Subclass-wise Logit Perturbation for Multi-label Learning

YU ZHU, Center for Applied Mathematics,Tianjin University, China and College of Sciences, Tianjin University of Commerce, China

OU WU*, Center for Applied Mathematics,Tianjin University, China and HIAS, University of Chinese Academy of Sciences, China

FENGGUANG SU, Center for Applied Mathematics,Tianjin University, China

Logit perturbation refers to adding perturbation on logit, which has been shown to be capable of enhancing the robustness and generalization capabilities of deep neural networks in machine learning. However, studies on logit perturbation for multi-label learning are limited and they only consider the issue of class imbalance in the training data. Furthermore, the logit perturbation vectors in these methods are identical for negative classes containing different subclasses when multi-label learning is viewed as a multiple binary classification problem. This study investigates logit perturbation by exploring the characteristics of subclass-wise multi-label training data. First, the influence of the characteristics of multi-label training data on classification performance is analyzed in terms of the three data characteristics, namely, proportion, variance, and co-occurrence for each category (or subclass). Quantitative analyses reveal that variance differences among the subclasses in the negative class of a decomposed binary task also negatively impact the training performance, and if multiple characteristics affect simultaneously, the performance deterioration will be more severe. Second, theoretical analysis is performed for subclass-wise logit perturbation and a new subclass-wise logit perturbation method is proposed for multi-label learning. In our method, each class/subclass has a carefully designed perturbation implementation according to its proportion, variance, and co-occurrence. Finally, our proposed method is further explained through a regularization view. Extensive experiments demonstrate that our method consistently enhances the generalization performance of popular depth networks on multi-label benchmark datasets.

Additional Key Words and Phrases: Logit perturbation, long-tailed classification, multi-label learning, subclass-wise.

ACM Reference Format:

Yu Zhu, Ou Wu, and Fengguang Su. 2024. Subclass-wise Logit Perturbation for Multi-label Learning. *ACM Trans. Knowl. Discov. Data.* 1, 1 (January 2024), 42 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Multi-label learning is a crucial research field in machine learning, as a sample is associated with multiple classes rather than just a single one in numerous real-world machine learning scenarios. Compared with single-label learning, multi-label learning (MLL) presents more challenges. One of the most significant issues in multi-label training data,

*Corresponding authors.

Authors' addresses: Yu Zhu, yuzhu@tjcu.edu.cn, Center for Applied Mathematics,Tianjin University, Tianjin, China and College of Sciences, Tianjin University of Commerce, Tianjin, China; Ou Wu, wuou@tju.edu.cn, Center for Applied Mathematics,Tianjin University, Tianjin, China and HIAS, University of Chinese Academy of Sciences, Hangzhou, China; Fengguang Su, su.scenery@outlook.com, Center for Applied Mathematics,Tianjin University, Tianjin, China.

Manuscript submitted to ACM

such as VOC-MLT [36], COCO-MLT [36], and Reuters-21578 [12], is the presence of class imbalance. Many MLL algorithms focus on this challenge [10, 12, 16, 27, 36]. In addition, label co-occurrence is a typical characteristic of MLL and it can be challenging to mitigate the impact of label co-occurrence proportion imbalance. For two classes with a high co-occurrence proportion, when only one class appears in a sample, the existing class may be missed or the non-existing class may be incorrectly detected. Conversely, for two classes with a low co-occurrence proportion, when two classes appear simultaneously in a sample, it can be easy for one of the classes to be ignored [39]. However, only a few studies have focused on the bias caused by the imbalanced co-occurrence proportion of labels in MLL [36, 39].

The technical approaches in multi-label studies are similar to those in single-label studies, encompassing the following paradigms (which may overlap): new backbone network, new basic training loss, new training data perturbation scheme, and new learning strategy (e.g., weighting) [16, 21, 26, 27]. Training data perturbation mainly refers to feature [14, 38], logit [10, 16], and label [30, 37] perturbation. In this study, our focus is on designing more effective logit perturbation schemes for MLL.

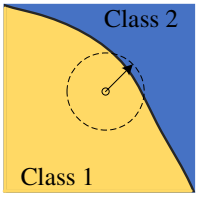


Fig. 1. Illustration of the decision boundary between Class1,and Class2. The circle with solid line represents the logit u_i, the circle with dotted line represents the perturbation bound, and the arrow indicates any perturbation direction.

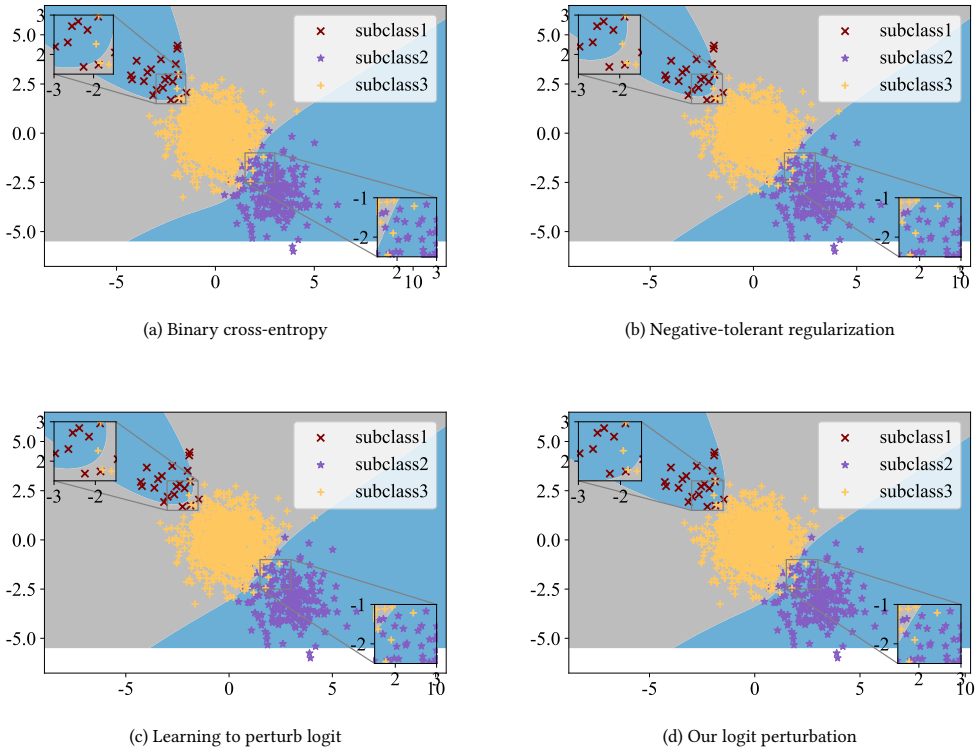
Logit vectors (or logits) are the outputs of the final feature encoding layer in almost all deep neural networks. Let $S = \{\boldsymbol{x_i}, \boldsymbol{y_i}\}_{i=1}^{N}$ be a corpus of N training samples. Let $\boldsymbol{u_i}$ and $\boldsymbol{y_i}$ denote the logit vector and the label of a given training sample $\boldsymbol{x_i}$, respectively. It can be obtained by $\boldsymbol{u_i} = f(\boldsymbol{x_i}, \boldsymbol{W})$, where $f(\cdot,\cdot)$ is the deep neural network with parameter $\boldsymbol{W}$. Employing logit perturbation during training can be described using the following formula:

$$\mathcal{L} = \sum_i l(\zeta(\boldsymbol{u_i} + \Delta\boldsymbol{u_i}), \boldsymbol{y_i}), \tag{1}$$

where $\Delta\boldsymbol{u_i}$ is the perturbation term for $\boldsymbol{u_i}$, $\zeta(\cdot)$ is the sigmoid activation function, $l(\cdot,\cdot)$ is the loss function, and $\mathcal{L}$ refers to the overall training loss. In addition, $\Delta\boldsymbol{u_i} = \epsilon \cdot \boldsymbol{v}$, where $\epsilon \in \mathbb{R}$ is the perturbation magnitude and $\boldsymbol{v} \in \mathbb{R}^d$ is the perturbation direction. Perturbation bound is the maximal allowed magnitude of the add perturbation on logit. For better understanding, in Fig. 1, we illustrate the perturbation bound and direction using geometric diagrams. Li et al. [16] showed that several classical learning methods [2, 25, 34], which are based on distinct motivations and theoretical inspirations, essentially belong to logit perturbation methods in single-label learning. There are also a few logit perturbation-based MLL methods. Wu et al. [36] proposed a negative-tolerant regularization (NTR) to handle class imbalance that occurs in the decomposed binary learning tasks. The NTR loss function actually incorporates an implicit logit perturbation term. Guo and Wang [10] assumed that the logits obey the Gaussian distribution, and utilized the distribution's mean and variance to perform logit compensation (LC) for the positive and negative samples, respectively. Experiments for these two studies indicate the potential of the logit perturbation for MLL.

However, two major shortcomings can be identified in current MLL logit perturbation researches. First, the logit perturbation vectors developed in existing studies are designed exclusively to address the class imbalance. We will demonstrate that there exist other significant characteristics that should also be taken into consideration. Second, the logit perturbation vectors (including both the bound and the direction) in studies we have retrieved so far are identical

[1] Assume that subclass 1, subclass 2, and subclass 3 follow two-dimensional Gaussian distributions with different means but the same variance. The distances from subclass 1 and subclass 3 to the class center of subclass 2 are equal. The number of subclass samples is 20, 200, and 1000, respectively, with 80% used as training data and 20% as test data. For more specific experimental settings, refer to the last paragraph of Section 3.2 (Analysis on Toy Datasets).



(a) Binary cross-entropy

(b) Negative-tolerant regularization

(c) Learning to perturb logit

(d) Our logit perturbation

Fig. 2. Illustrating the behavior of different decomposed binary classifiers in different scenarios where negative subclasses are located in different directions around the positive class. Subclasses 1 and 2, as negative classes, have smaller sample sizes compared to subclass 3, which serves as the positive class. Obviously, the classification margin (the minimum distance between samples in the sample set and the classification boundary) of the subclass 1 in Fig. 2(a) is larger than that in Fig. 2(b) and (c), which are all smaller negative numbers. This indicates that negative-tolerant regularization (NTR) [36] and learning to perturb logit (LPL) [16] methods implemented consistent directional perturbations on all negative subclasses, further damaging the performance of subclass 1 (low proportion, named weak subclass), compared to binary cross-entropy (BCE) loss. Our proposed logit perturbation method applies different directional perturbations to different negative subclasses, thereby ensuring that the classification margin of the weaker negative subclasses does not decrease[1].

for all training samples in the same category or even the entire training corpus [36][2]. To illustrate the drawbacks of applying the same directional perturbations to subclasses, we draw the behavior of different decomposed binary classifiers in Fig. 2, in which all models are trained using polynomial logistic regression. It is noticeable that those methods of adding perturbations of the same direction for the negative subclasses[3] further harms the weak subclasses in cases where each negative class is located in a different direction around the positive class, as shown in Fig. 2(b) and (c). We will demonstrate that using different perturbation vectors for different subclasses can be more effective.

In this study, we delve into the data characteristics of MLL and propose a new logit perturbation method that applies varying logit perturbations to different subclasses in the training of decomposed binary classifiers. First, the data

[2]In fact, this conclusion holds true for almost all existing logit perturbation methods in single-label learning.
[3]In decomposed binary learning tasks for MLL, the negative class can contain numerous categories, which are referred to as "subclasses" in this study.

characteristics of multi-label training data are explored in terms of three crucial characteristics, namely, proportion, variance, and co-occurrence of different categories during the training process. Quantitative analysis is performed on real and toy datasets, and several valuable findings are revealed. Variance differences among categories may have a more serious negative impact on classification performance. If two or three characteristics simultaneously affect, then the performance will decrease much. Second, theoretical analysis for subclass-wise logit perturbation is performed and a new subclass-wise logit perturbation method is proposed by accounting for the three characteristics mentioned earlier. Specifically, perturbation bounds and directions are determined based on the three characteristics for both the positive clasting multi-label logit perturbation methods. It reveals that our proposed method enforces intra-subclass compactness by minimizing the variance of the subclasses' mapped inputs, while also encouraging larger subclass-wise margin. Third, we provide an explanation with a regularization view for both our proposed and existing methods.

Extensive experiments are trained on benchmark datasets for MLL, and the results demonstrate that our methods are highly competitive compared to existing methods.

Our main contributions are summarized as follows:

- We perform a quantitative analysis of the training data characteristics in MLL, and find that all three characteristics, namely proportion, variance, and co-occurrence, have a significant impact on the performance of both the positive class and negative subclasses. Thus, they should not be ignored during the training process.
- Motivated by the defects of existing methods and our quantitative findings, we perform theoretical analysis for subclass-wise logit perturbation and propose a novel subclass-wise logit perturbation method and empirically demonstrate its effectiveness on MLL benchmarks.
- We explain our and several typical multi-label logit perturbation methods in a regularization view. The results reveal that our method has more theoretical merits in feature learning.

2 RELATED WORK

2.1 Multi-label Learning (MLL)

MLL has been widely applied in emotion classification [7, 40], text classification [13, 41], and image recognition [10, 27]. Compared with single-label learning, MLL is more prevalent, since some objects belonging to different classes usually co-occur in the real world. Modeling label co-occurrence relationships is important in MLL, as simply decomposing it into independent binary classification tasks may not be appropriate in cases where label co-occurrence is dense. To overcome this problem, recent researches have explored various approaches for capturing label dependencies. Probabilistic graphical model-based approaches [18, 20] are proposed to explicitly model label dependencies. Recurrent neural networks (RNNs) [33], graph convolutional networks (GCNs) [3, 4], and BERT [1] are utilized to learn label co-occurrence relationships and label embedding in multi-label image/text learning. Attention mechanisms [42] are also widely applied to implicitly capture the label co-occurrence relationships in the MLL. Lin et al. [23] proposed a multi-label-specific feature space ensemble, which creates features customized to each label and utilizes the label correlation to optimize the margin distribution of the base classifiers. However, the above-mentioned current methods neglect the bias of co-occurrence proportion imbalance between the subclass of the negative class and the current positive class. Ye et al. [39] modeled semantic relations for each input image by estimating an image-specific dynamic graph, which helps overcome the co-occurrence proportion imbalance bias that exists in constructing a global graph based on the entire dataset [3, 4]. Re-weighting [36] is shown to be an effective method for mitigating the bias caused

by co-occurrence proportion imbalance. Song et al. [29] proposed a simple sampling strategy, i.e., copy-decoupling re-sampling which converts a multi-label image into multiple single-label images with special labels, eliminating the effect of label co-occurrence on the re-sampling strategy. For the first time, we investigate the use of logit perturbation to mitigate the bias of co-occurrence proportion imbalance.

2.2 Long-tailed Learning

Current research on deep long-tailed learning is generally categorized into three main approaches: class re-balancing [2, 8, 25], information augmentation [5, 19], and module improvement [15]. Class re-balancing is the dominant paradigm in long-tailed learning and can be further subdivided into three methods: re-sampling [8], cost-sensitive learning [2], and logit adjustment (LA) [25], all of which balance the number of training samples of different classes during model training. Information augmentation is a strategy that enhances model performance by introducing additional information. Head-to-tail knowledge transfer [5] and head-to-tail data augmentation [19] are typical of this type of method. Wang et al. [31] designed an effective manner to transfer the statistics from relevant head classes to infer the distribution of tail classes and sample from calibrated distribution further facilitates additional features for tail classes. Compared with long-tailed single-label learning [2, 16, 25], the category labels in MLL may exhibit an even more severe long-tailed distribution [12]. To solve the long-tailed distribution problem in multi-label classification, existing work also mainly uses information augmentation [32], re-sampling [10, 36], cost-sensitive re-weighting [36, 40], and logit perturbation [10]. Specifically, Wang et al. [32] proposed a multiple-stage training framework to exploit both model- and feature-level knowledge from the head labels, to improve both the representation and generalization ability of multi-label text classification models. Wu et al. [36] extended the re-balanced sampling and cost-sensitive re-weighting methods to handle long-tailed multi-labels, resulting in significant performance improvements. Yilmaz et al. [40] performed a novel approach to multi-label emotion classification by dynamically weighting method that balances the contribution from each class during training. Guo and Wang [10] proposed a new collaborative training approach to multi-label classification that leverages two branches: one takes the uniform sampling as input while the other takes the re-balanced sampling as the input. For each branch, they conducted binary classification using a binary-cross-entropy-based classification loss with learnable logit perturbation. Label co-occurrence tends to be harmful to the logit perturbation algorithm. Nevertheless, the designing of logit perturbation strategies considering label co-occurrence for MLL tasks is rarely explored.

2.3 Logit Perturbation

Logit perturbation, which involves modifying the model logits based on various research goals such as data augmentation or long-tailed learning, etc., is a classic idea to adjust the whole loss [16]. Implicit semantic data augmentation (ISDA) [35] acquires a perturbation item associated with the intra-class covariance matrix of each class by positing an infinitely large sample size. Label-distribution-aware margin (LDAM) [2] is a class-wise perturbation method that considers the proportion for long-tailed single-label classification. Logit compensation (LC) [10] is a corpus-wise perturbation method for long-tailed multi-label classification that takes variance into account. The learning of learning to perturb logits (LPL) [16] perturbation term draws on the idea of adversarial training and controls the magnitude and direction of the perturbation using the proportion and logit variance of positive to negative classes. Negative-tolerant Regularization (NTR) [36] is a corpus-wise logit perturbation work for MLL. Let C be the number of classes. NRT

Table 1. Comparison of existing logit perturbation methods.

Method	Application task	Perturbation granularity	Perturbation factor
LA [25]	single-label learning	corpus-wise	class proportion
ISDA [35]	single-label learning	class-wise	variance
LDAM [2]	single-label learning	class-wise	class proportion
NTR [36]	multi-label learning	corpus-wise	class proportion
LC [10]	multi-label learning	corpus-wise	variance
LPL [16]	multi-label learning	class-wise	class proportion and variance

decomposes MLL into C independent binary classification tasks and defines the negative-tolerant binary loss as follows:

$$\mathcal{L}_{NTR} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{C}\sum_{c=1}^{C}[y_{i,c}\log(1-\zeta(u_{i,c}+\Delta u_{i,c})) + \frac{1}{s}(1-y_{i,c})\log(\zeta(s(u_{i,c}+\Delta u_{i,c})))] \quad . \tag{2}$$

In the above equation, s is hyper-parameter, and $\Delta u_{i,c} = -\varphi\log(\frac{N}{N_c} - 1)$. N and N_c denote the total number of samples and the number of samples belonging to the c^{th} category, respectively. Besides, $u_{i,c}$ and $y_{i,c}$ represent the c^{th} elements of the predicted logits $\boldsymbol{u}_i$ and the ground-truth label $\boldsymbol{y}_i$, respectively. The function $\zeta(\cdot)$ is the sigmoid function maps logits in $\mathbb{R}$ to probabilities in the range of $(0, 1)$ by

$$\zeta(u_{i,c}) = 1/(1+e^{-u_{i,c}}). \tag{3}$$

In Eq. (2), logit perturbation term ($\Delta\boldsymbol{u}_i$) can also be expressed as follows:

$$\Delta\boldsymbol{u}_i = \Delta\tilde{\boldsymbol{u}} = -\psi[\log(\frac{N}{N_1} - 1), \cdots, (\frac{N}{N_C} - 1)]^T, \tag{4}$$

where $\Delta\tilde{\boldsymbol{u}}$ is corpus-wise vector and ψ is non-negative in Wu et al.'s experiments [36]. Thus, if $N < 2N_c$, samples with label c are dominant, and the value of $\Delta\boldsymbol{u}_i$ in Eq. (2) is negative. In other words, the loss term corresponding to $y_{i,c} = 1$ decreases, and the loss term corresponding to $y_{i,c} = 0$ increases. However, if $N > 2N_c$, the opposite is true.

