ROBUST LONG-TAILED LEARNING UNDER LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-tailed learning has attracted much attention recently, with the goal of improving generalisation for tail classes. Most existing works use supervised learning without considering the prevailing noise in the training dataset. To move long-tailed learning towards more realistic scenarios, this work investigates the label noise problem under long-tailed label distribution. We first observe the negative impact of noisy labels on the performance of existing methods, revealing the intrinsic challenges of this problem. As the most commonly used approach to cope with noisy labels in previous literature, we then find that the small-loss trick fails under long-tailed label distribution. The reason is that deep neural networks cannot distinguish correctly-labeled and mislabeled examples on tail classes. To overcome this limitation, we establish a new prototypical noise detection method by designing a distance-based metric that is resistant to label noise. Based on the above findings, we propose a robust framework, RoLT, that realizes noise detection for long-tailed learning, followed by soft pseudo-labeling via both label smoothing and diverse label guessing. Moreover, our framework can naturally leverage semi-supervised learning algorithms to further improve the generalisation. Extensive experiments on both benchmark and real-world datasets demonstrate substantial improvement over many existing methods. For example, RoLT outperforms baselines by more than 5% in test accuracy.

1 INTRODUCTION

Classification problems in real-world typically exhibit a long-tailed label distribution, where most classes are associated with only a few examples, e.g., visual recognition Horn et al. (2018); Liu et al. (2019); Tan et al. (2020), instance segmentation Gupta et al. (2019), and text categorization Wei & Li (2020). Due to the paucity of training examples, generalisation for tail classes is challenging; moreover, naïve learning on such data is susceptible to an undesirable bias towards head classes. Recently, long-tailed learning (LTL) has gained renewed interest in the context of deep neural networks Wang et al. (2017); Cui et al. (2019); Kang et al. (2020); Wu et al. (2020); Yang & Xu (2020); Wu et al. (2021); Wang et al. (2021). 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 Menon et al. (2021).

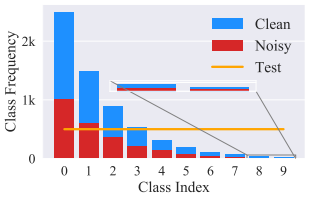
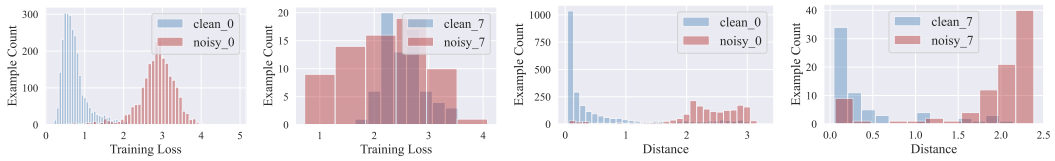


Figure 1: Problem setup.

The above-mentioned 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 Li et al. (2017). While the existing works have shown advantages in various applications Li et al. (2014); Ma et al. (2018), web data is naturally class-imbalanced Li et al. (2020; 2021) and accompanied with label noise Xu et al. (2019); Li et al. (2020); Yao et al. (2020); Xia et al. (2021). As a result, it is crucial that deep neural networks can harvest noisy and class-imbalanced training data.

Although the LTL and noisy label problems have been extensively studied in previous literature, it is still poorly explored when the training dataset follows a long-tailed label distribution and contains



(a) Loss for head class (b) Loss for tail class (c) Distance for head class (d) Distance for tail class

Figure 2: (a-b) Training losses for examples of head class and tail class, respectively. (c-d) Distance distribution between examples and their class prototype for head class and tail class, respectively.

label noise. We provide a simple visualization of the studied problem in Figure 1. Without considering label noise, we show that LTL methods severely degrade their performance in experiments. To address this problem, a direct approach is to apply methods for learning with noisy labels to LTL. One of the most commonly used approaches for learning with noisy labels is DivideMix Li et al. (2020), which uses the small-loss criterion to detect label noise. However, we note that using such approach leads to unsatisfactory results in long-tailed label distribution, as shown in Figure 2a and 2b. Therefore, it remains a challenge to obtain models that can cope with LTL under label noise.

