Learning from Long-Tailed Noisy Data with Sample Selection and Balanced Loss

Lefan Zhang, **Zhang-Hao Tian**, **Wujun Zhou** and **Wei Wang**
National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{zhanglf, tianzh, zhouwujun, wangw}@lamda.nju.edu.cn

Abstract

The success of deep learning depends on large-scale and well-curated training data, while data in real-world applications are commonly long-tailed and noisy. Existing methods are usually dependent on label frequency to tackle class imbalance, while the model bias on different classes is not directly related to label frequency and the true label frequency is inaccessible under label noise. To solve this, we propose a robust method for learning from long-tailed noisy data with sample selection and balanced loss. Specifically, we separate the noisy training data into clean labeled set and unlabeled set with sample selection, and train the deep neural network in a semi-supervised manner with a balanced loss based on model bias. Extensive experiments on benchmarks demonstrate that our method outperforms existing state-of-the-art methods.

1 Introduction

Deep neural networks have made great successes in machine learning applications [He *et al.*, 2016] but require well-curated data for training. These data, such as ImageNet [Russakovsky *et al.*, 2015] and MS-COCO [Lin *et al.*, 2014], are usually artificially balanced across classes with clean labels obtained by manual labeling, which is costly and time-consuming. However, the data in real-world applications are long-tailed and noisy, since data from specific classes are difficult to acquire and labels are usually collected without expert annotations. To take WebVision dataset as an example, it exhibits long-tailed distribution, where the sample size of each class varies from 362 (Windsor tie) to 11,129 (ashcan), and contains about 20% noisy labels [Li *et al.*, 2017]. Thus, developing robust learning methods for long-tailed noisy data is a great challenge.

Many methods have been proposed for long-tailed learning or learning with noisy labels. In terms of long-tailed learning, re-sampling methods [Jeatrakul *et al.*, 2010], re-weighting methods [Cui *et al.*, 2019; Cao *et al.*, 2019; Menon *et al.*, 2021], transfer learning methods [Liu *et al.*, 2019] and two-stage methods [Kang *et al.*, 2019; Cao *et al.*, 2019] are included; in terms of learning with noisy labels, designing noise-robust loss functions [Ghosh *et al.*, 2017], constructing unbiased loss terms with the transition matrix [Patrini *et al.*, 2017], filtering clean samples based on small-loss criterion [Han *et al.*, 2018; Li *et al.*, 2020] and correcting the noisy labels [Tanaka *et al.*, 2018; Yi and Wu, 2019] are included. Despite learning from long-tailed or noisy data has been well studied, these methods cannot tackle long-tailed noisy data in real-world applications. A few methods are proposed to deal with long-tailed noisy data [Karthik *et al.*, 2021; Yi *et al.*, 2022; Fang *et al.*, 2023; Zhao *et al.*, 2023], which count on observed label frequency to handle class imbalance. Nevertheless, the model bias on different classes is not merely relevant to label frequency and the true frequency is unavailable owing to label noise.

To deal with long-tailed noisy data, an intuitive way is to select clean samples with small-loss criterion and then apply long-tailed learning methods. For the sample selection process, [Gui *et al.*, 2021] revealed that the losses of samples with different labels are incomparable and chose each class a threshold for applying the small-loss criterion. For long-tailed learning, existing methods are commonly based on label frequency to prevent head classes from dominating the training process. However, it is shown that the model bias on different classes may not be directly related to label frequency (see Appendix[1] E), and the true label frequency is also unknown under label noise. In this paper, we propose a robust method for learning from long-tailed noisy data. Specifically, we separate the noisy training data into clean labeled set and unlabeled set with class-aware sample selection and then train the model with a balanced loss based on model bias in a semi-supervised manner. Experiments on the long-tailed versions of CIFAR-10 and CIFAR-100 with synthetic noise and the long-tailed versions of mini-ImageNet-Red, Clothing1M, Food-101N, Animal-10N and WebVision with real-world noise demonstrate the superiority of our method.

2 Related Work

Learning from Long-Tailed Data. The purpose of long-tailed learning is to alleviate the performance degradation caused by the small sample sizes of tail classes. Popular methods include re-sampling/re-weighting methods which commonly simulate a balanced training set by paying more attention to tail classes [Jeatrakul *et al.*, 2010; Cui *et al.*,

[1]The appendix is available at https://arxiv.org/abs/2211.10906

2019], transfer learning methods which boost the recognition performance on tail classes by utilizing the knowledge learned from head classes [Liu *et al.*, 2019] and two-stage methods which argue that decoupling the representation learning and the classifier learning leads to better performance [Kang *et al.*, 2019]. Recently, a new line of works are proposed, which utilize the contrastive learning to learn a class-balanced feature space and further achieve better long-tailed learning performance [Kang *et al.*, 2021].

Learning from Noisy Data. The purpose of learning from noisy data is to achieve good generalization capability in the presence of noisy labels [Song *et al.*, 2022]. Works proposed to address the problem of learning with noisy labels can be divided into three categories: 1) methods based on robust loss [Ghosh *et al.*, 2017], 2) methods based on loss correction [Patrini *et al.*, 2017] and 3) methods based on noise cleansing [Han *et al.*, 2018; Tanaka *et al.*, 2018; Yi and Wu, 2019]. According to whether remove or correct the noisy labels, the third category can be further divided into sample selection methods and label correction methods. Sample selection methods try to identify clean samples, which are commonly based on small-loss criterion [Han *et al.*, 2018; Gui *et al.*, 2021]. Label correction methods try to improve the quality of raw labels [Tanaka *et al.*, 2018; Yi and Wu, 2019]. Sample selection can also be combined with label correction, e.g., DivideMix [Li *et al.*, 2020] conducts sample selection via a two-component gaussian mixture model and then applies the semi-supervised learning technique MixMatch with label correction, NDCC [Qi and Chelmis, 2023] automatically chooses a loss value threshold for sample selection and uses counterfactual learning to correct the noisy labels.

Learning from Long-Tailed Noisy Data. Methods designed for learning from long-tailed noisy data are under-explored. [Shu *et al.*, 2019] and [Jiang *et al.*, 2021] learned a weighting function from training data in a meta-learning manner, which can be applied to deal with long-tailed noisy data, yet these methods require extra unbiased meta-data. [Zhang and Pfister, 2021] used a dictionary to store valuable training samples as a proxy of meta-data. These meta-learning methods attempt to learn a weighting function mapping training loss into sample weight and assign large weights to clean samples and tail samples. Nevertheless, in the learning process, clean samples usually have small losses and tail samples usually have large losses. [Cao *et al.*, 2021] proposed a heteroskedastic adaptive regularization, which regularizes different regions of input space differently. However, they assigned the same regularization strength to samples with the same observed label without considering label noise, which made the method sensitive to noisy data. [Karthik *et al.*, 2021] followed a two-stage training approach by using self-supervised learning to train an unbiased feature extractor and fine-tuning with an appropriate loss. [Yi *et al.*, 2022] proposed an iterative framework H2E, which first trains a noise identifier invariant to the class and context distributional change and then learns a robust classifier. These methods use the observed class distribution to handle class imbalance, while in real-world applications the class distribution is inaccessible under label noise. [Lu *et al.*, 2023] devised the two-stage bi-dimensional sample selection and then trained the model in a semi-supervised manner. However, they did not consider long-tailed distribution in the semi-supervised learning process. [Fang *et al.*, 2023] proposed a cross-augmentation matching criterion and a regularization to detect the noisy samples and eliminate their effects, then a penalization term based on confidence level was used for rebalancing, where the estimation of confidence level was still influenced by the corrupted label frequency. [Zhao *et al.*, 2023] adopted sample selection with harmonizing factor strategy and dynamic cost-sensitive learning based on the estimated clean sample frequency of each class. Nevertheless, the estimation of clean sample frequency may not be accurate and the selected noisy samples were not used in the following process. Generally, most existing methods depend on label frequency to deal with long-tailed distribution, while the model bias on different classes is not solely related to label frequency and the observed frequency is corrupted due to label noise.

