Robust Long-Tail Learning in the Presence of Noisy Labels

Tong Wei, Jiang-Xin Shi, and Yu-Feng Li

Abstract—Long-tail learning has attracted much attention recently, with the goal of improving generalization for tail classes. Most existing works use supervised learning without considering the prevailing noisy labels in the training dataset. To move long-tail learning towards more realistic scenarios, we investigate this underexplored yet realistic problem in this paper. As the most popular approach to detect noisy labels, we find that the loss-based criterion fails under long-tailed class distribution, because deep neural networks are biased to misclassify tail class examples as head classes yielding unreliable losses. To overcome this limitation, we establish a new distance-based criterion that can better select correctly-labeled examples for both head and tail classes by observing the representation is more resistant to noisy labels and long-tailed class distribution than the classifier. To encourage the tail classes training, we incorporate label distributions, rather than discrete pseudo-labels, for examples that are likely mislabeled, such that the underrepresented tail classes will receive significant improvements. Based on the above findings, we propose the **Ro**bust **L**ong-**T**ail learning framework, ROLT, to train unbiased models from long-tail and noisy data. Extensive experiments on benchmark and real-world datasets demonstrate substantial improvements over state-of-the-art methods.

Index Terms—Machine learning, weakly-supervised learning, long-tail learning, learning with noisy labels, semi-supervised learning.

1 Introduction

CLASSIFICATION problems in real-world typically exhibit a long-tailed class distribution, where most classes are associated with only a few examples, e.g., visual recognition [1], [2], [3], instance segmentation [4], and text categorization [5], [6]. Due to the paucity of training examples, generalization for tail classes is challenging; moreover, naïve learning on such data is susceptible to an undesirable bias towards head classes. Recently, *long-tail learning (LTL)* has gained renewed interest in the community [7], [8], [9], [10], [11], [12], [13]. Two active strands of work involve normalisation of the classifier's weights, and modification of the underlying loss to account for different class penalties. Each of these strands is intuitive, and has been empirically shown to be effective [14], [15], [16].

Existing LTL methods with remarkable performance are mostly trained on *clean datasets* with high-quality human annotations. However, in real-world machine learning applications, annotating a large-scale dataset is costly and time-consuming. Some recent works resort to the large amount of web data as a source of supervision for training deep neural networks [17]. While the existing works have shown advantages in various applications [18], [19], web data is naturally under long-tailed class distribution accompanied with noisy labels [20], [21], [22], [23]. As a result, it is crucial that deep neural networks can harvest noisy and long-tailed training data. Nevertheless, deep neural networks (DNNs) have been shown to be prone to overfitting to noisy labels. This problem has been widely studied in the literature [24] on learning from noisy labels. Without considering noisy labels, we show that LTL methods severely degrade their performance in experiments.

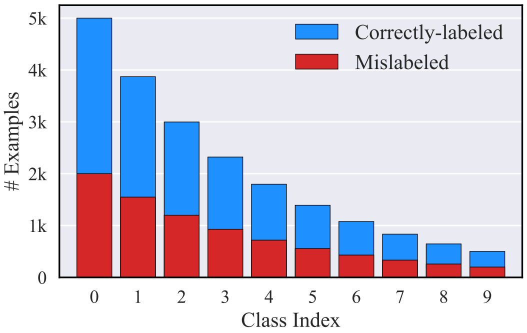
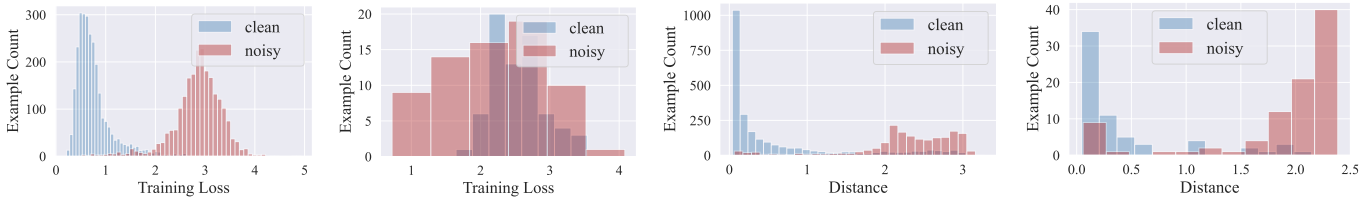


Fig. 1: Problem illustration.

In this paper, we investigate the problem of learning from noisy and long-tailed data, which is a more realistic setting but still underexplored. We provide a simple visualization of the studied problem in Figure 1. To reduce the negative impact of noisy labels, learning with noisy labels has gained a lot of attention in recent years and a lot of approaches have been proposed [20], [21], [22], [25], [26], [27], [28]. Existing works can be roughly divided into two strands, i.e., noise transition matrix estimation [29], [30] and sample selection [31], [32], [33]. Since the noise transition matrix is hard to be estimated especially when the number of classes is large, sample selection is a more promising way of handling noisy labels and is our focus in this paper. In

- *Tong Wei is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China.*
  E-mail: weit@seu.edu.cn
- *Jiang-Xin Shi and Yu-Feng Li are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, China.*
  E-mail: {shijx, liyf}@lamda.nju.edu.cn

Manuscript received April 19, 2005; revised August 26, 2015.
(Co-first author: Tong Wei, Jiang-Xin Shi.)
(Corresponding author: Yu-Feng Li.)



(a) loss of head class (b) loss of tail class (c) distance of head class (d) distance of tail class

Fig. 2: (a-b) Training losses for examples of head class and tail class. (c-d) Distance distribution between examples and their class prototype for head class and tail class. Experiments are conducted on CIFAR-10 with noise level $\gamma = 50\%$ and imbalance ratio $\rho = 100$.

sample selection methods, the *small-loss* criterion is one of the most popular approaches [34], [35], [36], [37], [38]. It selects samples with small losses and treats these samples as correctly annotated for robust training. In recent years, the small-loss criterion has been demonstrated to be effective in many works.

However, small-loss criterion selects possibly clean samples using a constant threshold [35], [38] or mixture distribution model [36], [37] for all classes, thus failing to consider different learning status and learning difficulties of different classes, which is very important in LTL. Owing to this paucity of samples for tail classes, naïve method is susceptible to an undesirable bias towards head classes. Specifically, due to the misclassification of samples with tail classes (large losses), the small-loss criterion cannot distinguish clean samples of tail classes and samples with noisy labels. Once wrong selection is made, the inferiority of accumulated errors will arise. We further confirm this standpoint by experiments as shown in Figure 2a and 2b.

In previous literature, Meta-Weight-Net [39] was proposed to address this challenging problem. It learns a weighting function that produces high weights for samples of tail classes. To this end, Meta-Weight-Net solves a complex bi-level optimization problem using an auxiliary validation set, similar to previous work [40]. This validation set needs to be clean and class-balanced, which is hard to obtain in practice. Even equipped with this unbiased validation set, we find that Meta-Weight-Net yields limited improvement.

When handling noisy labels in long-tailed data, we believe it is important to keep the approach simple and free of auxiliary supervision. The benefit of doing this is that the approach can be easily absorbed by many existing frameworks for learning with long-tailed data. Guided by this belief, we propose the *class-wise small-distance* criterion. For each individual class, it selects small-distance samples as clean where distance is calculated between the sample and its class prototype in the embedding space. The intuition why the *class-wise small-distance* criterion can be more robust than *small-loss* is briefly explained as following: (1) *why is sample-distance better than small-loss?* As confirmed by many previous literature [33], [41], [42], it is reasonable to assume that clean examples tend to be clustered around their prototypes even when training with noisy labels. (2) *why can be the sample selection work well in a class-wise manner?* As the number of classes can be large and the population of classes varies significantly in training data, the variance of distances between samples and class prototypes becomes large. We show the distance distribution for both head and tail classes in Figure 2c and 2d. Moreover, the proposed *class-wise small-distance* is general and can be combined with semi-supervised learning to improve the generalization.

In typical sample selection approaches, such as MentorNet [34] and Co-training [35], only samples flagged as possibly clean are used for training. However, when learning from long-tail data, we believe that even mislabeled samples are valuable, especially for the underrepresented tail classes. To this end, we incorporate label distributions, but not discrete pseudo-labels, for examples that are likely mislabeled, such that the underrepresented tail classes will receive significant improvements, which is crucial in LTL. By incorporating this with the *class-wise small-distance*, we propose a **Ro**bust **L**ong-**T**ail learning framework, ROLT, for training an unbiased model from long-tail and noisy data.

Our main contributions are summarized as follows.

1) We investigate the problem of learning from long-tail in the presence of noisy data, which is underexplored but is a considerable step towards real-world applications;
2) We find the popular small-loss criterion fails under long-tailed class distribution, and establish a new *class-wise small-distance* criterion to fill the gap;
3) We propose a robust framework, ROLT. It incorporates label distributions for mislabeled examples to encourage the tail classes training.
4) Our framework can be built on top of semi-supervised learning methods without much extra overhead, yielding an improved approach ROLT+.
5) We present a new noise generation method which incorporates the long-tailed class distribution.
6) The proposed method can significantly outperform state-of-the-art methods on both benchmark and real-world datasets.

The rest of this paper is organized as follows. Section 2 briefly reviews related works. Section 3 presents details of the proposed framework ROLT. Section 4 reports experimental results over a wide range of datasets with both label noise and class imbalance. Section 5 concludes this paper.

2 RELATED WORK

Our work is closely related to the following directions.