To achieve performance improvement, it is a natural idea to detect noisy data while accommodating class imbalance. It is known that a classifier trained on long-tailed data yields higher accuracy for head classes but hurts tail classes Kang et al. (2020). Thus, to detect label noise, it is not trustworthy to use predictions and training losses produced by the biased classifier. Another commonly used classifier for LTL is the nearest class mean (NCM) classifier that computes class prototypes and performs nearest neighbour search in embedding space Kang et al. (2020). As many previous literature, it is reasonable to assume clean examples tend to be clustered around their prototypes even when training with noisy labels. This inspires us to design a *class-independent* noise detector by treating examples closed to their corresponding prototypes as clean, while others as noisy. As a comparison with the small-loss trick, we demonstrate the distance distribution for both head and tail classes in Figure 2c and 2d. Unlike learning from balanced datasets where noisy data can be removed from training Pleiss et al. (2020), we claim that each example is significant, especially for tail classes. To this end, we introduce a new soft pseudo-labeling mechanism that uses both label smoothing and label guessing to guide the learning of networks. Thanks to the generality of our proposed noise detection method, we can also interpret noisy examples as unlabeled data and incorporate well-established semi-supervised learning techniques to further improve the generalisation.

Our main contributions are: (i) We study the problem of long-tailed learning under label noise, which is less explored and is a significant step towards real-world applications; (ii) We find that the commonly used small-loss trick fails in long-tailed learning. Thus, we establish a novel prototypical noise detection method that overcomes the limitations of small-loss trick; (iii) We propose a robust framework, RoLT. It realizes noise detection that is immune to label distribution, and compensates the problem of data scarcity for tail classes. Our framework can be built on top of semi-supervised learning methods without much extra overhead, leading to an improved approach RoLT+. The proposed methods achieve strong empirical performance on benchmark and real-world datasets.

2 Related Work

Our work is closely related to the following directions.

Long-Tailed Learning. Recently, many approaches have been proposed to cope with long-tailed learning. Most extant approaches can be categorized into three types by modifying (i) the inputs to a model by re-balancing the training data Shen et al. (2016); Liu et al. (2019); Zhou et al. (2020); (ii) the outputs of a model, for example by post-hoc adjustment of the classifier Kang et al. (2020); Tang et al. (2020); Menon et al. (2021); and (iii) the internals of a model by modifying the loss function Cao et al. (2019); Shu et al. (2019); Jamal et al. (2020); Ren et al. (2020). 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 under label noise. Although the presence of label noise in class-imbalanced dataset has also been mentioned in

HAR Cao et al. (2021), 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 long-tailed learning methods. More importantly, many existing methods can be easily integrated into our framework, leading to noticeable performance improvement.

Label Noise Detection. Plenty of methods have been proposed to detect noisy examples Jiang et al. (2018); Han et al. (2018); Li et al. (2020); Nguyen et al. (2020). Many works adopt the small-loss trick, which treats examples with small training losses as correctly-labeled. In particular, MentorNet Jiang et al. (2018) reweights samples with small loss so that noisy samples contribute less to the loss. Co-teaching Han et al. (2018) trains two networks where each network selects small-loss samples in a mini-batch to train the other. DivideMix Li et al. (2020) fits a Gaussian mixture model on per-sample loss distribution to divide the training data into clean set and noisy set. In addition, AUM Pleiss et al. (2020) 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 Li et al. (2021) 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-independent prototypical noise detection method that works well in long-tailed learning.

3 Robust Long-Tailed Learning under Label Noise

In this section, we first introduce the problem setting. Then, we present our method for long-tailed learning under label noise.

