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 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., 2019].

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

### 3 Methodology

Let \( \mathcal{X} \) denote the instance space, for each \( x \in \mathcal{X} \), there exists a true label \( y \in \mathcal{Y} = \{1, \ldots, C\} \), where \( C \) is the number of classes. Let \( \mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \) denote the training data, where \( x_i \in \mathcal{X}, y_i \in \mathcal{Y} \) is the observed label of \( x_i \) 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 \( \mathcal{D}_c = \{(x_i, y_i) \mid (x_i, y_i) \in \mathcal{D} \wedge y_i = c\} \) and \( n_c = |\mathcal{D}_c| \). 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 \( \mathcal{D} \), where \( \theta \) represents the model parameter. Given \( x \in \mathcal{X} \), the network outputs \( f(x; \theta) = [f_1(x; \theta), \ldots, f_C(x; \theta)]^T \) and \( f_i(x; \theta) \) represents the output logit of class \( i \). We omit the model parameter \( \theta \) and denote \( f(x; \theta) \) as \( f(x) \) for brevity. Let \( p_{\text{model}}(x) = \frac{\exp(f_i(x))}{\sum_{j=1}^C \exp(f_j(x))} \) be the classifier induced by \( f \) is \( \phi_f(x) = \arg\max_{y \in \{1, \ldots, C\}} p_{\text{model}}(x, y) \). For \( (x, y) \), the loss is calculated as \( \ell(f(x), y) \) with a loss function \( \ell(\cdot, \cdot) \). The empirical loss of \( f \) on \( \mathcal{D} \) is \( \frac{1}{N} \sum_{i=1}^N \ell(f(x_i), y_i) \). We focus on learning the optimal model parameter \( \theta^* \) which minimizes the expected loss, i.e., \( \theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)}[\ell(f(x; \theta), y)] \).

The training data in real-world applications usually con-
tain noisy labels, while deep neural networks are gener-
ally learned by minimizing the empirical risk on the train-
ing data. [Zhang et al., 2021] demonstrated that noisy la-
bes 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 popu-
lar 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:

$$L_{reg} = \sum_{i=1}^{C} \frac{n_{C}}{n_{i}} \log \left( \frac{\pi_i}{1 - \sum_{j=1}^{C} \sum_{(x, y) \in D_i} 1/n_j p_{model}(x)} \right) ,$$

where $\pi_i = \frac{1}{n_i}$. $L_{reg}$ 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 $1/n_j$ to samples
from class $j$. We pay more attention to tail classes by assign-
ing a regularization strength $\frac{\lambda_w}{n_i}$ to class $i$, which guarantees a
decent performance on tail classes. During warming up,
hyper-parameter $\lambda_w$ controls the strength of $L_{reg}$:

$$L = L_{CE} + \lambda_w L_{reg},$$

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

After warming up, small-loss criterion can be used to se-
lect 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 pro-
posed 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 ($g_i$ 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 $g_i$ represents the clean distribution and $g_0$ represents
the noisy distribution. For class $i$, let $L(D_i)$ denote the sam-
ple’s cross-entropy loss, and it reflects how well the model
fits the training samples:

$$L(D_i) = \left\{ -\log \left( \frac{\exp(f_i(x_j))}{\sum_{k=1}^{C} \exp(f_k(x_j))} \right) \left| (x_j, y_j) \in D_i \right. \right\} .$$

A two-component GMM can be fitted with respect to
$L(D_i)$, and the clean probability of a sample $(x_j, y_j)$ is
given by its posterior probability of belonging to $g_i$.

$$P(g_i | L(D_i), (x_j, y_j)).$$

The sample with clean probability $P(g_i | L(D_i), (x_j, y_j)) > 0.5$ 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 0.5 as the threshold. Here, we do not need to know
the proportion of clean samples which depends on the inac-
cessible noise rate. In this way, the clean labeled set $\mathcal{L}$
can be selected from the training data $D$, and an unlabeled set
$\mathcal{U} = \{(x, y) \notin \tilde{D} \setminus \mathcal{L} \}$ is constructed. This class-aware sample selection is described in Procedure 1. We further plot
the loss distributions of different classes in order to demon-
strate 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 $q \in [0, 1]^C$ and then utilizes the cross-entropy
loss and the mean squared error on $\mathcal{L}'$ and $\mathcal{U}'$ respectively:

$$L_{\mathcal{L}} = \frac{1}{|\mathcal{L}'|} \sum_{(x, q) \in \mathcal{L}'} L(x, q) ,$$

$$= -\frac{1}{|\mathcal{L}'|} \sum_{(x, q) \in \mathcal{L}'} \sum_{i=1}^{C} q_i \log \left( \frac{\exp(f_i(x))}{\sum_{j=1}^{C} \exp(f_j(x))} \right) ,$$

$$= \frac{1}{|\mathcal{L}'|} \sum_{(x, q) \in \mathcal{L}'} \sum_{i=1}^{C} q_i \log \left[ 1 + \sum_{j \neq i} \frac{\exp(f_j(x))}{\exp(f_i(x))} \right] ,$$