Long-tail learning. Recently, many approaches have been proposed in LTL [12], [16], [43], [44]. Most extant approaches can be categorized into three types by modifying (i) the inputs to a model by re-balancing the training data [2], [45], [46]; (ii) the outputs of a model, for example by post-hoc adjustment of the classifier [14], [47], [48], [49]; and (iii) the internals of a model by modifying the loss function [11], [15], [39], [50], [51], [52]. Each of the above methods are intuitive, and have shown strong empirical performance. However, these methods assume the training examples are correctly-labeled, which is often difficult to obtain in many real-world applications. Instead, we study a realistic problem to learn from long-tailed data with label noise. Although the presence of label noise in class-imbalanced dataset has also been mentioned in HAR [53], they only consider a specialized noise setup. In this work, we provide a more general simulation of label noise, as well as systematic studies for LTL methods.

Label noise detection. Plenty of methods have been proposed to detect noisy examples [34], [35], [37], [54], [55], [56], [57], [58]. Many works adopt the small-loss trick, which treats examples with small training losses as correctly-labeled. In particular, MentorNet [34] reweights samples with small loss so that noisy samples contribute less to the loss. Co-teaching [35] trains two networks where each network selects small-loss samples in a mini-batch to train the other. DivideMix [37] fits a Gaussian mixture model on per-sample loss distribution to divide the training data into clean set and noisy set. In addition, AUM [59] introduces a margin statistic to identify noisy samples by measuring the average difference between the logit values for a sample's assigned class and its highest non-assigned class. The above methods only consider training datasets that are class-balanced, thus is not applicable for long-tailed label distribution. Recent work [23] observes the real-world dataset with label noise also has imbalanced number of samples per-class. Nevertheless, they only inspect a particular setup, while we provide a systematic study of learning with noisy labels under various long-tailed scenarios. In contrast to previous works, we propose a class-wise prototypical noise detection method that works well in LTL.

3 Long-Tail Learning meets Noisy Labels

In this section, we first introduce the problem setting and some background. Then we discuss the disadvantages of small-loss criterion under long-tailed class distribution. Finally, we present the proposed framework RoLT, which exploits the *class-wise small-distance* criterion and soft pseudo-labeling to combat noisy labels and improve the performance. We illustrate our method in Figure 3.

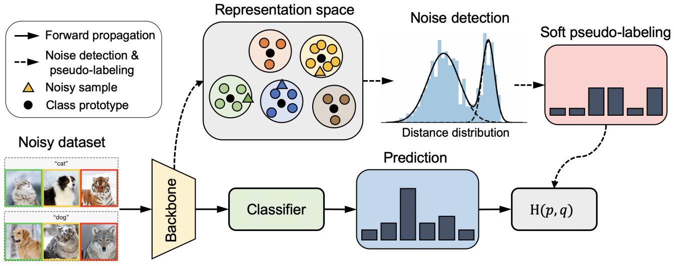


Fig. 3: The proposed framework RoLT.

3.1 Problem Setting & Background

Given a training dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i$ is an instance feature vector and $y_i \in \mathcal{C} = [K] = \{1, \ldots, K\}$ is the class label assigned to it. The training examples $(\boldsymbol{x}_i, y_i), 1 \leq i \leq N$ consists of two types: i) a *correctly-labeled example* whose assigned label matches the ground-truth label, i.e., $y_i = y_i^*$, where y_i^* denotes the ground-truth label of $\boldsymbol{x}_i$, ii) a *mislabeled example* whose assigned label does not match the ground-truth label, but the input matches one of the classes in $\mathcal{C}$, i.e., $y_i \neq y_i^*$ and $y_i^* \in \mathcal{C}$. Moreover, the data follows a long-tailed class distribution where the class prior distribution $\mathbb{P}(y)$ is highly skewed, so that many underrepresented classes have a very low probability of occurrence. Specifically, we denote the imbalance ratio as $\rho = \max_y \mathbb{P}(y) / \min_y \mathbb{P}(y)$ to indicate the skewness of data. Classes with high $\mathbb{P}(y)$ are referred to as *head classes*, and others are referred to as *tail classes*.

In practice, since the data distribution is unknown, Empirical Risk Minimization (ERM) uses the training data to achieve an empirical estimate of the underlying data distribution. Typically, one minimizes the softmax cross-entropy as following

$$\ell(y, f(\boldsymbol{x})) = \log\left[\sum_{y' \in [K]} e^{f_{y'}(\boldsymbol{x})}\right] - f_y(\boldsymbol{x}), \tag{1}$$

where $f_y(\boldsymbol{x})$ denotes the predictive probability of model f on class y. This ubiquitous approach neglects the issue of class imbalance, and makes the model biased toward head classes. Moreover, it assumes training examples are correctly-labeled. Commonly used approaches for learning with noisy labels cannot work well in a long-tailed class distribution.

To reduce the impact of noisy labels, sample selection based on the small-loss criterion is one of the most popular approaches. It selects "easy" samples for training based on the outputs of the current networks. Specifically, given an arbitrary training example $(\boldsymbol{x}_i, y_i)$, we can obtain a loss ℓ_i, i.e., $\ell_i = \ell\left(f\left(\boldsymbol{x}_i\right), y_i\right)$. Then a certain proportion of samples with small losses are selected as probably clean samples. The selection can be achieved by a manually set hyperparameter [35] or a mixture distribution model that fits two-component mixtures on the loss distribution [36].

The small-loss criterion was demonstrated to be effective in the existing literature, we find that the it does not work well on long-tail dataset. It is known to us that DNNs learn "easy" samples first, and on long-tail dataset, it first learns to recognize head classes ahead of tail classes. In light of that, DNNs tend to misclassify tail class examples as head classes. This phenomenon has been widely revealed in recent works [14], [46]. Therefore, the losses cannot reflect the probability that a sample is mislabeled, especially for tail classes as shown in Figure 2a and 2b.

3.2 Robust Distance-based Sample Selection

We present a new criterion for selecting noisy labels, called *small-distance*. Formally, we model clean examples of class $k \in [K]$ as if they were distributed around prototype $\boldsymbol{c}_k \in \mathbb{R}^D$, and the likelihood of an example $\boldsymbol{x}$ belonging to class k decays exponentially with its distance from the prototype $\boldsymbol{c}_k$, i.e., $\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{c}_k) \propto e^{-dist(\boldsymbol{c}_k,\boldsymbol{x})}$, which is a common assumption about the data distribution [60], [61]. Here, $dist$ is a distance measure in the embedding space and is typically set to be the Euclidean distance. Considering the variance of data distribution of different classes, we propose to inspect the distance statistics in a *class-wise* manner.

To justify the feasibility of using distance to select clean examples, we compute the AUC based on the distance between examples and their class prototypes for each class separately, and report the average value of *Many* (more than 100 images), *Medium* (20∼100 images), *Few* (less than 20 images) and *All* shots in Table 1. The experiment is done on CIFAR-100 with imbalance ratio $\rho = 100$, noise level $\gamma = 20\%$ and $\gamma = 50\%$. It can be seen that the AUC is high even for tail classes, indicating the distance measure may be a useful approach to distinguish clean and noisy examples. Notice that, some previous literature [33], [41], [42] also confirms that the representations are resilient to noisy labels.

	Many	Medium	Few	All
$\gamma = 20\%$	95.16	93.38	82.81	90.64
	Many	Medium	Few	All
$\gamma = 50\%$	92.00	87.43	73.60	85.20

TABLE 1: Average AUC of different shots.

As aforementioned, the sample selection can be achieved using a thresholding hyperparamter or a mixture distribution model. Following previous works [36], [37], we will use the latter approach. Specifically, for class k, we employ a two components GMM [62] to model the distance distribution of clean and noisy samples, i.e., $d \sim \sum_{j=1}^{2} \phi_j \mathcal{N}(\mu_j, \sigma_j^2)$, where $d = dist(\boldsymbol{c}_k, \boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{D}_k$ and ϕ_j denotes weight of the j-th component. Note that we have $\sum_{j=1}^{2} \phi_j = 1$. Without loss of generality, we assume $\mu_1 < \mu_2$. Since clean examples locate around the prototype while noisy examples spread out, we flag $\boldsymbol{x}$ as clean if and only if $\mathbb{P}(d \mid \mu_1, \sigma_1) > \mathbb{P}(d \mid \mu_2, \sigma_2)$. We thus perform noise detection by estimating the Gaussians' parameters from distance statistics. In particular, for each class $k \in [K]$, we compute its prototype as the normalized average of the embeddings for training examples by

$$\boldsymbol{c}_k \leftarrow \text{Normalize}\left(\frac{1}{|\mathcal{D}_k|} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} f_\theta(\boldsymbol{x}_i)\right), \tag{2}$$

where $f_\theta(\boldsymbol{x})$ denotes the extracted feature of $\boldsymbol{x}$ and $\forall \boldsymbol{x}_i \in \mathcal{D}_k$. Given $\boldsymbol{c}_k$, the distances between $\boldsymbol{c}_k$ and examples of class k are obtained by

$$dist(\boldsymbol{c}_k, \boldsymbol{x}_i) = \left\|\boldsymbol{c}_k - f_\theta(\boldsymbol{x}_i)\right\|_2^2, \tag{3}$$

We then fit a two-component Gaussian mixture model to maximize the log-likelihood value by optimizing $\max \sum_{i=1}^{|\mathcal{D}_k|} \log(\sum_{j=1}^{2} \phi_j \mathbb{P}(d_k(\boldsymbol{x}_i) \mid \mu_j, \sigma_j))$, where $d_k(\boldsymbol{x}_i) = dist(\boldsymbol{c}_k, \boldsymbol{x}_i)$ for $\boldsymbol{x}_i \in \mathcal{D}_k$.