3.1 Problem Formulation

Given a training dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i$ is an instance feature vector and $y_i \in \mathcal{C} = [K] = \{1, \dots, K\}$ is the class label assigned to it. We assume that 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 $y_i^* \in \mathcal{C}$. The setting of long-tailed learning is where the class prior distribution $\mathbb{P}(y)$ is highly skewed, so that many tail labels have a very low probability of occurrence. Specifically, we define the imbalance ratio (IR) as $\rho = \max_y \mathbb{P}(y) / \min_y \mathbb{P}(y)$.

In practice, since the data distribution is not known, Empirical Risk Minimization (ERM) uses the training data to achieve an empirical estimate of the data distribution. Typically, one minimizes the softmax cross-entropy as

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

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. In the following, we adapt the ERM to address these problems without introducing much extra training efforts.

3.2 Class-Independent Prototypical Noise Detection

In this work, we find that this commonly used method does not fit well with long-tailed learning. The reason is that the training loss of an example can also be large because it belongs to the tail classes and the small-loss trick is not able to distinguish mislabeled examples from tail classes examples. In contrast, we show that the estimate of class prototypes is robust to label noise and can be used to detect noisy labels.

Considering the discrepancy of data distribution of each class, we propose to inspect the distance statistics in a *class-independent* manner. 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 Goldberger et al. (2004); Samuel & Chechik (2021). Here, $dist$ is a distance measure in the embedding space and is typically set to be the Euclidean distance. 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, Medium, Few, and All classes in Table 1. The experiment is done on CIFAR-100 with imbalance ratio $\rho = 100$, noise ratio $\gamma = 0.2$ and $\gamma = 0.5$. It can be seen that the AUC is high even for tail classes, indicating the effectiveness of distance measure at distinguishing clean and noisy examples.

$\gamma = 0.2$	Many	Medium	Few	All	$\gamma = 0.5$	Many	Medium	Few	All
	95.16	93.38	82.81	90.64		92.00	87.43	73.60	85.20

Table 1: Average AUC for Many, Medium, Few, and All classes.

To separate clean examples from noisy data, we assume that, for training examples of class k, the distance statistics follow a mixture of two Gaussians Permuter et al. (2006), 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 class-independent 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

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

where $f_\theta(\boldsymbol{x})$ denotes the extracted feature representation of $\boldsymbol{x}$. Based on $\boldsymbol{c}_k$, the distances between $\boldsymbol{c}_k$ and examples of class k are obtained by

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

We then fit a two-component Gaussian mixture model to maximize the log-likelihood value by $\max \sum_{i=1}^{|\mathcal{D}_k|} \log(\sum_{j=1}^{2} \phi_j \mathbb{P}(d_i \mid \mu_j, \sigma_j))$, where $d_i = dist(\boldsymbol{c}_k, \boldsymbol{x}_i)$ 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 Li et al. (2020). 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, we believe that the obtained $\mathcal{X}_k$ retains most of correctly-labeled examples of class k as well as less mislabeled examples.

3.3 Soft Pseudo-Labeling via Label Smoothing and Label Guessing

For each noisy example, we aims to refine its training label by generating a soft pseudo-label. A direct approach is to leverage the prediction of ERM model. However, the ERM is known to be biased toward head classes Zhong et al. (2021). Hence, refining noisy labels using the predictions of ERM may be sub-optimal for examples of tail classes. In contrast, the NCM classifier can yield balanced classification boundary Kang et al. (2020). 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 Zhong et al. (2021).