Procedure 1 Class-Aware Sample Selection (CASS)

Require: Parameter of model θ, training data $(\mathcal{X}, \mathcal{Y})$.
1: **for** c = 1 to C **do**
2: Draw $\tilde{D}_c = \{(\boldsymbol{x}_i, \tilde{y}_i) | (\boldsymbol{x}_i, \tilde{y}_i) \in (\mathcal{X}, \mathcal{Y}) \wedge \tilde{y}_i = c\}$ from $\tilde{D} = (\mathcal{X}, \mathcal{Y})$
3: $\mathcal{W}_c = \text{GMM}(\{\text{SoftmaxCE}(\boldsymbol{x}_i, \tilde{y}_i; \theta) | (\boldsymbol{x}_i, \tilde{y}_i) \in \tilde{D}_c\})$
4: $\mathcal{L}_c = \{(\boldsymbol{x}_i, \tilde{y}_i) | (\boldsymbol{x}_i, \tilde{y}_i, w_i) \in (\tilde{D}_c, \mathcal{W}_c) \wedge w_i > \frac{1}{2}\}$
5: $\mathcal{U}_c = \{\boldsymbol{x}_i | (\boldsymbol{x}_i, \tilde{y}_i, w_i) \in (\tilde{D}_c, \mathcal{W}_c) \wedge w_i \leq \frac{1}{2}\}$
6: **end for**
Ensure: $\mathcal{L} = \cup_{i=1}^{C} \mathcal{L}_i, \mathcal{U} = \cup_{i=1}^{C} \mathcal{U}_i$.

3 Methodology

Let $\mathcal{X}$ denote the instance space, for each $\boldsymbol{x} \in \mathcal{X}$, there exists a true label $y \in \mathcal{Y} = \{1, \ldots, C\}$, where C is the number of classes. Let $\tilde{D} = \{(\boldsymbol{x}_1, \tilde{y}_1), \ldots, (\boldsymbol{x}_N, \tilde{y}_N)\}$ denote the training data, where $\boldsymbol{x}_i \in \mathcal{X}$, $\tilde{y}_i \in \mathcal{Y}$ is the observed label of $\boldsymbol{x}_i$ that may be corrupted and N is the number of training samples. In real-world applications, the data may follow a long-tailed distribution. Let $\tilde{D}_c = \{(\boldsymbol{x}_i, \tilde{y}_i) | (\boldsymbol{x}_i, \tilde{y}_i) \in \tilde{D} \wedge \tilde{y}_i = c\}$ and $n_c = |\tilde{D}_c|$. Without loss of generality, the classes are sorted by their cardinalities n_c in the decreasing order, i.e., $n_1 > n_2 > \cdots > n_C$, and the imbalance ratio $\rho = n_1/n_C$. The deep neural network $f(\cdot; \theta) : \mathcal{X} \to \mathbb{R}^C$ is learned from the long-tailed and noisy data $\tilde{D}$, where θ represents the model parameter. Given $\boldsymbol{x} \in \mathcal{X}$, the network outputs $f(\boldsymbol{x}; \theta) = [f_1(\boldsymbol{x}; \theta), \ldots, f_C(\boldsymbol{x}; \theta)]^T$ and $f_i(\boldsymbol{x}; \theta)$ represents the output logit of class i, we omit the model parameter θ and denote $f(\boldsymbol{x}; \theta)$ as $f(\boldsymbol{x})$ for brevity. Let $p^i_{model}(\boldsymbol{x}) = \frac{\exp(f_i(\boldsymbol{x}))}{\sum_{j=1}^{C} \exp(f_j(\boldsymbol{x}))}$, the classifier induced by f is $\phi_f(\boldsymbol{x}) = \arg\max_{i \in \{1,\ldots,C\}} p^i_{model}(\boldsymbol{x})$. For $(\boldsymbol{x}, \tilde{y})$, the loss is calculated as $\ell(f(\boldsymbol{x}), \tilde{y})$ with a loss function $\ell(\cdot, \cdot)$. The empirical loss of f on $\tilde{D}$ is $\frac{1}{N} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), \tilde{y}_i)$. We focus on learning the optimal model parameter θ^* which minimizes the expected loss, i.e., $\theta^* = \arg\min_\theta \mathbb{E}_{(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}; \theta), y)]$.

The training data in real-world applications usually contain noisy labels, while deep neural networks are generally learned by minimizing the empirical risk on the training data. [Zhang *et al.*, 2021] demonstrated that noisy labels can be easily fitted by deep neural networks, which harms the generalization of the neural networks. In order to alleviate the effect of noisy labels, we adopt the popular strategy which first warms up the model and then selects clean samples with small-loss criterion [Han *et al.*, 2018; Li *et al.*, 2020]. Since the training data are long-tailed in real-world applications, we introduce a regularization term L_{reg} in the warm up process to prevent the model from being influenced by the long-tailed distribution:

$$L_{reg} = \sum_{i=1}^{C} \frac{n_C}{n_i} \pi_i \log \left(\pi_i \Big/ \frac{1}{C} \sum_{j=1}^{C} \sum_{(\boldsymbol{x},\tilde{y}) \in \tilde{D}_j} \frac{1}{n_j} p^i_{model}(\boldsymbol{x}) \right),$$

where $\pi_i = \frac{1}{C}$. L_{reg} forces the model's average output on all classes to be the uniform distribution. Considering that head classes have more samples and the calculation of the model's average output may be biased, we conduct the calculation in a class-balanced manner by assigning a weight $\frac{1}{n_j}$ to samples from class j. We pay more attention to tail classes by assigning a regularization strength $\frac{n_C}{n_i}$ to class i, which guarantees a decent performance on tail classes. During warming up, hyper-parameter λ_{warm} controls the strength of L_{reg}:

$$L = L_{CE} + \lambda_{warm} L_{reg}, \tag{1}$$

where L_{CE} denotes the cross-entropy loss.