For simplicity, we denote the clean (noisy) data of class k as $\mathcal{X}_k$ ($\mathcal{S}_k$). Note that we have $\mathcal{D}_k = \mathcal{X}_k \bigcup \mathcal{S}_k$. Therefore, we obtain a subset of clean examples by $\mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}_k$ and noisy examples by $\mathcal{S} = \bigcup_{k=1}^{K} \mathcal{S}_k$. It is also verified that Gaussian mixture model can be used to distinguish clean and noisy data because of its flexibility in the sharpness of distribution in previous literature [37]. We also observe that the proposed method works well on real-world datasets where the Gaussian distribution assumption does not perfectly satisfied. Recall that $\mathcal{D}_k$ may contain noisy labels, the estimate of $\boldsymbol{c}_k$ in equation 2 is inaccurate and the split of $\mathcal{D}_k = \mathcal{X}_k \bigcup \mathcal{S}_k$ is problematic. To remedy this, we refine class prototypes using $\mathcal{X}_k$ rather than $\mathcal{D}_k$, and acquire a new split of $\mathcal{D}_k$. By doing this, the obtained $\mathcal{X}_k$ retains most of correctly-labeled examples of class k as well as less mislabeled examples.

Algorithm 1: ROLT

```
Input: training dataset $\{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$, initial learning
  rate $\eta_0$, number of warm-up iterations $T_0$
// Warm-up Stage: run SGD for $T_0$ iterations
for $t = 1, \dots, T_0$ do
    Sample $m_0$ examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m_0}$ from $\mathcal{D}$
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_0 \tilde{\boldsymbol{g}}_t$, where
      $\tilde{\boldsymbol{g}}_t = \frac{1}{m_0} \sum_{i=1}^{m_0} \nabla \ell(\boldsymbol{w}_t; \boldsymbol{x}_i)$
end
// Robust Learning Stage: run SGD for $T$ iterations
for $t = 1, \dots, T$ do
    $\mathcal{X} = \varnothing, \mathcal{S} = \varnothing$
    for $k = 1, \dots, K$ do
        Compute class prototype $\boldsymbol{c}_k$ as in equation 2
        Compute distance between the prototype $\boldsymbol{c}_k$
          and each of $\boldsymbol{x}_i \in \mathcal{D}_k$ as in equation 3
        Fit GMM and divide $\mathcal{D}_k$ into clean set $\mathcal{X}_k$ and
          noisy set $\mathcal{S}_k$
        $\mathcal{X} = \mathcal{X} \bigcup \mathcal{X}_k, \mathcal{S} = \mathcal{S} \bigcup \mathcal{S}_k$   // collect clean and
          noisy examples of class $k$
        Refine class prototype
          $\boldsymbol{c}_k \leftarrow \text{Normalize}\left(\frac{1}{|\mathcal{X}_k|} \sum_{i \in \mathcal{X}_k} f_\theta(\boldsymbol{x}_i)\right)$
    end
    Compute soft pseudo-labels $\tilde{\boldsymbol{y}}$ as in equation 4
    Compute stochastic gradient $\boldsymbol{g}_t$ as
      $\boldsymbol{g}_t = \frac{\sum_{i=1}^{|\mathcal{X}|} \nabla H(\boldsymbol{y}_i, f(\boldsymbol{x}_i)) + \sum_{j=1}^{|\mathcal{S}|} \nabla H(\tilde{\boldsymbol{y}}_j, f(\boldsymbol{x}_j))}{|\mathcal{X}| + |\mathcal{S}|}$
    Update model parameters using $\boldsymbol{g}_t$ and learning
      rate $\eta$ : $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$
end
```

3.3 Soft Pseudo-Labeling with Label Distribution

For each example that are likely to be mislabeled, we convert its original discrete noisy label to a label distribution by incorporating the uncertainty of the ground-truth label. The benefits of using label distributions are two-folds. First, it mitigates the influence of noisy labels [63]; Second, it compensates the learning of data scarcity tail classes [64]. To this end, the underrepresented tail classes will receive significant improvements, which is crucial in LTL. To generate label distributions, a direct approach is to leverage the prediction of ERM (Empirical Risk Minimization) model. However, the ERM is known to be biased toward head classes [64]. Hence, refining noisy labels using the predictions of ERM may be sub-optimal for examples of tail classes. In contrast, the NCM (Nearest Class Mean) classifier can yield balanced classification boundary [14]. Specifically, we find that the NCM classifier produces much higher recall on tail classes than the ERM in experiments. By aggregating the predictive information from the ERM and NCM classifiers, we construct diverse soft pseudo-labels for detected noisy examples. To amend the misflag of noisy detector, we also take account of the original labels as a source of soft pseudo-labels. Moreover, since it is not impossible that both ERM and NCM classifiers produce incorrect predictions, we further remedy this by the label smoothing technique [64].

Putting together, given the predictions $\hat{y}^{erm} = \arg\max_k f(\boldsymbol{x})$, $\hat{y}^{ncm} = \arg\min_k \|\boldsymbol{c}_k - f_\theta(\boldsymbol{x})\|_2$, and original noisy label y, we construct the guessing label set $\mathcal{G} = \{\hat{y}^{erm}, \hat{y}^{ncm}, y\}$ and generate the label distribution $\tilde{\boldsymbol{y}} \in \mathbb{R}^K$ for $\boldsymbol{x}$. For $k \in [K]$, we compute

$$\tilde{y}_k = \begin{cases} \frac{1}{4}\sum_{\hat{y}\in\mathcal{G}} \mathbb{I}(\hat{y}=k) + \frac{1}{4K} & \text{if } k \in \mathcal{G} \\ \frac{1}{4K} & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\mathbb{I}(\cdot)$ is an indicator which returns 1 if the condition is true, otherwise 0. Considering the classification task with four classes (i.e., $K=4$) and $\mathcal{G} = \{1, 4, 2\}$, the soft pseudo-label would be $\tilde{\boldsymbol{y}} = [\frac{5}{16}, \frac{5}{16}, \frac{1}{16}, \frac{5}{16}]$. The targets $\hat{y}^{erm}$ and $\hat{y}^{ncm}$ can be set equal to the model output, but using a running average is more effective which is known as temporal ensembling [65] in semi-supervised learning. For ERM or NCM classifier, let $\boldsymbol{z}_i(t) \in \mathbb{R}^K$ be the output logits vector (pre-softmax output) for example $\boldsymbol{x}_i$ at iteration t of training, we update the momentum logits by

$$\boldsymbol{q}_i(t) = \alpha \boldsymbol{q}_i(t-1) + (1-\alpha)\boldsymbol{z}_i(t), \tag{5}$$

where $0 \leq \alpha < 1$ is the combination weight. For each iteration t, we then obtain $\hat{y}^{erm}$ and $\hat{y}^{ncm}$ using softmax of $\boldsymbol{q}_i(t)$. Having acquired $\mathcal{X}$, $\mathcal{S}$, and soft pseudo-labels, we first compute the cross-entropy loss for clean examples using original training labels by

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}|}\sum_{\boldsymbol{x}_i \in \mathcal{X}} H(\boldsymbol{y}_i, f(\boldsymbol{x}_i)), \tag{6}$$

where $\boldsymbol{y}_i$ is the one-hot label vector for $\boldsymbol{x}_i$. For noisy examples, the loss function is computed by

$$\mathcal{L}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}_i \in \mathcal{S}} H(\tilde{\boldsymbol{y}}_i, f(\boldsymbol{x}_i)), \tag{7}$$

where $H(\boldsymbol{q}, \boldsymbol{p}) = -\sum_{k=1}^{K} q_k \log\left(\frac{\exp p_k}{\sum_{j=1}^{K}\exp p_j}\right)$ is the cross-entropy between distributions $\boldsymbol{q}$ and $\boldsymbol{p}$. Overall, the training objective is $\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \mathcal{L}_{\mathcal{S}}$. Details of the method are presented in Algorithm 1.

3.4 Combing with Semi-Supervised Learning

The proposed method can be further improved by using well-established semi-supervised learning approach, where clean and noisy examples are viewed as labeled and unlabeled data, respectively. Inspired by DivideMix [37] and ELR+ [66], we use two separate neural networks, where the target of each network is computed from the output of the other network. For fair comparison, we replace the sample selection module of DivideMix by our proposed class-wise prototypical noise detector. We call this improved method RoLT+.

3.5 Difference with Prior Works using Class Prototypes

Class prototypes are employed in some previous literature and we discuss the differences between this paper and some related works. In few-shot learning, prototypical networks [67] learn a metric space in which classification can be performed by computing distances to prototype representations of each class. Compared to other methods, prototypical networks reflect a simpler inductive bias that is beneficial in this limited-data regime, and achieve excellent results. In long-tailed recognition, OLTR [2] proposes to use the distances between samples and prototypes to handle open-set recognition. In self-supervised learning, PCL [68] proposes the ProtoNCE loss which encourages representations to be closer to their assigned prototypes. However, the above-mentioned works do not consider using class prototypes to detect noisy labels in long-tailed class distribution. Moreover, the distances calculated between examples and their class prototypes are utilized in a *class-wise* manner to mitigate the influence of long-tailed class distribution.