Algorithm 1: Robust Long-Tailed Learning under Label Noise (ROLT)

Input: training dataset $\{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$, initial learning rate η_0, number of warm-up iterations T_0
// Warm-up Stage: run SGD for T_0 iterations
for $t = 1, \ldots, T_0$ **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 $t = 1, \ldots, T$ **do**
  $\mathcal{X} = \varnothing, \mathcal{S} = \varnothing$
  for $k = 1, \ldots, K$ **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 $\boldsymbol{c}_k \leftarrow \text{Normalize}\left(\frac{1}{|\mathcal{X}_k|} \sum_{i \in \mathcal{X}_k} f_\theta(\boldsymbol{x}_i)\right)$ // In practice, class prototype computed from $\mathcal{X}_k$ is more accurate than that from $\mathcal{D}_k$
  end
  Compute soft pseudo-labels $\tilde{\boldsymbol{y}}$ for $\boldsymbol{x} \in \mathcal{S}$ 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 η : $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$
end

Put 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 label y, we form the guessing label set $\mathcal{G} = \{\hat{y}^{erm}, \hat{y}^{ncm}, y\}$ and generate soft pseudo-label $\tilde{\boldsymbol{y}} \in \mathbb{R}^K$ of $\boldsymbol{x} \in \mathcal{S}$ as follows. For class $k \in [K]$, 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. 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 Laine & Aila (2017) 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

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

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

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

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. Moreover, $\mathcal{X}$ and $\mathcal{S}$ can be viewed as labeled and unlabeled data respectively, and semi-supervised learning Berthelot et al. (2019); Li et al. (2020) can be leveraged to train networks, which is further validated in experiments.

4 EXPERIMENTS

We now present experiments that confirm our main claims: (i) on benchmark datasets, we demonstrate the efficacy of our method by comparing with both methods for long-tailed learning and learning with noisy labels; (ii) on a real-world dataset with natural label noise, we compare with existing methods by controlling the imbalance ratio; (iii) we provide detailed studies for each proposed component in our framework and analyze their effectiveness.

4.1 SIMULATING NOISY AND LONG-TAILED DATASETS ON CIFAR

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 Cao et al. (2019). The long-tailed imbalance follows an exponential decay in sample sizes across different classes. We then inject label noise according to the noise transition matrix equation 8 to the long-tailed dataset to form the training set. In particular, we consider imbalance ratio to be $\rho \in \{10, 50, 100\}$ and noise level to be $\gamma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Due to space constraints, we defer the results for $\gamma = 0$ and $\rho = 50$ to the supplementary material.

Label Noise Generation. We propose a new label noise generation method. To generate noisy labels, the most basic idea is to utilize the noise transition matrix Liu & Tao (2016), 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/feature, 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 present a new noise generation approach by setting $T(X = x)$ according to the estimated class priors $\mathbb{P}(y)$, e.g., the empirical class frequencies in the training dataset. Formally, given the noise proportion $\gamma \in [0, 1]$, we define

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

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, we believe that examples are more likely to be mislabeled as frequent ones in real-world situations.

Result. Table 2 and Table 3 summarize the results for CIFAR-10 and CIFAR-100. We compare our methods with several commonly used baselines for long-tailed learning and learning with noisy labels. As shown in the results, previous methods dreadfully degrade their performance as the noise level and imbalance ratio increase, while our methods retain robust performance. In particular, compared with ERM, 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) Cao et al. (2019) enhances the performance by favoring the tail classes. Note that, ERM-DRW achieves even lower accuracy than LDAM Cao et al. (2019) in many cases, while ROLT-DRW outperforms LDAM-DRW by 5% on average. This clearly demonstrates the importance of correcting noisy labels in the training data. Intriguingly, our experiments reveal that NCM Kang et al. (2020), which is mostly overlooked in previous literature on long-tailed learning, performs better than cRT Kang et al. (2020) in most cases, especially in scenarios with high noise levels. This also provides evidence for us to develop geometry-based noise detection method.

We further compare ROLT+ with DivideMix Li et al. (2020), one of the most popular methods for learning with noisy labels. We use the same experimental setups for these two methods. The results are given in Table 3. It can be observed that DivideMix performs worse as the training dataset becomes more class-imbalanced. In contrast, our method ROLT+ achieves an improvement in test accuracy by 2.57% on average. This validates the superiority of our prototypical noise detector over the small-loss trick. In the supplementary material, we further show that DivideMix flags most example of tail classes as noisy, which is the main reason accounting for its failure.