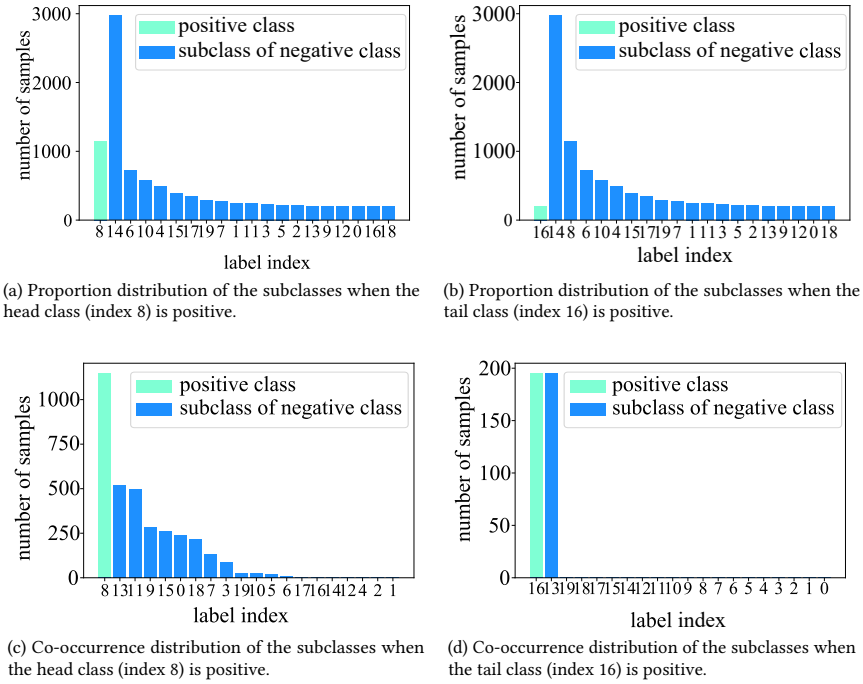
Table 1 summarizes the comparison of existing logit perturbation methods. It can be clearly seen from Table 1 that the existing logit perturbation methods for multi-label learning still only consider the label proportion and variance like single-label learning. In fact, due to the correlation between category labels in the scenario of multi-label classification, categories with different co-occurrence proportions should not be treated equally. To treat classes with different co-occurrence proportions inconsistently, we will study subclass-wise logit perturbation for MLL.

3 QUANTITATIVE ANALYSIS ON SUBCLASS-WISE MULTI-LABEL DATA CHARACTERISTICS

This section conducts quantitative analysis for the influence of three subclass-wise characteristics, namely, proportion, variance, and co-occurrence of multi-label training data on the model performance.

3.1 Analysis on Real Dataset

To analyze how the data characteristics of a subclass in a real dataset affect the model, we first perform statistical analyses of subclass proportion, label co-occurrence, and logit variance on data from various classes in the VOC-MLT dataset. To facilitate presentation, we randomly select the head (index 8) and tail (index 16) classes of the VOC-MLT dataset as positive classes in two decomposed binary learning tasks.



(a) Proportion distribution of the subclasses when the head class (index 8) is positive.

(b) Proportion distribution of the subclasses when the tail class (index 16) is positive.

(c) Co-occurrence distribution of the subclasses when the head class (index 8) is positive.

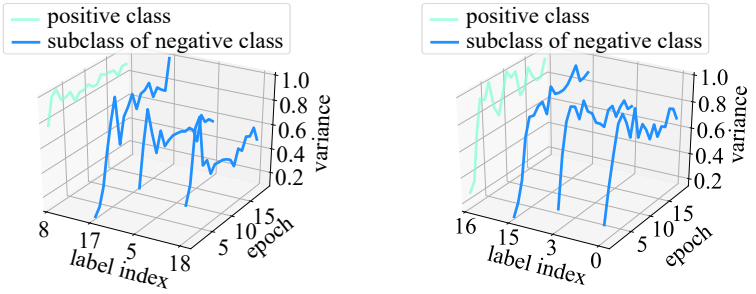
(d) Co-occurrence distribution of the subclasses when the tail class (index 16) is positive.

Fig. 3. Proportion and co-occurrence distributions of the subclasses.

We first conduct a statistic on the subclass-wise proportion and co-occurrence, as illustrated in Fig. 3. Our statistics indicate that the proportion exhibits a long-tailed distribution, and similarly, the co-occurrence also demonstrates an imbalanced state.

To record the logit variance of each positive and negative subclass, we use ResNet-50 [11], pre-trained on ImageNet, as the backbone feature extractor on the VOC-MLT dataset. Standard binary cross-entropy (BCE) loss is employed. To illustrate the logit variance changes for head, middle, and tail subclasses, we perform uniform sampling with a step size of 6 on the proportionally arranged negative subclasses from two randomly selected binary tasks (index 8 or 16 as positive), and plot the normalized logit variance curves of these subclasses over training epochs, as shown in Fig. 4. We observe that the logit variance between the sampled subclasses has a significant difference in the later stages of training, no matter whether the positive class comes from the head or the tail.

To analyze how the features of a training set influence the model, we calculate the correlation coefficient between the proportion of medium and tail subclasses in the training set and the F1 scores for medium and tail subclasses in the test set. We also performed similar calculations for the features of logit variance and label co-occurrence proportion. The results, as shown in Fig. 5, indicate that the most of them are negatively correlated, which means that in most cases, the lower the class proportion, the poorer the performance is. The situations with logit variance and label co-occurrence proportion are opposite to this. This is consistent with the conclusions of previous researches, which considers long-tailed [36] and high label co-occurrence frequency [39] to reduce the model's recognition performance.



(a) Normalized variance variation of the subclasses in epochs when the head class (index 8) is positive.

(b) Normalized variance variation of the subclasses in epochs when the tail class (index 16) is positive.

Fig. 4. Normalized variance variation of the subclasses in epochs.

In the succeeding subsection, to facilitate variable control, we will construct toy datasets to analyze the influence of the three characteristics on the performance of existing multi-label logit perturbation methods.

3.2 Analysis on Toy Datasets

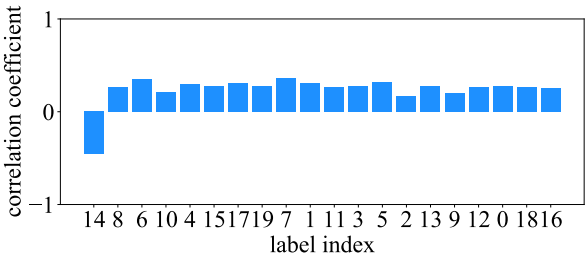
To observe the effects of subclass proportion, variance, and label co-occurrence proportion on different algorithms, as well as the impact of mixed factors, we construct four typical cases by controlling variables. We design toy datasets with well-defined data typical characteristics and training classifiers using existing multi-label logit perturbation methods. Logistic regression is employed as the basic classifier network.

The first case explores how class (proportion) imbalance among the negative subclasses affects performance. The toy dataset is simulated as follows. Let subclass 1, subclass 2, and subclass 3 obey the two-dimensional Gaussian distribution of $\mathcal{N}(\boldsymbol{\mu_1}, \sigma_1^2\boldsymbol{I})$, $\mathcal{N}(\boldsymbol{\mu_2}, \sigma_2^2\boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{\mu_3}, \sigma_3^2\boldsymbol{I})$, respectively. Assuming subclasses have equidistant class center $\boldsymbol{\mu_1} = (0, 2\sqrt{3})$, $\boldsymbol{\mu_2} = (-2, 0)$, $\boldsymbol{\mu_3} = (2, 0)$[4], and the same covariance coefficient $\sigma_1 = \sigma_2 = \sigma_3 = 1$. The number of samples of the subclasses is n_1, n_2, and n_3 of which 80% is used as training data and 20% is used as test data. Let $n_1 = 1000$, $n_2 = 200$, $n_3 = 1000$[5]. The classification results of different methods in the binary classification with subclass 3 as the positive class are shown in Fig. 6(a). Fig. 6(a) shows the AUC (Area Under the Curve) values of various methods in the legend. In the other subfigures, the AUC values of different methods are also shown in the legend. The BCE loss has a significant negative impact on the negative subclass with fewer samples (subclass 2). Although the NTR [36]method considers the overall positive-to-negative sample proportion, it still harms subclass 2 due to not accounting for subclass imbalance, compared to BCE.

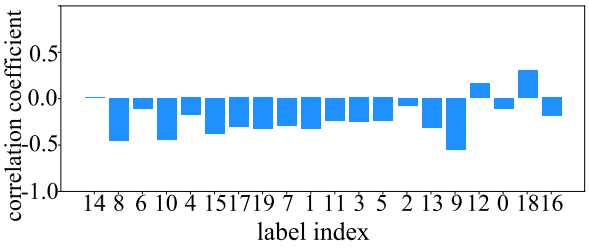
The second case explores how differences in variance among the negative subclasses affect performance. The toy dataset is simulated as follows. Assuming the mean same as in the previous case, but different covariance coefficient $\sigma_1 = 1$, $\sigma_2 = 5$, $\sigma_3 = 1$, and the same amount of data $n_1 = n_2 = n_3 = 1000$. The classification results of different methods in binary classification, with subclass 3 as the positive class, are presented in Fig. 6(b). The BCE loss has a detrimental

[4]The mean values are set so that the distances between the class centers of the subclasses are consistent, meaning the coordinates of the means form a simplex.

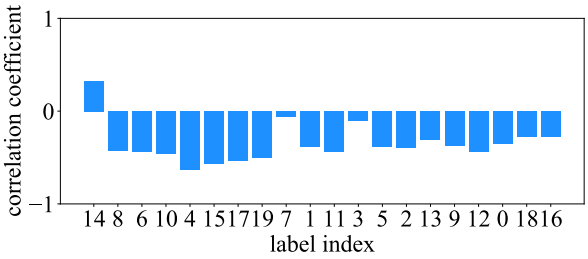
[5]In the following three cases, when considering the impact of the covariance coefficient and label co-occurrence proportion, the ratio of the covariance coefficient or label co-occurrence between subclasses is also set to 1:5.



(a) Class proportion.



(b) Logit variance.



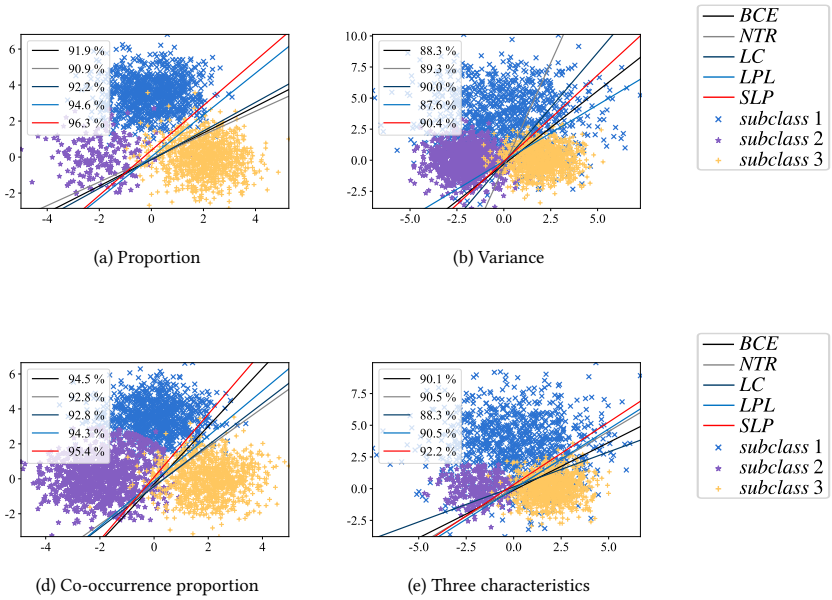
(c) Label co-occurrence proportion.

Fig. 5. Correlation coefficient between each of the three data characteristics of the medium and tail subclasses in the training set and F1 scores of the medium and tail subclasses of the test set.

effect on the performance of the negative subclass with large variance. Although the LC [10] method considers the overall variance of positive and negative classes, it does not show a significant performance improvement.

The third case explores how the co-occurrence proportion imbalance among the negative subclasses affects performance. The toy dataset is simulated as follows. Assuming the same mean and data size as stated previously, the covariance coefficient is also identical ($\sigma_1 = \sigma_2 = \sigma_3 = 1$). Furthermore, there are 100 samples that belong to both subclass 1 and subclass 3, and 20 samples that fall under both subclass 2 and subclass 3, meaning that they have two labels simultaneously. The classification results of different methods in the binary classification with subclass 3 as the positive class are shown in Fig. 6(c). The NTR [36], LC [10], and LPL [16] methods all cause significant harm to subclass 2, which has a high co-occurrence proportion with subclass 3.

The fourth case examines how the simultaneous occurrence of three different characteristics, namely proportion, variance, and co-occurrence, in negative subclasses affects performance. The toy data is simulated as follows. Assuming



(a) Proportion (b) Variance

(d) Co-occurrence proportion (e) Three characteristics

Fig. 6. The influence of three characteristics difference in the subclass of negative class on the existing methods.

the mean same as above, different covariance coefficient $\sigma_1 = 1$, $\sigma_2 = 5$, $\sigma_3 = 1$, and different amounts of data $n_1 = 1000$, $n_2 = 200$, $n_3 = 1000$. In addition, the label co-occurrence situation is the same as above. The classification results of different methods in the binary classification with subclass 3 as the positive class are shown in Fig. 6(d). In the case where all three factors are present simultaneously in the negative subclasses, BCE and LC [10] show a significant misclassification.

Based on the above analysis, the following conclusions are obtained:

- Existing multi-label logit perturbation methods (e.g., NTR [36]) employ undiscriminating perturbation bounds and directions for each negative subclass, which brings more harm to the minority subclasses.
- In addition to the commonly considered class imbalance, differences in variance and co-occurrence proportion also affect performance.

Current multi-label logit perturbation methods not only neglect subclass imbalance but also fail to consider variance and co-occurrence differences among subclasses. We also design a toy data sample to demonstrate the influence of perturbation direction is also explored. Polynomial logistic regression is employed as a basic network for investigating the influence of the direction of perturbation on the toy dataset, which is simulated as follows. Let subclass 1, subclass 2, and subclass 3 obey the two-dimensional Gaussian distribution of $\mathcal{N}(\boldsymbol{\mu_1}, \sigma_1^2\boldsymbol{I})$, $\mathcal{N}(\boldsymbol{\mu_2}, \sigma_2^2\boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{\mu_3}, \sigma_3^2\boldsymbol{I})$, respectively. The number of subclass samples is $n_1 = 20$, $n_2 = 200$, and $n_3 = 1000$, respectively, of which 80% is used as training data and 20% is used as test data. Assuming the mean $\boldsymbol{\mu_1} = (-3, 3)$, $\boldsymbol{\mu_2} = (0, 0)$, $\boldsymbol{\mu_3} = (3, -3)$, the same variance $\sigma_1 = \sigma_2 = \sigma_3 = 1$. The classification results of different methods in the binary classification with subclass 3 as the positive class are shown in Fig. 2. Since the perturbation directions of the negative subclasses are the same, the NTR [36] and LPL [16] methods are more beneficial to the negative subclass 2 with a large sample ratio, but both hurt the

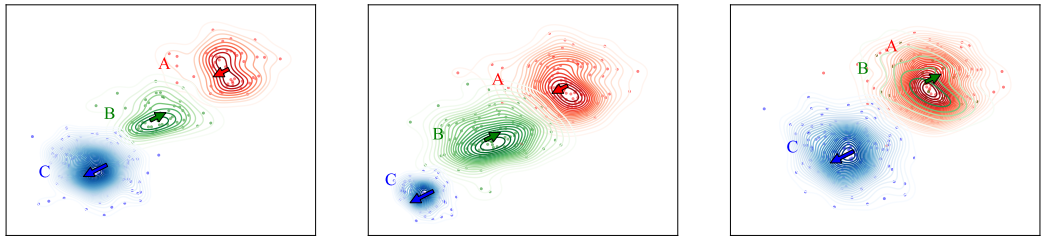
performance of negative subclass 1. It is necessary to explore a subclass-wise logit perturbation method that takes into account all three characteristics.

4 METHODOLOGY

In this section, we will first invest the relationships among loss variations, performance improvements, and subclass-wise logit perturbations. Then, we will introduce the proposed subclass-wise logit perturbation loss and illustrate the differences between the proposed loss and the current multi-label perturbation loss. Next, we will deduce the perturbation coefficient in the proposed loss and provide the method for its dynamic estimation. Subsequently, we will describe the optimization procedure for the logit perturbation term in the proposed loss. Finally, we will describe the overall optimization procedure for the proposed loss.

4.1 Theoretical Analysis for Subclass-wise Logit Perturbation

Under the settings of subclass proportion, variance, and co-occurrence proportion imbalance, we respectively employ simple binary classification tasks to quantitatively investigate the relationships among loss variations, performance improvements, and subclass-wise logit perturbations.



(a) Proportion imbalance. (b) Variance imbalance. (c) Co-occurrence proportion imbalance.

Fig. 7. Illustrative examples for subclass-wise logit perturbation in different scenarios. Different subclasses apply perturbations with different directions and magnitudes.

For proportion imbalance, the data from each of the three classes $\{A, B, C\}$ follow three Gaussian distributions, which are centered on $\boldsymbol{\theta} = [\eta, ..., \eta]$ (d−dimensional vector and $\eta > 0$), $\boldsymbol{0}$, and -$\boldsymbol{\theta}$, respectively. The data follow

$$y_c \overset{u.a.r}{\sim} \{A, B, C\} \tag{5}$$

$$\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) & \text{if } y_c = A \\ \mathcal{N}\left(\boldsymbol{0}, \sigma^2 \boldsymbol{I}\right) & \text{if } y_c = B \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) & \text{if } y_c = C \end{cases} \tag{6}$$

For a classifier f , the overall natural error is defined as $\mathcal{R}_{nat}\left(f\right) = \Pr.\left(f\left(\boldsymbol{x}\right) \neq y\right)$. We use $\mathcal{R}_{nat}\left(f, y\right)$ to denote the natural error conditional on a specific class y. The overall natural error is defined as $\mathcal{R}_{rob}\left(f\right) = \Pr.\left(f\left(\boldsymbol{w}^T\boldsymbol{x} + b + \Delta u\right) \neq y\right)$, where Δu represents logit perturbation. We use $\mathcal{R}_{rob}\left(f, y\right)$ to denote the robust error conditional on a specific class y.

Following the work of Zhang et al. [43], we decompose the robust error ($\mathcal{R}_{rob}$) into natural error ($\mathcal{R}_{nat}$) and boundary error ($\mathcal{R}_{bdy}$), and use the boundary error to assess the classifier's sensitivity to logit perturbation. It can be easily obtained that $\mathcal{R}_{bdy}(f) = \mathcal{R}_{rob}(f) - \mathcal{R}_{nat}(f)$.

In our theoretical analysis, we define subclass-wise logit perturbation as follows:

$$\Delta u_c^* = \arg \min_{\substack{|\Delta u_c| \le |\epsilon_c| \\ \Delta u_c \cdot \epsilon_c \ge 0}} \mathrm{E}_{(\boldsymbol{x},y):y_c=c} \left[l\left(u + \Delta u_c, c\right) \right], \tag{7}$$

where $u = \boldsymbol{w}^T \boldsymbol{x} + b$. The prior probabilities of the three classes $P_A := P(y_c = A)$, $P_B := P(y_c = B)$ and $P_C := P(y_c = C)$ are assumed to be different. Without loss of generality, we assume $P_A : P_B : P_C = \Gamma : \Gamma : 1 - 2\Gamma$ and $0 < \Gamma < \frac{1}{3}$. An illustrative example of subclass-wise logit perturbation at corresponding proportion is shown in Fig. 7 (a). We have the following theorem:

Theorem 1. *For the binary classification task where class A is the positive class, and subclasses B and C are the negative classes with logit perturbation $\rho_1 \cdot \epsilon$, $\rho_1 \cdot \epsilon$, and $\rho_2 \cdot \epsilon$, respectively. The optimal robust linear classifier f_{rob} that minimizes the average classification error is*

$$f_{rob} = \arg\min_f \Pr.(\mathbb{S}\left(u + \Delta u_c^*\right) \neq y_c), \tag{8}$$

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 natural error for the positive class and two negative subclasses:

$$\begin{aligned}
\mathcal{R}_{nat}\left(f_{rob}, A\right) &= \Pr.\left\{ \mathcal{N}(0,1) < -\frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} \right\}, \\
\mathcal{R}_{nat}\left(f_{rob}, B\right) &= \Pr.\left\{ \mathcal{N}(0,1) < \frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} \right\}, \\
\mathcal{R}_{nat}\left(f_{rob}, C\right) &= \Pr.\left\{ \mathcal{N}(0,1) < \frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{3\sqrt{d}\eta}{2\sigma} \right\},
\end{aligned} \tag{9}$$

where $\Lambda = \frac{2\epsilon \cdot \rho_1 - d\eta}{\sqrt{d}\sigma}$.