3.6 Label Noise Generation

To simulated label noise in long-tail data, we propose a new label noise generation method. To generate noisy labels, the most basic idea is to utilize the noise transition matrix [29], denoting the probabilities that clean labels flip into noisy labels. Let Y denote the variable for the clean label, $\bar{Y}$ the noisy label, and X the instance, the transition matrix $T(X = x)$ is defined as $T_{ij}(X) = \mathbb{P}(\bar{Y} = j \mid Y = i, X = x)$. In this work, we propose a new noise generation method by setting $T(X = x)$ according to the estimated class priors $\mathbb{P}(y)$, e.g., the empirical class frequencies in the training set. Formally, given noise proportion $\gamma \in [0, 1]$, we define

$$T_{ij}(X) = \mathbb{P}(\bar{Y} = j \mid Y = i, X = x) = \begin{cases} 1-\gamma & \text{if } i = j \\ \frac{N_j}{N-N_i}\gamma & \text{else.} \end{cases} \tag{8}$$

Here, N denotes the total number of training examples and N_j is frequency of class j. In contrast to commonly used uniform label noise, examples are more likely to be mislabeled as frequent ones in real-world situations.

4 Experiments

We now present experiments that confirm our main claims:

1) on benchmark datasets, we demonstrate the efficacy of our methods by comparing with approaches for both long-tail learning and learning with noisy labels;
2) on real-world datasets with natural label noise and long-tailness, our methods consistently outperform many existing methods;
3) we provide detailed studies for each component of our framework to show its effectiveness.

4.1 Datasets and Implementation Details

CIFAR. We follow the simple data augmentation used in [69] with only random crop and horizontal flip. For experiments of RoLT, we use ResNet-32 as the backbone network and train it using standard SGD with a momentum of 0.9, a

		CIFAR-10						CIFAR-100					
Noise Level		20%			50%			20%			50%		
Imbalance Ratio		10	50	100	10	50	100	10	50	100	10	50	100
CE	Best	75.61	63.60	62.17	63.25	43.38	38.71	43.27	30.27	26.21	26.92	16.97	14.23
	Last	73.86	58.60	54.15	48.39	32.73	27.05	43.06	30.04	26.08	26.60	15.59	13.47
LDAM	Best	82.37	71.34	66.26	60.30	42.95	36.66	48.14	33.43	29.70	29.62	17.51	14.19
	Last	82.07	71.11	65.88	59.10	38.33	33.38	47.89	33.30	29.50	29.38	17.29	13.24
LDAM-DRW	Best	83.73	76.41	72.28	67.93	48.88	43.23	50.44	36.60	32.27	32.24	19.48	15.21
	Last	83.67	75.67	71.08	67.68	47.38	41.45	50.29	36.16	32.05	31.72	19.23	14.75
BBN	Best	77.81	68.01	64.51	64.71	46.22	36.72	47.60	31.07	28.79	30.01	19.75	14.56
	Last	76.81	67.48	64.24	53.76	43.35	34.83	47.26	30.76	28.56	29.42	19.55	14.34
cRT	Best	76.15	65.02	59.92	64.15	43.26	36.73	42.56	30.23	26.31	25.55	17.47	16.01
	Last	75.05	64.22	58.47	62.75	41.87	34.55	41.56	30.08	26.18	23.94	17.34	15.94
MW-Net†	Best	82.19	71.63	67.26	72.12	56.09	46.36	50.20	36.68	31.77	37.50	23.99	**21.24**
	Last	77.67	64.12	58.23	59.68	45.39	37.05	47.82	34.45	29.57	33.14	20.33	18.82
HAR-DRW	Best	82.43	67.44	67.88	67.39	52.35	42.80	46.24	28.86	26.29	31.30	16.75	14.78
	Last	78.44	61.08	62.73	64.75	45.06	40.07	43.04	26.11	24.71	26.96	13.87	12.42
RoLT	Best	85.03	75.80	71.83	76.72	55.38	49.62	51.83	36.28	31.10	37.58	24.25	19.56
	Last	84.71	75.31	71.36	76.28	54.54	48.77	51.63	36.09	30.98	37.37	23.96	19.23
RoLT-DRW	Best	**85.04**	**77.86**	**73.84**	**77.11**	**60.15**	**55.32**	**53.41**	**38.94**	**33.36**	**39.22**	**25.51**	20.61
	Last	**84.95**	**77.65**	**73.54**	**76.94**	**59.59**	**54.55**	**53.22**	**38.77**	**33.20**	**39.01**	**25.35**	**20.45**

† MW-Net uses an auxiliary $1k$ clean and class-balanced validation set.

TABLE 2: Test accuracy (%) on CIFAR datasets with different imbalanced ratio and noise level.

		CIFAR-10						CIFAR-100					
Noise Level		20%			50%			20%			50%		
Imbalance Ratio		10	50	100	10	50	100	10	50	100	10	50	100
ELR+	Best	**88.96**	**80.21**	69.60	85.02	56.96	48.72	54.01	49.64	38.40	49.53	30.12	21.58
	Last	88.09	**79.69**	66.67	84.08	48.14	43.11	53.32	48.37	38.12	49.06	29.68	20.47
DivideMix	Best	88.79	75.34	66.90	87.54	67.92	61.81	63.79	49.64	43.91	49.35	36.52	31.82
	Last	**88.10**	73.48	63.76	86.88	65.22	59.65	63.17	48.37	42.59	48.87	35.72	31.05
RoLT+	Best	87.95	77.26	**72.31**	**88.17**	**75.11**	**64.42**	**64.22**	**51.01**	**45.35**	**53.31**	**39.78**	**35.29**
	Last	87.54	75.90	**69.12**	**87.45**	**73.92**	**61.15**	**63.31**	**49.40**	**43.16**	**52.44**	**39.27**	**34.43**

TABLE 3: Test accuracy (%) on CIFAR datasets with different imbalanced ratio and noise level.

weight decay of 2×10^{-4}, a batch size of 128, and an initial learning rate of 0.1. The model is trained for 200 epochs. We perform noise detection and soft pseudo-labeling after a warm up period of 80 epochs, and anneal the learning rate by a factor of 100 at 160 and 180 epochs. For experiments of RoLT+, we use the same settings as [37], which trains two 18-layer PreAct Resnet. The model is trained for 300 epochs, and the warm up period has 50 epochs. We train each model with 1 NVIDIA GeForce RTX 2070.

mini WebVision. Following previous work [37], we use two Inception-Resnet V2 for RoLT+. The model is trained for 100 epochs. We set the initial learning rate as 0.01, and reduce it by a factor of 10 after 50 epochs. The warm up period is 40 epochs. We train each model with 2 NVIDIA Tesla V100 GPUs.

ImageNet-LT. We use the long-tailed version of ImageNet dataset [70] produced by [2]. The imbalance ratio of ImageNet-LT is 200 and we simulate different level of label noise to compare their performance. We train ResNet-10 for all methods, following previous literature [14]. We use standard SGD with a momentum of 0.9, a weight decay of 5×10^{-4}. The batch size is set as 512. The model is trained for 90 epochs. We set the initial learning rate as 0.2, and reduce it by a factor of 10 every 30 epochs. We perform noise detection and soft pseudo-labeling after a warm up period of 40 epochs. Each model is trained with 2 NVIDIA Tesla M40 GPUs.

4.2 Results on Simulated Datasets

Setting. We test RoLT on CIFAR-10 and CIFAR-100 under various imbalanced ratio ρ and noise level γ. For each dataset, we first simulate the long-tailed dataset following the same setting as LDAM [50]. The long-tailed imbalance follows an exponential decay in sample sizes across different classes. To inject noisy labels, we use the noise transition matrix equation 8 defined in Section 3.6 or the asymmetric noise to form the training set. In particular, we consider imbalance ratio to be $\rho \in \{10, 50, 100\}$ and noise level to be $\gamma \in \{20\%, 50\%\}$.

Results under simulated noise in Section 3.6. Table 2 and Table 3 summarize the results for CIFAR-10 and CIFAR-100. As shown in the results, previous LTL methods (i.e., LDAM [50], BBN [46], cRT [14]) dreadfully degrade their performance as the noise level and imbalance ratio increase, while our methods retain robust performance. In particular, compared with CE, RoLT improves the test accuracy by 8% on average. It can be observed that the improvement becomes more significant at high noise levels, benefiting from proposed noise detection and soft pseudo-labeling. Further application of Deferred Re-Weighting (DRW) [50] enhances the performance by favoring the tail classes. This clearly demonstrates the importance of correcting noisy labels in the training data. Moreover, MW-Net [39] and HAR-DRW [53] are proposed to handle label noise and class imbalance. Our method consistently outperforms them by a large margin.

We further compare RoLT+ with DivideMix [37] and ELR+ [66], which are the most popular methods for learning with noisy labels. The results are given in Table 3. First, we can see that the performance of ELR+ significantly drops as the training set becomes class-imbalanced. DivideMix is relatively robust to class imbalance than ELR+ by imposing the uniform predictions regularization in its objective. In contrast, our method RoLT+ achieves performance improvements in test accuracy by 2.57% on average. This validates the superiority of our noise detector over the small-loss trick. In the supplementary material, we further show that DivideMix flags most examples of tail classes as noisy, which is the main reason accounting for its failure.

Results under asymmetric noise. We further verify the effectiveness of the proposed detection method and the robust framework under asymmetric label noise. We conduct experiments on long-tail CIFAR-10 dataset and the noise injection rules are illustratetd in Figure 4, following the previous works [37], [66]. From Table 4, it can be seen that RoLT+ consistently outperforms DivideMix in all cases. It is interesting to observe that the performance gap between *best* and *last* widens as the imbalance ratio becomes large. This reflects that the model is easy to collapse under asymmetric noise and class imbalance. The proposed method can alleviate this issue substantially.

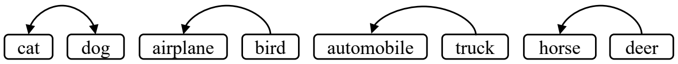


Fig. 4: Transit directions of asymmetric label noise.