4.2 EVALUATION ON REAL-WORLD CLASS-IMBALANCED AND NOISY DATASET

We test the performance of our method on a real-world dataset. WebVision Li et al. (2017) contains 2.4 million images collected from Flickr and Google with real noisy and class-imbalanced

CIFAR-10										
Imbalance Ratio	10					100				
Noise Level	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
ERM	80.41	75.61	71.94	70.13	63.25	64.41	62.17	52.94	48.11	38.71
ERM-DRW	81.72	77.61	71.94	70.13	63.25	66.74	62.17	52.94	48.11	38.71
LDAM	84.59	82.37	77.48	71.41	60.30	71.46	66.26	58.34	46.64	36.66
LDAM-DRW	85.94	83.73	80.20	74.87	67.93	**76.58**	72.28	66.68	57.51	43.23
BBN	83.59	80.35	72.94	70.04	63.63	70.05	64.51	56.86	44.30	36.72
cRT	80.22	76.15	74.17	70.05	64.15	61.54	59.92	54.05	50.12	36.73
NCM	82.33	74.73	74.76	68.43	64.82	68.09	66.25	60.91	55.47	42.61
HAR-DRW	84.09	82.43	80.41	77.43	67.39	70.81	67.88	48.59	54.23	42.80
RoLT	86.18	85.03	83.53	**81.53**	76.72	72.38	71.83	68.15	59.80	49.62
RoLT-DRW	**86.27**	**85.04**	**83.58**	81.40	**77.11**	75.33	**73.84**	**70.21**	**64.99**	**55.32**
CIFAR-100										
Imbalance Ratio	10					100				
Noise Level	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
ERM	48.65	43.27	37.43	32.94	26.92	31.81	26.21	21.79	17.91	14.23
ERM-DRW	50.38	45.24	39.02	34.78	28.50	34.49	28.67	23.84	19.47	14.76
LDAM	51.77	48.14	43.27	36.66	29.62	34.77	29.70	25.04	19.72	14.19
LDAM-DRW	54.01	50.44	45.11	39.35	32.24	37.24	32.27	27.55	21.22	15.21
BBN	53.50	47.91	42.81	35.17	28.60	34.39	27.84	23.38	18.20	15.47
cRT	49.13	42.56	37.80	32.18	25.55	32.25	26.31	21.48	20.62	16.01
NCM	50.76	45.15	41.31	35.41	29.34	34.89	29.45	24.74	21.84	16.77
HAR-DRW	51.04	46.24	41.23	37.35	31.30	33.21	26.29	22.57	18.98	14.78
RoLT	54.61	51.83	47.43	42.46	37.58	35.30	31.10	27.72	24.80	19.56
RoLT-DRW	**55.68**	**53.41**	**48.77**	**44.18**	**39.22**	**37.70**	**33.36**	**30.02**	**26.46**	**20.61**

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

		CIFAR-10						CIFAR-100					
Noise Level		0.2			0.5			0.2			0.5		
Imbalance Ratio		10	50	100	10	50	100	10	50	100	10	50	100
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.

data. Following previous literature, we train on a subset, mini WebVision, which contains the first 50 classes. In Table 4, we report results comparing against state-of-the-art approaches, including D2L Ma et al. (2018), MentorNet Jiang et al. (2018), Co-teaching Han et al. (2018), Iterative-CV Chen et al. (2019), HAR Cao et al. (2021), and DivideMix Li et al. (2020).

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 5. From the results, we can see that the superiority of our method is more significant as the imbalance ratio increases.

4.3 Further Analysis and Ablation Studies

We study the effectiveness of the two main modules of our method.