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

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 mo-
tivates us to develop a loss to correct the model bias. That is,
we introduce $\alpha_{ij}$ into $L(x, q)$ in Eq. (2):

$$L_{\alpha}(x, q) = -\sum_{i=1}^{C} q_i \log \left( \frac{\exp(f_i(x))}{\sum_{j=1}^{C} \alpha_{ij} \exp(f_j(x))} \right) ,$$

$$= \sum_{i=1}^{C} q_i \log \left( \alpha_{ii} + \sum_{j \neq i} \alpha_{ij} \frac{\exp(f_j(x))}{\exp(f_i(x))} \right) .$$

Specifically, for a pair of classes $(i, j)$, if the learned model
prefers class $j$, then a weight $\alpha_{ij} > 1$ is assigned to $\exp(f_j(x)) / \exp(f_i(x))$ for samples from class $i$ to suppress class $j$, and a weight $\alpha_{ji} < 1$ is assigned to $\exp(f_i(x)) / \exp(f_j(x))$ for samples from class $j$ to relax class $i$. Many methods use label frequency as the esti-
mation of model bias in long-tailed learning [Cui et al., 2019;
Menon et al., 2021]. However, the model bias on differ-
cent classes is not directly related to label frequency (see Ap-
pendix 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 $t$, $M_{ij}^{t+1} = M_{ij}^{t} + \frac{1}{|\mathcal{L}|} \sum_{(x, y) \in \mathcal{L}} \frac{p_{model}(x)}{L(x, y)}$, where $\mathcal{L} = \{(x, y) | (x, y) \in L \wedge y = i\}$. Inspired by [Laine and Aila, 2017], we fur-
ther 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., $\bar{M} = \sigma \bar{M}^{t-1} + (1 - \sigma) M^t$, where $\sigma$ controls the con-
tribution of the estimated matrix from each epoch. If $M_{ij}$ is

5473
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 \(\lambda_u\), regularization term weight \(\lambda_{reg}\), hyper-paramter of EMA \(\sigma\).

1: \(\theta = \text{WarmUp}(\mathcal{X}, \mathcal{Y}, \theta)\) \(\triangleright\) training with Eq. (1)
2: Fill \(M \in \mathbb{R}^{C \times C}\) with zeros
3: while \(e < E\) do
4: Fill \(M^c \in \mathbb{R}^{C \times C}\) with zeros
5: \((\mathcal{L}, \mathcal{U}) = \text{CASS}(\mathcal{X}, \mathcal{Y}, \theta)\) \(\triangleright\) apply Procedure 1
6: for \(i = 1\) to num_iters do
7: Draw a mini-batch \(L_B = \{(x_b, y_b)| b \in \{1, ..., B\}\}\) from \(\mathcal{L}\)
8: Draw a mini-batch \(U_B = \{u_b| b \in \{1, ..., B\}\}\) from \(\mathcal{U}\)
9: for \(c = 1\) to \(C\) do
10: Draw \(M_{B,c} = \{(x, y)| (x, y) \in \mathcal{L} \land y = c\}\) from \(\mathcal{L}_{B,c}\)
11: \(M^c = M^c + \sum_{(x, y) \in \mathcal{L}_{B,c}} \text{Softmax}(f(x; \theta))\) \(\triangleright\) update the \(c\)-th row of \(M^c\)
12: end for
13: \((\mathcal{L}', \mathcal{U}') = \text{MixMatch}(L_B, U_B)\) \(\triangleright\) apply MixMatch with augmented \(L_B\) and \(U_B\)
14: \(L_C, L_d, \text{SoftmaxCE}(\mathcal{L}', \theta), \text{SoftmaxMSE}(\mathcal{U}', \theta)\)
15: \(L = L_C + \lambda_u L_d + \lambda_{reg} L_{reg}\)
16: \(\theta = \text{SGD}(L, \theta)\) \(\triangleright\) update the parameters of \(f(\cdot; \theta)\) with Stochastic Gradient Descent
17: end for
18: Normalize \(M^c\)
19: \(M = \sigma M + (1 - \sigma)M^c\) \(\triangleright\) update the model bias matrix \(M\)
20: end while
21: \(R_{ij} = \frac{M_{ij}}{M_{ji}} R \in \mathbb{R}^{C \times C}\) \(\triangleright\) compute \(R\)
22: while \(e < T\) do
23: \((\mathcal{L}, \mathcal{U}) = \text{CASS}(\mathcal{X}, \mathcal{Y}, \theta)\) \(\triangleright\) apply Procedure 1
24: for \(i = 1\) to num_iters do
25: Draw a mini-batch \(L_B = \{(x_b, y_b)| b \in \{1, ..., B\}\}\) from \(\mathcal{L}\)
26: Draw a mini-batch \(U_B = \{u_b| b \in \{1, ..., B\}\}\) from \(\mathcal{U}\)
27: \((\mathcal{L}', \mathcal{U}') = \text{MixMatch}(L_B, U_B)\) \(\triangleright\) apply MixMatch with augmented \(L_B\) and \(U_B\)
28: \(L_C, L_d, \text{SoftmaxCE}(\mathcal{L}', \theta, R), \text{SoftmaxMSE}(\mathcal{U}', \theta)\)
29: \(L = L_C + \lambda_u L_d + \lambda_{reg} L_{reg}\)
30: \(\theta = \text{SGD}(L, \theta)\) \(\triangleright\) update the parameters of \(f(\cdot; \theta)\) with Stochastic Gradient Descent
31: end for
32: end while