After warming up, small-loss criterion can be used to select small-loss samples as clean ones following [Arpit *et al.*, 2017]. Recently, [Gui *et al.*, 2021] revealed that the losses of samples with different labels may not be comparable and proposed to select samples class by class accordingly. They set each class a threshold based on the noise rate. Unfortunately, each class's noise rate is unknown in practice. Instead, we adopt a two-component (g_1 and g_0) Gaussian Mixture Model (GMM) for each class in the sample selection process to fit the loss as that in [Arazo *et al.*, 2019] and [Li *et al.*, 2020], where g_1 represents the clean distribution and g_0 represents the noisy distribution. For class i, let $L(\tilde{D}_i)$ denote the samples' cross-entropy loss, and it reflects how well the model fits the training samples:

$$L(\tilde{D}_i) = \left\{ -\log \left(\frac{\exp(f_i(\boldsymbol{x}_j))}{\sum_{k=1}^{C} \exp(f_k(\boldsymbol{x}_j))} \right) \middle| (\boldsymbol{x}_j, \tilde{y}_j) \in \tilde{D}_i \right\}_{j=1}^{n_i}.$$

A two-component GMM can be fitted with respect to $L(\tilde{D}_i)$, and the clean probability of a sample $(\boldsymbol{x}_j, \tilde{y}_j)$ is given by its posterior probability of belonging to g_1 $P(g_1|L(\tilde{D}_i), (\boldsymbol{x}_j, \tilde{y}_j))$. The sample with clean probability $P(g_1|L(\tilde{D}_i), (\boldsymbol{x}_j, \tilde{y}_j)) > 1/2$ is selected as clean; otherwise, the sample is regarded as noisy. We determine whether a sample is clean according to the posterior probability, so we choose $1/2$ as the threshold. Here, we do not need to know the proportion of clean samples which depends on the inaccessible noise rate. In this way, the clean labeled set $\mathcal{L}$ can be selected from the training data $\tilde{D}$, and an unlabeled set $\mathcal{U} = \{\boldsymbol{x}_i | (\boldsymbol{x}_i, \tilde{y}_i) \in \tilde{D} \backslash \mathcal{L}\}$ is constructed. This class-aware sample selection is described in Procedure 1. We further plot the loss distributions of different classes in order to demonstrate that the distributions follow the two-component GMM (see Appendix C for details).

With clean labeled samples $\mathcal{L}$ and unlabeled samples $\mathcal{U}$, the widely-used semi-supervised learning method MixMatch [Berthelot *et al.*, 2019] can be adopted to learn the model. MixMatch uses the learned model to generate pseudo labels for $\mathcal{U}$, applies MixUp to transform $\mathcal{L}$ and $\mathcal{U}$ into $\mathcal{L}'$ and $\mathcal{U}'$ with soft labels $\boldsymbol{q} \in [0,1]^C$ and then utilizes the cross-entropy loss and the mean squared error on $\mathcal{L}'$ and $\mathcal{U}'$ respectively:

$$\begin{aligned} L_{\mathcal{L}} &= \frac{1}{|\mathcal{L}'|} \sum_{(\boldsymbol{x},\boldsymbol{q}) \in \mathcal{L}'} L(\boldsymbol{x}, \boldsymbol{q}) \\ &= -\frac{1}{|\mathcal{L}'|} \sum_{(\boldsymbol{x},\boldsymbol{q}) \in \mathcal{L}'} \sum_{i=1}^{C} q_i \log \frac{\exp(f_i(\boldsymbol{x}))}{\sum_{j=1}^{C} \exp(f_j(\boldsymbol{x}))} \\ &= \frac{1}{|\mathcal{L}'|} \sum_{(\boldsymbol{x},\boldsymbol{q}) \in \mathcal{L}'} \sum_{i=1}^{C} q_i \log \left[1 + \sum_{j \neq i} \frac{\exp(f_j(\boldsymbol{x}))}{\exp(f_i(\boldsymbol{x}))} \right], \end{aligned} \tag{2}$$

$$L_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{(\boldsymbol{x},\boldsymbol{q}) \in \mathcal{U}'} \| \boldsymbol{q} - p_{model}(\boldsymbol{x}) \|_2^2. \tag{3}$$

In Eq. (2) of MixMatch, training with long-tailed data leads to model bias, and the model with bias prefers to predict head classes and has a poor performance on tail classes. This motivates us to develop a loss to correct the model bias. That is, we introduce α_{ij} into $L(\boldsymbol{x}, \boldsymbol{q})$ in Eq. (2):

$$\begin{aligned} L_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{q}) &= -\sum_{i=1}^{C} q_i \log \frac{\exp(f_i(\boldsymbol{x}))}{\sum_{j=1}^{C} \alpha_{ij} \exp(f_j(\boldsymbol{x}))} \\ &= \sum_{i=1}^{C} q_i \log \left[\alpha_{ii} + \sum_{j \neq i} \alpha_{ij} \frac{\exp(f_j(\boldsymbol{x}))}{\exp(f_i(\boldsymbol{x}))} \right]. \end{aligned} \tag{4}$$

Specifically, for a pair of classes (i, j), if the learned model prefers class j, then a weight $\alpha_{ij} > 1$ is assigned to $\frac{\exp(f_j(\boldsymbol{x}))}{\exp(f_i(\boldsymbol{x}))}$ for samples from class i to suppress class j, and a weight $\alpha_{ji} < 1$ is assigned to $\frac{\exp(f_i(\boldsymbol{x}))}{\exp(f_j(\boldsymbol{x}))}$ for samples from class j to relax class i. Many methods use label frequency as the estimation of model bias in long-tailed learning [Cui *et al.*, 2019; Menon *et al.*, 2021]. However, the model bias on different classes is not directly related to label frequency (see Appendix E), and the true label frequency is also unknown under label noise. To characterize the model bias, we calculate the matrix $M \in \mathbb{R}^{C \times C}$ before each training epoch, where the entry M_{ij} represents the probability that the learned model predicts a sample from class i to class j. For epoch t, $M^t_{ij} = \frac{1}{|\mathcal{L}_i|} \sum_{(\boldsymbol{x},\tilde{y}) \in \mathcal{L}_i} p^{j,t}_{model}(\boldsymbol{x})$, where $\mathcal{L}_i = \{(\boldsymbol{x}, \tilde{y}) | (\boldsymbol{x}, \tilde{y}) \in \mathcal{L} \wedge \tilde{y} = i\}$. Inspired by [Laine and Aila, 2017], we further temporally average the calculated matrices for a stable estimation of model bias with Exponentially Moving Average (EMA) and obtain the averaged model bias matrix $\bar{M}$, i.e., $\bar{M}^t = \sigma \bar{M}^{t-1} + (1-\sigma) M^t$, where σ controls the contribution of the estimated matrix from each epoch. If $\bar{M}_{ij}$ is

Algorithm 1 Learning with class-aware Sample Selection and Balanced Loss (SSBL).

Require: Deep neural network $f(\cdot;\theta)$, training data $(\mathcal{X},\mathcal{Y})$, total training epochs T, model bias estimation epochs E, unsupervised loss weight λ_u, regularization term weight λ_{reg}, hyper-paramter of EMA σ.