Efficacy of the noise detector. To further support our motivation, we compare the performance of the ERM and NCM classifiers in Figure 3. It can be seen that NCM produces more balanced recall

		D2L	MentorNet	Co-teaching	Iterative-CV	HAR	DivideMix	**RoLT+**
Webvision	top1	62.68	63.00	63.58	65.24	75.5	77.32	**77.64**
	top5	84.00	81.40	85.20	85.34	90.7	91.64	**92.44**
ImageNet	top1	57.80	57.80	61.48	61.60	70.3	**75.20**	74.64
	top5	81.36	79.92	84.70	84.98	90.0	90.84	**92.48**

Table 4: Accuracy (%) on mini WebVision and ImageNet validation sets.

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 5: Top-1 (Top-5) test accuracy on mini Webvision and ImageNet.

across classes, while ERM tends to predict examples as head classes, resulting in low recall for tail classes. Figure 4 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∼100 images), and few (less than 20 images) shots; and CIFAR-10 by: many ($\{0, 1\}$), medium($\{2, \ldots, 6\}$), 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 $\gamma = 0.3$.

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:

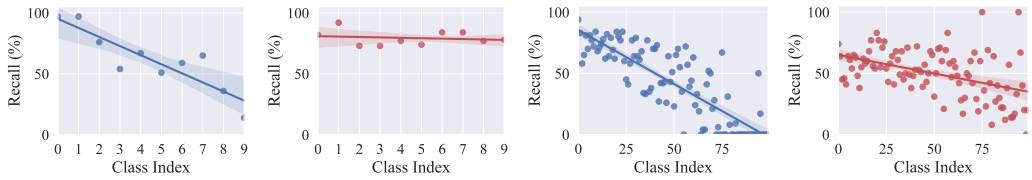
$$\tilde{y}_k = \begin{cases} \frac{1}{3}\sum_{\hat{y}\in\mathcal{G}} \mathbb{I}(\hat{y} = k) & \text{if } k \in \mathcal{G} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We report the results in Table 6 with respect to noise level $\gamma \in \{0.2, 0.5\}$ 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.

4.4 Discussion and Limitations

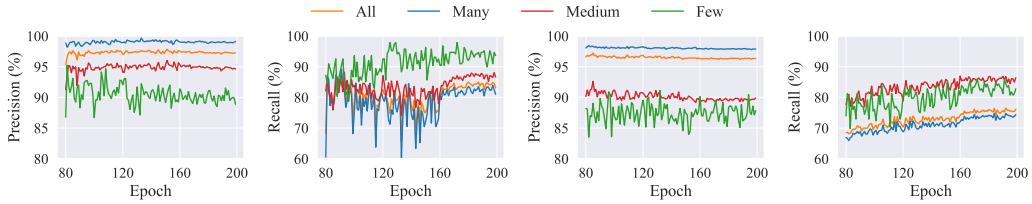
One may be interested in combining the proposed method RoLT with other loss functions. In particular, we attempt to optimize LDAM loss Cao et al. (2019) during training and the results are reported in the supplementary material. Indeed, LDAM encourages the model to yield balanced classification boundaries. 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 long-tailed learning. 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



(a) ERM on CIFAR-10 (b) NCM on CIFAR-10 (c) ERM on CIFAR-100 (d) NCM on CIFAR-100

Figure 3: Per-class recall of ERM and NCM classifiers on CIFAR-10 and CIFAR-100 datasets. It can be clearly seen that NCM produces more balanced predictions than ERM across classes.



(a) CIFAR-10 Precision (b) CIFAR-10 Recall (c) CIFAR-100 Precision (d) CIFAR-100 Recall

Figure 4: Precision and Recall of selected clean examples by our method.

DRW	Classifier	Pseudo-Label	$\gamma = 0.2$				$\gamma = 0.5$			
			Many	Med.	Few	All	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 6: Ablation studies on pseudo-labeling. Test accuracy on CIFAR-100 is reported.

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

We study the long-tailed learning under label noise and a robust framework is proposed to tackle this challenging problem. We reveal the failure of small-loss trick in long-tailed learning, and establish a prototypical noise detection method that is immune to label distribution. We provide systematic studies on benchmark and real-world datasets to verify the superiority of our methods by comparing to state-of-the-art methods in the strands of long-tailed learning and learning with noisy labels.