Noise Level		20%			40%		
Imbalance Ratio		10	50	100	10	50	100
DivideMix	Best	83.48	73.13	66.51	78.85	67.74	59.63
	Last	78.77	56.90	44.50	74.50	46.48	32.68
RoLT+	Best	**84.05**	**75.93**	**68.53**	**79.98**	**71.44**	**65.56**
	Last	**79.81**	**67.13**	**60.38**	**74.68**	**60.07**	**54.05**

TABLE 4: Accuracy (%) on CIFAR-10 with asymmetric noise.

4.3 Results on Real-World Dataset

4.3.1 Results on WebVision dataset

We test the performance of our method on a real-world dataset. WebVision [17] contains 2.4 million images collected from Flickr and Google with real noisy and class-imbalanced data. The noise level of WebVision is estimated at 20%. Following previous literature, we train on a subset, mini WebVision, which contains the first 50 classes. In Table 5, we report results comparing against state-of-the-art approaches, including D2L [19], MentorNet [34], Co-teaching [35], Iterative-CV [71], HAR [53], DivideMix [37], and ELR+ [66]. RoLT produces superior results than DivideMix and ELR+, particularly in terms of top-5 accuracy.

	Webvision		ImageNet	
	top1	top5	top1	top5
D2L	62.68	84.00	57.80	81.36
MentorNet	63.00	81.40	57.80	79.92
Co-teaching	63.58	85.20	61.48	84.70
Iterative-CV	65.24	85.34	61.60	84.98
HAR	75.50	90.70	70.30	90.00
DivideMix	77.32	91.64	**75.20**	90.84
ELR+	**77.78**	91.68	70.29	89.76
RoLT+	77.64	**92.44**	74.64	**92.48**

TABLE 5: Accuracy (%) on WebVision and ImageNet.

To further uncover the advantages of our method, we run experiments by controlling the imbalance ratio of Webvision dataset. The test accuracy is reported in the Table 6. From the results, we can see that the superiority of our method is more significant as the imbalance ratio increases.

Imbalance ratio	Method	Webvision	ImageNet
$\rho = 50$	DivideMix	64.56 (83.56)	62.68 (85.24)
	w/ DRW	68.16 (84.92)	66.12 (85.40)
	RoLT+	66.28 (88.68)	64.76 (89.96)
	w/ DRW	70.08 (88.52)	67.28 (90.12)
$\rho = 100$	DivideMix	55.76 (73.48)	53.92 (74.00)
	w/ DRW	60.28 (74.60)	59.04 (75.68)
	RoLT+	60.68 (87.84)	59.68 (88.52)
	w/ DRW	65.48 (87.32)	64.80 (87.08)

TABLE 6: Top1(top5) accuracy on Webvision and ImageNet.

4.3.2 Results on ImageNet-LT dataset

We compare our method with baselines (CE and ELR) on a large long-tail benchmark, i.e. *ImageNet-LT*, by combining with methods and strategies for class-imbalanced datasets, i.e. Classifier Re-training (cRT) [14] and Logit Adjustment (LA) [49]. From the comparison results in Table 7, we can clearly see that our method outperforms other methods, particularly under high noise level.

	20% label noise			50% label noise		
Method	-	w/ cRT	w/ LA	-	w/ cRT	w/ LA
CE	28.18	34.15	34.06	17.80	21.85	22.71
ELR	26.58	35.21	34.05	17.33	22.80	22.60
RoLT	**29.57**	**35.76**	**35.09**	**21.53**	**25.61**	**25.50**

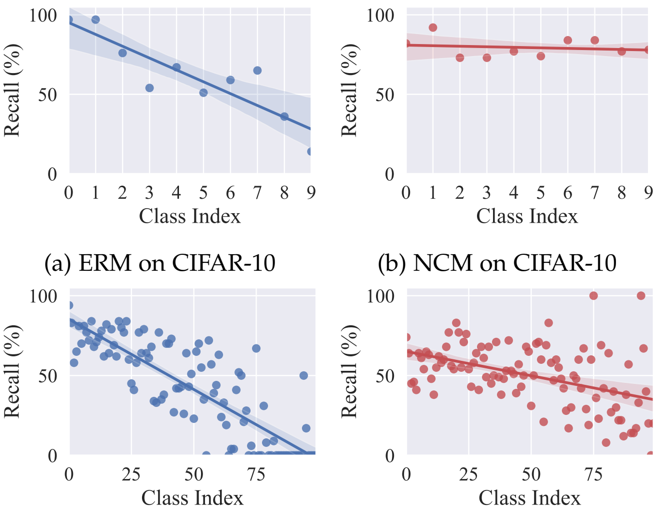
TABLE 7: Accuracy (%) on *ImageNet-LT* dataset

4.4 Further Analysis and Ablation Studies

Efficacy of the noise detector. To further support our motivation, we compare the performance of the ERM and NCM classifiers in Figure 5. It can be seen that NCM produces more balanced recall across classes, while ERM tends to predict examples as head classes, resulting in low recall for tail classes. Figure 6 shows the precision and recall of selected clean examples by our method. To better understand RoLT, we construct three groups of classes for CIFAR-100 by: many (more than 100 images), medium (20$\sim$100 images), and few (less than 20 images) shots; and CIFAR-10 by: many ($\{0, 1\}$), medium($\{2, \dots, 6\}$), and few ($\{7, 8, 9\}$) shots according to class indices. RoLT maintains high precision and recall, which validates the effectiveness of our method. This experiment is conducted under imbalance ratio $\rho = 100$ and noise level $\gamma = 30\%$.

DRW	Classifier	Pseudo-Label	$\gamma = 20\%$ Many	Med.	Few	All	$\gamma = 50\%$ Many	Med.	Few	All
✗	Linear	Noisy	49.38	21.42	4.57	26.21	32.06	7.89	0.04	14.23
✗	Linear	ERM	58.79	26.50	4.21	**31.24**	38.83	12.05	0.89	18.41
✗	Linear	Soft (w/o LS)	54.59	26.47	7.25	30.65	36.03	15.11	2.19	18.94
✗	Linear	Soft (w/ LS)	56.59	26.95	5.79	31.10	37.20	15.97	1.74	**19.56**
✗	NCM	Noisy	44.09	32.03	12.00	30.52	26.86	17.89	5.59	17.71
✗	NCM	ERM	49.06	34.92	13.07	**33.61**	31.11	21.05	5.63	**20.41**
✗	NCM	Soft (w/o LS)	45.32	31.05	14.14	31.17	29.14	21.13	5.41	19.69
✗	NCM	Soft (w/ LS)	47.65	32.16	13.93	32.32	29.80	20.74	5.85	19.89
✓	Linear	Noisy	45.82	26.50	10.79	28.67	23.77	14.53	3.41	14.76
✓	Linear	ERM	50.62	31.55	11.64	32.46	32.80	17.05	2.30	18.58
✓	Linear	Soft (w/o LS)	44.21	31.76	15.39	31.41	30.31	18.92	4.93	19.13
✓	Linear	Soft (w/ LS)	47.85	32.68	16.68	**33.36**	30.94	21.32	6.22	**20.61**
✓	NCM	Noisy	43.21	31.95	12.61	30.36	26.86	17.89	5.59	17.71
✓	NCM	ERM	43.53	33.21	11.07	30.52	26.83	19.45	5.52	18.27
✓	NCM	Soft (w/o LS)	45.09	30.26	12.25	30.26	26.80	20.50	5.56	18.67
✓	NCM	Soft (w/ LS)	45.41	32.34	14.39	**31.76**	29.37	21.29	5.74	**19.92**

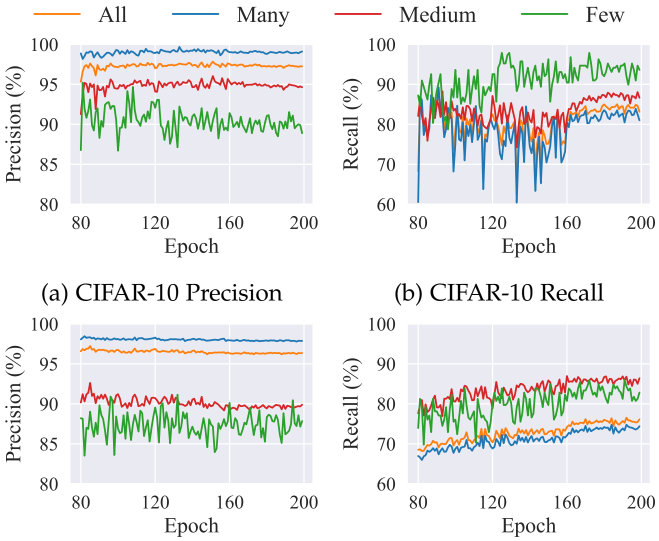
TABLE 8: Ablation studies on pseudo-labeling. Test accuracy on CIFAR-100 is reported.



(a) ERM on CIFAR-10 (b) NCM on CIFAR-10

(c) ERM on CIFAR-100 (d) NCM on CIFAR-100

Fig. 5: Per-class recall of ERM and NCM classifiers



(a) CIFAR-10 Precision (b) CIFAR-10 Recall

(c) CIFAR-100 Precision (d) CIFAR-100 Recall

Fig. 6: Precision and Recall of selected samples.