θ = WarmUp$(\mathcal{X},\mathcal{Y},\theta)$ ▷ training with Eq. (1)
Fill $\bar{M} \in \mathbb{R}^{C\times C}$ with zeros
while $e < E$ **do**
  Fill $M^e \in \mathbb{R}^{C\times C}$ with zeros
  $(\mathcal{L},\mathcal{U})$ = CASS$(\mathcal{X},\mathcal{Y},\theta)$ ▷ apply Procedure 1
  for iter = 1 to num_iters **do**
    Draw a mini-batch $\mathcal{L}_B = \{(\boldsymbol{x}_b, \tilde{y}_b)|b \in \{1, ..., B\}\}$ from $\mathcal{L}$
    Draw a mini-batch $\mathcal{U}_B = \{\boldsymbol{u}_b|b \in \{1, ..., B\}\}$ from $\mathcal{U}$
    for c = 1 to C **do**
      Draw $\mathcal{L}_{B,c} = \{(\boldsymbol{x},\tilde{y})|(\boldsymbol{x},\tilde{y}) \in \mathcal{L}_B \wedge \tilde{y} = c\}$ from $\mathcal{L}_B$
      $M^e_c = M^e_c + \sum_{(\boldsymbol{x},\tilde{y})\in\mathcal{L}_{B,c}} \text{Softmax}(f(\boldsymbol{x};\theta))$ ▷ update the c-th row of M^e
    end for
    $(\mathcal{L}',\mathcal{U}')$ = MixMatch$(\mathcal{L}_B,\mathcal{U}_B)$ ▷ apply MixMatch with augmented $\mathcal{L}_B$ and $\mathcal{U}_B$
    $L_{\mathcal{L}}, L_{\mathcal{U}}$ = SoftmaxCE$(\mathcal{L}';\theta)$, SoftmaxMSE$(\mathcal{U}';\theta)$
    $L = L_{\mathcal{L}} + \lambda_u L_{\mathcal{U}} + \lambda_{reg} L_{reg}$
    $\theta = SGD(L,\theta)$ ▷ update the parameters of $f(\cdot;\theta)$ with Stochastic Gradient Descent
  end for
  Normalize M^e
  $\bar{M} = \sigma\bar{M} + (1-\sigma)M^e$ ▷ update the model bias matrix $\bar{M}$
end while
$R_{ij} = \frac{\bar{M}_{ij}}{\bar{M}_{ji}}, R \in \mathbb{R}^{C\times C}$ ▷ compute R
while $e < T$ **do**
  $(\mathcal{L},\mathcal{U})$ = CASS$(\mathcal{X},\mathcal{Y},\theta)$ ▷ apply Procedure 1
  for iter = 1 to num_iters **do**
    Draw a mini-batch $\mathcal{L}_B = \{(\boldsymbol{x}_b, \tilde{y}_b)|b \in \{1, ..., B\}\}$ from $\mathcal{L}$
    Draw a mini-batch $\mathcal{U}_B = \{\boldsymbol{u}_b|b \in \{1, ..., B\}\}$ from $\mathcal{U}$
    $(\mathcal{L}',\mathcal{U}')$ = MixMatch$(\mathcal{L}_B,\mathcal{U}_B)$ ▷ apply MixMatch with augmented $\mathcal{L}_B$ and $\mathcal{U}_B$
    $L'_{\mathcal{L}}, L_{\mathcal{U}} = L_{\boldsymbol{\alpha}}(\mathcal{L}';\theta,R)$, SoftmaxMSE$(\mathcal{U}';\theta)$
    $L = L'_{\mathcal{L}} + \lambda_u L_{\mathcal{U}} + \lambda_{reg} L_{reg}$
    $\theta = SGD(L,\theta)$ ▷ update the parameters of $f(\cdot;\theta)$ with Stochastic Gradient Descent
  end for
end while

larger than $\bar{M}_{ji}$, then the model has a bias to class j. Let $R_{ij} = \bar{M}_{ij}/\bar{M}_{ji}$, $g(\cdot) = \gamma_{sup}\cdot\mathbb{I}(R_{ij} > 1) + \gamma_{rel}\cdot\mathbb{I}(R_{ij} \le 1)$ and $\alpha_{ij} = R_{ij}^{g(R_{ij})}$, where $g(\cdot)$ is a function of R_{ij} in which γ_{sup} and γ_{rel} are hyper-parameters for controlling the power of suppressing and relaxing respectively. Formally, we have

$$
\begin{aligned}
L'_{\mathcal{L}} &= \frac{1}{|\mathcal{L}'|}\sum_{(\boldsymbol{x},\boldsymbol{q})\in\mathcal{L}'} L_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{q}) \\
&= -\frac{1}{|\mathcal{L}'|}\sum_{(\boldsymbol{x},\boldsymbol{q})\in\mathcal{L}'}\sum_{i=1}^{C} q_i \log\frac{\exp(f_i(\boldsymbol{x}))}{\sum_{j=1}^{C} R_{ij}^{g(R_{ij})}\exp(f_j(\boldsymbol{x}))} \\
&= \frac{1}{|\mathcal{L}'|}\sum_{(\boldsymbol{x},\boldsymbol{q})\in\mathcal{L}'}\sum_{i=1}^{C} q_i \log\left[1 + \sum_{j\neq i} R_{ij}^{g(R_{ij})}\frac{\exp(f_j(\boldsymbol{x}))}{\exp(f_i(\boldsymbol{x}))}\right].
\end{aligned}
$$

Since the training data are in long-tailed distribution and there are only a few samples in tail classes, we augment clean labeled samples in $\mathcal{L}$ and re-exploit L_{reg} as a regularization on $\mathcal{L}'$ with its hard labels. Thus, we use the balanced loss $L = L'_{\mathcal{L}} + \lambda_u L_{\mathcal{U}} + \lambda_{reg} L_{reg}$ in the training process, where λ_u controls the unsupervised loss and λ_{reg} controls the regularization term. The whole training process of our SSBL is described in Algorithm 1.

4 Experiment

4.1 Setup

Datasets. We validate our method on seven benchmark datasets, namely CIFAR-10, CIFAR-100 [Krizhevsky *et al.*, 2009], mini-ImageNet-Red [Jiang *et al.*, 2020], Clothing1M [Xiao *et al.*, 2015], Food-101N [Lee *et al.*, 2018], Animal-10N [Song *et al.*, 2019] and WebVision [Li *et al.*, 2017]. On CIFAR, we consider two kinds of label noise, i.e., symmetric noise and asymmetric noise. For symmetric noise, we first construct long-tailed versions of CIFAR with different imbalance ratios ρ following [Cui *et al.*, 2019]. In specific, we reduce the sample size per class according to an exponential function $n_i = O_i \frac{1}{\rho}^{\frac{i-1}{C-1}}$, where O_i is the original sample size of class i, $\rho = \frac{n_1}{n_C}$ and $i \in \{1, \ldots, C\}$. Then, we generate symmetric noise in long-tailed CIFAR according to the fol-

