the methods on the iNat2017 and iNat2018. For real-world long-tail datasets, it still exceeds LA 0.89% and 0.97%, respectively. On all the comparisons, the semantic augmentation method ISDA obtains poor results. On CIFAR100-LT, ISDA achieves the worst performances on both ratios. This result is expected because ISDA aims to positively augment all categories equally and does not favor tail categories, which may lead to tail categories suffering from this positive 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 the head ones. That is, larger augmentations are exerted on tail categories.

We listed the Top-1 errors of LA and LPL (varied τ + varied ϵ_c) on Table VII 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 one, indicating that there is no imbalance, LA will lose effect; however, our LPL can still augment the training data effectively.

TABLE VII
THE ERROR REDUCTION OF LPL (VARIED τ+VARIED ϵ) OVER LA ON THE TWO DATA SETS.

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%)

C. Experiments on Multi-label Classification

Datasets and competing methods. In this part, the long-tail multi-label versions of VOC [45] and MS-COCO [46] compiled by Wu et al. [19] are used and called VOC-MLT and COCO-MLT, respectively. The training and test configurations used in [19] are followed. The training set of VOC-MLT is sampled from train-val set of VOC2012, containing 1142 images from 20 categories, with a maximum of 775 images per category and a minimum of 4 images per category. A toal of 4952 images from the VOC2007 test set are used for evaluation. COCO-MLT is sampled from MS COCO-2017 dataset, containing 1909 images from 80 categories, with a maximum of 1128 images per category and a minimum of 6 images per category. 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 comparison experiments on the basis of R-BCE [19]. Several classical and state-of-the-art robust loss functions and multi-label methods are compared: Empirical Risk Minimization (ERM), Re-Weighting (RW), Focal Loss [23], Re-Sampling (RS) [47], ML-GCN [48], OLTR [49], LDAM [28], CB-Focal [30], R-BCE [19], R-BCE-Focal [19], R-BCE + NTR [19], R-BCE-Focal + NTR [19], R-BCE + LC [20], R-BCE-Focal + LC [20].

Wu et al. [19] released the training data and code. Hence, their data and reported results for the above competing methods are directly presented. The experimental results of LC are reimplemented from the original paper's formula. Similar to the experiments in [19], ResNet-50 [40] is used as the base network.

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

TABLE VIII
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%**
Wide-ResNet-16-8+ISDA	11.0M	4.04 ± 0.29%	19.91 ± 0.21%
Wide-ResNet-16-8+LPL (mean + fixed ϵ_c)	11.0M	3.97 ± 0.09%	19.87 ± 0.02%
Wide-ResNet-16-8+LPL (mean + varied ϵ_c)	11.0M	**3.93 ± 0.10%**	**19.83 ± 0.09%**

in $\{0.05, 0.1, 0.1\}$. 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 and optimizer follow NTR.

Results. The evaluation metric mAP is used. Table V 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, respexctively. 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 for Our Method

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:

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

TABLE IX
TEST TOP-1 ERRORS OF THREE METHODS ON TWO DATA SETS.

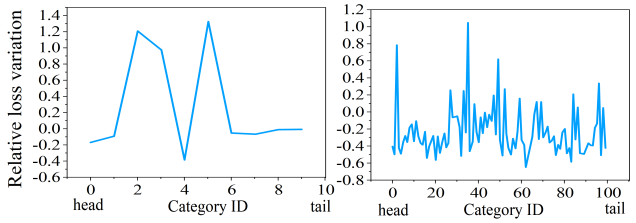
Method	CIFAR10-LT100	CIFAR100-LT100
LA	22.33%	56.11%
LPL	22.05%	55.75%
LA+LPL	**21.46%**	**53.89%**

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

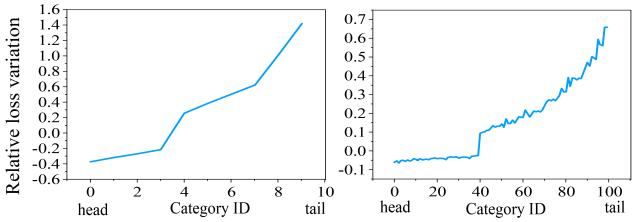
Method	MS-COCO
R-BCE+NTR	83.7%
R-BCE+LC	84.5%
R-BCE+LPL(varied τ + varied ϵ_c)	**85.4%**

where $\log\boldsymbol{\pi} = [\log\pi_1, \cdots, \log\pi_C]$. 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 IX. 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.