The proof is attached in the appendix. Theorem 1 indicates that logit perturbation parameterized by ϵ, ρ_1, and ρ_2 influences performance of positive class and all subclasses. We then show how the classification errors of the positive class and negative subclasses change as ρ_1 or ρ_2 increases.

Corollary 1. *For the binary classification task investigated in Theorem 1, when $0 < \Gamma < \frac{1}{3}$, as ρ_1 or ρ_2 increases, the logit perturbations in Theorem 1 will decrease the error for the positive class, and increase the error for the negative subclasses B and C.*

Corollary 2. *For the binary classification task investigated in Theorem 1, $\mathcal{R}_{bdy}\left(f_{rob}, A\right)$, $\mathcal{R}_{bdy}\left(f_{rob}, B\right)$, and $\mathcal{R}_{bdy}\left(f_{rob}, C\right)$ represent the boundary errors for the positive class and the two negative subclasses, respectively. The total boundary error of the classifier is $\mathcal{R}_{bdy}\left(f_{rob}\right)$. The upper bound of $\mathcal{R}_{bdy}\left(f_{rob}\right)$ can be obtained as:*

$$\mathcal{R}_{bdy}\left(f_{rob}\right) < 2\Pr.\left\{ 0 < \mathcal{N}(0,1) < \frac{\epsilon \cdot \rho_1}{\sqrt{d}\sigma} \right\} + \Pr.\left\{ -\frac{\sqrt{d}\eta}{\sigma} < \mathcal{N}(0,1) < \frac{\epsilon \cdot \rho_1}{\sqrt{d}\sigma} \right\} \tag{10}$$

For variance imbalance, the data from each of the three classes $\{A, B, C\}$ follow three Gaussian distributions, which are centered on $\boldsymbol{\theta} = [\eta, ..., \eta]$ (d-dimensional vector and $\eta > 0$), $\boldsymbol{0}$, and $\boldsymbol{\theta}$, respectively. The data follow

$$y_c \overset{u.a.r}{\sim} \{A, B, C\} \tag{11}$$

$$\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, K^2\sigma^2\boldsymbol{I}\right) & \text{if } y_c = A \\ \mathcal{N}\left(\boldsymbol{0}, K^2\sigma^2\boldsymbol{I}\right) & \text{if } y_c = B \\ \mathcal{N}\left(-\boldsymbol{\theta}, (1-2K)^2\sigma^2\boldsymbol{I}\right) & \text{if } y_c = C \end{cases} \tag{12}$$

where $\frac{1}{3} < K < \frac{1}{2}$. An illustrative example of subclass-wise logit perturbation at corresponding variance proportion is shown in Fig. 7 (b).

In our theoretical analysis, we define subclass-wise logit perturbation as follows:

$$\Delta u_c^* = \arg \min_{\substack{|\Delta u_c| \le |\epsilon_c| \\ \Delta u_c \cdot \epsilon_c \ge 0}} \mathrm{E}_{(\boldsymbol{x}, y): y_c = c} \left[l\left(u + \Delta u_c, c\right) \right], \tag{13}$$

where $u = \boldsymbol{w}^T\boldsymbol{x} + b$. We have the following theorem:

Theorem 2. *For the binary classification task where class A is the positive class, and subclasses B and C are the negative classes with logit perturbation $\rho_1 \cdot \epsilon$, $\rho_1 \cdot \epsilon$, and $\rho_2 \cdot \epsilon$, respectively. The optimal robust linear classifier f_{rob} that minimizes the average classification error is*

$$f_{rob} = \arg\min_f \Pr.(\mathbb{S}\left(u + \Delta u_c^*\right) \neq y_c), \tag{14}$$

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 natural error $\mathcal{R}_{nat}$ for the positive class and two negative subclasses:

$$\begin{aligned} \mathcal{R}_{nat}\left(f_{rob}, A\right) &= \Pr.\{\mathcal{N}(0,1) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\}, \\ \mathcal{R}_{nat}\left(f_{rob}, B\right) &= \Pr.\{\mathcal{N}(0,1) < \frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\}, \\ \mathcal{R}_{nat}\left(f_{rob}, C\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{K}{1-2K}\frac{\log 2}{\Lambda} - \frac{3\sqrt{d}\eta}{2(1-2K)\sigma}\right\}, \end{aligned} \tag{15}$$

where $\Lambda = \frac{d\eta - 2\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}$.

The proof is attached in the appendix. Theorem 2 indicates that logit perturbation parameterized by ϵ, ρ_1, and ρ_2 influences performance of positive class and all subclasses. We then show how the classification errors of the positive class and negative subclasses change as ρ_1 or ρ_2 increases.

Corollary 3. *For the binary classification task investigated in Theorem 2, when $\frac{1}{3} < K < \frac{1}{2}$, as ρ_1 or ρ_2 increases, the logit perturbations in Theorem 2 will decrease the error for the positive class A, and increase the error for the negative subclasses B and C.*

Corollary 4. *For the binary classification task investigated in Theorem 2, $\mathcal{R}_{bdy}\left(f_{rob}, A\right)$, $\mathcal{R}_{bdy}\left(f_{rob}, B\right)$, and $\mathcal{R}_{bdy}\left(f_{rob}, C\right)$ represent the boundary errors of the positive class and two negative subclasses, respectively. The total boundary error of the classifier is $\mathcal{R}_{bdy}\left(f_{rob}\right)$. The upper bound of $\mathcal{R}_{bdy}\left(f_{rob}\right)$ can be obtained as:*

$$\mathcal{R}_{bdy}\left(f_{rob}\right) < 2\Pr.\left\{0 < \mathcal{N}(0,1) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} + \Pr.\left\{-\frac{3\sqrt{d}\eta}{2(1-2K)\sigma} < \mathcal{N}(0,1) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma} - \frac{\sqrt{d}\eta}{2K\sigma}\right\} \tag{16}$$

For co-occurrence proportion imbalance, the data from each of the three classes $\{A, B, C\}$ follow two Gaussian distributions, which are centered on $\boldsymbol{\theta} = [\eta, ..., \eta]$ (d-dimensional vector and $\eta > 0$), and -$\boldsymbol{\theta}$, respectively. The data follow

$$\boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) & \text{if } Y = \{A, B\} \text{ with probability } P, Y = \{B\} \text{ with probability } (1-P) \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) & \text{if } Y = \{C\} \end{cases} \tag{17}$$

In our theoretical analysis, we define subclass-wise logit perturbation as follows:

$$\Delta u_c^* = \arg \min_{\substack{|\Delta u_c| \leq |\epsilon_c| \\ \Delta u_c \cdot \epsilon_c \geq 0}} \mathrm{E}_{(\boldsymbol{x}, y): y_c = c} \left[l\left(u + \Delta u_c, c\right) \right], \tag{18}$$

where $u = \boldsymbol{w}^T \boldsymbol{x} + b$. We have the following theorem:

Theorem 3. *For the binary classification task where class A is the positive class, and subclasses B and C are the negative classes with logit perturbation 0, $\rho_1 \cdot \epsilon$, and $\rho_2 \cdot \epsilon$, respectively. The optimal robust linear classifier f_{rob} that minimizes the average classification error is*

$$f_{rob} = \arg\min_f \Pr.\left(\mathbb{S}\left(u + \Delta u_c^*\right) \neq y_c\right), \tag{19}$$

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 natural error $\mathcal{R}_{nat}$ for the positive class and two negative subclasses:

$$\begin{aligned} \mathcal{R}_{nat}\left(f_{opt}, A\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{\Lambda}{2} - \frac{\log(\frac{1-P}{1+P})}{\Lambda}\right\}, \\ \mathcal{R}_{nat}\left(f_{opt}, B\right) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\Lambda}{2} + \frac{\log(\frac{1-P}{1+P})}{\Lambda}\right\}, \\ \mathcal{R}_{nat}\left(f_{opt}, C\right) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\Lambda}{2} + \frac{\log(\frac{1-P}{1+P})}{\Lambda} - \frac{2\sqrt{d}\eta}{\sigma}\right\}, \end{aligned} \tag{20}$$

where $\Lambda = \frac{\epsilon \cdot \rho_1}{\sqrt{d}\sigma}$.

The proof is attached in the appendix. Theorem 3 indicates that logit perturbation parameterized by ϵ, ρ_1, and ρ_2 influences performance of positive class and all subclasses. We then show how the classification errors of the positive class and negative subclasses change as ρ_1 or ρ_2 increases.

Corollary 5. *For the binary classification task investigated in Theorem 3, when $\frac{1-P}{1+P} > e^{-\frac{(\epsilon \cdot \rho_1)^2}{2d\sigma^2}}$, as ρ_1 or ρ_2 increases, the logit perturbations in Theorem 3 will decrease the error for the positive class A, and increase the error for the negative subclasses B and C.*

Corollary 6. *For the binary classification task investigated in Theorem 3, $\mathcal{R}_{bdy}\left(f_{rob}, A\right), \mathcal{R}_{bdy}\left(f_{rob}, B\right)$, and $\mathcal{R}_{bdy}\left(f_{rob}, C\right)$ represent the boundary errors of the positive class and two negative subclasses, and the total boundary error of the classifier is $\mathcal{R}_{bdy}\left(f_{rob}\right)$. The upper bound of $\mathcal{R}_{bdy}\left(f_{rob}\right)$ can be obtained as:*

$$\mathcal{R}_{bdy}\left(f_{rob}\right) < \Pr.\left\{0 < \mathcal{N}(0,1) < \frac{\epsilon \cdot \rho_1}{\sqrt{d}\sigma}\right\} + \Pr.\left\{0 < \mathcal{N}(0,1) < \frac{2\sqrt{d}\eta}{\sigma}\right\} \tag{21}$$

Based on the above theoretical analysis, it can be concluded that adding positive perturbations to weak negative subclasses, negative perturbations to strong negative subclasses, and applying the opposite strategy to positive classes with varying data characteristics will help reduce classification errors.

4.2 The Proposed Loss

Based on the theoretical analyses under the three data characteristics (class proportion, variance, and label co-occurrence proportion) imbalance settings and inspired by the logit perturbation method used in LPL [16], we establish the following new logit perturbation for MLL, named subclass-wise logit perturbation (SLP). It can be represented as a unified end-to-end training loss, as shown in Fig. 8. This loss allows for the virtual generation and deletion of samples[6] at the classification boundary for each decomposed binary classifier. Virtual generation or deletion samples are used to simulate and study the model's behavior near decision boundaries, helping to understand and improve the model's sensitivity to boundary changes and dependence on class divisions. The generation and removal of these samples assist in optimizing the model's decision boundaries. Let C be the number of categories. The proposed SLP loss function is as follows:

$$\mathcal{L}_{SLP} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{C}\sum_{c=1}^{C}[\min_{\substack{|\Delta u_c^+|\le|\epsilon_c^+| \\ \Delta u_c^+\cdot\epsilon_c^+\ge 0}} y_{i,c} log(\zeta(u_{i,c}+\Delta u_c^+)) + \frac{1}{\mathrm{wt}(\boldsymbol{y_i})}\sum_{j=1}^{C}\min_{\substack{|\Delta u_{c,j}^-|\le|\epsilon_{c,j}^-| \\ \Delta u_{c,j}^-\cdot\epsilon_{c,j}^-\ge 0}}(1-y_{i,c})y_{i,j} log(1-\zeta(u_{i,c}+\Delta u_{c,j}^-))], \tag{22}$$

where $\zeta(\cdot)$ is the sigmoid function; $\boldsymbol{y_i} = \{y_{i,1}, ..., y_{i,c}, ...y_{i,C}\}$, where $y_{i,c} = 1$ if label c is associated with given training sample $\boldsymbol{x}_i$, and is otherwise zero[7]; $\mathrm{wt}(\boldsymbol{y_i})$ denotes the Hamming weight of $\boldsymbol{y_i}$, i.e., the number of elements with a value of 1 in $\boldsymbol{y_i}$; $u_{i,c}$ and $y_{i,c}$ represent the c^{th} elements of the predicted logits $\boldsymbol{u}_i$ and the ground-truth label $\boldsymbol{y_i}$, respectively; ϵ_c^+ and $\epsilon_{c,j}^-$ are used to determine perturbation bounds for the positive class and negative subclasses, respectively. The '+' sign in ϵ_c^+ is used to indicate that the perturbation bound is related to the positive class, regardless of the actual sign of its value. While the '−' sign in $\epsilon_{c,j}^-$ indicates that the perturbation bound is related to the negative class. Thus, the perturbation terms Δu_c^+ and $\Delta u_{c,j}^-$ belong to the intervals $[min(\epsilon_c^+, 0), max(\epsilon_c^+, 0)]$ and $[min(\epsilon_{c,j}^-, 0), max(\epsilon_{c,j}^-, 0)]$, respectively. We can see that in the second term of Eq. (22), $\epsilon_{c,j}^-$ indicates that the perturbation term of each negative subclass is different. In addition, the direction of the perturbation is determined by the sign of $\epsilon_{c,j}^-$, and the bound of the perturbation is related to the value of $|\epsilon_{c,j}^-|$. Define the perturbation bound of the positive sample for the c^{th} classifier as follows:

$$\epsilon_c^+ = -log(\frac{cof_{c,c}}{\eta})\Delta\epsilon. \tag{23}$$

The perturbation bounds of samples of different subclasses in the negative class for the c^{th} class classifier are as follows:

$$\epsilon_{c,j}^- = log(\frac{cof_{c,j}}{\eta})\Delta\epsilon, \tag{24}$$

where $cof_{c,j}$ indicates the perturbation coefficient of the j^{th} subclass when the c^{th} class is the positive class; $\eta > 0$ is a variable threshold. The specific calculation of the perturbation coefficients and the perturbation optimization process will be introduced in the next subsection. The use of the logarithmic function is inspired by Wu et al. [36]. Eq. (23) indicates that when the subclass perturbation coefficient is less than the threshold η, the subclass perturbation bound is

[6] Virtual generation samples refer to those samples that, after the perturbation term is applied, are moved closer to the decision boundary, assuming they are generated. Conversely, virtual deletion samples refer to those samples that, after the perturbation term is applied, are moved further away from the decision boundary, assuming they are deleted.

[7] If a text belongs to classes 1 and 3, then $\boldsymbol{y_i} = \{1, 0, 1\}$.

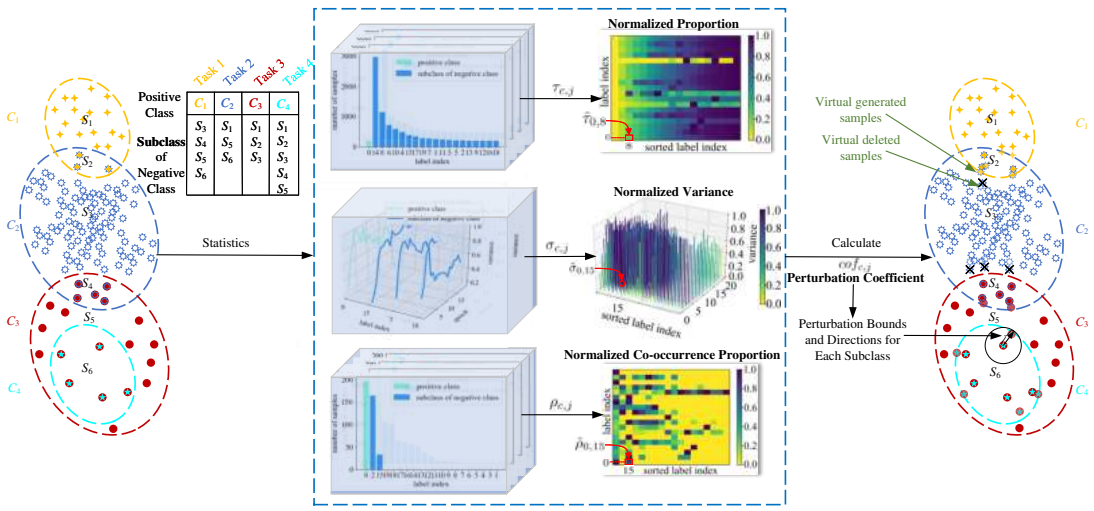


Fig. 8. Our subclass-wise logit perturbation loss performs calculating the perturbation coefficient on the base of three characteristics, namely proportion, variance, and co-occurrence proportion, to avoid consistent perturbations that only consider a single factor in negative class. Assuming there are C classes in the data, it is divided into C tasks. For each task, we can calculate various statistics, including proportion, logit variance, and co-occurrence proportion of the positive class and negative subclasses. Based on the normalized proportion, logit variance, and co-occurrence proportion, we can obtain the perturbation coefficient for each positive class and negative subclass.

a positive number; when it is greater than the threshold η, the subclass perturbation bound is a negative number; when it is equal to the threshold η, the perturbation bound is 0. Eq. (24) is the opposite of the above. $\eta = (\sum_{c=1}^{C}\sum_{j=1}^{C} cof_{c,j})/C^2$ can be used as a simple form of threshold selection. We also provide a varying form threshold selection method in the experimental section, as detailed in Section 6.3. $\Delta\epsilon$ is a hyper-parameter.

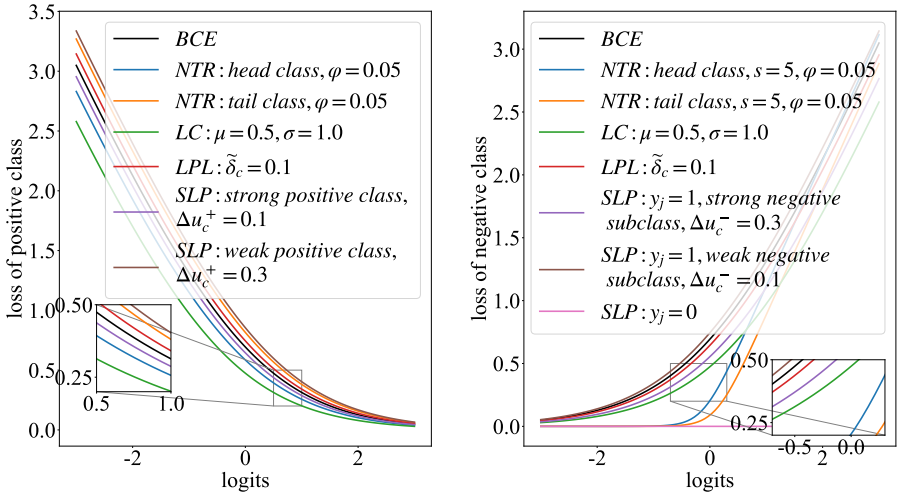
For positive classes with a perturbation coefficient greater than a threshold, the first term loss increases; for negative subclasses with a perturbation coefficient greater than a threshold, the second term loss increases. The opposite is true for the case where the perturbation coefficient is less than a threshold. This shows that our proposed subclass-wise logit perturbation loss increases the attention to positive classes and negative subclasses with large perturbation coefficients, and decreases the attention to positive classes and negative subclasses with small perturbation coefficients.

As shown in Fig. 9, we present the curves of the SLP loss and the existing losses. It can be observed that, compared to the BCE loss, our proposed SLP loss suppresses and encourages the strong and weak positive classes and negative subclasses separately. However, other losses adopt a single approach of either suppression or encouragement.

4.3 Perturbation Coefficient Calculation

Based on the analysis on both toy and real datasets in the aforementioned section, we have obtained that the perturbation bound and direction for the positive class and each negative subclass are determined according to the following principles:

- The smaller the proportion of subclasses, the larger the perturbation bound, and the more the perturbation direction tends to increase the loss.