Dataset		CIFAR-10						CIFAR-100					
Noise Rate		0.2			0.5			0.2			0.5		
Imbalance Ratio		10	50	100	10	50	100	10	50	100	10	50	100
ERM	Best	76.90	65.35	60.82	65.75	48.76	39.70	45.83	35.05	29.96	28.96	19.88	16.80
	Last	73.02	61.35	54.48	45.85	33.05	28.79	45.64	34.93	29.88	24.33	17.77	14.47
Co-teaching [Han *et al.*, 2018]	Best	78.50	45.86	39.59	34.60	23.58	17.45	43.81	30.58	28.08	14.58	11.62	9.69
	Last	77.51	44.91	38.07	31.71	22.57	14.88	43.69	30.22	28.08	14.49	10.54	9.35
HAR-DRW [Cao *et al.*, 2021]	Best	73.83	61.30	60.92	59.10	43.69	36.31	37.31	29.29	26.00	23.22	17.04	12.45
	Last	71.16	60.27	56.94	40.10	34.19	35.91	36.92	29.02	25.69	19.01	14.29	11.11
MW-Net [Shu *et al.*, 2019]	Best	83.78	67.97	50.96	71.81	38.45	27.15	51.09	37.88	33.41	31.83	18.68	14.34
	Last	73.25	62.63	48.01	50.02	36.14	20.90	50.24	37.75	33.24	27.72	13.61	13.20
H2E [Yi *et al.*, 2022]	Best	79.40	59.49	52.80	62.03	36.29	31.79	48.66	34.86	29.26	33.38	22.92	19.15
	Last	78.76	52.10	49.95	43.66	33.57	31.10	48.48	34.65	28.95	33.17	19.73	14.69
SSBL	Best	**91.60**	**86.30**	**79.84**	**88.51**	**77.72**	**72.36**	**62.78**	**51.25**	**45.68**	**55.95**	**42.19**	**36.87**
	Last	**91.49**	**85.83**	**78.39**	**88.17**	**75.76**	**68.44**	**62.45**	**50.70**	**43.89**	**55.65**	**40.82**	**36.49**
ELR+ [Liu *et al.*, 2020]	Best	88.34	76.31	68.39	64.61	16.40	15.65	53.11	39.17	34.35	35.97	25.27	20.49
	Last	87.68	74.62	66.60	60.30	10.03	10.03	48.95	35.20	31.36	22.73	18.78	16.36
DivideMix [Li *et al.*, 2020]	Best	91.03	83.07	70.08	85.97	69.73	52.06	63.08	49.76	43.71	55.98	41.79	35.03
	Last	90.71	82.16	69.82	85.79	68.19	51.72	62.54	49.45	42.84	55.56	41.25	34.51
DivideMix-LA	Best	91.68	85.16	80.06	85.84	56.28	49.48	64.76	53.31	47.52	57.11	40.72	35.18
	Last	91.65	84.15	78.40	85.47	54.67	46.66	64.33	52.83	46.21	56.57	40.21	34.73
DivideMix-DRW	Best	90.44	84.64	80.50	85.40	73.90	47.31	63.65	50.28	45.23	56.77	42.24	36.17
	Last	89.33	82.99	80.40	84.88	72.92	45.90	63.24	50.04	44.95	56.58	41.61	35.74
TABASCO [Lu *et al.*, 2023]	Best	83.67	68.43	57.82	88.01	74.15	66.12	-	-	-	-	-	-
	Last	83.33	67.29	56.93	87.32	73.38	64.22	-	-	-	-	-	-
SSBL_2	Best	**92.47**	**87.14**	**81.98**	**89.41**	**78.65**	**72.96**	**65.09**	**53.52**	**47.87**	**57.95**	**44.37**	**39.49**
	Last	**92.25**	**86.87**	**81.29**	**89.09**	**76.44**	**69.11**	**64.60**	**53.35**	**47.64**	**57.80**	**43.47**	**38.64**

Table 1: Comparison with baselines in test accuracy (%) on long-tailed versions of CIFAR-10 and CIFAR-100 with symmetric noise. SSBL represents our performance of single model, while SSBL_2 represents our ensemble performance of two models.

Imbalance Ratio	10	50	100
ERM	77.45	71.01	69.00
Co-teaching [Han *et al.*, 2018]	77.06	69.49	67.34
HAR-DRW [Cao *et al.*, 2021]	82.69	72.17	69.32
MW-Net [Shu *et al.*, 2019]	81.42	72.82	68.92
H2E [Yi *et al.*, 2022]	76.80	68.89	68.08
SSBL	**87.88**	**72.95**	**70.16**
ELR+ [Liu *et al.*, 2020]	74.09	68.33	67.04
DivideMix [Li *et al.*, 2020]	88.07	69.19	67.46
DivideMix-LA	83.88	72.11	66.41
DivideMix-DRW	81.56	71.02	69.80
TABASCO [Lu *et al.*, 2023]	84.86	67.26	65.50
SSBL_2	**88.83**	**73.96**	**71.43**

Table 2: Comparison with baselines in test accuracy (%) on step-imbalanced versions of CIFAR-10 with asymmetric noise. SSBL represents our performance of single model, while SSBL_2 represents our ensemble performance of two models.

Noise Rate	0.2			0.5		
Imbalance Ratio	20	50	100	20	50	100
Co-teaching [Han *et al.*, 2018]	22.72	20.09	17.05	17.97	14.74	13.86
HAR-DRW [Cao *et al.*, 2021]	29.78	25.94	23.30	24.10	21.34	19.14
MW-Net [Shu *et al.*, 2019]	33.56	27.74	24.46	26.72	23.10	20.82
H2E [Yi *et al.*, 2022]	26.26	24.56	20.82	25.76	20.50	17.80
SSBL	36.02	31.38	28.20	30.04	26.36	23.98
SSBL+RandAug	**37.52**	**32.54**	**29.54**	**31.14**	**27.94**	**25.34**
ELR+ [Liu *et al.*, 2020]	31.86	26.33	24.39	26.23	23.30	20.61
DivideMix [Li *et al.*, 2020]	35.96	29.02	27.36	30.10	26.42	24.20
DivideMix-LA	37.36	32.12	29.66	32.34	28.12	24.78
DivideMix-DRW	36.56	30.14	28.48	30.64	26.62	24.34
SSBL_2	**37.92**	**32.90**	**30.74**	**32.84**	**29.06**	**26.26**

Table 3: Comparison with baselines in test accuracy (%) on long-tailed versions of mini-ImageNet-Red with real-world noise. SSBL represents our performance of single model, while SSBL_2 represents our ensemble performance of two models.

lowing noise transition matrix N:

$$N_{ij}(\boldsymbol{x}, y) = \mathcal{P}(\tilde{y} = j | y = i, \boldsymbol{x}) = \begin{cases} 1 - r & i = j \\ r \dfrac{n_j}{\sum_{k \neq i} n_k} & i \neq j, \end{cases}$$

where y denotes the ground truth of $\boldsymbol{x}$, $\tilde{y}$ denotes the corrupted label of $\boldsymbol{x}$ and $r \in [0, 1]$ denotes the noise rate. We set imbalance ratio to be $\rho \in \{10, 50, 100\}$ and set noise rate to be $r \in \{0.2, 0.5\}$. In the experiments of learning from long-tailed noisy data, the noise rate is usually lower than 0.5, as that in [Zhang and Pfister, 2021] and [Cao *et al.*, 2021]. For asymmetric noise, we construct step-imbalanced versions of CIFAR-10 with asymmetric noise following [Cao *et al.*, 2021]. In specific, we corrupt semantically-similar classes by exchanging 40% of the labels between 'cat' and 'dog', and by exchanging 40% of the labels between 'truck' and 'automobile'. Then, we remove samples from the corrupted classes. Here, the imbalance ratio ρ is the sample size ratio between the frequent (and clean) classes and the rare (and noisy) classes. Mini-ImageNet-Red, Clothing1M, Food-101N, Animal-10N and WebVision are datasets with real-world noise, so we directly create their long-tailed versions following [Cui *et al.*, 2019]. For mini-ImageNet-Red, Clothing1M, Food-101N and Animal-10N, we set imbalance ratio to be $\rho \in \{20, 50, 100\}$. For Clothing1M, we randomly select a small balanced subset of Clothing1M following [Yi and Wu, 2019] and construct its long-tailed versions. For WebVision, we train and test on the subset mini WebVision which contains the first 50 classes of WebVision following [Li *et al.*, 2020]. The original imbalance ratio of mini WebVision is approximately 7, we further create long-tailed versions of mini WebVision with $\rho \in \{50, 100\}$ following [Cui *et al.*, 2019].