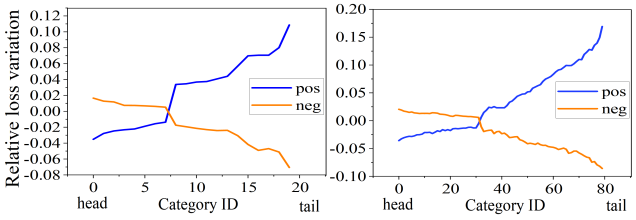
The combination method LA+LPL achieves the lowest errors on both comparisons, indicating that our LPL can further improve the performances of existing SOTA methods. ISDA can likewise be improved with the same manner.



(a) Balanced data sets on CIFAR10 (left) and CIFAR100 (right)



(b) Long-tail data sets on CIFAR10-LT (left) and CIFAR100-LT (right)



(c) Multi-label data sets on VOC-MLT (left) and COCO-MLT (right)

Fig. 11. Relative loss variations of our LPL on two balanced data sets, two long-tail data sets, and two multi-label data sets. "pos" means the relative loss variations of positive samples. "neg" means the relative loss variations of negative samples.

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 VIII shows the comparisons between ISDA and LPL (two variants) on three additional base DNN models, namely, SE-ResNet110 [50], Wide-ResNet-16 [39], 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 data sets under all the five base models.

Loss variations of LPL during training. For single-label classification, we plot the loss variations of LPL on two balanced and two long-tail data sets to assess whether our method LPL is in accordance with the two conjectures. The curves are shown in Fig. 11 (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 multi-label classification, Fig. 11 (c) shows the results. In comparison with NTR and LC, our method LPL focuses more on the tail categories according to the trends of relative loss reduction.

Performances of LPL under different τ and ϵ_c. Both the threshold for category set split and the bound for augmentation extent are two important hyper-parameters in LPL. Based on our experiments, the following observations are obtained. On the balanced data sets, the results are relatively stable when the bound locates in [0.1, 0.5]; when the threshold is searched around the mean$(\bar{q}_c)$, the results are usually better. On the long-tail data sets, the results are relatively stable when the bound locates in [1.5, 5.0]. When the threshold is searched in $\{0.4C,\ 0.5C,\ 0.6C\}$, the results are usually good in our experiment. Long-tail problems require larger extent of data augmentation.

More comparisons with NTR and LC. We also compare our method with NTR and LC on the original multi-label dataset MS-COCO. MS-COCO contains 122,218 images with 80 different labels, which is divided to a training set with 82,081 images and a test set with 40,137 images. In this part, ResNet-110 is used as backbone network and the input size is 448×448. Other setups follow Subsection C in Section IV. Table X shows the results. The evaluation metric mAP is used. Again, our method achieves the competitive results. R-BCE+LPL(varied τ + varied ϵ_c) exceeds R-BCE+NTR and R-BCE+LC 1.7 % and 0.9 % respectively.

V. Conclusions

This study investigates the class-level logit perturbation in deep learning. Two conjectures for the relationship between (logit perturbation-incurred) loss increment/decrement and positive/negative 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 (LPL) during DNN training for both single-label and multi-label learning tasks. Two key components of LPL, namely, category-set split and boundary calculation, are investigated. Extensive experiments on data augmentation (for balanced classification), long-tail classification, and multi-label classification are conducted. LPL achieves the best performances in both situations under different basic networks. Existing methods with logit perturbation (e.g. LA) can also be improved by using our method.

References

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

[2] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *CVPR*, 2020, pp. 819–828.

[3] 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 *AAAI*, 2020, pp. 8018–8025.

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

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

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

[7] G. Patrini, A. Rozzah, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *CVPR*, 2017, pp. 1944–1952.

[8] 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 *CVPR*, 2021, pp. 752–761.