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

\[
L_C' = \frac{1}{|C'|} \sum_{(x,q) \in C'} L_C(x, q) = \frac{1}{|C'|} \sum_{(x,q) \in C'} \sum_{i=1}^{C} q_i \log \left[ 1 + R_{ij} \cdot \frac{\exp(f_i(x))}{\exp(f_j(x))} \right].
\]

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 \(L'\) with its hard labels. Thus, we use the balanced loss \(L = L_C' + \lambda_u L_d + \lambda_{reg} L_{reg}\) in the training process, where \(\lambda_u\) controls the unsupervised loss and \(\lambda_{reg}\) 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 \(\rho\) following [Cui et al., 2019]. In specific, we reduce the sample size per class according to an exponential function \(n_i = O_i \frac{1}{\rho^{1/3}}\), where \(O_i\) is the original sample size of class \(i\), \(\rho = \frac{n_i}{n_c}\) and \(i \in \{1, \ldots, C\}\). Then, we generate symmetric noise in long-tailed CIFAR according to the fol-
<table>
<thead>
<tr>
<th>Imbalance Ratio</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

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

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 SSBL₂ represents our ensemble performance of two models.

The following noise transition matrix $N$:

$$N_{ij}(x, y) = P(y = j | y = i, x) = \begin{cases} 1 - r_{ij} & i = j \\ r_{ij} / \sum_{k \neq i} n_k & i \neq j \end{cases}$$

where $y$ denotes the ground truth of $x$, $y$ denotes the corrupted label of $x$ and $r \in [0, 1]$ denotes the noise rate. We set imbalance ratio to $\rho \in \{10, 50, 100\}$ and set noise rate to $r \in \{0.2, 0.5\}$. 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 $\rho$ 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 $\rho \in \{20, 50, 100\}$. 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].
**Baseline Methods.** 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 \cite{han2018co} which trains two networks simultaneously and updates one network on the data selected by the other with small-loss criterion, ELR+ \cite{liu2020 ELR+} which capitalizes on early learning via regularization, DivideMix \cite{Li2020 DivideMix}, DivideMix-LA which combines DivideMix and Logit Adjustment (LA) \cite{Menon2021 DivideMix-LA}, DivideMix-DRW which combines DivideMix and Deferred Re-Weighting (DRW), and TABASCO \cite{lu2023 TABASCO} 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 \cite{Li2020 ELR+}. 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, $\gamma_{sup}$ is set as 3 and $\gamma_{rel}$ is set as 1 in $L_{\gamma}$ (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 SSSLB 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 SSSLB$_2$ and also report its performance in Tables 1~5. For a fair comparison with H2E, which uses strong augmentation technique RandAugment \cite{cubuk2020 RandAugment} in training, we also report the performance of SSSLB with RandAugment (SSSLB+RandAug).