(a) The curves of positive class loss.

(b) The curves of negative class loss.

Fig. 9. Illustrative the curves of the SLP and existing loss.

- The larger the logit variance of the subclasses, the larger the perturbation bound, and the more the perturbation direction tends to increase the loss.
- The more co-occurrences of the subclasses and the positive class, the larger the perturbation bound, and the more the perturbation direction tends to increase the loss.

Therefore, we calculate the perturbation coefficient. The specific calculation process is as follows:

Statistic three characteristics and normalize. Let Y_c be the sample set containing the c class. Let $\tau_c = (N - |Y_c|)/|Y_c|$, which indicates the ratio of negative samples to positive samples when c is a positive class. Let $\tau_{c,j} = (N - |Y_c|)/(|Y_j \backslash (Y_c \cap Y_j)| + \varepsilon)$, which indicates the reciprocal of the proportion of negative subclass j when c is the positive class. $\varepsilon = 10^{-8}$. Let σ_c and $\sigma_{c,j}$ be the logit variance of positive class and negative subclass j in the c classifier, respectively. Let $\rho_{c,j} = |Y_c \cap Y_j|/|Y_j|$, which indicates the co-occurrence proportion with the positive class c in negative subclass j. $\hat{\tau}_c$ and $\hat{\sigma}_c$ indicate the Min-Max normalized value of τ_c and σ_c, respectively. The normalization method used below is the same. $\hat{\tau}_{c,j}$, $\hat{\sigma}_{c,j}$, and $\hat{\rho}_{c,j}$ indicate the normalized value of $\tau_{c,j}$, $\sigma_{c,j}$, and $\rho_{c,j}$, respectively.

Calculate perturbation coefficient. The perturbation coefficients of positive class c and negative subclass j are defined as follows:

$$cof_{c,c} = [1 - (1 - \alpha)\beta]\hat{\tau}_c + (1 - \alpha)\beta\hat{\sigma}_c, \tag{25}$$

$$cof_{c,j} = \alpha\hat{\tau}_{c,j} + (1 - \alpha)[\beta\hat{\sigma}_{c,j} + (1 - \beta)\hat{\rho}_{c,j}], \tag{26}$$

where α and β are hyper-parameters. In order to avoid complicated parameter tuning, we regard three characteristics as equally important in our experimental settings, namely, $\alpha = 1/3$, $\beta = 1/2$.

Table 2. Symbol explanations used in this section.

Quantity	Formula	Description
τ_c	$(N-\|Y_c\|)/\|Y_c\|$	Ratio of negative samples to positive samples in the c classifier
$\hat{\tau}_c$	$(\tau_c - min(\boldsymbol{\tau}))/(max(\boldsymbol{\tau}) - min(\boldsymbol{\tau}))$	Normalized ratio of negative samples to positive samples in the c classifier
$\tau_{c,j}$	$(N-\|Y_c\|)/(\|Y_j \backslash (Y_c \cap Y_j)\| + \varepsilon)$	Reciprocal of the proportion of negative subclass j in the c classifier
$\hat{\tau}_{c,j}$	$(\tau_{c,j} - min(\boldsymbol{\tau}))/(max(\boldsymbol{\tau}) - min(\boldsymbol{\tau}))$	Normalized reciprocal of the proportion of negative subclass j in the c classifier.
σ_c	$\frac{1}{\|Y_c\|}\sum_{i=1}^{\|Y_c\|}(u_{i,c} - \frac{1}{\|Y_c\|}\sum_{i=1}^{\|Y_c\|}u_{i,c})^2$	Logit variance of positive class in the c classifier
$\hat{\sigma}_c$	$(\sigma_c - min(\boldsymbol{\sigma}))/(max(\boldsymbol{\sigma}) - min(\boldsymbol{\sigma}))$	Normalized logit variance of positive class in the c classifier
$\sigma_{c,j}$	$\frac{\sum_{i=1}^{\|Y_j(\backslash(Y_c\cap Y_j))\|}(u_{i,j} - \frac{\sum_{i=1}^{\|Y_j(\backslash(Y_c\cap Y_j))\|}u_{i,j}}{\|Y_j(\backslash(Y_c\cap Y_j))\|})^2}{\|Y_j(\backslash(Y_c\cap Y_j))\|}$	Logit variance of negative subclass j in the c classifier
$\hat{\sigma}_{c,j}$	$(\sigma_{c,j} - min(\boldsymbol{\sigma}))/(max(\boldsymbol{\sigma}) - min(\boldsymbol{\sigma}))$	Normalized logit variance of negative subclass j in the c classifier
$\rho_{c,j}$	$\|Y_c \cap Y_j\|/\|Y_j\|$	Co-occurrence proportion with the positive class c in negative subclass j.
$\hat{\rho}_{c,j}$	$(\rho_{c,j} - min(\boldsymbol{\rho}))/(max(\boldsymbol{\rho}) - min(\boldsymbol{\rho}))$	Normalized co-occurrence proportion with the positive class c in negative subclass j
$cof_{c,c}$	$[1-(1-\alpha)\beta]\hat{\tau}_c + (1-\alpha)\beta\hat{\sigma}_c$	Perturbation coefficient of positive class c
$cof_{c,j}$	$cof_{c,j} = \alpha\hat{\tau}_{c,j} + (1-\alpha)[\beta\hat{\sigma}_{c,j} + (1-\beta)\hat{\rho}_{c,j}]$	Perturbation coefficient of the negative subclass j in the c classifier
ϵ_c^+	$log(cof_{c,c}/\eta)\Delta\epsilon$	Perturbation bound of the positive class c
$\epsilon_{c,j}^-$	$-log(cof_{c,j}/\eta)\Delta\epsilon$	Perturbation bound of the negative subclass j in the c classifier

The perturbation coefficient matrix $\boldsymbol{cof}$ composed of $cof_{c,j}$ is as follows:

$$\boldsymbol{cof} = \begin{bmatrix} cof_{1,1} & \cdots & cof_{1,C} \\ \vdots & \ddots & \vdots \\ cof_{C,1} & \cdots & cof_{C,C} \end{bmatrix}. \tag{27}$$

For clarity, Table 2 summarizes the symbol explanations used in this section. In the table, $min(\cdot)/max(\cdot)$ denotes obtaining the minimum/maximum value from a vector.

Dynamic estimation of perturbation coefficient. During implementation, all three characteristics that affect the perturbation coefficient are computed online from the summary statistics of each mini-batch. Their estimation method follows the strategy leveraged in MetaSAug [19]. The estimation process is as follows:

$$Y_c^{(t)} = \frac{n^{(t-1)}Y_c^{(t-1)} + m^{(t)}Y_c^{'(t)}}{n^{(t-1)} + m^{(t)}}, \tag{28}$$

$$\sigma_c^{+(t)} = \frac{n_c^{(t-1)}\sigma_c^{+(t-1)} + m_c^{(t)}\sigma_c^{+'(t)}}{n_c^{(t-1)} + m_c^{(t)}}, \tag{29}$$

$$\sigma_c^{-(t)} = \frac{(n^{(t-1)} - n_c^{(t-1)})\sigma_c^{-(t-1)} + (m^{(t)} - m_c^{(t)})\sigma_c^{-'(t)}}{(n^{(t)} - n_c^{(t-1)}) + (m^{(t)} - m_c^{(t)})}, \tag{30}$$

$$\sigma_{c,j}^{(t)} = \frac{n_{c,j}^{(t-1)}\sigma_{c,j}^{(t-1)} + m_{c,j}^{(t)}\sigma_{c,j}^{'(t)}}{n_{c,j}^{(t-1)} + m_{c,j}^{(t)}}, \tag{31}$$

where $Y_c^{(t)}$ and $Y_c^{'(t)}$ are the estimated and true values of the number of c^{th} class samples of the t^{th} mini-batch; $\sigma_c^{+(t)}$ and $\sigma_c^{+'(t)}$ are the estimated and true values of the logit variance of the positive class at the t^{th} step when the c^{th} class is positive class; $\sigma_c^{-(t)}$ and $\sigma_c^{-'(t)}$ are the estimated and true values of the logit variance of the negative class at the t^{th} step when the c^{th} class is positive class; $\sigma_{c,j}^{(t)}$ and $\sigma_{c,j}^{'(t)}$ are the estimated and true values of the logit variance of subclass j^{th} in the negative class at the t^{th} step when c^{th} is a positive class.

Algorithm 1 Perturbation optimization algorithm

Input: Logit vector $\boldsymbol{u}_i$, Perturbation bound ϵ_c^+, $\epsilon_{c,j}^-$, Hyper-parameter λ.

Output: Perturbation $\Delta \boldsymbol{u}_c^{K_c}$.

Let $\boldsymbol{u}_i^0 = \boldsymbol{u}_i$
Calculate $K_c = max\{\lfloor \frac{|\epsilon_c^+|}{\lambda} \rfloor, \lfloor \frac{|\epsilon_{c,1}^-|}{\lambda} \rfloor, ..., \lfloor \frac{|\epsilon_{c,j}^-|}{\lambda} \rfloor, ..., \lfloor \frac{|\epsilon_{c,C}^-|}{\lambda} \rfloor\}$
for $k = 1$ to K_c **do**
  Calculate $\frac{\partial l(\zeta(\boldsymbol{u}_i), \boldsymbol{y}_i)}{\partial \boldsymbol{u}_i} = \zeta(\boldsymbol{u}_i) - \boldsymbol{y}_i$
  if $k \leq \lfloor \frac{|\epsilon_c^+|}{\lambda} \rfloor$ **then**
    Calculate perturbation item $\Delta u_c^{k,+}$ by Eq. (33)
  else
    $\Delta u_c^{k,+} = \Delta u_c^{k-1,+}$
  end if
  if $k \leq \lfloor \frac{|\epsilon_{c,j}^-|}{\lambda} \rfloor$ **then**
    Calculate perturbation item $\Delta u_{c,j}^{k,-}$ by Eq. (34)
  else
    $\Delta u_{c,j}^{k,-} = \Delta u_{c,j}^{k-1,+}$
  end if
  Calculate perturbation vector $\Delta \boldsymbol{u}_c^k$
  Update $\boldsymbol{u}_i^{k+1} := \boldsymbol{u}_i^k + \Delta \boldsymbol{u}_c^k$
end for

4.4 Perturbation Optimization

The perturbation term in Eq. (22) can be solved by an optimization method similar to PGD [24]. Algorithm 1 gives the specific optimization process. The BCE loss function calculates the derivative of the logit vector, resulting in:

$$\frac{\partial l(\zeta(\boldsymbol{u}_i), \boldsymbol{y}_i)}{\partial \boldsymbol{u}_i} = \zeta(\boldsymbol{u}_i) - \boldsymbol{y}_i. \tag{32}$$

Δu_c^+ in the Eq. (22) represents the c^{th} element of $\boldsymbol{\delta}_c$. Δu_c^+ is updated by the following formula:

$$\Delta u_c^+ = \frac{\texttt{sign}(\epsilon_c^+)\lambda}{|Y_c|} \sum_{i:y_{i,c}=1} (\zeta(\boldsymbol{u}_i) - \boldsymbol{y}_i), \tag{33}$$

where λ is a hyper-parameter, and $\texttt{sign}(\cdot)$ represents a symbolic function. In Eq. (22), $\Delta u_{c,j}^-$ represents the j^{th} element of $\Delta \boldsymbol{u}_c$. $\Delta u_{c,j}^-$ is updated by the following formula:

$$\Delta u_{c,j}^- = \frac{\texttt{sign}(\epsilon_{c,j}^-)\lambda}{|Y_j \backslash (Y_c \cap Y_j)| + \varepsilon} \sum_{i:(y_{i,j}=1, y_{i,c}=0)} (\zeta(\boldsymbol{u}_i) - \boldsymbol{y}_i). \tag{34}$$

4.5 The Overall Learning Procedure

The overall learning procedure within each mini-batch consists of four steps: (1) Dynamically estimate the perturbation coefficients. (2) Solve the perturbation bounds. (3) Perform perturbation optimization to update logits. (4) Update the network parameters. Algorithm 2 presents the overall optimization steps of our SLP for MLL.

Algorithm 2 Subclass-wise logit perturbation algorithm

Input: S, max iteration T, hyper-parameters for perturbation optimization algorithm, batch size, hyper-parameter $\Delta\epsilon$ and threshold η for calculating the perturbation bound, momentum and weight decay for calculating SGD.
Output: Deep neural network parameters $\boldsymbol{W}$.

```
for t = 1 to T do
    Sample a mini-batch from S.
    Dynamic estimation of perturbation coefficient by Eqs. (28)- (31).
    Calculate perturbation bound by Eqs. (23) and (24).
    Update logits using Algorithm 1.
    Update W with SGD.
end for
```

5 EXPLANATION IN REGULARIZATION VIEW

This section conducts a comprehensive analysis of NTR [36], LC [10], LPL [16], and our SLP from the perspective of regularization. To the best of our knowledge, this is the first study that uses regularization to explain these multi-label logit perturbation methods. Our findings suggest that our SLP has more theoretical merits.

Table 3. Regularization terms and reflected generalization factors of four losses (NTR [36], LC [10], LPL [16], and SLP).

Loss	Regularization term	Generalization factor
NTR [36]	$\frac{1}{C}\sum_{c=1}^{C}\{[y_{i,c}\zeta(u_{i,c})-(1-y_{i,c})(1-\zeta(su_{i,c}))](-\psi\log(\frac{N}{N_c}-1))\}$	✓Class-wise margin
LC [10]	$-\frac{1}{C}\sum_{c=1}^{C}[y_{i,c}(1-\zeta(\sigma_c^{+}u_{i,c})\mu_c^{+})+(1-y_{i,c})\zeta(\sigma_c^{-}u_{i,c})\mu_c^{-}]$	✓Intra-class compactness
LPL [16]	$\frac{1}{C}\sum_{c=1}^{C}\mathbb{S}(c-\tau)(\zeta(u_{i,c})-y_{i,c})^2\epsilon_c$	✓Class-wise margin
Our SLP	$\frac{1}{C}\sum_{c=1}^{C}[(y_{i,c}(\zeta(u_{i,c})-1))^2\epsilon_c^{+}+\frac{1}{\lvert\boldsymbol{y_i}\rvert}\sum_j((1-y_{i,c})y_{i,j}\zeta(u_{i,c}))^2\epsilon_{c,j}^{-}]$	✓Class-wise/subclass-wise margin ✓Intra-class/intra-subclass compactness

Using the first-order Taylor expansion of the loss, we have

$$\ell_{BCE}(\boldsymbol{u}+\Delta\boldsymbol{u})\approx\ell_{BCE}(\boldsymbol{u})+(\frac{\partial\ell_{BCE}}{\partial\boldsymbol{u}})^T\Delta\boldsymbol{u}=\ell_{BCE}(\boldsymbol{u})+(\zeta(\boldsymbol{u})-\boldsymbol{y})^T\Delta\boldsymbol{u}, \tag{35}$$

where $\boldsymbol{y}$ is the label. Considering $R_{BCE}=(\zeta(\boldsymbol{u})-\boldsymbol{y})^T\Delta\boldsymbol{u}$, the underlying regularizers of all approaches can be derived. The regularization terms of the four losses are presented in Table 3.

NTR [36] loss for sample $\boldsymbol{x_i}$ is

$$\ell_{NTR}=-\frac{1}{C}\sum_{c=1}^{C}[y_{i,c}log(1-\zeta(u_{i,c}+\Delta u_c))+\frac{1}{s}(1-y_{i,c})log(\zeta(s(u_{i,c}+\Delta u_c)))], \tag{36}$$

where $\Delta u_c=-\psi\log(\frac{N}{N_c}-1)$ is the perturbation term of the positive and negative classes. According to Eq. (35), we can derive its regularization term as

$$R_{NTR}=\frac{1}{C}\sum_{c=1}^{C}\{[y_{i,c}\zeta(u_{i,c})-(1-y_{i,c})(1-\zeta(su_{i,c}))](-\psi\log(\frac{N}{N_c}-1))\}. \tag{37}$$

R_{NTR} increases the class-wise margin of the positive class of the tail classifier.

LC [10] loss for sample $\boldsymbol{x_i}$ is

$$\ell_{LC} = -\frac{1}{C}\sum_{c=1}^{C}[y_{i,c}log(\zeta(\sigma_c^+(u_{i,c}+\Delta u_c^+))) + (1-y_{i,c})log(1-\zeta(\sigma_c^-(u_{i,c}+\Delta u_c^-)))], \tag{38}$$

where $\Delta u_c^+ = \frac{\mu_c^+}{\sigma_c^+}$ and $\Delta u_c^- = \frac{\mu_c^-}{\sigma_c^-}$ are the perturbation terms of the positive and negative classes, respectively. According to Eq. (35), we can derive its regularization term as

$$R_{LC} = -\frac{1}{C}\sum_{c=1}^{C}[y_{i,c}(1-\zeta(\sigma_c^+u_{i,c})\mu_c^+) + (1-y_{i,c})\zeta(\sigma_c^-u_{i,c})\mu_c^-], \tag{39}$$

where $-y_{i,c}(1-\zeta(\sigma_c^+u_{i,c}))\mu_c^+$ and $-(1-y_{i,c})\zeta(\sigma_c^-u_{i,c})\mu_c^-$ will force the model to decrease the logit variance (Fig. 10, and thus increase intra-class compactness.

LPL [16] loss for sample $\boldsymbol{x_i}$ is

$$\ell_{LPL} = \frac{1}{C}\sum_{c=1}^{C}\{\mathbb{S}(c-\tau)\times\max_{||\Delta u_c||\le\epsilon_c}[\ell_{BCE}(\zeta(u_{i,c}+\Delta u_c),y_{i,c})\mathbb{S}(c-\tau)]\}, \tag{40}$$

where $\Delta u_c = \frac{\partial \ell_{BCE}}{\partial u_{i,c}}\epsilon_c$ or $\Delta u_c = -\frac{\partial \ell_{BCE}}{\partial u_{i,c}}\epsilon_c$ is the perturbation term when $c \ge \tau$ or $c < \tau$. According to Eq. (35), we can derive its regularization term as

$$R_{LPL} = \frac{1}{C}\sum_{c=1}^{C}\mathbb{S}(c-\tau)(\zeta(u_{i,c})-y_{i,c})^2\epsilon_c. \tag{41}$$

LPL [16] increases the class-wise margin of the positive and negative classes of the tail classifier, which is beneficial to the learning of tail classes.