Efficacy of the soft pseudo-labeling. We investigate the effectiveness of soft pseudo-labeling by comparing it with three other methods: (i) keep the noisy labels, (ii) rectify it via the ERM predictions, (iii) use the soft label without label smoothing (w/o LS) as follow:

$$\tilde{y}_k = \begin{cases} \frac{1}{3}\sum_{\hat{y}\in\mathcal{G}} \mathbb{I}(\hat{y}=k) & \text{if } k \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We report the results in Table 8 with respect to noise level $\gamma \in \{20\%, 50\%\}$ and imbalance ratio $\rho = 100$. We observe that ERM and soft pseudo-labeling significantly improve the performance by over 4% in test accuracy, and the improvement is more significant under high noise levels. Moreover, the soft pseudo-labeling outperforms its ERM and "w/o LS" counterpart in most cases, demonstrating that label smoothing and label guessing can provide diverse and informative supervision under imperfect training labels. We also investigate the effectiveness of learned representations with NCM for classification. It can be observed that NCM with soft labels outperforms the one using original noisy labels, which confirms that our soft pseudo-labeling facilitates representation learning.

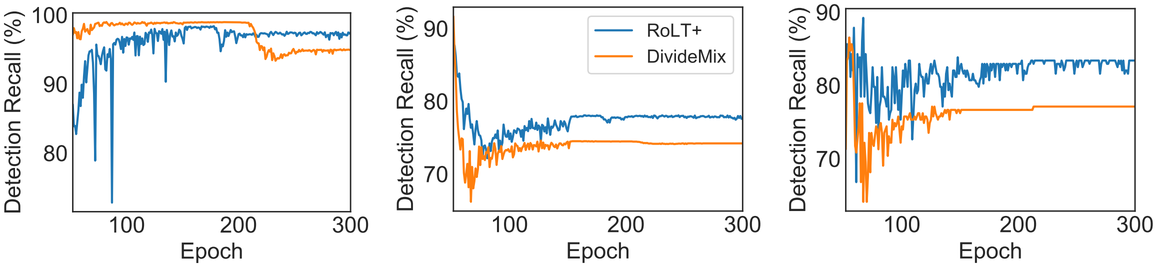
		CIFAR-10		CIFAR-100	
Noise Level		20%	50%	20%	50%
DivideMix	Best	**92.79**	**95.03**	77.25	73.84
	Last	**92.41**	**94.63**	77.03	**73.42**
RoLT+	Best	92.46	94.59	**78.60**	**74.11**
	Last	92.01	94.41	**78.14**	73.35

TABLE 9: Test accuracy (%) on class-balanced datasets.

4.5 Comparison RoLT+ with DivideMix

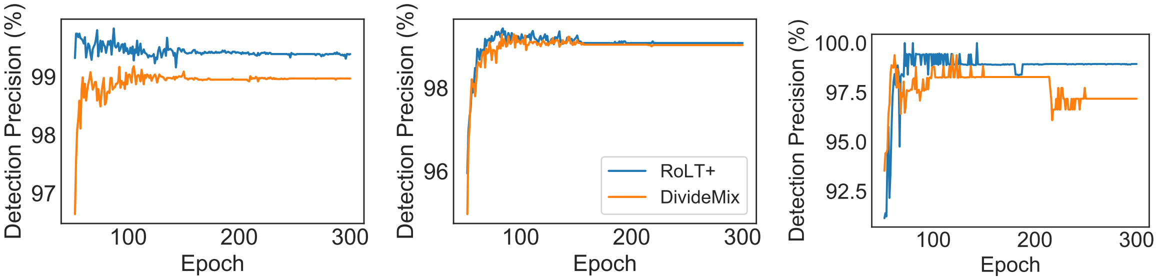
4.5.1 Comparison with DivideMix w.r.t. Noise Detection

To further demonstrate our proposed noise detection that is tailored for LTL, we compare it with DivideMix and the results are shown in Figure 7∼10. This experiment



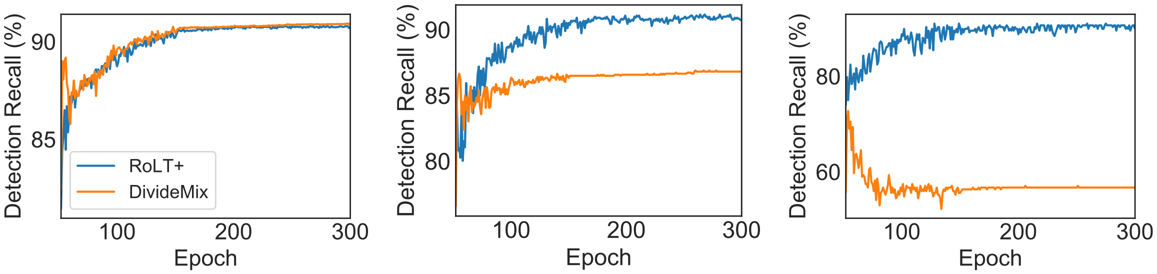
(a) Many Shot (b) Medium Shot (c) Few Shot

Fig. 7: Comparison of detection recall between RoLT+ and DivideMix on CIFAR-10.



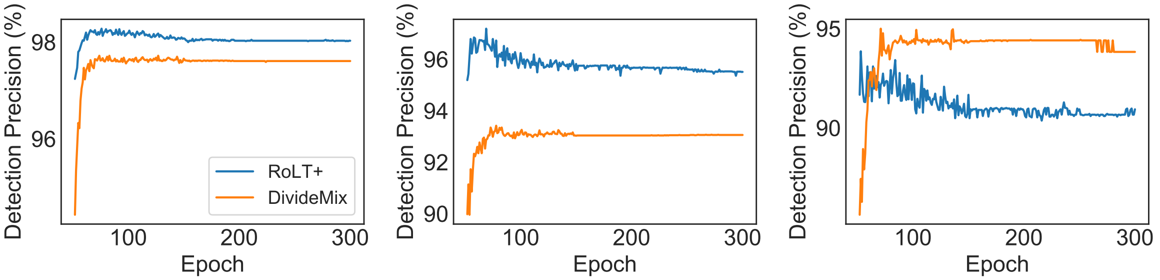
(a) Many Shot (b) Medium Shot (c) Few Shot

Fig. 8: Comparison of detection precision between RoLT+ and DivideMix on CIFAR-10.



(a) Many Shot (b) Medium Shot (c) Few Shot

Fig. 9: Comparison of detection recall between RoLT+ and DivideMix on CIFAR-100.



(a) Many Shot (b) Medium Shot (c) Few Shot

Fig. 10: Comparison of detection precision between RoLT+ and DivideMix on CIFAR-100.

is conducted under imbalance ratio $\rho = 100$ and noise level $\gamma = 20\%$. We partition classes into three splits, i.e., Many, Medium, and Few-shots, and report the recall and precision of examples that are flagged as clean for each split. It can be observed that the detection recall of DivideMix is smaller than RoLT+ on Medium and Few shot. This also explains that, DivideMix trains networks that are biased towards head classes, thus leading to poor overall performance. Moreover, the detection precision of RoLT+ is larger than DivideMix in all cases, except the CIFAR-

DRW	Classifier	Pseudo-Label	$\gamma = 20\%$				$\gamma = 50\%$			
			Many	Med.	Few	All	Many	Med.	Few	All
✗	Linear	Noisy	54.06	26.53	4.43	29.70	31.03	8.42	0.48	14.19
✗	Linear	ERM	61.47	29.32	4.96	33.43	46.60	14.18	1.11	**22.00**
✗	Linear	Soft (w/o LS)	59.94	32.39	9.71	**35.41**	38.06	16.05	1.70	19.88
✗	Linear	Soft (w/ LS)	60.03	30.74	8.89	34.58	34.03	14.66	1.48	17.88
✗	NCM	Noisy	49.82	27.82	11.14	30.63	25.60	16.37	6.59	16.96
✗	NCM	ERM	58.53	30.11	12.46	**34.83**	42.57	16.50	4.41	**22.36**
✗	NCM	Soft (w/o LS)	54.06	31.26	14.86	34.42	31.23	16.13	4.37	18.24
✗	NCM	Soft (w/ LS)	52.71	27.89	12.57	32.04	24.46	13.34	3.26	14.51
✓	Linear	Noisy	49.53	30.34	13.93	32.27	24.83	13.53	5.11	15.21
✓	Linear	ERM	54.41	34.00	19.61	**36.91**	39.34	20.08	7.04	**23.30**
✓	Linear	Soft (w/o LS)	52.26	36.00	18.57	36.65	30.31	19.92	8.74	20.54
✓	Linear	Soft (w/ LS)	52.76	34.84	18.93	36.48	28.14	19.32	6.78	19.02
✓	NCM	Noisy	49.50	29.45	12.00	31.38	25.60	16.37	6.59	16.96
✓	NCM	ERM	56.09	32.39	13.75	**35.23**	41.00	17.89	4.74	**22.43**
✓	NCM	Soft (w/o LS)	53.06	32.37	14.43	34.38	29.97	16.71	4.22	17.98
✓	NCM	Soft (w/ LS)	50.76	28.29	13.18	31.70	24.06	13.76	3.33	14.55

TABLE 10: Ablation studies on pseudo-labeling based on models that optimize LDAM loss. Test accuracy on CIFAR-100 dataset with imbalance ratio $\rho = 100$ is reported.

	CIFAR-10			CIFAR-100		
Imbalance Ratio	10	50	100	10	50	100
CE	86.75	77.38	71.83	56.31	44.15	38.88
LDAM	86.38	77.62	74.31	55.66	43.61	39.25
LDAM-DRW	87.29	81.25	78.78	57.21	**47.30**	**42.93**
BBN	87.83	81.19	78.87	58.08	45.62	40.09
cRT	86.78	77.30	71.18	56.62	43.01	39.44
MW-Net[†]	**87.99**	79.58	74.92	**58.66**	46.72	42.10
HAR-DRW	87.81	79.82	75.99	56.89	43.34	40.78
RoLT	**87.99**	80.50	77.70	57.47	45.38	39.35
RoLT-DRW	87.75	**83.02**	**80.57**	57.48	47.21	41.70

[†] MW-Net uses a $1k$ clean and class-balanced validation set.