<table>
<thead>
<tr>
<th>Baselines</th>
<th>Co-teaching \cite{han2018co}</th>
<th>HAR-DRW \cite{cao2021 HAR-DRW}</th>
<th>MW-Net \cite{sha2019 MW-Net}</th>
<th>H2E \cite{yi2022 H2E}</th>
<th>SSBL</th>
<th>SSBL-RandAug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clothing1M</td>
<td>51.97 (43.23)</td>
<td>65.20 (54.08)</td>
<td>66.60 (57.15)</td>
<td>77.04 66.58</td>
<td>71.74 69.37</td>
<td>71.66 71.04 67.65</td>
</tr>
<tr>
<td>Food-101N</td>
<td>62.56 (50.12)</td>
<td>71.76 (64.20)</td>
<td>76.74 (69.12)</td>
<td>87.42 75.60</td>
<td>71.70 70.09 67.20</td>
<td>71.56 68.93 64.76</td>
</tr>
<tr>
<td>Animal-10N</td>
<td>60.04 (51.98)</td>
<td>70.41 63.58</td>
<td>76.88 69.12</td>
<td>79.32 74.14</td>
<td>70.16 68.66 61.12</td>
<td>70.09 67.81 61.58</td>
</tr>
</tbody>
</table>

**Table 1:** Comparison baselines in Top-1 (Top-5) test accuracy (%) on original and long-tailed versions of miniWebVision with real-world noise. SSSLB represents our performance of single model, while SSSLB$_2$ represents our ensemble performance of two models. Results of H2E on Food-101N and Animal-10N are borrowed from its original paper.

<table>
<thead>
<tr>
<th>Imbalance Ratio</th>
<th>Dataset</th>
<th>Original</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 50 100</td>
<td>WebVision</td>
<td>ImageNet</td>
<td>WebVision</td>
<td>ImageNet</td>
</tr>
<tr>
<td>Co-teaching \cite{han2018co}</td>
<td>63.60 (85.20)</td>
<td>61.30 (84.70)</td>
<td>43.11 (56.13)</td>
<td>40.95 (55.69)</td>
</tr>
<tr>
<td>HAR-DRW \cite{cao2021 HAR-DRW}</td>
<td>71.96 (90.14)</td>
<td>71.19 (89.36)</td>
<td>75.16 (82.80)</td>
<td>74.20 (80.80)</td>
</tr>
<tr>
<td>MW-Net \cite{sha2019 MW-Net}</td>
<td>71.76 (90.40)</td>
<td>76.74 (89.08)</td>
<td>76.12 (84.08)</td>
<td>75.20 (82.28)</td>
</tr>
<tr>
<td>H2E \cite{yi2022 H2E}</td>
<td>71.32 (90.34)</td>
<td>71.29 (91.28)</td>
<td>72.88 (84.70)</td>
<td>72.64 (83.04)</td>
</tr>
<tr>
<td>SSBL</td>
<td>67.42 (91.00)</td>
<td>75.00 (91.22)</td>
<td>68.12 (87.24)</td>
<td>64.92 (86.32)</td>
</tr>
<tr>
<td>SSBL-RandAug</td>
<td>68.12 (93.40)</td>
<td>74.00 (92.28)</td>
<td>69.20 (90.44)</td>
<td>65.72 (89.04)</td>
</tr>
<tr>
<td>H2E (200 epochs) \cite{yi2022 H2E}</td>
<td>71.04 (91.16)</td>
<td>76.56 (92.04)</td>
<td>72.10 (89.00)</td>
<td>69.52 (89.04)</td>
</tr>
<tr>
<td>SSBL (200 epochs)</td>
<td>80.48 (91.96)</td>
<td>76.88 (93.72)</td>
<td>73.08 (90.16)</td>
<td>70.16 (89.44)</td>
</tr>
<tr>
<td>SSBL+RandAug (200 epochs)</td>
<td>81.12 (93.68)</td>
<td>76.88 (93.72)</td>
<td>73.08 (90.16)</td>
<td>70.16 (89.44)</td>
</tr>
<tr>
<td>ELR+ \cite{liu2020 ELR+}</td>
<td>78.21 (91.50)</td>
<td>73.43 (90.50)</td>
<td>55.24 (82.65)</td>
<td>52.98 (80.88)</td>
</tr>
<tr>
<td>DivideMix \cite{li2020 DivideMix}</td>
<td>71.06 (92.74)</td>
<td>75.84 (92.84)</td>
<td>61.40 (82.12)</td>
<td>62.08 (82.64)</td>
</tr>
<tr>
<td>DivideMix-LA</td>
<td>71.44 (91.50)</td>
<td>75.06 (91.24)</td>
<td>69.80 (88.24)</td>
<td>68.00 (87.32)</td>
</tr>
<tr>
<td>DivideMix-DRW</td>
<td>78.25 (92.04)</td>
<td>75.44 (92.56)</td>
<td>65.20 (81.44)</td>
<td>65.40 (82.56)</td>
</tr>
<tr>
<td>SSSLB$_2$</td>
<td>78.56 (91.92)</td>
<td>76.64 (91.92)</td>
<td>70.40 (88.20)</td>
<td>69.12 (88.64)</td>
</tr>
</tbody>
</table>