Our SLP loss for sample $\boldsymbol{x_i}$ is

$$\ell_{SLP} = -\frac{1}{C}\sum_{c=1}^{C}[\min_{\substack{|\Delta u_c^+|\le|\epsilon_c^+|\\ \Delta u_c^+\cdot\epsilon_c^+\ge 0}} y_{i,c}log(\zeta(u_{i,c}+\Delta u_c^+)) + \frac{1}{\mathrm{wt}(\boldsymbol{y_i})}\sum_j \min_{\substack{|\Delta u_{c,j}^-|\le|\epsilon_{c,j}^-|\\ \Delta u_{c,j}^-\cdot\epsilon_{c,j}^-\ge 0}}(1-y_{i,c})y_{i,j}log(1-\zeta(u_{i,c}+\Delta u_{c,j}^-))], \tag{42}$$

where $\Delta u_c^+ = -\frac{\partial(y_{i,c}log(\zeta(u_{i,c})))}{\partial u_{i,c}}\epsilon_c^+ = y_{i,c}(\zeta(u_{i,c})-1)\epsilon_c^+$ and $\Delta u_{c,j}^- = -\frac{\partial((1-y_{i,c})y_{i,j}log(1-\zeta(u_{i,c})))}{\partial u_{i,c}}\epsilon_{c,j}^- = (1-y_{i,c})y_{i,j}\zeta(u_{i,c})\epsilon_{c,j}^-$ are the perturbation terms of the positive class and negative subclasses. Fig. 10 shows the necessity of varying amplitude and direction of perturbations. According to Eq. (35), we can derive its regularization term as

$$\begin{aligned}
&\ell_{SLP}(\boldsymbol{u_i}+\Delta\boldsymbol{u_i})\\
&\approx \ell_{SLP}(\boldsymbol{u_i}) + \frac{1}{C}\sum_{c=1}^{C}[y_{i,c}(\zeta(u_{i,c})-1)\Delta u_c^+ + \frac{1}{\mathrm{wt}(\boldsymbol{y_i})}\sum_j(1-y_{i,c})y_{i,j}\zeta(u_{i,c})\Delta u_{c,j}^-]\\
&= \ell_{SLP}(\boldsymbol{u_i}) + \frac{1}{C}\sum_{c=1}^{C}[(y_{i,c}(\zeta(u_{i,c})-1))^2\epsilon_c^+ + \frac{1}{\mathrm{wt}(\boldsymbol{y_i})}\sum_j((1-y_{i,c})y_{i,j}\zeta(u_{i,c}))^2\epsilon_{c,j}^-].
\end{aligned} \tag{43}$$

$$R_{SLP} = \frac{1}{C}\sum_{c=1}^{C}[(y_{i,c}(\zeta(u_{i,c})-1))^2\epsilon_c^+ + \frac{1}{\mathrm{wt}(\boldsymbol{y_i})}\sum_j((1-y_{i,c})y_{i,j}\zeta(u_{i,c}))^2\epsilon_{c,j}^-]. \tag{44}$$

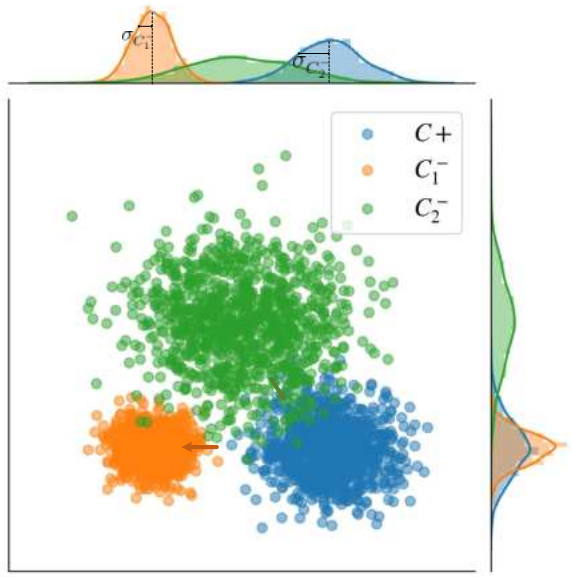


Fig. 10. Illustrative figure for the variance and the perturbation amplitude and direction. There are significant differences in both the variances ($\sigma_{C_1^-}$ and $\sigma_{C_2^-}$) and the perturbation amplitudes and directions (deep orange arrow and deep green arrow) between the two negative subclasses (C_1^- and C_2^-).

Our SLP increases the class-wise margin of weak positive classes and the subclass-wise margin of weak negative subclasses. Meanwhile, SLP makes ϵ_c^+ and $\epsilon_{c,j}^-$ decrease, which increases the intra-class compactness of weak positive classes and the intra-subclass compactness of weak negative subclasses.

6 EXPERIMENTS

6.1 Datasets

The proposed SLP is evaluated on five different datasets, namely Ren-CECps, SemEval18, Reuters-21578, VOC-MLT, and COCO-MLT. These datasets represent a diverse range of fields including emotion classification, text classification, and visual recognition. Figs.11 and 12 show the label distribution and label co-occurrence of these datasets.

Ren-CECps: Ren-CECps is a Chinese emotion corpus which is originally partitioned into two sets: training set and test set, and annotated on three levels, namely, document, paragraph, and sentence. Each level is annotated with eight emotion classes (“期待”, “高兴”, “喜爱”, “惊讶”, “焦虑”, “悲伤”, “愤怒”, “憎恨” (“expect", “joy", “love", “surprise", “anxiety", "sorrow", "angry", and "hate")) and discrete emotion intensities from 0.0 to 1.0. The eight labels are divided into three groups: head classes (“love", “anxiety", and “sorrow"); medium classes (“joy", “expect", and “hate"); tail classes (“angry" and “surprise"). As in [7], we follow the methodology where emotion intensity greater than 0.0 is set to 1, otherwise 0.

SemEval18: SemEval18 task 1 includes an array of subtasks on labeled multilingual tweets in English, Arabic and Spanish. The data is originally partitioned into three sets, namely, training set, validation set and test set. We utilize the English Emotion Classification (E-c) dataset from SemEval18 which comprises of 10,983 labeled samples with 11 different emotion categories: “anger", “anticipation", “disgust", “fear", “joy", “love", “optimism", “pessimism", “sadness",

"surprise", and "trust". The 11 labels of the SemEval18 dataset are divided into three groups: head classes ("disgust", "anger", and "joy"); medium classes ("sadness", "optimism", "fear", and "anticipation"); tail classes ("pessimism", "love", "surprise" and "trust").

Reuters-21578: Reuters-21578 (version 1.0) contains Reuters Newswire documents from 1987 that were manually annotated with 90 labels [12]. We follow the train-test split used by Huang et al. [12], obtaining 7,769 training documents (of which 1,000 are used for validation) and 3,019 test documents. Labels are divided into head classes (sample size $\geq$ 35), medium classes (sample size 8-35), and tail classes (sample size $\leq$ 8).

VOC-MLT: The long-tailed multi-label versions of VOC sampled and compiled from VOC-2012 and VOC-2007 datasets based on Pareto distribution by Wu et al. [36]. The training and test configurations used in [36] are followed. The training set contains 1,142 images and 20 classes, the number of images per class ranges from 4 to 775. Labels are divided into head classes (sample size $\geq$ 100), medium classes (sample size 20 - 100), and tail classes (sample size $\leq$ 20). The test set is built on VOC-2007 [9] test that contains 4,952 images.

COCO-MLT: The long-tailed multi-label versions of COCO sampled and compiled from MS-COCO-2017 dataset based on Pareto distribution by Wu et al. [36]. The training and test configurations used in [36] are followed. The training set contains 1,909 images and 80 classes, and the number of images per class ranges from 6 to 1,128. The test set consists of 5,000 images from the MS-COCO-2017 test set. The ratio of head, medium, and tail classes is 22:33:25 in COCO-MLT.

6.2 Experimental Settings

Evaluation Metrics. Following [1, 12], we evaluate micro-average F1-score (miF1) and macro-average F1-score (maF1) for all classes of RenCECps, SemEval18 and Reuters-21578 datasets, and we also report miF1 and maF1 for each subset. Micro-average F1 performs statistics on each sample in the dataset regardless of class to establish a global confusion matrix, and then calculates the F1-score. Micro-average F1 treats each sample equally, so its value is more affected by the head classes. Macro-average F1 calculates the F1-score for each class individually and then averages them. Macro-average F1 treats every class equally, so its value is mainly affected by a tail classes.

Following [36], we evaluate mean average precision(mAP) for all the classes of VOC-MLT and COCO-MLT datasets, and we also report mAP for each subset.

Competing methods. Several state-of-the-art MLL or logit perturbation methods are compared. For a fair comparison, their efficient combination results are also reported. The following methods are all modifications based on binary cross entropy (BCE).

(1) Empirical risk minimization (ERM): The plain model with equal weights and sampling probabilities for all samples.

(2) Re-weighting (RW) [36]: A smoothed version of re-weighting is performed that is inversely proportional to the square root of the class frequency and normalized in mini-batches.

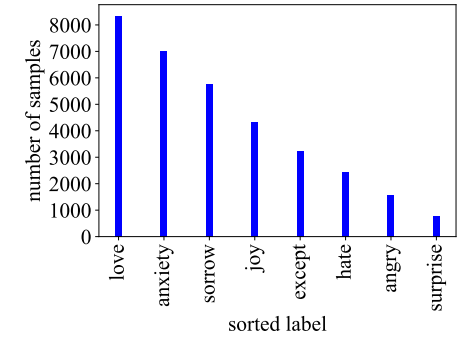
(3) Re-sampling (RS) [28]: A class-aware re-sampling without extra skills, which tries to make the probability of each class appear the same in each batch as much as possible; and avoids the same order of pictures.

(4) Focal loss (FL) [21]: As in [36], we use a focal loss with $\gamma = 2$ and a balance parameter of 2.

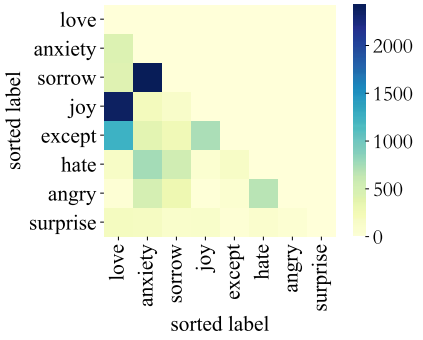
(5) ML-GCN [4]: A MLL method based on graph convolutional network (GCN).

(6) Class-balanced loss (CB) [6]: A class-wise re-weighting approach, guided by the effective number of each class $E_n = (1 - \beta^n)(1 - \beta)$.

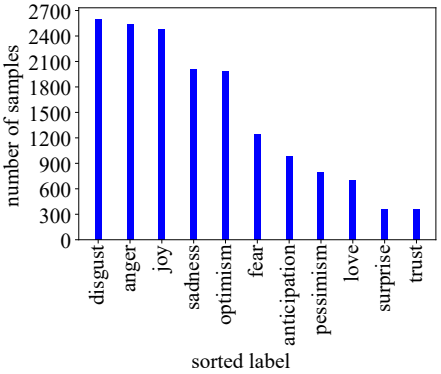
(7) Label-distribution-aware margin loss (LDAM) [2]: A recently proposed class-wise margin-loss that is motivated by minimizing a margin-based generalization bound.



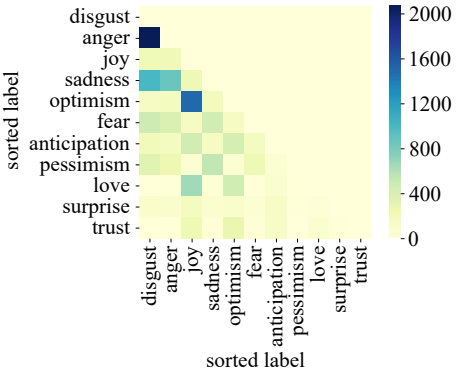
(a) Label distribution of the Ren-CECps dataset.



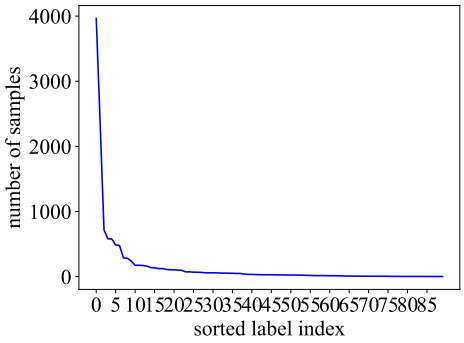
(b) Label co-occurrence of the Ren-CECps dataset.



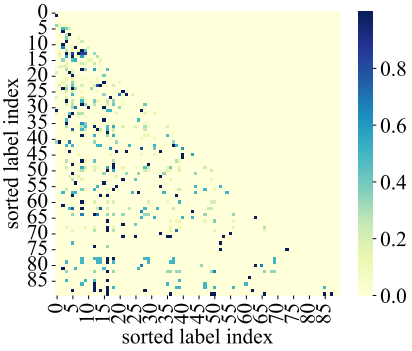
(c) Label distribution of the SemEval18 dataset.



(d) Label co-occurrence of the SemEval18 dataset.

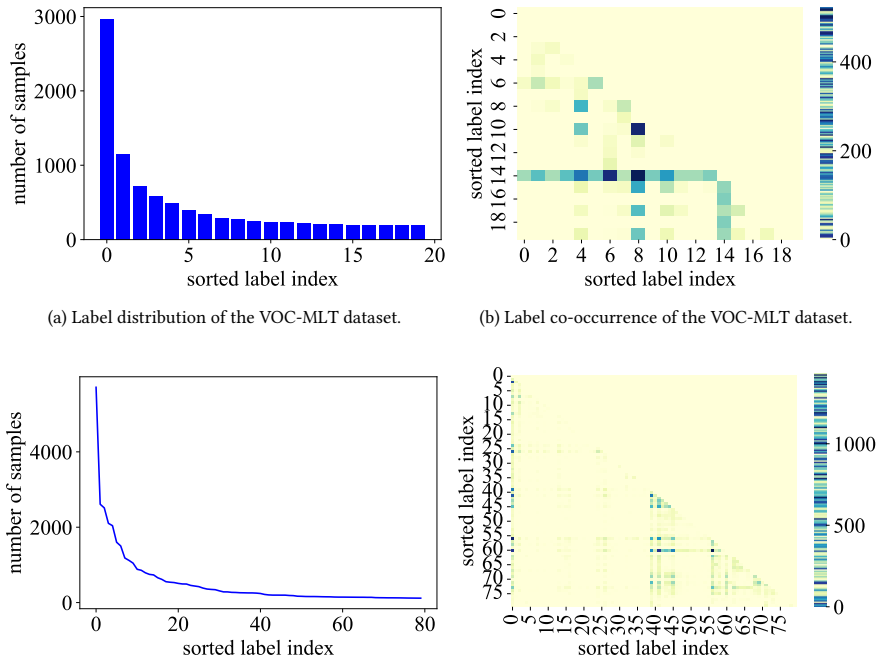


(e) Label distribution of the Reuters-21578 dataset.



(f) Label co-occurrence of the Reuters-21578 dataset. The co-occurrence matrix is row-normalized..

Fig. 11. Statistical charts of the text datasets.



(a) Label distribution of the VOC-MLT dataset.

(b) Label co-occurrence of the VOC-MLT dataset.

(c) Label distribution of the COCO-MLT dataset.

(d) Label co-occurrence of the COCO-MLT dataset.

Fig. 12. Statistical charts of the image datasets.

Table 4. Comparisons with state-of-the-art methods on the Ren-CECps dataset (%).

Methods	total miF1/maF1	head miF1/maF1	medium miF1/maF1	tail miF1/maF1
BCE	61.11±0.15/53.96±0.09	66.26±0.25/66.24±0.11	54.62±0.21/51.30±0.18	39.01±0.12/39.53±0.13
FL [21]	61.87±0.16/55.26±0.13	67.13±0.23/66.68±0.18	54.91±0.14/53.29±0.11	40.00±0.22/41.07±0.17
R-BCE [36]	61.79±0.14/55.16±0.12	67.06±0.19/66.61±0.18	54.83±0.24/53.23±0.13	39.86±0.17/40.86±0.20
R-BCE-Focal [36]	61.94±0.17/55.33±0.08	67.21±0.16/66.76±0.07	54.97±0.12/53.34±0.14	40.08±0.15/41.16±0.09
R-BCE+NTR [36]	61.90±0.12/55.29±0.16	67.11±0.11/66.65±0.05	55.06±0.16/53.47±0.21	40.03±0.15/40.99±0.13
R-BCE-Focal+NTR [36]	62.13±0.11/55.55±0.13	67.34±0.09/66.90±0.11	55.23±0.07/53.62±0.08	40.42±0.14/41.41±0.16
R-BCE+LC [10]	61.98±0.19/55.36±0.14	67.23±0.06/66.79±0.11	55.06±0.17/53.45±0.15	40.14±0.16/41.07±0.19
R-BCE-Focal+LC [10]	62.32±0.12/55.77±0.16	67.50±0.06/67.06±0.11	55.45±0.08/53.84±0.14	40.83±0.12/41.75±0.18
R-BCE+LPL [16]	62.34±0.12/55.87±0.08	67.65±0.11/67.19±0.17	55.48±0.07/53.99±0.13	40.59±0.14/41.68±0.12
R-BCE-Focal+LPL [16]	62.49±0.07/56.01±0.12	67.71±0.11/67.25±0.09	55.54±0.15/54.09±0.17	40.94±0.04/42.04±0.13
R-BCE+LPLE [17]	62.48±0.06/56.01±0.14	67.65±0.09/67.18±0.13	55.67±0.15/54.11±0.07	40.97±0.05/42.10±0.07
R-BCE-Focal+LPLE [17]	63.03±0.07/56.63±0.14	68.12±0.11/67.68±0.06	56.36±0.13/54.84±0.16	41.54±0.03/42.76±0.11
R-BCE+SLP(M)	63.02±0.06/56.65±0.07	68.16±0.05/67.69±0.09	56.24±0.13/54.81±0.08	41.67±0.03/42.86±0.06
R-BCE+SLP(V)	63.27±0.08/56.93±0.12	68.38±0.04/67.91±0.12	56.58±0.08/55.12±0.07	41.88±0.03/43.16±0.05
R-BCE-Focal+SLP(M)	63.91±0.03/57.63±0.07	68.92±0.12/68.47±0.08	57.42±0.05/56.05±0.06	42.48±0.04/43.74±0.05
R-BCE-Focal+SLP(V)	**64.02**±0.04/**57.77**±0.10	**68.97**±0.02/**68.55**±0.09	**57.59**±0.08/**56.16**±0.04	**42.92**±0.10/**44.04**±0.07

(8) Re-balanced weighting binary cross entropy (R-BCE) [36]: A way to re-balance the weights that takes into account the impact caused by label co-occurrence.

(9) Negative-tolerant regularization (NTR) [36]: A regularization to mitigate the over-suppression of negative labels.

(10) Logit compensation (LC) [10]: A corpus-wise logit perturbation method, which assumes that logit obeys a Gaussian distribution. The logit variance of positive or negative samples is used as a multiplicative perturbation, and the logit mean is used as an additive perturbation.

(11) Learning to perturb logits (LPL) [16]: A class-wise logit perturbation method, which adopts an idea similar to adversarial training to implement positively/negatively augmented based on low/high performance or tail/head class.

(12) Learning to perturb logits extension (LPLE) [17]: An extended version of LPL in multi-label classification. Through threshold adjustment, when a multi-label task is converted into multiple binary classification tasks, the binary classification task of the head class pays more attention to variance imbalance, while the tail class pays more attention to class imbalance.

(13) CD-RS + AFL (Ensemble) [29]: An ensemble is obtained by averaging the predictions of the trained models on the datasets with and without copy-decoupling re-sampling (CD-RS), these two models use adaptively focal loss (AFL). Specifically, CD-RS converts a multi-label image into multiple single-label images with special labels, eliminating the effect of label co-occurrence on the re-sampling strategy.

Implementation Details. For RenCECps, SemEval18 two emotion classification datasets and Reuters-21578 text classification dataset, we choose *BertForSequenceClassification* as the backbone in the *transformer* library. The `bert-base-uncased`, `bert-base-chinese`, and `bert-base-case` pre-trained models are used in SemEval18, RenCECps, and Reuters-21578, respectively. The maximum length of the pre-trained model is 512, the training data larger than the maximum length will be truncated, and the batch size is 32. AdamW with a weight decay of 0.01 is used as the optimizer, and the learning rate is determined by hyper-parameter search.

For the two multi-label visual recognition datasets VOC-MLT and COCO-MLT, we use the same configuration as Wu et al. [36], Guo and Wang [10], Li et al. [16] advanced methods for comparison. Specifically, we use ResNet50 pre-trained on ImageNet as the backbone, followed by global average pooling and 2048×256 fully connected layers to obtain image-level features. The final classifier outputs logit through a fully connected layer of 256 × C. The input images are resized to a spatial dimension of 224 × 224 and organized into batches of size 32 using standard data augmentation methods [36]. We use SGD with a momentum of 0.9 and a weight decay of 0.0001 as the optimizer.

6.3 Comparisons with State-of-the-Arts

Our method has two variants SLP(M) and SLP(V). SLP(M) means that the threshold η in Eqs. (23) and (24) take the mean value of the elements of $\boldsymbol{cof}$. SLP(V) represents the threshold $\eta = (\upsilon * min(\boldsymbol{cof}) + (5 - \upsilon) * max(\boldsymbol{cof}))/5$. υ searches from $\{0, 1, 2, 3, 4, 5\}$.