Dataset	Clothing1M			Food-101N			Animal-10N		
Imbalance Ratio	20	50	100	20	50	100	20	50	100
Co-teaching [Han *et al.*, 2018]	51.97	43.23	36.48	45.67	38.90	35.13	55.06	34.51	32.63
HAR-DRW [Cao *et al.*, 2021]	62.56	54.08	51.09	54.12	48.25	41.81	71.96	64.22	57.66
MW-Net [Shu *et al.*, 2019]	60.04	57.15	51.98	56.73	47.44	41.89	74.72	64.26	53.86
H2E [Yi *et al.*, 2022]	70.41	67.81	61.58	70.35	63.69	58.66	77.04	74.94	66.58
SSBL	**71.74**	69.37	66.80	72.05	66.78	61.62	80.20	75.06	69.10
SSBL+RandAug	71.66	**71.04**	**67.65**	**74.06**	**68.56**	**64.64**	**81.58**	**78.82**	**71.84**
ELR+ [Liu *et al.*, 2020]	67.46	61.24	54.83	58.56	48.98	44.05	63.37	49.28	47.53
DivideMix [Li *et al.*, 2020]	71.38	69.27	66.63	64.02	54.65	52.57	77.64	69.74	64.72
DivideMix-LA	71.20	70.09	67.20	71.48	63.73	59.01	78.62	74.90	66.08
DivideMix-DRW	71.56	69.83	66.85	64.21	53.98	53.65	79.32	74.14	66.14
TABASCO [Lu *et al.*, 2023]	-	-	-	-	-	-	67.38	54.46	48.94
SSBL_2	**72.41**	**70.76**	**67.91**	**74.39**	**69.06**	**63.67**	**81.20**	**75.84**	**69.36**

Table 4: Comparison with baselines in test accuracy (%) on long-tailed versions of Clothing1M, Food-101N and Animal-10N with real-world noise. SSBL represents our performance of single model, while SSBL_2 represents our ensemble performance of two models. Results of H2E on Food-101N and Animal-10N are borrowed from its original paper.

Imbalance Ratio	Original		50		100	
Dataset	WebVision	ImageNet	WebVision	ImageNet	WebVision	ImageNet
Co-teaching [Han *et al.*, 2018]	63.60 (85.20)	61.50 (84.70)	43.11 (56.13)	40.95 (55.69)	40.42 (52.72)	37.78 (51.16)
HAR-DRW [Cao *et al.*, 2021]	72.96 (90.12)	67.12 (89.36)	58.16 (82.60)	54.20 (80.80)	50.24 (78.16)	47.44 (75.96)
MW-Net [Shu *et al.*, 2019]	71.76 (90.40)	67.36 (89.08)	60.12 (84.08)	55.20 (82.28)	49.96 (80.04)	46.84 (77.80)
H2E [Yi *et al.*, 2022]	72.32 (90.84)	69.24 (**91.28**)	57.88 (84.76)	56.64 (85.04)	52.04 (81.16)	49.72 (82.28)
SSBL	**76.72** (**91.00**)	**75.00** (90.92)	**68.32** (**87.24**)	**64.92** (**86.32**)	**63.80** (**86.12**)	**61.00** (**86.28**)
H2E (200 epochs) [Yi *et al.*, 2022]	78.12 (93.40)	74.00 (92.28)	69.20 (**90.44**)	65.72 (89.04)	64.64 (88.56)	61.68 (86.88)
SSBL (200 epochs)	80.48 (93.16)	76.56 (92.04)	72.08 (89.00)	69.52 (89.04)	66.60 (88.56)	63.32 (87.88)
SSBL+RandAug (200 epochs)	**81.12** (**93.68**)	**76.88** (**93.72**)	**73.08** (90.16)	**70.16** (**89.44**)	**68.24** (**88.92**)	**65.08** (**89.68**)
ELR+ [Liu *et al.*, 2020]	78.21 (91.50)	73.43 (90.50)	55.24 (82.65)	52.98 (80.88)	51.40 (72.97)	50.47 (72.52)
DivideMix [Li *et al.*, 2020]	77.68 (**92.72**)	75.44 (92.44)	61.40 (82.12)	62.08 (82.64)	54.80 (73.76)	54.60 (74.96)
DivideMix-LA	77.44 (91.96)	75.96 (91.24)	69.80 (**88.24**)	68.00 (**89.32**)	64.64 (86.44)	63.04 (86.92)
DivideMix-DRW	78.28 (92.04)	75.44 (**92.56**)	65.20 (81.44)	65.40 (82.36)	57.44 (72.76)	57.76 (73.68)
SSBL_2	**78.56** (91.92)	**76.64** (91.92)	**70.40** (88.20)	**69.12** (88.64)	**66.08** (**87.20**)	**63.20** (**87.40**)

Table 5: Comparison with baselines in Top-1 (Top-5) test accuracy (%) on original and long-tailed versions of mini WebVision with real-world noise. SSBL represents our performance of single model, while SSBL_2 represents our ensemble performance of two models.

Baselines. We compare our method with state-of-the-art label noise learning methods and long-tailed label noise learning methods, including Empirical Risk Minimization (ERM) which trains the model with the cross-entropy loss, Co-teaching [Han *et al.*, 2018] which trains two networks simultaneously and updates one network on the data selected by the other with small-loss criterion, ELR+ [Liu *et al.*, 2020] which capitalizes on early learning via regularization, DivideMix [Li *et al.*, 2020] which conducts sample selection via a two-component gaussian mixture model and then applies the semi-supervised learning technique MixMatch with label correction, HAR-DRW [Cao *et al.*, 2021] which regularizes different regions of input space differently to handle heteroskedasticity and imbalance issues simultaneously, MW-Net [Shu *et al.*, 2019] which learns a weighting function in a meta-learning manner to assign each sample a weight, H2E [Yi *et al.*, 2022] which first trains a noise identifier and then learns a robust classifier, DivideMix-LA which combines DivideMix and Logit Adjustment (LA) [Menon *et al.*, 2021], DivideMix-DRW which combines DivideMix and Deferred Re-Weighting (DRW), and TABASCO [Lu *et al.*, 2023] which proposes a two-stage bi-dimensional sample selection and trains the model in a semi-supervised manner. We set the hyper-parameters of the baselines by following their original papers or source codes.