[9] Y. Wu, J. Shu, Q. Xie, Q. Zhao, and D. Meng, "Learning to purify noisy labels via meta soft label corrector," in *AAAI*, 2021, pp. 10 388–10 396.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[11] M. Li, F. Su, O. Wu, and J. Zhang, "Logit perturbation," in *AAAI*, 2022, pp. 10 388–10 396.

[12] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, and C. Wu, "Implicit semantic data augmentation for deep networks," in *NeurIPS*, 2019, pp. 12 635–12 644.

[13] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta semantic augmentation for long-tailed visual recognition," in *CVPR*, 2021, pp. 5212–5221.

[14] Z. Hu, B. Tan, R. Salakhutdinov, T. Mitchell, and E. P. Xing, "Learning data manipulation for augmentation and weighting," in *NeurIPS*, 2019, pp. 15 738–15 749.

[15] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *CVPR*, 2020, pp. 9719–9728.

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

[17] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *ICLR*, 2021.

[18] T. Wu, Z. Liu, Q. Huang, Y. Wang, and D. Lin, "Adversarial robustness under long-tailed distribution," in *CVPR*, 2021, pp. 8659–8668.

[19] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classiffication in long-tailed datasets," in *ECCV*, 2020.

[20] H. Guo and S. Wang, "Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings," in *CVPR*, 2021, pp. 15 089–15 098.

[21] C. Li, C. Liu, L. Duan, P. Gao, and K. Zheng, "Reconstruction regularized deep metric learning for multi-label image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2294–2303, 2020.

[22] T. Wei and Y.-F. Li, "Does tail label help for large-scale multi-label learning?" *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2315–2324, 2020.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[24] Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, and J. Y. Zou, "Adversarial training helps transfer learning via better representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25 179–25 191, 2021.

[25] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *ICCV*, 2021, pp. 15 721–15 730.

[26] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *ECCV*. Springer, 2020, pp. 484–501.

[27] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges, "Adversarial example detection for dnn models: A review and experimental comparison," *Artificial Intelligence Review*, pp. 1–60, 2022.

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

[29] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[30] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277.

[31] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *NeurIPS*, 2021, pp. 11 492–11 501.

[32] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, 2016, pp. 507–516.

[33] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "Disturblabel: Regularizing cnn on the loss layer," in *CVPR*, 2016, pp. 4753–4762.

[34] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*. Springer, 2016, pp. 499–515.

[35] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *NeurIPS*, 2018, pp. 8778–8788.

[36] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[37] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017, pp. 2642–2651.

[38] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016, pp. 2180–2188.

[39] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016, pp. 87.1–87.12.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[41] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *CVPR*, 2018, pp. 8769–8778.

[42] "inaturalist 2018 competition dataset," https://github.com/visipedia/inat_comp, 2018.

[43] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *CVPR*, 2018, pp. 4109–4118.

[44] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *NeurIPS*, 2019, pp. 1917–1928.

[45] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[47] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *ECCV*. Springer, 2016, pp. 467–482.

[48] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *CVPR*, 2019, pp. 5177–5186.

[49] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *CVPR*, 2019, pp. 2537–2546.

[50] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

APPENDIX

A. Proof for Theorem 1

Proof. Xu et al. [31] proved that $\boldsymbol{w} = \mathbf{1}$ when the data distribution in Eq. (17) is given (Lemma 1 in [31]). According to Lemma 1 in [31], 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 Eq. (20) can be written as follows.

$$b^* = \arg\min_b \Pr.(\mathbb{S}(\sum_{i=1}^{d} x_i + b + \tilde{\delta}_c^*) \neq y). \tag{42}$$

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.