Results on Ren-CECps, SemEval18 and Reuters-21578. Given that original code of comparison method have been open-sourced, we used the original open-source code provided by the authors, and the results shown are the averages and standard deviations obtained from five runs. The results on Ren-CECps comparing our proposed method with other traditional and state-of-the-art methods are shown in Table 4, where the underlined and bolded are the best results among other traditional methods, and the best among all methods results, respectively. Other tables also have similarly underlined and bolded. Compared to the best results of other methods, the best results achieved by the proposed SLP improve by 0.99%/1.14% in the overall miF1/maF1, and by 0.85%/0.87%, 1.23%/1.32%, 1.38%/1.28% for head, medium, and tail classes, respectively.

Table 5. Comparisons with state-of-the-art methods on the SemEval18 dataset (%).

Methods	total miF1/maF1	head miF1/maF1	medium miF1/maF1	tail miF1/maF1
BCE	70.06±0.20/54.24±0.23	80.22±0.21/79.86±0.16	66.12±0.17/60.36±0.13	38.85±0.21/28.89±0.24
FL [21]	70.33±0.14/54.80±0.20	80.35±0.15/80.01±0.16	66.32±0.14/60.35±0.17	40.16±0.17/30.33±0.11
R-BCE [36]	70.12±0.13/54.45±0.10	80.20±0.11/79.86±0.14	66.08±0.12/60.11±0.16	39.70±0.10/29.73±0.15
R-BCE-Focal [36]	71.14±0.09/56.36±0.11	80.81±0.17/80.46±0.12	67.29±0.19/61.69±0.21	42.34±0.20/32.97±0.17
R-BCE+NTR [36]	71.21±0.11/56.56±0.14	80.86±0.08/80.50±0.12	67.34±0.06/61.73±0.10	42.63±0.13/33.44±0.11
R-BCE-Focal+NTR [36]	71.43±0.11/56.79±0.10	81.01±0.07/80.56±0.15	67.62±0.14/61.97±0.14	43.00±0.12/33.73±0.17
R-BCE+LC [10]	71.24±0.13/56.30±0.14	81.00±0.04/80.65±0.05	67.58±0.05/62.34±0.12	41.30±0.09/31.98±0.10
R-BCE-Focal+LC [10]	71.49±0.14/56.92±0.12	81.04±0.10/80.69±0.09	67.73±0.11/62.21±0.13	43.09±0.07/33.80±0.05
R-BCE+LPL [16]	71.69±0.15/57.19±0.14	81.32±0.09/80.96±0.11	67.67±0.06/62.19±0.12	43.84±0.05/34.35±0.06
R-BCE-Focal+LPL [16]	71.92±0.03/57.77±0.11	81.47±0.08/81.12±0.09	67.95±0.10/62.50±0.06	44.37±0.13/35.53±0.12
R-BCE+LPLE [17]	71.90±0.10/57.60±0.13	81.28±0.16/80.93±0.10	68.25±0.08/62.83±0.12	44.01±0.04/34.88±0.11
R-BCE-Focal+LPLE [17]	72.42±0.12/58.41±0.05	81.58±0.11/81.22±0.13	68.96±0.10/63.60±0.07	45.10±0.13/36.12±0.07
R-BCE+SLP(M)	72.56±0.05/58.73±0.11	81.96±0.02/81.61±0.05	69.04±0.05/64.24±0.09	45.59±0.04/36.55±0.07
R-BCE+SLP(V)	72.79±0.10/59.31±0.09	82.10±0.12/81.76±0.11	68.96±0.06/64.06±0.12	45.32±0.06/37.70±0.13
R-BCE-Focal+SLP(M)	73.23±0.11/60.50±0.09	**82.28±0.10/81.94±0.11**	69.41±0.05/64.75±0.06	47.79±0.09/40.16±0.06
R-BCE-Focal+SLP(V)	**73.49±0.04/60.91±0.05**	82.15±0.03/81.79±0.09	**70.23±0.11/65.34±0.10**	**48.04±0.08/40.82±0.12**

Given that original code of comparison method have been open-sourced, we used the original open-source code provided by the authors, and the results shown are the averages and standard deviations obtained from five runs. Table 5 shows the results of all method SemEval18. The proposed method performs similarly to the state-of-the-art method on the head class; but shows a significant improvement in both the medium and tail classes. For example, compared to the best results of other methods, the best results achieved by the proposed SLP improve by 1.07%/2.50% in the overall miF1/maF1, and by 0.70%/0.72%, 1.27%/1.74%, 2.94%/4.70% for head, medium, and tail classes, respectively. Compared to the Ren-CECps dataset, this dataset shows greater improvement. We analyze this in relation to two factors: 1) the higher frequency of label co-occurrence in the SemEval18 dataset, and 2) the greater impact of the proposed method on the subclass variance (intra-class compactness) of the SemEval18 dataset.

Table 6 shows the results of all methods on Reuters-21578. The results marked with an asterisk in Table 6 are directly from the paper by Huang et al. [12], while the methods without an asterisk also used the authors'open-source code, with results being the averages and standard deviations from five runs. The proposed method significantly improves the performance of tail classes, demonstrating its advantage in dealing with imbalanced data. Compared to the best results of other methods, the proposed SLP achieved an improvement of 0.53%/1.89% in terms of the overall miF1/maF1, with improvements of 0.44%/1.66%, 0.80%/1.52%, and 0.91%/2.56% for the head, medium, and tail classes, respectively.

Table 7 presents the performance of our proposed approaches in terms of micro recall (miR), micro precision (miP) and Jaccard index score (JacS), and compares them to the baseline and state-of-the-art models on Ren-CECps, SemEval18, and Reuters-21578 datasets. It can be seen that our proposed method R-BCE-Focal+SLP(V) shows more significant improvements in miR and JacS metrics across the three datasets.

Results on VOC-MLT and COCO-MLT. In the VOC-MLT and COCO-MLT datasets, we evaluate the mAP for all classes and report the mAP for the head, medium, and tail. The experimental results compared with other methods are shown in Table 8, where the results of other methods are directly from the papers of Song et al. [29] and Li et al. [17].

Table 6. Comparisons with state-of-the-art methods on the Reuters-21578 dataset(%).

Methods	total miF1/maF1	head miF1/maF1	medium miF1/maF1	tail miF1/maF1
BCE*	89.14/47.32	91.75/82.81	66.28/57.26	0.00/0.00
FL [21]*	89.97/56.83	91.83/82.64	76.16/70.63	27.40/15.37
CB-Focal [6]*	89.23/52.96	91.56/80.44	71.64/66.61	23.08/9.93
R-BCE [36]	89.14/53.61	91.59/80.27	72.38/66.67	24.51/12.08
R-BCE-Focal [36]*	89.47/54.35	91.59/80.39	72.86/66.69	25.00/14.22
R-BCE+NTR [36]*	89.45/57.98	91.21/82.05	77.33/71.11	31.17/19.05
R-BCE-Focal+NTR [36]*	90.62/64.47	92.14/83.48	80.25/77.01	48.89/31.39
R-BCE+LC [10]	90.36±0.14/64.03±0.16	91.97±0.16/83.17±0.17	79.28±0.12/75.39±0.18	44.94±0.15/32.09±0.13
R-BCE-Focal+LC [10]	90.70±0.12/68.61±0.20	92.18±0.16/83.41±0.15	80.00±0.14/76.83±0.12	53.76±0.08/44.51±0.11
R-BCE+LPL [16]	90.65±0.07/65.91±0.12	92.53±0.15/83.50±0.14	79.82±0.07/76.21±0.13	48.38±0.10/34.05±0.16
R-BCE-Focal+LPL [16]	90.85±0.05/69.12±0.14	92.28±0.12/83.65±0.11	80.50±0.09/77.14±0.10	55.27±0.13/45.51±0.16
R-BCE+LPLE [17]	91.02±0.11/66.06±0.12	92.61±0.09/83.57±0.13	79.88±0.06/76.58±0.04	49.02±0.12/36.69±0.17
R-BCE-Focal+LPLE [17]	91.10±0.13/69.58±0.07	92.55±0.11/83.86±0.15	80.58±0.06/77.91±0.07	55.36±0.13/45.88±0.14
R-BCE+SLP(M)	91.18±0.06/66.13±0.10	92.80±0.04/84.39±0.06	79.88±0.11/76.10±0.08	48.33±0.08/36.56±0.10
R-BCE+SLP(V)	91.48±0.03/67.31±0.06	93.00±0.12/85.45±0.11	81.05±0.06/77.25±0.08	50.11±0.04/37.92±0.12
R-BCE-Focal+SLP(M)	91.22±0.10/69.90±0.11	92.69±0.12/84.36±0.14	80.44±0.05/76.82±0.06	54.83±0.11/47.51±0.06
R-BCE-Focal+SLP(V)	**91.63**±0.06/**71.47**±0.04	**93.05**±0.08/**85.52**±0.14	**81.38**±0.08/**79.43**±0.07	**56.27**±0.05/**48.44**±0.13

Table 7. Mean values and standard deviations on Ren-CECps, SemEval18 and Reuters-21578 in terms of miR, miP and JacS(%).

Methods	Ren-CECps			SemEval18			Reuters-21578		
	miR	miP	JacS	miR	miP	JacS	miR	miP	JacS
R-BCE+LC [10]	58.12±0.15	66.40±0.21	49.17±0.11	65.22±0.12	78.49±0.20	59.44±0.12	86.08±0.09	95.07±0.21	91.16±0.07
R-BCE-Focal+LC [10]	58.47±0.24	66.70±0.21	49.48±0.15	65.50±0.13	78.69±0.23	59.72±0.09	86.62±0.16	95.19±0.19	91.57±0.07
R-BCE+LPL [16]	58.48±0.11	66.75±0.26	49.50±0.04	65.70±0.06	78.88±0.17	60.02±0.13	86.51±0.23	95.21±0.31	91.53±0.13
R-BCE-Focal+LPL [16]	58.66±0.02	66.86±0.13	49.67±0.05	65.93±0.07	79.11±0.16	60.30±0.06	86.75±0.08	95.36±0.18	91.76±0.09
R-BCE+LPLE [17]	58.64±0.14	66.86±0.25	49.62±0.07	65.97±0.14	79.00±0.26	60.22±0.18	86.89±0.13	95.56±0.31	91.94±0.22
R-BCE-Focal+LPLE [17]	59.18±0.06	67.42±0.12	50.27±0.02	66.49±0.04	79.51±0.04	60.94±0.14	87.10±0.05	95.49±0.08	91.95±0.05
R-BCE+SLP(M)	59.15±0.06	67.43±0.07	50.27±0.06	66.67±0.05	79.59±0.13	61.13±0.09	87.13±0.07	95.63±0.07	92.09±0.09
R-BCE+SLP(V)	59.43±0.07	67.63±0.21	50.52±0.06	66.88±0.06	79.85±0.16	61.43±0.03	87.58±0.07	95.74±0.18	92.34±0.08
R-BCE-Focal+SLP(M)	60.04±0.08	68.30±0.16	51.25±0.09	67.35±0.03	80.23±0.16	61.90±0.09	87.21±0.11	95.64±0.17	92.12±0.06
R-BCE-Focal+SLP(V)	**60.22**±0.03	**68.33**±0.16	**51.27**±0.06	**67.66**±0.04	**80.41**±0.15	**62.18**±0.04	**87.69**±0.06	**95.94**±0.12	**92.53**±0.04

Compared to the COCO-MLT dataset, the VOC-MLT dataset has a higher frequency of co-occurrence between the tail and head classes. The proposed SLP loss, which considers label co-occurrence, results in a more significant improvement for this dataset compared to other methods. Compared with the best results of other methods, the proposed SLP improved by 1.33% and 0.50% in overall mAP of VOC-MLT and COCO-MLT, respectively.

Table 8. Comparison results of mAP with state-of-the-art methods on the VOC-MLT and COCO-MLT datasets (%).

Methods	VOC-MLT				COCO-MLT			
	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 [36]	74.70	67.58	82.81	73.96	42.27	48.62	45.80	32.02
FL [21]	73.88	69.41	81.43	71.56	49.46	49.80	54.77	42.14
RS [28]	75.38	70.95	82.94	73.05	46.97	47.58	50.55	41.70
RS-Focal [28]	76.45	72.05	83.42	74.52	51.14	48.90	54.79	48.30
ML-GCN [4]	68.92	70.14	76.41	62.39	44.24	44.04	48.36	38.96
LDAM [2]	70.73	68.73	80.38	69.09	40.53	48.77	48.38	22.92
CB-Focal [6]	75.24	70.30	83.53	72.74	49.06	47.91	53.01	44.85
R-BCE [36]	76.34	71.40	82.76	75.22	49.43	48.77	53.00	45.33
R-BCE-Focal [36]	77.39	72.44	83.16	76.77	52.75	50.20	56.52	50.02
R-BCE+NTR [36]	78.65	73.16	84.11	78.66	52.53	50.25	56.33	49.54
R-BCE-Focal+NTR [36]	78.94	73.22	84.18	79.30	53.55	51.13	57.05	51.06
R-BCE+LC [10]	78.08	73.10	83.49	77.75	53.68	50.58	57.10	51.90
R-BCE-Focal+LC [10]	78.66	72.74	83.45	79.52	53.94	50.99	57.47	51.88
R-BCE+LPL [16]	79.07	73.68	82.86	80.28	54.27	51.15	57.83	52.34
R-BCE-Focal+LPL [16]	79.34	73.01	83.08	81.27	54.61	51.45	58.37	52.42
R-BCE+LPLE [17]	79.02	72.39	82.14	81.64	54.35	51.48	57.72	52.42
R-BCE-Focal+LPLE [17]	79.57	73.47	83.95	80.87	54.76	50.78	58.12	53.81
CD-RS+AFL (Ensemble) [29]	78.96	73.35	85.03	78.63	55.35	52.45	59.48	52.46
R-BCE+SLP(M)	79.79$\pm$0.06	73.68$\pm$0.04	83.94$\pm$0.07	81.27$\pm$0.06	55.04$\pm$0.05	51.50$\pm$0.04	58.21$\pm$0.09	53.96$\pm$0.09
R-BCE+SLP(V)	79.78$\pm$0.05	73.56$\pm$0.04	83.45$\pm$0.05	81.69$\pm$0.08	55.04$\pm$0.04	51.54$\pm$0.02	58.11$\pm$0.07	54.06$\pm$0.13
R-BCE-Focal+SLP(M)	80.23$\pm$0.05	73.71$\pm$0.03	84.52$\pm$0.07	81.90$\pm$0.05	55.38$\pm$0.06	**51.67$\pm$0.05**	58.57$\pm$0.07	54.44$\pm$0.11
R-BCE-Focal+SLP(V)	**80.90$\pm$0.05**	**74.21$\pm$0.04**	**85.04$\pm$0.07**	**82.81$\pm$0.08**	**55.85$\pm$0.06**	51.55$\pm$0.05	**59.20$\pm$0.08**	**55.21$\pm$0.09**

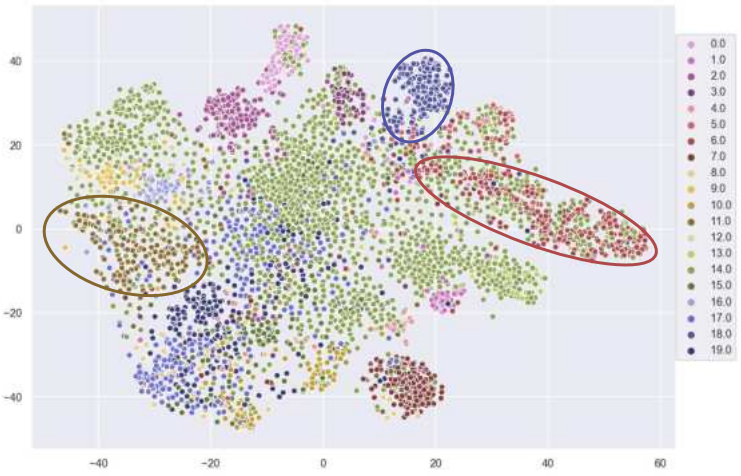
6.4 Feature Space Visualization

To gain additional insight, we look at the t-SNE [48] projection of learned representations and compared vanilla BCE loss with our proposed method. Fig. 13 shows that our learned feature space is more compact low sample proportion, large varianvce, and high co-occurrence proportion classes. Tail (low sample proportion) classes have larger margins.

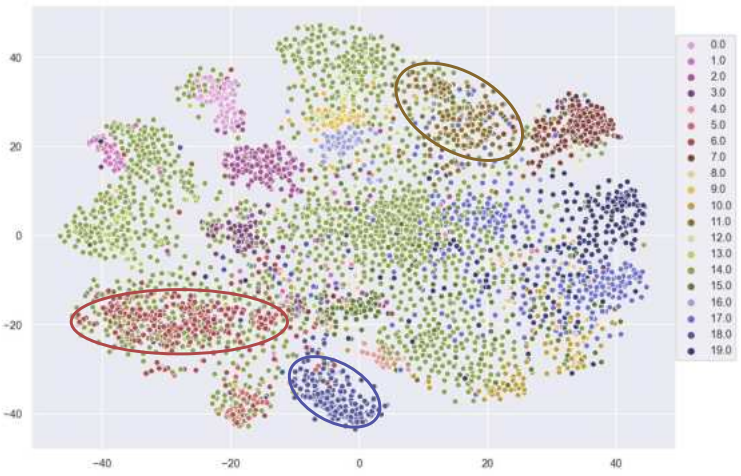
6.5 Quantitative Analysis

Ablation analysis. To further analyze the influence of proportion, variance, and co-occurrence on the proposed method of improving the performance of long-tailed MLL, we conduct a set of ablation studies and report the results in Table 9. It can be seen that the removal of any one of the three characteristics has an effect on the performance of the proposed method. The results show that variance has the largest impact on performance, which is consistent with our observations on toy datasets. In addition, co-occurrence has a significant impact on tail classes, to verify that the proposed method is effective for subclass friendly with large variance.

Class-wise analysis. In Fig. 14, we show that class average precision (AP) increments are computed by only the perturbation coefficients for proportion, variance, and co-occurrence, respectively. As shown in Fig. 14(a) and (b), compared with not adding the perturbation term, calculating the perturbation coefficient by proportion is not friendly to the head classes, because its perturbation direction reduces the class-wise margin of the head classes classifier. As



(a) R-BCE.



(b) R-BCE+SLP(V).

Fig. 13. t-SNE visualization of embedding space of VOC-MLT obtained using BCE loss and BCE+SLP(V) Loss method. The feature embedding of our model is more compact for low sample proportion (index 18), large varianvce (index 5), and high co-occurrence proportion (index 11) classes and better separated.

Table 9. Ablation study on the VOC-MLT dataset (%).