Implementation Details. On CIFAR-10, CIFAR-100, mini-ImageNet-Red and Animal-10N, we use an 18-layer PreAct ResNet and train for 200 epochs. On Clothing1M and Food-101N, we use a ResNet-50 and train for 200 epochs from scratch. On WebVision, we use an Inception-ResNet v2 and train for 100 epochs following [Li *et al.*, 2020]. Since the baseline H2E trains for 200 epochs in its original paper, we also conduct experiments on WebVision with H2E for 200 epochs. On all datasets, γ_{sup} is set as 3 and γ_{rel} is set as 1 in $L'_{\mathcal{L}}$ (refer to Appendix A for more details). The baselines ELR+, DivideMix, DivideMix-LA, DivideMix-DRW and TABASCO are based on two models. We also run our method SSBL twice with random initialization and use the ensemble of two runs for a fair comparison with them. We refer to our method with two models as SSBL_2 and also report its performance in Tables 1∼5. For a fair comparison with H2E, which uses strong augmentation technique RandAugment [Cubuk *et al.*, 2020] in training, we also report the performance of SSBL with RandAugment (SSBL+RandAug).

4.2 Evaluation on Benchmarks

Table 1 summarizes the results on long-tailed versions of CIFAR-10 and CIFAR-100 with symmetric noise, and Table 2 summarizes the results on step-imbalanced CIFAR-10 with asymmetric noise by following [Cao *et al.*, 2021]. From

Dataset	CIFAR-100						
Noise Rate	mean	0.2			0.5		
Imbalance Ratio		10	50	100	10	50	100
SSBL	**48.33**	**62.45**	**50.70**	**43.89**	**55.65**	40.82	**36.49**
w/o rebalancing	44.68	60.98	46.55	40.19	53.05	36.41	30.91
rebalancing with label frequency	46.59	61.99	48.86	43.14	54.32	38.30	32.95
w/o L_{reg} in warm up	47.85	61.82	49.87	43.32	54.93	**41.02**	36.16
w/o L_{reg} in the whole process	36.66	61.26	14.95	12.08	55.00	40.65	36.00
w/o class-aware sample selection	38.74	59.92	26.56	20.29	53.64	38.73	33.32

Table 6: Ablation study. All experiments are conducted with single model.

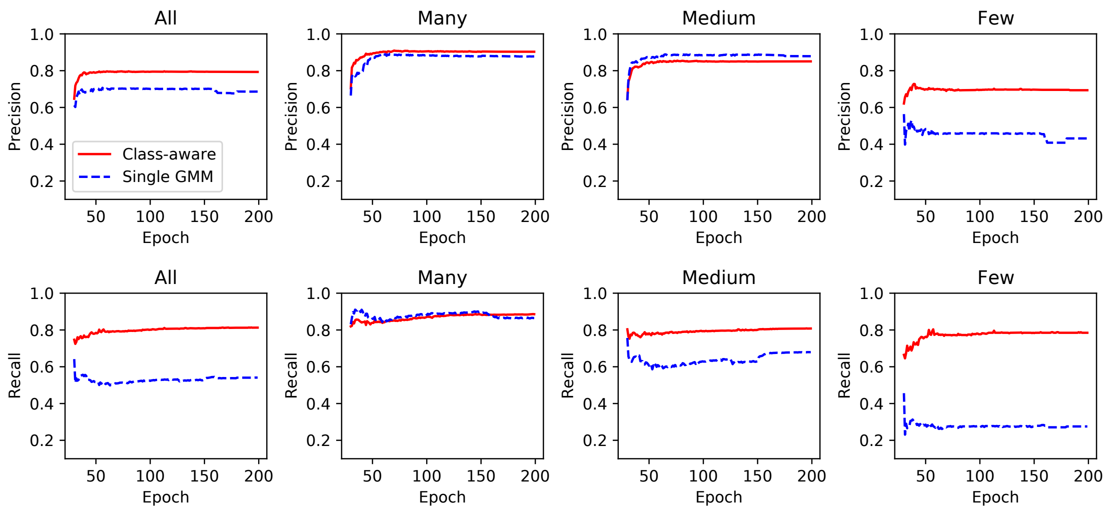

Figure 1: Comparison of precision and recall between class-aware sample selection and sample selection with single GMM. We visualize the precision and recall of samples selected as clean for all and three splits of classes: Many (more than 100 images), Medium (20∼100 images) and Few (less than 20 images). The experiment is conducted on long-tailed CIFAR-100 with symmetric noise. The imbalance ratio $\rho = 100$ and the noise rate $r = 0.5$, which is an extremely hard setting.

these tables, it can be found that SSBL with single model outperforms ERM, Co-teaching, HAR-DRW, MW-Net and H2E under all settings, and SSBL_2 with two models outperforms ELR+, DivideMix, DivideMix-LA, DivideMix-DRW and TABASCO under all settings.

Tables 3 and 4 summarize the results on long-tailed mini-ImageNet-Red, Clothing1M, Food-101N and Animal-10N with real-world noise. From these tables, it can be found that SSBL with single model outperforms Co-teaching, HAR-DRW, MW-Net and H2E under all settings, and its performance can be further boosted with RandAugment. SSBL_2 with two models outperforms ELR+, DivideMix, DivideMix-LA, DivideMix-DRW and TABASCO under all settings.

Table 5 summarizes the results on original and long-tailed versions of mini WebVision with real-world noise, where we report Top-1 and Top-5 test accuracy on both WebVision and ImageNet following [Li *et al.*, 2020]. From Table 5, it can be found that SSBL with single model outperforms Co-teaching, HAR-DRW, MW-Net and H2E in Top-1 test accuracy under all settings, and has the comparable or better performance in Top-5 test accuracy. The performance of SSBL with single model can be further boosted with RandAugment. SSBL_2 with two models also outperforms ELR+, DivideMix, DivideMix-LA and DivideMix-DRW in Top-1 test accuracy under all settings, and has the comparable or better performance in Top-5 test accuracy.

4.3 Ablation Study

In order to study the effects of removing components of our method SSBL with single model, we conduct the ablation study on long-tailed versions of CIFAR-100 with symmetric noise and summarize the results in Table 6.

- To study the effect of rebalancing the model bias after having estimated the model bias matrix $\bar{M}$, we replace our loss $L'_{\mathcal{L}}$ with the cross-entropy loss $L_{\mathcal{L}}$. The decrease in test accuracy suggests that $L'_{\mathcal{L}}$ is effective for learning from long-tailed noisy data.
- To study the effect of rebalancing the model bias with the estimated model bias matrix $\bar{M}$ instead of label frequency, we test the performance of rebalancing with label frequency. The degradation in performance verifies the advantage of tackling long-tailed noisy data with the estimated model bias.
- To study the effect of the regularization term L_{reg}, we test our method without L_{reg} in warm up and without

Noise Rate		0.2		0.5	
Imbalance Ratio		50	100	50	100
$\gamma_{sup}=3, \gamma_{rel}=0$	Best	84.77	79.21	75.58	70.51
	Last	82.44	77.00	74.86	66.76
$\gamma_{sup}=3, \gamma_{rel}=0.5$	Best	85.71	**80.73**	76.48	71.93
	Last	85.01	**78.77**	75.19	68.12
$\gamma_{sup}=3, \gamma_{rel}=1$	Best	**86.30**	79.84	77.72	72.36
	Last	**85.83**	78.39	**75.76**	68.44
$\gamma_{sup}=3, \gamma_{rel}=2$	Best	79.72	73.90	**79.08**	72.79
	Last	50.70	41.86	74.54	59.28
$\gamma_{sup}=3, \gamma_{rel}=3$	Best	79.72	73.90	78.73	**73.22**
	Last	46.29	42.03	64.68	60.22
$\gamma_{sup}=1, \gamma_{rel}=1$	Best	85.67	80.39	75.50	69.84
	Last	85.16	78.76	73.64	67.99
$\gamma_{sup}=2, \gamma_{rel}=1$	Best	85.43	80.24	76.95	71.56
	Last	85.19	77.82	73.39	**68.78**
$\gamma_{sup}=4, \gamma_{rel}=1$	Best	85.55	79.60	78.21	72.88
	Last	85.55	73.08	66.27	58.26

Table 7: Ablation study of the hyper-parameters γ_{sup} and γ_{rel} in test accuracy (%) on long-tailed CIFAR-10 with symmetric noise.