TABLE 11: Test accuracy (%) on clean datasets.

100 Few shot. However, in this case, DivideMix has a low detection recall, so the high precision is meaningless. This experiment demonstrates the superiority of our prototypical noise detection method.

4.5.2 Comparison with DivideMix on Balanced Datasets

We compare the performance of our method with DivideMix on balanced datasets with noise level $\rho \in \{20\%, 50\%\}$. The results are reported in Table 9 and our method is comparable with DivideMix. This shows that the proposed prototypical noise detector also works well on balanced datasets.

4.5.3 Results on Clean CIFAR Datasets

Although our method is particularly designed for long-tail learning in the presence of noisy labels, it is interesting to study its performance on clean datasets. We report the results in Table 11. Intriguingly, RoLT consistently outperforms vanilla CE in all cases, showing the benefit of the proposed soft pseudo-labeling approach. Additionally, our method achieves comparable performance with the popular baseline LDAM-DRW. In comparison with the HAR-DRW, which is also proposed to cope with class imbalance and label noise problems, our method improves the performance by over 2% on average. By using an auxiliary $1k$ clean and class-balanced validation set, MW-Net is able to achieve comparable performance with RoLT-DRW in 4 out of 6 cases, however in other 2 cases, the performance gain of RoLT-DRW is significant. Moreover, an auxiliary $1k$ clean and class-balanced validation set is hard to obtain in practice. This validates the robustness of our method, which does not hurt the performance in the corner case.

4.6 Results for Optimizing LDAM Loss

In the main text, we optimize the cross-entropy loss and report its performance for comparison. One may interested in if other loss functions can be integrated into our framework. To this end, we leverage the LDAM loss, which is particularly designed for LTL, and report the results in Table 10. This indeed produces different results with the cross-entropy. It is known that LDAM can prevent the networks from being biased toward tail classes and yield balanced predictions. Therefore, it is reasonable to use predictions of the ERM for pseudo-labeling. By further applying the soft pseudo-labels, it puts much focus on tail classes and results in performance deterioration.

4.7 Decoupling Representation and Classifier Learning

In Table 12∼15, we study the impact of label noise for two-stage LTL methods, i.e., Classifier Re-Training (cRT) and Nearest Classifier Mean (NCM), which disentangle the representation and classifier learning. In this setup, γ_r and γ_c are the noise level when performing representation and classifier learning, respectively.

We have the following observations from the results. In particular, when $\gamma_c = 0$, the performance of both cRT and NCM drop significantly as γ_r increases, revealing the negative impact of label noise on representation learning. With respect to classifier learning, it can be seen that cRT further suffers from inaccurate supervision. In contrast, NCM classifier retains high performance as γ_c grows. The results validate our finding that NCM is more robust to label noise, which motivates us to investigate distance-based method for noise detection. Moreover, in order to improve the representation learning, one may remove noisy data or rectify noisy labels during training. In this work, we provide two ways of achieving this, by pseudo-labeling

		$\rho = 1$								$\rho = 10$								$\rho = 100$					
		γ_c								γ_c								γ_c					
		0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%
γ_r	0%	93.15	92.85	92.76	92.55	92.56	92.40	γ_r	0%	86.78	85.90	85.43	85.21	83.13	81.49	γ_r	0%	71.18	68.59	66.31	65.92	61.83	57.58
	10%	91.43	91.37	91.36	91.31	91.33	91.41		10%	81.13	80.22	78.84	77.60	77.05	75.13		10%	62.48	61.54	59.91	58.70	55.57	53.17
	20%	90.40	90.42	90.31	90.33	90.35	90.24		20%	76.91	76.48	76.15	75.09	75.20	73.66		20%	61.33	60.18	59.92	57.98	56.34	52.82
	30%	88.74	88.80	88.77	88.58	88.73	88.54		30%	75.64	74.60	74.36	74.17	72.76	71.17		30%	55.26	55.05	53.79	54.05	50.45	47.74
	40%	87.00	86.91	86.82	86.89	86.85	86.75		40%	72.26	71.61	70.95	69.96	70.05	67.83		40%	51.98	51.22	51.05	50.36	50.12	46.28
	50%	84.57	84.53	84.46	84.38	84.29	83.95		50%	67.01	67.04	66.83	64.68	64.16	64.15		50%	41.70	40.90	40.75	40.07	38.61	36.73

TABLE 12: Accuracy (%) of cRT on CIFAR-10 with different imbalanced ratio ρ and noise level γ.

		$\rho = 1$								$\rho = 10$								$\rho = 100$					
		γ_c								γ_c								γ_c					
		0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%
γ_r	0%	92.77	92.75	92.69	92.67	92.55	92.54	γ_r	0%	88.14	88.08	87.97	87.89	87.72	87.45	γ_r	0%	79.59	79.64	79.67	79.64	79.57	78.63
	10%	91.29	91.29	91.28	91.31	91.25	91.24		10%	82.23	82.33	82.09	82.05	81.91	81.91		10%	68.21	68.09	67.06	66.19	65.35	64.53
	20%	90.20	90.24	90.23	90.26	90.31	90.24		20%	75.27	75.02	74.73	74.37	73.82	73.25		20%	66.80	66.59	66.25	65.98	64.95	63.70
	30%	88.51	88.51	88.48	88.55	88.53	88.53		30%	74.99	75.01	74.98	74.76	74.52	74.09		30%	61.68	61.22	61.06	60.91	60.04	59.19
	40%	86.77	86.80	86.78	86.80	86.79	86.76		40%	70.45	69.75	69.40	69.07	68.43	67.97		40%	56.57	56.46	56.21	55.92	55.47	54.60
	50%	83.78	83.78	83.78	83.77	83.79	83.77		50%	66.16	65.82	65.62	65.40	65.07	64.82		50%	44.66	44.08	43.98	43.18	43.10	42.61

TABLE 13: Accuracy (%) of NCM on CIFAR-10 with different imbalanced ratio ρ and noise level γ.

		$\rho = 1$								$\rho = 10$								$\rho = 100$					
		γ_c								γ_c								γ_c					
		0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%
γ_r	0%	69.73	68.90	68.09	67.49	66.61	65.70	γ_r	0%	56.62	53.55	52.21	50.52	49.09	47.34	γ_r	0%	39.44	35.43	33.93	32.34	31.32	29.68
	10%	68.55	68.03	67.38	66.58	66.48	65.87		10%	50.55	49.13	47.89	46.23	44.55	43.36		10%	33.19	32.25	30.69	28.76	27.61	25.97
	20%	65.51	65.21	64.86	64.45	64.46	63.96		20%	45.31	44.22	42.56	41.73	40.27	38.61		20%	27.77	27.02	26.31	24.57	23.79	22.82
	30%	63.01	62.74	62.32	62.02	61.65	60.96		30%	41.72	40.82	39.77	37.80	37.84	36.38		30%	24.91	23.83	23.61	21.48	21.28	19.61
	40%	60.78	60.42	60.30	59.73	59.19	58.93		40%	37.33	36.76	35.31	34.46	32.18	32.68		40%	23.02	22.38	22.04	21.49	20.62	19.48
	50%	57.88	57.51	56.98	56.83	55.97	55.14		50%	32.07	31.09	30.29	29.90	28.58	25.55		50%	19.05	18.60	17.93	17.67	16.89	16.01

TABLE 14: Accuracy (%) of cRT on CIFAR-100 with different imbalanced ratio ρ and noise level γ.

		$\rho = 1$								$\rho = 10$								$\rho = 100$					
		γ_c								γ_c								γ_c					
		0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%			0%	10%	20%	30%	40%	50%
γ_r	0%	66.94	66.71	66.37	65.79	64.84	64.07	γ_r	0%	56.05	55.63	55.22	54.14	53.18	51.78	γ_r	0%	41.73	41.18	40.59	39.81	38.56	37.95
	10%	65.72	65.84	65.66	65.27	64.83	64.16		10%	50.49	50.76	50.14	49.53	49.51	48.16		10%	35.43	34.89	34.49	33.77	32.93	32.11
	20%	63.08	62.92	63.26	62.61	62.67	62.15		20%	45.22	45.05	45.15	44.83	44.21	43.22		20%	30.47	29.95	29.45	28.74	28.56	28.13
	30%	60.82	60.64	60.62	60.81	60.29	60.16		30%	41.82	41.66	41.23	41.31	40.27	39.68		30%	25.97	25.50	25.17	24.74	23.96	22.59
	40%	57.87	58.00	57.81	57.82	57.91	57.55		40%	36.13	36.32	36.19	35.81	35.41	34.84		40%	23.89	23.47	22.80	22.29	21.84	20.50
	50%	55.24	55.25	55.05	55.01	54.64	54.95		50%	30.85	30.63	30.50	30.07	29.84	29.34		50%	19.16	18.63	18.47	18.15	16.89	16.77

TABLE 15: Accuracy (%) of NCM on CIFAR-100 with different imbalanced ratio ρ and noise level γ.

using either ERM predictions or soft pseudo-labels. Recall that, NCM computes the classification vectors for each class by taking the mean of all vectors belonging to that class. Thus, the classification accuracy is directly related to the feature representation quality. By observing considerable performance gains for NCM, it shows the effectiveness of our pseudo-labeling method for representation learning.