Methods	total	head	medium	tail
R-BCE-Focal+SLP(V)	80.90	74.21	85.04	82.81
w/o proportion	80.41	74.02	84.65	82.03
w/o variance	80.05	73.57	84.11	81.86
w/o co-occurrence	80.44	74.17	84.86	81.82

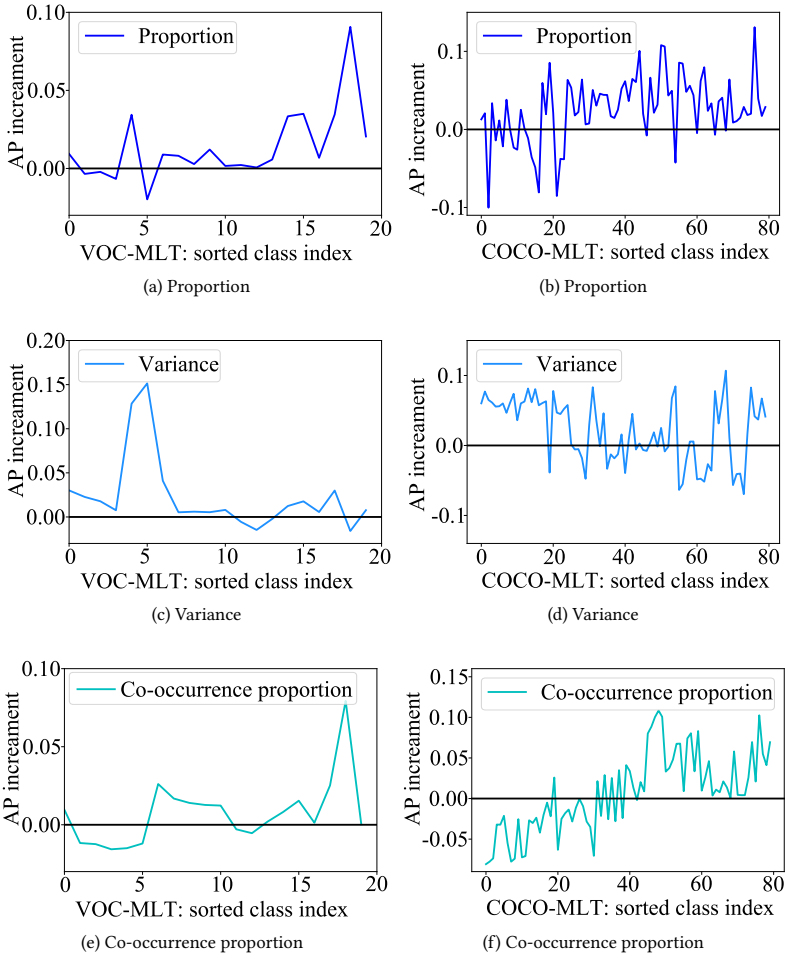
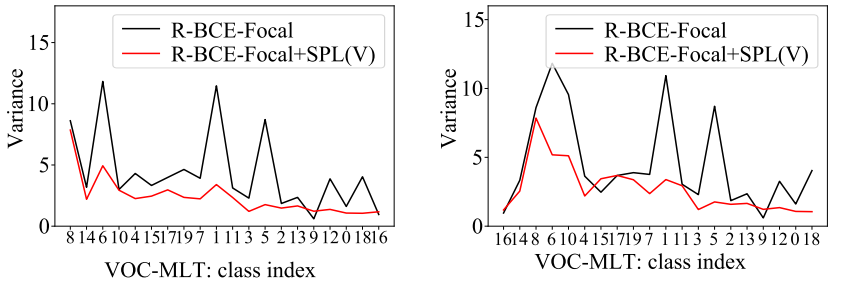


Fig. 14. Class-wise AP increment of perturbation coefficients by only proportion, variance, and co-occurrence proportion, respectively. Class labels are sorted from head to tail classes left-right.

shown in Fig. 14(c) and (d), the head classes and most of the medium and tail classes benefit from calculating the perturbation coefficient by variance, as it increases the intra-class compactness. As shown in Fig. 14(e) and (f), compared



(a) Variance of each subclass in the last epoch when the head class (index 8) is positive.

(b) Variance of each subclass in the last epoch when the tail class (index 16) is positive.

Fig. 15. Variance of each subclass in the last epoch.

with no perturbation term added, calculating the perturbation coefficient only from the co-occurrence presents a similar trend to the proportion.

Variance analysis. We plot the change in logit variance of the last epoch of positive class and negative subclasses on the VOC-MLT dataset when the head or tail class (index 8 and 16) is positive, to verify that the proposed method is effective for subclass with large variance friendly. The curves are shown in Fig. 15(a) and (b). For the head class (index 8) as the positive class, the proposed method reduces the logit variance of most negative subclasses. For the tail class (index 16) as the positive class, our method also significantly reduces the logit variance of the medium and tail classes.

6.6 Qualitative Analysis

To better understand how our method handles the long tail multi-label data, we performed qualitative experiments with NTR [36], LC [10], LPL [16], and our SLP on VOC-MLT. Fig. 16 presents several examples showing the predictions of different models. For example, in the middle column, all methods other than ours miss recognition of the "horse" (belongs to the tail class and has a high degree of co-occurrence with "cow"). A similar problem exists in the third example, which is a common challenge in MLL. Our model takes into account the degree of label co-occurrence, so that the logit is well compensated. In addition, our model also shows better results in head-tail classification.

6.7 Space and Time Complexity Analysis

It can be seen from Algorithm 2 that our logit perturbation algorithm mainly adds 4, 5, and 6 compared to the original algorithm. Among them, steps 4 and 5 are to estimate the perturbation coefficient and calculate the perturbation bound. During the training process of the original algorithm, the proportions, variances, and co-occurrence proportions of different classes of each batch are recorded to obtain the perturbation coefficient, and then the perturbation bound is calculated.

Space complexity analysis. The space overhead of Algorithm 2 is mainly step 4, corresponding to the Eqs.(11)-(14), that is, the process of estimating the perturbation coefficient. A total of 14 C-dimensional vectors need to be stored, where C is the number of classes in the dataset. Therefore, the space complexity of our logit perturbation algorithm is O(14C).

Time complexity analysis. There is no additional time overhead in steps 4 and 5 of Algorithm 2. Compared with the original algorithm, the extra time overhead is mainly in step 6 of Algorithm 2, which is the process of Algorithm 1.

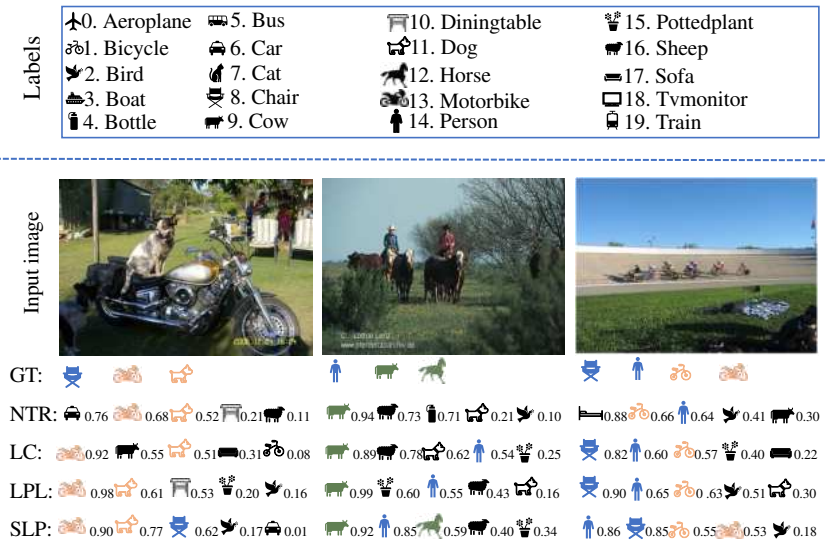
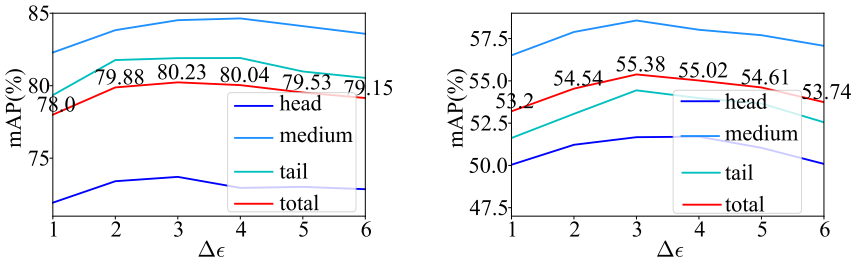


Fig. 16. Example decisions from our SLP, NTR [36], LC [10], and LPL [16] on VOC-MLT dataset. GT indicates the ground truth. *Cornflower blue*, *orange*, *sea green* are the head, medium, and tail classes, respectively.

It can be seen that step 2 in Algorithm 1 is the number of perturbation updates K_c. Therefore, the time complexity of our perturbation algorithm is $O(K_c)$, and the size of K_c can be controlled by hyper-parameters. Specifically, on the NVIDIA RTX 3080, the training times for the baseline algorithm (R-BCE) and our proposed algorithm (R-BCE+SLP(V)) on the VOC-MLT dataset are approximately 280.16 seconds and 548.03 seconds, respectively.

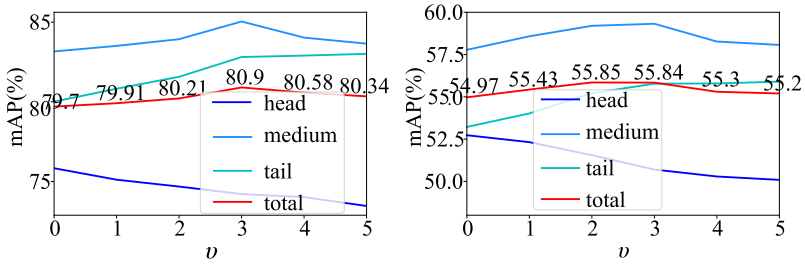
6.8 Effect of Hyper-parameter

Effect of hyper-parameter $\Delta\epsilon$**.** To understand how $\Delta\epsilon$ of Eq. (23) and Eq. (24) affect the results, we first vary $\Delta\epsilon$ in a set of {1,2,3,4,5,6} when the threshold η in Eq. (22) takes the mean value of the elements of $\boldsymbol{cof}$. The results are shown in Fig. 17. We can see that when $\Delta\epsilon = 3$, it can achieve the best performance on both VOC-MLT and COCO-MLT. If $\Delta\epsilon$ is too small, the perturbation term will not be effective for most samples. However, if $\Delta\epsilon$ is too large, it will increase the possibility of overlap between classes.



(a) VOC-MLT (b) COCO-MLT

Fig. 17. The effect of hyper-parameter $\Delta\epsilon$ to the mAP performance on VOC-MLT and COCO-MLT datasets.



(a) VOC-MLT (b) COCO-MLT

Fig. 18. The effect of hyper-parameter υ to the mAP performance on VOC-MLT and COCO-MLT datasets.

Effect of hyper-parameter υ. To explore the effects of different values of η in Eq. (23) and Eq. (24), we vary υ in a set of {0,1,2,3,4,5} to adjust η ($\eta = (\upsilon * min(\boldsymbol{cof}) + (5 - \upsilon) * max(\boldsymbol{cof}))/5$), and show the results for fixing $\Delta\epsilon = 3$ in Fig. 17. When $\upsilon = 0$, it means that all samples have added the perturbation term in the direction of loss increase; when $\upsilon = 5$, it is just the opposite. We can observe that as υ increases, the performance of the head classes gradually decreases because a smaller υ prevents the underfitting of the head classes; while the performance of the tail classes gradually increases because a larger υ prevents overfitting of tail classes. At $\upsilon = 3$ and $\upsilon = 2$, the proposed model achieves the best results on two datasets, VOC-MLT and COCO-MLT, respectively.

7 CONCLUSION AND FUTURE WORK

This study focuses on logit perturbation in MLL. We have analyzed the impact of the characteristics of multi-label training data on classification performance from three statistical characteristics: category proportion, variance, and co-occurrence. Based on the above quantitative analysis, this study proposes a new subclass-wise logit perturbation (SLP) that takes the above three characteristics into consideration. SLP implements different perturbations (in terms of magnitude and direction) to the negative subclass and alleviates the differences in proportion, variance, and co-occurrence within the negative class. Furthermore, we have theoretically analyzed existing and our proposed multi-label logit perturbation methods from a regularization view. Extensive experimental comparison results on several typical multi-label datasets demonstrate the proposed method's effectiveness.

Similar to existing logit perturbation methods, if the statistical distribution of data changes, the method proposed here may experience a decline in generalization performance. However, strategies such as continual learning [45] can be adopted. Our method can be naturally extended to continual learning, such as [46, 47]. In the future, we will focus more on the problem of multi-label learning under concept drift and consider designing more effective logit perturbation methods.

REFERENCES

[1] Alhuzali, H., and Ananiadou, S. (2021). SpanEmo: Casting multi-label emotion classification as span-prediction. *arXiv*2101.10038. https://doi:10.18653/v1/2021.eacl-main.135

[2] Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*. https://doi:10.1109/ICIP42928.2021.9506389

[3] Chen, T., Xu, M., Hui, X., Wu, H., and Lin, L. (2019). Learning semantic-specific graph representation for multi-label image recognition. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 522-531. https://doi:10.1109/ICCV.2019.00061

[4] Chen, Z. M., Wei, X. S., Wang, P., and Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5177-5186. https://doi:10.1109/CVPR.2019.00532

[5] Chu, P., Bian, X., Liu, S., and Ling, H. (2020). Feature space augmentation for long-tailed data. *In Computer Vision–ECCV 2020: 16th European Conference*, 694-710.

[6] Cui, Y., Jia, M., Lin, T. Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9268-9277. https://doi:10.1109/CVPR.2019.00949

[7] Deng, J., and Ren, F. (2020). Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. *IEEE Transactions on Affective Computing*, https://doi: 10.1109/TAFFC.2020.3034215

[8] Dong, Q., Gong, S., and Zhu, X. (2017). Class rectification hard mining for imbalanced deep learning. *In Proceedings of the IEEE International Conference on Computer Vision*, 1851-1860. https://doi:10.1109/ICCV.2017.205

[9] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2007). The pascal visual object classes challenge 2007 results.

[10] Guo, H., and Wang, S. (2021). Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15089-15098. https://doi:10.1109/CVPR46437.2021.01484

[11] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

[12] Huang, Y., Giledereli, B., Köksal, A., Özgür, A., and Ozkirimli, E. (2021). Balancing methods for multi-label text classification with long-tailed class distribution. *arXiv*2109.04712. https://doi:10.18653/v1/2021.emnlp-main.643

[13] Jiang, T., Wang, D., Sun, L., Yang, H., Zhao, Z., and Zhuang, F. (2021, May). Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 7987-7994. https://doi:10.1609/aaai.v35i9.16974

[14] Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. (2020, April). Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *In Proceedings of the AAAI conference on artificial intelligence*, 8018-8025. https://doi:10.1609/aaai.v34i05.6311

[15] Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. (2019). Decoupling representation and classifier for long-tailed recognition. *arXiv*1910.09217.

[16] Li, M., Su, F., Wu, O., and Zhang, J. (2022, June). Logit perturbation. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 1359-1366. https://doi:10.1609/aaai.v36i2.20024

[17] Li, M., Su, F., Wu, O., and Zhang, J. (2023). Class-level logit perturbation. *IEEE Transactions on Neural Networks and Learning Systems*.

[18] Li, Q., Qiao, M., Bian, W., and Tao, D. (2016). Conditional graphical lasso for multi-label image classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2977-2986. https://doi:10.1109/CVPR.2016.325

[19] Li, S., Gong, K., Liu, C. H., Wang, Y., Qiao, F., and Cheng, X. (2021). Metasaug: Meta semantic augmentation for long-tailed visual recognition. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5212-5221. https://doi:10.1109/CVPR46437.2021.00517

[20] Li, X., Zhao, F., and Guo, Y. (2014, January). Multi-label image classification with a probabilistic label enhancement model. *In UAI*, 1-10.

[21] Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. *In Proceedings of the IEEE international conference on computer vision*, 2980-2988. https://doi:10.1109/ICCV.2017.324

[22] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *In Computer Vision–ECCV 2014: 13th European Conference*, 740-755. https://doi:10.1007/978-3-319-10602-1

[23] Lin, Y., Hu, Q., Liu, J., Zhu, X., and Wu, X. (2021). MULFE: multi-label learning via label-specific feature space ensemble. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1-24. https://doi:10.1145/3451392

[24] Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A., Training deep neural networks on noisy labels with bootstrapping. *arXiv*1412.6596.

[25] Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. (2020). Long-tail learning via logit adjustment.*arXiv*2007.07314.
[26] Nguyen, H. D., Vu, X. S., and Le, D. T. (2021, May). Modular graph transformer networks for multi-label image classification. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 9092-9100. https://doi:10.1609/aaai.v35i10.17098
[27] Ridnik, T., Ben-Baruch, E., Zamir, N., Noy, A., Friedman, I., Protter, M., and Zelnik-Manor, L. (2021). Asymmetric loss for multi-label classification. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 82-91.
[28] Shen, L., Lin, Z., and Huang, Q. (2016). Relay backpropagation for effective learning of deep convolutional neural networks. *In Computer Vision–ECCV 2016: 14th European Conference*, 467-482. https://doi:10.1007/978-3-319-46478-7
[29] Song, P., Ju, A., Xu, W., and Guo, F. (2023, August). Adaptively weighted copy-decoupling resampling strategy for long-tailed multi-label classification. *In 2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, 437-442.
[30] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818-2826. https://doi:10.1109/CVPR.2016.308
[31] Wang, C., Gao, S., Wang, P., Gao, C., Pei, W., Pan, L., and Xu, Z. (2022). Label-aware distribution calibration for long-tailed classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1-13. https://doi:10.1109/TNNLS.2022.3213522
[32] Wang, H., Peng, C., Dong, H., Feng, L., Liu, W., Hu, T., and Chen, G. (2024). On the value of head labels in multi-label text classification. *ACM Transactions on Knowledge Discovery from Data*.
[33] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2285-2294.
[34] Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y., and Wu, C. (2021). Regularizing deep networks with semantic data augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3733-3748. https://doi:10.1109/TPAMI.2021.3052951
[35] Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G., and Wu, C. (2019). Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32.
[36] Wu, T., Huang, Q., Liu, Z., Wang, Y., and Lin, D. (2020). Distribution-balanced loss for multi-label classification in long-tailed datasets. *The European Conference on Computer Vision*, 162-178.
[37] Wu, Y., Shu, J., Xie, Q., Zhao, Q., and Meng, D. (2021, May). Learning to purify noisy labels via meta soft label corrector. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 10388-10396. https://doi:10.1609/aaai.v35i12.17244
[38] Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. L., and Le, Q. V. (2020). Adversarial examples improve image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 819-828. https://doi: 10.1109/CVPR42600.2020.00090
[39] Ye, J., He, J., Peng, X., Wu, W., and Qiao, Y. (2020). Attention-driven dynamic graph convolutional network for multi-label image recognition. *In Computer Vision–ECCV 2020: 16th European Conference*, 649-665.
[40] Yilmaz, S. F., Kaynak, E. B., Koç, A., Dibeklioğlu, H., and Kozat, S. S. (2021). Multi-label sentiment analysis on 100 languages with dynamic weighting for label imbalance. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi:10.1109/TNNLS.2021.3094304
[41] You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H., and Zhu, S. (2019). Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in Neural Information Processing Systems*.
[42] Zhu, F., Li, H., Ouyang, W., Yu, N., and Wang, X. (2017). Learning spatial regularization with image-level supervisions for multi-label image classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5513-5522. https://doi:10.1109/CVPR.2017.219
[43] Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019, May). Theoretically principled trade-off between robustness and accuracy. *In International Conference on Machine Learning*, 7472-7482. PMLR.
[44] Xu, H., Liu, X., Li, Y., Jain, A., and Tang, J. (2021, July). To be robust or to be fair: Towards fairness in adversarial training. *In International conference on machine learning*, 11492-11501. PMLR.
[45] Zhou, D. W., Sun, H. L., Ning, J., Ye, H. J., and Zhan, D. C. (2024). Continual learning with pre-trained models: A survey. *arXiv*2401.16386.
[46] Du, K., Lyu, F., Li, L., Hu, F., Feng, W., Xu, F., Xi, X., and Cheng, H. (2023). Multi-label continual learning using augmented graph convolutional network. *IEEE Transactions on Multimedia*.
[47] Chrysakis, A., and Moens, M. F. (2020, November). Online continual learning from imbalanced data. *In International Conference on Machine Learning*, 1952-1961.
[48] Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

A. PROOF FOR THEOREM 1

Proof. Xu et al. [44] proved that $\boldsymbol{w} = \mathbf{1}$ when the data distribution in Eq. (4.1) is given (Lemma 1 in [44]). According to Lemma 1 [44], we can easily prove that when $P_A : P_B : P_C = \Gamma : \Gamma : 1 - 2\Gamma$ and $0 < \Gamma < \frac{1}{3}$, $\boldsymbol{w} = \mathbf{1}$ holds. Thus,

$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i + b$. Then Eq. (4.1) can be written as follows.

$$b^* = \arg\min_b \Pr.\left(\mathbb{S}\left(\sum_{i=1}^{d} x_i + b + \Delta u_c^*\right) \neq y_c\right). \tag{A.1}$$

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 $\mathcal{R}_{rob}(f_{\text{rob}})$ to denote the robust error after logit perturbation.