**Table 2:** Comparison baselines in test accuracy (%) on long-tailed versions of Clothing1M, Food-101N and Animal-10N with real-world noise. SSSLB represents our performance of single model, while SSSLB$_2$ represents our ensemble performance of two models.
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 SSBL2 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 versions of CIFAR-100 with symmetric 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. SSBL2 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 ImageNet-Red, Clothing1M, Food-101N and Animal-10N with real-world noise. These tables, 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. SSBL2 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 $M$, we replace our loss $L_{\bar{C}}$ with the cross-entropy loss $L_C$. The decrease in test accuracy suggests that $L_{\bar{C}}$ is effective for learning from long-tailed noisy data.
- To study the effect of rebalancing the model bias with the estimated model bias matrix $\hat{M}$ 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 rebalancing with label frequency. The degradation in performance verifies the advantage of tackling long-tailed noisy data with the estimated model bias.

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

<table>
<thead>
<tr>
<th>Recall</th>
<th>All</th>
<th>Many</th>
<th>Medium</th>
<th>Few</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class-aware</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>Single GMM</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
</tbody>
</table>

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.
The hyper-parameters $\gamma_{sup}$ and $\gamma_{rel}$ 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.

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 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


---

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

<table>
<thead>
<tr>
<th>Noise Rate</th>
<th>$\gamma_{sup} = 3, \gamma_{rel} = 0$</th>
<th>$\gamma_{sup} = 3, \gamma_{rel} = 0.5$</th>
<th>$\gamma_{sup} = 3, \gamma_{rel} = 1$</th>
<th>$\gamma_{sup} = 3, \gamma_{rel} = 2$</th>
<th>$\gamma_{sup} = 3, \gamma_{rel} = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}$ balanced</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
</tr>
<tr>
<td>0.2</td>
<td>84.77</td>
<td>79.21</td>
<td>75.38</td>
<td>70.51</td>
<td>85.71</td>
</tr>
<tr>
<td>0.5</td>
<td>82.44</td>
<td>77.00</td>
<td>74.86</td>
<td>66.76</td>
<td>85.01</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
</tr>
<tr>
<td>0.2</td>
<td>86.30</td>
<td>79.84</td>
<td>77.72</td>
<td>72.36</td>
<td>85.83</td>
</tr>
<tr>
<td>0.5</td>
<td>85.83</td>
<td>78.39</td>
<td>75.76</td>
<td>68.44</td>
<td>85.71</td>
</tr>
<tr>
<td>$\mathcal{L}$ balanced</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
</tr>
<tr>
<td>0.2</td>
<td>87.92</td>
<td>73.90</td>
<td>79.08</td>
<td>72.79</td>
<td>50.70</td>
</tr>
<tr>
<td>0.5</td>
<td>86.29</td>
<td>42.03</td>
<td>74.54</td>
<td>59.28</td>
<td>50.70</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
</tr>
<tr>
<td>0.2</td>
<td>85.67</td>
<td>78.53</td>
<td>73.64</td>
<td>67.99</td>
<td>85.55</td>
</tr>
<tr>
<td>0.5</td>
<td>85.43</td>
<td>80.24</td>
<td>76.95</td>
<td>71.56</td>
<td>85.19</td>
</tr>
<tr>
<td>$\mathcal{L}$ balanced</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
<td>Last</td>
<td>Best</td>
</tr>
<tr>
<td>0.2</td>
<td>85.55</td>
<td>79.60</td>
<td>78.21</td>
<td>72.88</td>
<td>85.55</td>
</tr>
<tr>
<td>0.5</td>
<td>85.55</td>
<td>73.08</td>
<td>66.27</td>
<td>58.26</td>
<td>85.55</td>
</tr>
</tbody>
</table>