$$\begin{aligned}
\mathcal{R}_{\text{lp}}(f) &\propto \Gamma \cdot \Pr.(\exists \|\tilde{\delta}_-\| \leq \epsilon,\ \mathbb{S}(u+\tilde{\delta}_-) \neq -1 \mid y=-1) \\
&+ \Pr.(\exists \|\tilde{\delta}_+\| \leq \epsilon \cdot \rho_+,\ \mathbb{S}(u+\tilde{\delta}_+) \neq +1 \mid y=+1) \\
&= \Gamma \cdot \max_{\|\tilde{\delta}_-\| \leq \epsilon} \Pr.(\mathbb{S}(u+\tilde{\delta}_-) \neq -1 \mid y=-1) \\
&+ \max_{\|\tilde{\delta}_+\| \leq \epsilon \cdot \rho_+} \Pr.(\mathbb{S}(u+\tilde{\delta}_+) \neq +1 \mid y=+1) \\
&= \Gamma \cdot \Pr.(\mathbb{S}(u+\epsilon) \neq -1 \mid y=-1) \\
&+ \Pr.(\mathbb{S}(u-\epsilon \cdot \rho_+) \neq +1 \mid y=+1) \\
&= \Gamma \cdot \Pr.\left\{\sum_{i=1}^{d} x_i + b + \epsilon > 0 \mid y=-1\right\} \\
&+ \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon \cdot \rho_+ < 0 \mid y=+1\right\} \\
&= \Gamma \cdot \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b+\epsilon}{\sqrt{d}\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}(0,1) < -(\frac{\sqrt{d}\eta}{\sigma} + \frac{b-\epsilon \cdot \rho_+}{\sqrt{d}\sigma})\right\}.
\end{aligned} \tag{43}$$

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

$$b^* = \frac{1}{2}\epsilon(\rho - 1) + \frac{d\sigma^2 \log\Gamma}{\epsilon - 2d\eta + \epsilon \cdot \rho_+}. \tag{44}$$

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

$$\begin{aligned}
\mathcal{R}\left(f_{\text{opt}}, -1\right) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} \\
&= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} + \frac{\log\Gamma}{A} - \frac{\epsilon}{\sqrt{d}\sigma}\right\}, \\
\mathcal{R}\left(f_{\text{opt}}, +1\right) &= \Pr.\left\{\mathcal{N}(0,1) < -(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma})\right\} \\
&= \Pr.\left\{\mathcal{N}(0,1) < \frac{A}{2} - \frac{\log\Gamma}{A} - \frac{\epsilon \cdot \rho_+}{\sqrt{d}\sigma}\right\},
\end{aligned} \tag{45}$$

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

B. Corollary 1

Proof. According to Eq. (45), we compute the partial derivatives of b^*_{rob} with respect to ρ to proof the corollary.

$$\frac{\partial b^*}{\partial \rho_+} = \frac{\epsilon}{2} - \frac{d\epsilon\sigma^2 \log\Gamma}{(\epsilon - 2d\eta + \epsilon \cdot \rho_+)^2}. \tag{46}$$

When $\frac{\partial b^*}{\partial \rho_+} > 0$, b^* increases as ρ_+ increases. We reorganize $\frac{\partial b^*}{\partial \rho} > 0$ to get the following equation.

$$\log\Gamma < \frac{(\epsilon + \epsilon \cdot \rho_+ - 2d\eta)^2}{2d\sigma^2}. \tag{47}$$

The minimum value of the right-hand term of inequality (47) is taken at $\rho_+ = \frac{2d\eta - \epsilon}{\epsilon}$. But obviously, we have $\frac{2d\eta-\epsilon}{\epsilon} > \frac{\eta}{\epsilon}$. So we bring $\rho_+ = \frac{\eta}{\epsilon}$ into the right-hand side of inequality (47), and we get the following inequality.

$$\Gamma < e^{\frac{((2d-1)\eta-\epsilon)^2}{2d\sigma^2}}. \tag{48}$$

When Eq. (48) holds, b^* is a monotonically increasing function of ρ_+. According to Eq. (45), the corollary holds. □