4.8 Discussion and Limitations

One may be interested in combining the proposed method RoLT with other loss functions. In particular, we attempt to optimize LDAM loss [50] during training and the results are reported in the supplementary material. Indeed, LDAM encourages the model to yield balanced classification boundaries. However, it slightly distort these boundaries when applied together with soft pseudo-labeling because too much focus has been put on tail classes. Our experimental finding suggests using the ERM predictions as pseudo-labels leading to more significant improvements.

Additionally, we admit that it is challenging to train networks that consistently performs well under various noise levels in LTL. Although RoLT can take both label noise and class imbalance into account, its improvement is less obvious when training on a clean dataset. We report the results in the supplementary material due to limited space. This is because that the noise detector inevitably fits a two-component GMM and flags some examples as noisy, leading to loss of accurate supervision. We believe this concern can be alleviated by estimating the noise proportion in training data, which is another interesting research problem, and leave this for future work.

5 Conclusion

In this paper, we propose to mitigate the influence of noisy labels in long-tail learning and present a robust learning framework. We reveal the failure of loss-based sample selection criterion under long-tailed class distribution, and establish a new distance-based criterion which can more accurately select correctly-labeled examples for both head and tail classes. Our method can be applied to many existing methods to improve their generalization. Moreover, we propose a new noise generation method which incorporates the class frequencies. We provide systematic studies on benchmark and real-world datasets to verify the superiority of the proposed framework by comparing to state-of-the-art methods in the strands of long-tail learning and learning with noisy labels. We believe that this work can motivate more future studies on this underexplored yet realistic task.

References

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

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

[3] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, "Equalization loss for long-tailed object recognition," in *CVPR*, 2020, pp. 11 659–11 668.

[4] A. Gupta, P. Dollár, and R. B. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *CVPR*, 2019, pp. 5356–5364.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

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

[7] C. Cardie and N. Nowe, "Improving minority class prediction using case-specific feature weights," in *ICML*, 1997, pp. 57–65.

[8] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2005.

[9] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[10] Y. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *NeurIPS*, 2017, pp. 7029–7039.

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

[12] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. Yu, "Long-tailed recognition by routing diverse distribution-aware experts," in *ICLR*, 2021.

[13] T. Wei, W.-W. Tu, Y.-F. Li, and G.-P. Yang, "Towards robust prediction on tail labels," in *SIGKDD*, 2021, pp. 1812–1820.

[14] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2020.

[15] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *ECCV*, vol. 12349, 2020, pp. 162–178.

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

[17] W. Li, L. Wang, W. Li, E. Agustsson, and L. V. Gool, "Webvision database: Visual learning and understanding from web data," *CoRR*, vol. abs/1708.02862, 2017.

[18] W. Li, L. Niu, and D. Xu, "Exploiting privileged information from web data for image categorization," in *ECCV*, 2014, pp. 437–452.

[19] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," in *ICML*, 2018, pp. 3355–3364.

[20] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L_DMI: An information-theoretic noise-robust loss function," in *NeurIPS*, 2019.

[21] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama, "Dual T: reducing estimation error for transition matrix in label-noise learning," in *NeurIPS*, 2020.

[22] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, "Robust early-learning: Hindering the memorization of noisy labels," in *ICLR*, 2021.

[23] J. Li, C. Xiong, and S. C. Hoi, "Mopro: Webly supervised learning with momentum prototypes," in *ICLR*, 2021.

[24] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR*, 2017.

[25] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *NeurIPS*, 2013.

[26] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Transactions on Neural Networks Learning Systems*, vol. 25, no. 5, pp. 845–869, 2013.

[27] D.-B. Wang, Y. Wen, L. Pan, and M.-L. Zhang, "Learning from noisy labels with complementary loss functions," in *AAAI*, 2021, pp. 10 111–10 119.

[28] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[29] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 447–461, 2015.

[30] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *NeurIPS*, 2018, pp. 10 477–10 486.

[31] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," in *CVPR*, 2018, pp. 8688–8696.

[32] Q. Yao, H. Yang, B. Han, G. Niu, and J. kwok, "Searching to exploit memorization effect in learning with noisy labels," in *ICML*, 2020, pp. 6033–6042.

[33] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, "Robust inference via generative classifiers for handling noisy labels," in *ICML*, vol. 97, 2019, pp. 3763–3772.

[34] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *ICML*, 2018, pp. 2304–2313.

[35] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NeurIPS*, 2018, pp. 8536–8546.

[36] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. Mcguinness, "Unsupervised label noise modeling and loss correction," in *ICML*, 2019, pp. 312–321.

[37] J. Li, R. Socher, and S. C. Hi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *ICLR*, 2020.

[38] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, and M. Sugiyama, "Sample selection with uncertainty of losses for learning with noisy labels," in *ICLR*, 2022.

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

[40] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*, 2018, pp. 4334–4343.

[41] P. Wu, S. Zheng, M. Goswami, D. N. Metaxas, and C. Chen, "A topological filter for learning with label noise," in *NeurIPS*, 2020.

[42] Z.-F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y.-F. Li, "Ngc: A unified framework for learning with open-world noisy data," in *ICCV*, 2021, pp. 62–71.

[43] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *NeurIPS*, 2020.

[44] M.-L. Zhang, Y.-K. Li, H. Yang, and X.-Y. Liu, "Towards class-imbalance aware multi-label learning," *IEEE Transactions on Cybernetics*, 2020.

[45] L. Shen, Z. Lin, and Q. Huang, "Relay backpropagation for effective learning of deep convolutional neural networks," in *ECCV*, vol. 9911, 2016, pp. 467–482.

[46] B. Zhou, Q. Cui, X. Wei, and Z. Chen, "BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition," in *CVPR*, 2020, pp. 9716–9725.

[47] H. Ye, H. Chen, D. Zhan, and W. Chao, "Identifying and compensating for feature deviation in imbalanced deep learning," *CoRR*, vol. abs/2001.01385, 2020.

[48] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," in *NeurIPS*, 2020.

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

[50] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *NeurIPS*, 2019, pp. 1565–1576.

[51] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *CVPR*, 2020, pp. 7610–7619.

[52] J. Ren, C. Yu, S. Sheng, X. Ma, H. Zhao, S. Yi, and H. Li, "Balanced meta-softmax for long-tailed visual recognition," in *NeurIPS*, 2020.

[53] K. Cao, Y. Chen, J. Lu, N. Arechiga, A. Gaidon, and T. Ma, "Heteroskedastic and imbalanced deep learning with adaptive regularization," in *ICLR*, 2021.

[54] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *CVPR*, 2018, pp. 5552–5560.

[55] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *ICCV*, 2019, pp. 322–330.

[56] Y. Kim, J. Yim, J. Yun, and J. Kim, "NLNL: negative learning for noisy labels," in *ICCV*, 2019, pp. 101–110.

[57] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *CVPR*, 2019, pp. 7017–7025.

[58] D. T. Nguyen, C. K. Mummadi, T. . P. . N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, "SELF: learning to filter noisy labels with self-ensembling," in *ICLR*, 2020.

[59] G. Pleiss, T. Zhang, E. R. Elenberg, and K. Q. Weinberger, "Identifying mislabeled data using the area under the margin ranking," in *NeurIPS*, 2020.

[60] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *NeurIPS*, 2004, pp. 513–520.

[61] D. Samuel and G. Chechik, "Distributional robustness loss for long-tail learning," *CoRR*, vol. abs/2104.03066, 2021.

[62] H. H. Permuter, J. M. Francos, and I. Jermyn, "A study of gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognition*, vol. 39, no. 4, pp. 695–706, 2006.

[63] M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar, "Does label smoothing mitigate label noise?" in *ICML*, 2020, pp. 6448–6458.

[64] Z. Zhong, J. Cui, S. Liu, and J. Jia, "Improving calibration for long-tailed recognition," in *CVPR*, 2021.

[65] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2017.

[66] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *NeurIPS*, 2020.

[67] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[68] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," in *ICLR*, 2021.

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

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[71] P. Chen, B. B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *ICML*, 2019.

Tong Wei received the PhD degrees in computer science from Nanjing University, China, in 2021. He joined the Southeast University in 2022, and is currently an associate researcher. His research interest is machine learning. Particularly, he is interested in weakly supervised learning and long-tail learning.

Jiang-Xin Shi received the BSc degree in 2020. He is currently working toward the PhD degree in the National Key Laboratory for Novel Software Technology at Nanjing University, China. His research interest is machine learning. Particularly, he is interested in weakly-supervised learning.

Yu-Feng Li received the BSc and PhD degrees in computer science from Nanjing University, China, in 2006 and 2013, respectively. He joined the National Key Laboratory for Novel Software Technology at Nanjing University in 2013, and is currently an associate professor. He is a member of the LAMDA group. His research interests include mainly in machine learning. Particularly, he is interested in weakly supervised learning, statistical learning and optimization. He has received outstanding doctoral dissertation award from China Computer Federation (CCF), outstanding doctoral dissertation award from Jiangsu Province and Microsoft Fellowship Award. He has published more than 50 papers in top-tier journals and conferences such as the Journal of Machine Learning Research, the IEEE Transactions on Pattern Analysis and Machine Intelligence, the Artificial Intelligence, the IEEE Transactions on Knowledge and Data Engineering, ICML, NIPS, AAAI, etc. He is served as an editorial board member of machine learning journal (since 2021-), neural network (since 2020-), etc., co-chair of ACML22/21 journal track and ACML19 tutorial, and an area chair/senior PC member of top-tier conferences such as ICML22/21, IJCAI'21.