$$\begin{aligned}
\mathcal{R}_{rob}(f_{\text{rob}}) \quad & \propto \Gamma \cdot \Pr.\left(\exists \left\|\Delta u_1^-\right\| \leq \epsilon \cdot \rho_1, \mathbb{S}\left(u + \Delta u_1^-\right) \neq +1 \mid y = +1\right) \\
& + \Gamma \cdot \Pr.\left(\exists \left\|\Delta u_1^-\right\| \leq \epsilon \cdot \rho_1, \mathbb{S}\left(u + \Delta u_1^-\right) \neq -1 \mid y = -1\right) \\
& + (1-2\Gamma) \cdot \Pr.\left(\exists \left\|\Delta u_2^-\right\| \leq \epsilon \cdot \rho_2, \mathrm{S}\left(u + \Delta u_2^-\right) \neq -1 \mid y = -1\right) \\
& = \Gamma \cdot \min_{\substack{|\Delta u_1^-| \leq |\epsilon \cdot \rho_1| \\ \Delta u_1^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u + \Delta u_1^-\right) \neq +1 \mid y = +1\right) + \Gamma \cdot \min_{\substack{|\Delta u_1^-| \leq |\epsilon \cdot \rho_1| \\ \Delta u_1^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u + \Delta u_1^-\right) \neq -1 \mid y = -1\right) \\
& + (1-2\Gamma) \cdot \min_{\substack{|\Delta u_2^-| \leq |\epsilon \cdot \rho_2| \\ \Delta u_2^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u + \Delta u_2^-\right) \neq -1 \mid y = -1\right) \\
& = \Gamma \cdot \Pr.\left(\mathbb{S}\left(u - \epsilon \cdot \rho_1\right) \neq +1 \mid y = +1\right) + \Gamma \cdot \Pr.\left(\mathbb{S}\left(u + \epsilon \cdot \rho_1\right) \neq -1 \mid y = -1\right) \\
& + (1-2\Gamma) \cdot \Pr.\left(\mathbb{S}\left(u - \epsilon \cdot \rho_2\right) \neq -1 \mid y = -1\right) \\
& = \Gamma \cdot \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon \cdot \rho_1 < 0 \mid y = +1\right\} + \Gamma \cdot \Pr.\left\{\sum_{i=1}^{d} x_i + b + \epsilon \cdot \rho_1 > 0 \mid y = -1\right\} \\
& + (1-2\Gamma) \cdot \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon \cdot \rho_2 > 0 \mid y = -1\right\} \\
& = \Gamma \cdot \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} - \frac{b - \epsilon \cdot \rho_1}{\sqrt{d}\sigma}\right\} + \Gamma \cdot \Pr.\left\{\mathcal{N}(0,1) < \frac{b + \epsilon \cdot \rho_1}{\sqrt{d}\sigma}\right\} \\
& + (1-2\Gamma) \cdot \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b - \epsilon \cdot \rho_2}{\sqrt{d}\sigma}\right\}.
\end{aligned} \tag{A.2}$$

For ease of computation, let $\rho_2 = \frac{2b}{\epsilon} - \rho_1$. The optimal b^* to minimize $\mathcal{R}_{\text{slp}}(f)$ is achieved at the point that $\frac{\partial R_{\text{slp}}(f)}{\partial b} = 0$. Then we can get the optimal b^* :

$$b^* = -\frac{d\eta}{2} + \frac{d\sigma^2 \log\left(\frac{\Gamma}{1-\Gamma}\right)}{2\epsilon \cdot \rho_1 - d\eta}. \tag{A.3}$$

By taking b^* into $\mathcal{R}_{nat}(f_{\text{rob}}, A)$, $\mathcal{R}_{nat}(f_{\text{rob}}, B)$, and $\mathcal{R}_{nat}(f_{\text{rob}}, C)$, we can get the theorem.

$$\begin{aligned}
\mathcal{R}_{nat}(f_{\text{rob}}, A) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} - \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < -\frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma}\right\}, \\
\mathcal{R}_{nat}(f_{\text{rob}}, B) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma}\right\}, \\
\mathcal{R}_{nat}(f_{\text{rob}}, C) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{\log(\frac{\Gamma}{1-\Gamma})}{\Lambda} - \frac{3\sqrt{d}\eta}{2\sigma}\right\},
\end{aligned} \tag{A.4}$$

where $\Lambda = \frac{2\epsilon \cdot \rho_1 - d\eta}{\sqrt{d}\sigma}$.

B. COROLLARY 1

Proof. According to Eq. (A.3), we compute the partial derivatives of b^* with respect to ρ_1 to proof the corollary.

$$\frac{\partial b^*}{\partial \rho_1} = -\frac{2d\epsilon\sigma^2 \log\left(\frac{\Gamma}{1-\Gamma}\right)}{(2\epsilon \cdot \rho_1 - d\eta)^2} > 0. \tag{A.5}$$

b^* is a monotonically increasing function of ρ_1. According to Eq. (A.4) and $\rho_2 = \frac{2b}{\epsilon} - \rho_1$, the corollary holds.

C. COROLLARY 2

Proof. by taking (A.3) into $\mathcal{R}_{bdy}\left(f_{\text{rob}}\right)$, we can get:

$$\begin{aligned}
\mathcal{R}_{bdy}\left(f_{\text{rob}}\right) &= \mathcal{R}_{rob}\left(f_{\text{rob}}\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}\right) \\
&= \mathcal{R}_{rob}\left(f_{\text{rob}}, A\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, A\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, B\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, B\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, C\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, C\right) \\
&= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} - \frac{b^* - \epsilon\cdot\rho_1}{\sqrt{d}\sigma}\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{b^* + \epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}(0,1) < \frac{b^*}{\sqrt{d}\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^* - \epsilon\cdot\rho_2}{\sqrt{d}\sigma}\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{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}(0,1) < -\frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}(0,1) < -\frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{3\sqrt{d}\eta}{2\sigma}\right\} \\
&< 2\Pr.\left\{0 < \mathcal{N}(0,1) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} + \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{\sqrt{d}\eta}{2\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{\Gamma}{1-\Gamma}\right)}{\Lambda} - \frac{3\sqrt{d}\eta}{2\sigma}\right\} \\
&< 2\Pr.\left\{0 < \mathcal{N}(0,1) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} + \Pr.\left\{-\frac{\sqrt{d}\eta}{\sigma} < \mathcal{N}(0,1) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\}.
\end{aligned} \tag{A.6}$$

The corollary holds.

D. PROOF FOR THEOREM 2

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

$$
\begin{aligned}
\mathcal{R}_{rob}\left(f_{\text{rob}}\right) \quad & \propto \Pr.\left(\exists \left\|\Delta u_1^-\right\| \leq \epsilon \cdot \rho_1, \mathbb{S}\left(u+\Delta u_1^-\right) \neq +1 \mid y=+1\right) \\
& +\Pr.\left(\exists \left\|\Delta u_1^-\right\| \leq \epsilon \cdot \rho_1, \mathbb{S}\left(u+\Delta u_1^-\right) \neq -1 \mid y=-1\right) \\
& +\Pr.\left(\exists \left\|\Delta u_2^-\right\| \leq \epsilon \cdot \rho_2, \mathbb{S}\left(u+\Delta u_2^-\right) \neq -1 \mid y=-1\right) \\
& = \min_{\substack{|\Delta u_1^-| \leq |\epsilon \cdot \rho_1| \\ \Delta u_1^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u+\Delta u_1^-\right) \neq +1 \mid y=+1\right) + \min_{\substack{|\Delta u_1^-| \leq |\epsilon \cdot \rho_1| \\ \Delta u_1^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u+\Delta u_1^-\right) \neq -1 \mid y=-1\right) \\
& + \min_{\substack{|\Delta u_2^-| \leq |\epsilon \cdot \rho_2| \\ \Delta u_2^- \cdot \epsilon \geq 0}} \Pr.\left(\mathbb{S}\left(u+\Delta u_2^-\right) \neq -1 \mid y=-1\right) \\
& = \Pr.\left(\mathbb{S}\left(u-\epsilon \cdot \rho_1\right) \neq +1 \mid y=+1\right) + \Pr.\left(\mathbb{S}\left(u+\epsilon \cdot \rho_1\right) \neq -1 \mid y=-1\right) \\
& +\Pr.\left(\mathbb{S}\left(u-\epsilon \cdot \rho_2\right) \neq -1 \mid y=-1\right) \\
& = \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon \cdot \rho_1 < 0 \mid y=+1\right\} + \Pr.\left\{\sum_{i=1}^{d} x_i + b + \epsilon \cdot \rho_1 > 0 \mid y=-1\right\} \\
& +\Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon \cdot \rho_2 > 0 \mid y=-1\right\} \\
& = \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{K\sigma} - \frac{b-\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} + \Pr.\left\{\mathcal{N}(0,1) < \frac{b+\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} + \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{(1-2K)\sigma} + \frac{b-\epsilon\cdot\rho_2}{\sqrt{d}(1-2K)\sigma}\right\}.
\end{aligned} \tag{A.7}
$$

For ease of computation, let $\rho_2 = \frac{(1-3K)d\eta+(2K-1)\epsilon\cdot\rho_1+(1-K)b}{K\epsilon}$. The optimal b^* to minimize $\mathcal{R}_{\text{slp}}(f)$ is achieved at the point that $\frac{\partial R_{\text{slp}}(f)}{\partial b} = 0$. Then we can get the optimal b^* :

$$
b^* = -\frac{d\eta}{2} + \frac{dK^2\sigma^2 \log 2}{d\eta - 2\epsilon\cdot\rho_1}. \tag{A.8}
$$

By taking b^* into $\mathcal{R}_{nat}\left(f_{\text{rob}}, A\right)$, $\mathcal{R}_{nat}\left(f_{\text{rob}}, B\right)$, and $\mathcal{R}_{nat}\left(f_{\text{rob}}, C\right)$, we can get the theorem.

$$
\begin{aligned}
\mathcal{R}_{nat}\left(f_{\text{rob}}, A\right) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{K\sigma} - \frac{b^*}{\sqrt{d}K\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\right\}, \\
\mathcal{R}_{nat}\left(f_{\text{rob}}, B\right) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{b^*}{\sqrt{d}K\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\right\}, \\
\mathcal{R}_{nat}\left(f_{\text{rob}}, C\right) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{(1-2K)\sigma} + \frac{b^*}{\sqrt{d}(1-2K)\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{K}{1-2K}\cdot\frac{\log 2}{\Lambda} - \frac{3\sqrt{d}\eta}{2(1-2K)\sigma}\right\},
\end{aligned} \tag{A.9}
$$

where $\Lambda = \frac{d\eta-2\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}$.

E. COROLLARY 3

Proof. According to Eq. (A.8), we compute the partial derivatives of b^* with respect to ρ_1 to proof the corollary.

$$
\frac{\partial b^*}{\partial \rho_1} = \frac{2\epsilon d K^2\sigma^2 \log 2}{\left(d\eta - 2\epsilon\cdot\rho_1\right)^2} > 0. \tag{A.10}
$$

b^* is a monotonically increasing function of ρ_1. According to Eq. (A.9) and $\rho_2 = \frac{(1-3K)d\eta+(2K-1)\epsilon\cdot\rho_1+(1-K)b}{K\epsilon}$, the corollary holds.

F. COROLLARY 4

Proof. by taking (A.8) into $\mathcal{R}_{bdy}\left(f_{\text{rob}}\right)$, we can get:

$$
\begin{aligned}
\mathcal{R}_{bdy}\left(f_{\text{rob}}\right) &= \mathcal{R}_{rob}\left(f_{\text{rob}}\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}\right) \\
&= \mathcal{R}_{rob}\left(f_{\text{rob}}, A\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, A\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, B\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, B\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, C\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, C\right) \\
&= \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{K\sigma} - \frac{b^*-\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{K\sigma} - \frac{b^*}{\sqrt{d}K\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{b^*+\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{b^*}{\sqrt{d}K\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{(1-2K)\sigma} + \frac{b^*-\epsilon\cdot\rho_2}{\sqrt{d}(1-2K)\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} \\
&= \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{K}{1-2K}\frac{\log 2}{\Lambda} - \frac{3\sqrt{d}\eta}{2(1-2K)\sigma}\right\} \\
&< 2\Pr.\left\{0 < \mathcal{N}\left(0,1\right) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} + \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log 2}{\Lambda} - \frac{\sqrt{d}\eta}{2K\sigma} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log 2}{\Lambda} - \frac{3\sqrt{d}\eta}{2(1-2K)\sigma}\right\} \\
&< 2\Pr.\left\{0 < \mathcal{N}\left(0,1\right) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma}\right\} + \Pr.\left\{-\frac{3\sqrt{d}\eta}{2(1-2K)\sigma} < \mathcal{N}\left(0,1\right) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}K\sigma} - \frac{\sqrt{d}\eta}{2K\sigma}\right\}.
\end{aligned}
\tag{A.11}
$$

The corollary holds.

G. PROOF FOR THEOREM 3

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

$$
\begin{aligned}
\mathcal{R}_{rob}\left(f_{\text{rob}}\right) &\propto P\cdot\Pr.\left(\mathrm{S}\left(u\right)\neq+1 \mid y=+1\right) + \left(1-P\right)\cdot\Pr.\left(\exists\left\|\Delta u_1^-\right\| \le \epsilon\cdot\rho_1, \mathbb{S}\left(u+\Delta u_1^-\right)\neq-1 \mid y=-1\right) \\
&+ \Pr.\left(\exists\left\|\Delta u_2^-\right\| \le \epsilon\cdot\rho_2, \mathrm{S}\left(u+\Delta u_2^-\right)\neq-1 \mid y=-1\right) \\
&= P\cdot\Pr.\left(\mathrm{S}\left(u\right)\neq+1 \mid y=+1\right) + \left(1-P\right)\cdot\min_{\substack{|\Delta u_1^-|\le|\epsilon\cdot\rho_1| \\ \Delta u_1^-\cdot\epsilon\ge 0}}\Pr.\left(\mathbb{S}\left(u+\Delta u_1^-\right)\neq-1 \mid y=-1\right) \\
&+ \min_{\substack{|\Delta u_2^-|\le|\epsilon\cdot\rho_2| \\ \Delta u_2^-\cdot\epsilon\ge 0}}\Pr.\left(\mathbb{S}\left(u+\Delta u_2^-\right)\neq-1 \mid y=-1\right) \\
&= P\cdot\Pr.\left(\mathrm{S}\left(u\right)\neq+1 \mid y=+1\right) + \left(1-P\right)\cdot\Pr.\left(\mathbb{S}\left(u+\epsilon\cdot\rho_1\right)\neq-1 \mid y=-1\right) \\
&+ \Pr.\left(\mathbb{S}\left(u-\epsilon\cdot\rho_2\right)\neq-1 \mid y=-1\right) \\
&= P\cdot\Pr.\left\{\sum_{i=1}^{d} x_i + b < 0 \mid y=+1\right\} + \left(1-P\right)\cdot\Pr.\left\{\sum_{i=1}^{d} x_i + b + \epsilon\cdot\rho_1 > 0 \mid y=-1\right\} \\
&+ \Pr.\left\{\sum_{i=1}^{d} x_i + b - \epsilon\cdot\rho_2 > 0 \mid y=-1\right\} \\
&= P\cdot\Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{\sigma} - \frac{b}{\sqrt{d}\sigma}\right\} + \left(1-P\right)\cdot\Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\sqrt{d}\eta}{\sigma} + \frac{b+\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b-\epsilon\cdot\rho_2}{\sqrt{d}\sigma}\right\}.
\end{aligned}
\tag{A.12}
$$

For ease of computation, let $\rho_2 = \frac{2b}{\epsilon}$. The optimal b^* to minimize $\mathcal{R}_{\text{slp}}(f)$ is achieved at the point that $\frac{\partial R_{\text{slp}}(f)}{\partial b} = 0$. Then we can get the optimal b^* :

$$b^* = -d\eta + \frac{d\sigma^2 \log\left(\frac{1-P}{1+P}\right)}{\epsilon \cdot \rho_1} - \frac{\epsilon \cdot \rho_1}{2}. \tag{A.13}$$

By taking b^* into $\mathcal{R}_{nat}(f_{\text{rob}}, A)$, $\mathcal{R}_{nat}(f_{\text{rob}}, B)$, and $\mathcal{R}_{nat}(f_{\text{rob}}, C)$, we can get the theorem.

$$\begin{aligned}
\mathcal{R}_{nat}(f_{\text{rob}}, A) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} - \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < -\frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} + \frac{\Lambda}{2}\right\}, \\
\mathcal{R}_{nat}(f_{\text{rob}}, B) &= \Pr.\left\{\mathcal{N}(0,1) < \frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} - \frac{\Lambda}{2}\right\}, \\
\mathcal{R}_{nat}(f_{\text{rob}}, C) &= \Pr.\left\{\mathcal{N}(0,1) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} = \Pr.\left\{\mathcal{N}(0,1) < \frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} - \frac{\Lambda}{2} - \frac{2\sqrt{d}\eta}{\sigma}\right\},
\end{aligned} \tag{A.14}$$

where $\Lambda = \frac{\epsilon \cdot \rho_1}{\sqrt{d}\sigma}$.

H. COROLLARY 5

Proof. According to Eq. (A.13), we compute the partial derivatives of b^* with respect to ρ_1 to proof the corollary.

$$\frac{\partial b^*}{\partial \rho_1} = -\frac{\epsilon}{2} - \frac{d\sigma^2 \log\left(\frac{1-P}{1+P}\right)}{\epsilon \cdot {\rho_1}^2}. \tag{A.15}$$

When $\frac{\partial b^*}{\partial \rho_1} > 0, b^*$ increases as ρ_1 increases. We reorganize $\frac{\partial b^*}{\partial \rho_1} > 0$ to get the following equation.

$$\frac{1-P}{1+P} < e^{-\frac{(\epsilon \cdot \rho_1)^2}{2d\sigma^2}}. \tag{A.16}$$

When Eq. (A.16) holds, b^* is a monotonically increasing function of ρ_1. According to Eq. (A.14) and $\rho_2 = \frac{2b}{\epsilon}$, the corollary holds.

I. COROLLARY 6

Proof. by taking (A.13) into $\mathcal{R}_{bdy}\left(f_{\text{rob}}\right)$, we can get:

$$
\begin{aligned}
\mathcal{R}_{bdy}\left(f_{\text{rob}}\right) &= \mathcal{R}_{rob}\left(f_{\text{rob}}\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}\right) \\
&= \mathcal{R}_{rob}\left(f_{\text{rob}}, A\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, A\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, B\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, B\right) + \mathcal{R}_{rob}\left(f_{\text{rob}}, C\right) - \mathcal{R}_{nat}\left(f_{\text{rob}}, C\right) \\
&= \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\sqrt{d}\eta}{\sigma} + \frac{b^* + \epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^* - \epsilon\cdot\rho_2}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\sqrt{d}\eta}{\sigma} + \frac{b^*}{\sqrt{d}\sigma}\right\} \\
&= \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} - \frac{\Lambda}{2} + \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} - \frac{\Lambda}{2}\right\} \\
&+ \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} + \frac{\Lambda}{2}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < \frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} - \frac{\Lambda}{2} - \frac{2\sqrt{d}\eta}{\sigma}\right\} \\
&< \Pr.\left\{0 < \mathcal{N}\left(0,1\right) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} + \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} + \frac{\Lambda}{2}\right\} - \Pr.\left\{\mathcal{N}\left(0,1\right) < -\frac{\log\left(\frac{1-P}{1+P}\right)}{\Lambda} + \frac{\Lambda}{2} - \frac{2\sqrt{d}\eta}{\sigma}\right\} \\
&< \Pr.\left\{0 < \mathcal{N}\left(0,1\right) < \frac{\epsilon\cdot\rho_1}{\sqrt{d}\sigma}\right\} + \Pr.\left\{0 < \mathcal{N}\left(0,1\right) < \frac{2\sqrt{d}\eta}{\sigma}\right\}.
\end{aligned}
\tag{A.17}
$$

The corollary holds.

A ONLINE RESOURCES

https://github.com/ruby-yu-zhu/Subclass/tree/master/slp