L_{reg} in the whole process. The results imply that L_{reg} in warm up usually brings better performance and L_{reg} in the whole process ensures the robustness under extremely long-tailed distribution.

- To study the effect of class-aware sample selection, we replace our class-aware GMM with a single GMM for all classes. The degeneration in performance justifies the superiority of class-aware sample selection over sample selection with single GMM. To further demonstrate that class-aware sample selection is robust to long-tailed noisy data, we visualize the precision and recall of samples selected as clean in Figure 1. In comparison with sample selection with single GMM, our method with class-aware sample selection achieves comparable or better performance in both precision and recall on all splits of classes.

The hyper-parameters γ_{sup} and γ_{rel} control the power of suppressing and relaxing, respectively. We conduct experiments on both of them on long-tailed versions of CIFAR-10 with symmetric noise and summarize the results in Table 7. $\gamma_{sup}=1$ (too little suppression for head classes) or $\gamma_{sup}=4$ (too much suppression for head classes) will degrade the performance; $\gamma_{rel}=0$ (without any relaxation for tail classes) or $\gamma_{rel}>1$ (too much relaxation for tail classes) will harm the performance. In general, $\gamma_{sup}=3$ and $\gamma_{rel}=1$ is an appropriate choice (see Appendix F for more discussion about other hyper-parameters).

In our method, we separate training data into a clean labeled set $\mathcal{L}$ and an unlabeled set $\mathcal{U}$ with class-aware sample selection, and then train the model in a semi-supervised manner with our balanced loss. There are other Class-Imbalanced Semi-Supervised Learning (CISSL) methods, e.g., DARP in [Kim *et al.*, 2020] and ABC in [Lee *et al.*, 2021]. We conduct the ablation study with the existing CISSL methods based on our $\mathcal{L}$ and $\mathcal{U}$ (see Appendix D). The results show that our method performs better than existing CISSL methods.

5 Conclusion

We propose the robust method for learning from long-tailed noisy data with sample selection and balanced loss. In specific, we separate the training data into the clean labeled set and the unlabeled set with class-aware sample selection, and then train the model in a semi-supervised manner with a novel balanced loss. Extensive experiments across benchmarks demonstrate that our method is superior to existing state-of-the-art methods.

Acknowledgments

This work is supported by the National Science Foundation of China (62276125, 61921006), the Fundamental Research Funds for the Central Universities (022114380013), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Contribution Statement

Lefan Zhang and Zhang-Hao Tian have equal contribution. Wei Wang is the corresponding author.

References

[Arazo *et al.*, 2019] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Unsupervised label noise modeling and loss correction. In *Proceedings of the 36th International Conference on Machine Learning*, pages 312–321, 2019.

[Arpit *et al.*, 2017] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242, 2017.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32*, pages 5050–5060, 2019.

[Cao *et al.*, 2019] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems 32*, pages 1567–1578, 2019.

[Cao *et al.*, 2021] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Aréchiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[Cubuk *et al.*, 2020] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[Cui *et al.*, 2019] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the 2019*

IEEE Conference on Computer Vision and Pattern Recognition, pages 9268–9277, 2019.

[Fang *et al.*, 2023] Chaowei Fang, Lechao Cheng, Yining Mao, Dingwen Zhang, Yixiang Fang, Guanbin Li, Huiyan Qi, and Licheng Jiao. Separating noisy samples from tail classes for long-tailed image classification with label noise. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Ghosh *et al.*, 2017] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1919–1925, 2017.

[Gui *et al.*, 2021] Xian-Jin Gui, Wei Wang, and Zhang-Hao Tian. Towards understanding deep learning from noisy labels with small-loss criterion. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 2469–2475, 2021.

[Han *et al.*, 2018] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31*, pages 8536–8546, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Jeatrakul *et al.*, 2010] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Classification of imbalanced data by combining the complementary neural network and smote algorithm. In *Proceedings of the 17th International Conference on Neural Information Processing*, pages 152–159, 2010.

[Jiang *et al.*, 2020] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4804–4815, 2020.

[Jiang *et al.*, 2021] Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. Delving into sample loss curve to embrace noisy and imbalanced data. *arXiv*, 2021.

[Kang *et al.*, 2019] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv*, 2019.

[Kang *et al.*, 2021] Bingyi Kang, Yu Li, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[Karthik *et al.*, 2021] Shyamgopal Karthik, Jérome Revaud, and Boris Chidlovskii. Learning from long-tailed data with noisy labels. *arXiv*, 2021.

[Kim *et al.*, 2020] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems 33*, pages 14567–14579, 2020.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[Laine and Aila, 2017] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[Lee *et al.*, 2018] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[Lee *et al.*, 2021] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *In Advances in Neural Information Processing Systems 34*, pages 7082–7094, 2021.

[Li *et al.*, 2017] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc V. Gool. Webvision database: Visual learning and understanding from web data. *arXiv*, 2017.

[Li *et al.*, 2020] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision*, pages 740–755, 2014.

[Liu *et al.*, 2019] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Liu *et al.*, 2020] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems 33*, pages 20331–20342, 2020.

[Lu *et al.*, 2023] Yang Lu, Yiliang Zhang, Bo Han, Yiu-ming Cheung, and Hanzi Wang. Label-noise learning with intrinsically long-tailed data. In *Proceedings of the 19th International Conference on Computer Vision*, pages 1369–1378, 2023.

[Menon *et al.*, 2021] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[Patrini *et al.*, 2017] Giorgio Patrini, Alessandro Rozza, Aditya K. Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2233–2241, 2017.

[Qi and Chelmis, 2023] Wenting Qi and Charalampos Chelmis. Noisy label detection and counterfactual correction. *IEEE Transactions on Artificial Intelligence*, 2023.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 211–252, 2015.

[Shu *et al.*, 2019] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems 32*, pages 1919–1930, 2019.

[Song *et al.*, 2019] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5907–5915, 2019.

[Song *et al.*, 2022] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Tanaka *et al.*, 2018] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[Xiao *et al.*, 2015] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[Yi and Wu, 2019] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

[Yi *et al.*, 2022] Xuanyu Yi, Kaihua Tang, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. Identifying hard noise in long-tailed sample distribution. In *Proceedings of the 17th European Conference on Computer Vision*, pages 739–756, 2022.

[Zhang and Pfister, 2021] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the 18th International Conference on Computer Vision*, pages 725–734, 2021.

[Zhang *et al.*, 2021] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[Zhao *et al.*, 2023] Qihao Zhao, Fan Zhang, Wei Hu, Songhe Feng, and Jun Liu. Ohd: An online category-aware framework for learning with noisy labels under long-tailed distribution. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.