C. Proof for Theorem 3

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

$$\begin{aligned}
&\mathcal{R}_{\text{lp}}(f) \propto \Pr.(\exists\|\tilde{\delta}_+\| \le \epsilon \cdot \rho_+, \mathbb{S}(u+\tilde{\delta}_+) \neq +1 \mid y=+1) \\
&+ \Gamma \cdot \Pr.(\exists\|\tilde{\delta}_-\| \le \epsilon \cdot \rho_-, \mathbb{S}(u+\tilde{\delta}_-) \neq -1 \mid y=-1) \\
&= \Gamma \cdot \max_{\|\tilde{\delta}_-\| \le \epsilon\cdot\rho_-} \Pr.(\mathbb{S}(u+\tilde{\delta}_-) \neq -1 \mid y=-1) \\
&+ \max_{\|\tilde{\delta}_+\| \le \epsilon\cdot\rho_+} \Pr.(\mathbb{S}(u+\tilde{\delta}_+) \neq +1 \mid y=+1) \\
&= \Gamma \cdot \Pr.(\mathbb{S}(u+\epsilon\cdot\rho_-) \neq -1 \mid y=-1) \\
&+ \Pr.(\mathbb{S}(u-\epsilon\cdot\rho_+) \neq +1 \mid y=+1) \\
&= \Gamma \cdot \Pr.\left\{\sum_{i=1}^{d} x_i + b + \epsilon\cdot\rho_+ > 0 \mid y=-1\right\} \\
&+ \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon\cdot\rho_+ < 0 \mid y=+1\right\} \\
&= \Gamma \cdot \Pr.\left\{\mathcal{N}(0,1) < \frac{1}{K}(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b+\epsilon\cdot\rho_-}{\sqrt{d}\sigma})\right\} \\
&+ \Pr.\left\{\mathcal{N}(0,1) < -(\frac{\sqrt{d}\eta}{\sigma} + \frac{b-\epsilon\cdot\rho_+}{\sqrt{d}\sigma})\right\}.
\end{aligned} \tag{49}$$

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

$$\begin{aligned}
b^* = &\frac{1}{K^2-1}(\epsilon(\rho_- + K^2\rho_+) - d\eta(K^2+1) \\
&+ K\sqrt{(\epsilon\rho_- + \epsilon\rho_+ - 2d\eta)^2 + 2d(K^2-1)\sigma^2\log(\frac{K}{\Gamma})}).
\end{aligned} \tag{50}$$

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

$$\begin{aligned}
&\mathcal{R}(f_{\text{opt}}, +1) = \Pr.\left\{\mathcal{N}(0,1) < -(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma})\right\} \\
&= \Pr.\left\{\mathcal{N}(0,1) < -K\sqrt{B^2 + q(K,\Gamma)} - B - \frac{\epsilon\rho_+}{\sqrt{d}\sigma})\right\}, \\
&\mathcal{R}(f_{\text{opt}}, -1) = \Pr.\left\{\mathcal{N}(0,1) < \frac{1}{K}(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma})\right\} \\
&= \Pr.\left\{\mathcal{N}(0,1) < KB + \sqrt{B^2 + q(K,\Gamma)} - \frac{\epsilon\rho_-}{K\sqrt{d}\sigma}\right\},
\end{aligned} \tag{51}$$

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

□

D. Corollary 3

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

$$\begin{aligned}
b^* = &\frac{1}{K^2-1}(-d\eta(K^2+1) \\
&+ K\sqrt{4d^2\eta^2 + 2d(K^2-1)\sigma^2\log(\frac{K}{\Gamma})}).
\end{aligned} \tag{52}$$

Let U_+ and U_- be as follows.

$$U_+ = -(\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma});\ U_- = \frac{1}{K}(-\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}). \tag{53}$$

It is easy to verify that when $Ke^{\frac{(2d\eta-\epsilon)^2}{2dK^2\sigma^2}} < \Gamma < Ke^{\frac{2d\eta^2}{(K^2-1)\sigma^2}}$, $U_+ > U_-$ 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 $Ke^{\frac{(2d\eta-\epsilon)^2}{2dK^2\sigma^2}} < \Gamma < Ke^{\frac{2d\eta^2}{(K^2-1)\sigma^2}}$, Eq. (54) holds.

$$\frac{\partial b^*}{\partial t} = \frac{K^2\epsilon + \frac{K^2\epsilon(\epsilon+\epsilon\rho_+-2d\eta)}{K\sqrt{(\epsilon+\epsilon\rho_+-2d\eta)^2+2d(K^2-1)\sigma^2\log(\frac{K}{\Gamma})}}}{K^2-1} \le 0. \tag{54}$$

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

□