ETHICS STATEMENT

This paper introduces a method to learning from noisy and long-tailed data. It can benefit the widespread use of "weakly-labeled" data Li et al. (2017; 2020; 2021), which are often cheap to acquire but have suffered from data quality issues. The proposed method is simple yet effective, which we believe will broadly benefit practitioners dealing with heavily imbalanced data in realistic applications.

In this work, we only extensively test our strategies on benchmark datasets. In many real-world applications such as autonomous driving, medical diagnosis, and healthcare, beyond being naturally noisy and imbalanced, the data may impose additional constraints on learning process and final models, e.g., being fair or private. We focus on standard accuracy as our measure and largely ignore other ethical issues in imbalanced data, especially in minor classes. As such, the risk of producing unfair or biased outputs reminds us to carry rigorous validations in critical, high-stakes applications.

REPRODUCIBILITY STATEMENT

We elaborate the implementation details in Section A. Our anonymous source code can be found in supplementary materials.

REFERENCES

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pp. 1565–1576, 2019.

Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. In *ICLR*, 2021.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pp. 9268–9277, 2019.

Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *NeurIPS*, pp. 513–520, 2004.

Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, pp. 5356–5364, 2019.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pp. 8536–8546, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pp. 8769–8778, 2018.

Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *CVPR*, pp. 7610–7619, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pp. 2304–2313, 2018.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Junnan Li, Richard Socher, and Steven CH Hi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

Junnan Li, Caiming Xiong, and Steven CH Hoi. Mopro: Webly supervised learning with momentum prototypes. In *ICLR*, 2021.

Wen Li, Li Niu, and Dong Xu. Exploiting privileged information from web data for image categorization. In *ECCV*, pp. 437–452, 2014.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *CoRR*, abs/1708.02862, 2017.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE TPAMI*, 38(3):447–461, 2016.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pp. 2537–2546, 2019.

Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pp. 3355–3364, 2018.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi - Phuong - Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. SELF: learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

Haim H. Permuter, Joseph M. Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4): 695–706, 2006.

Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020.

Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. *CoRR*, abs/2104.03066, 2021.

Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, volume 9911, pp. 467–482, 2016.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, pp. 1917–1928, 2019.

Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, pp. 11659–11668, 2020.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *ICLR*, 2021.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, pp. 7029–7039, 2017.

Tong Wei and Yu-Feng Li. Does tail label help for large-scale multi-label learning? *IEEE Transaction Neural Networks Learning Systems*, 31(7):2315–2324, 2020.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, volume 12349, pp. 162–178, 2020.

Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. In *CVPR*, 2021.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_DMI: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019.

Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual T: reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*, 2020.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, 2021.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, pp. 9716–9725, 2020.

A Implementation Details

We develop our core algorithm in PyTorch.

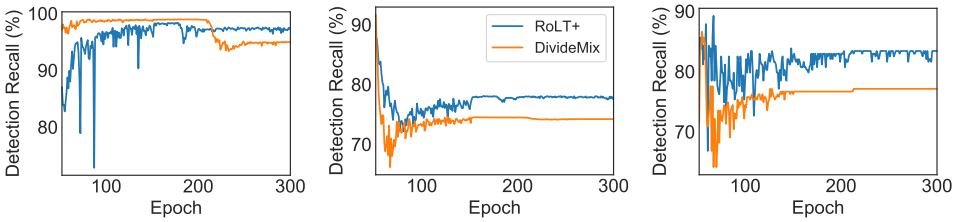
Implementation details for CIFAR. We follow the simple data augmentation used in He et al. (2016) with only random crop and horizontal flip. For experiments of RoLT, we use ResNet-32 as the backbone network and train it using standard SGD with a momentum of 0.9, a weight decay of 2×10^{-4}, a batch size 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 Li et al. (2020), 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.

Implementation details for mini WebVision. Following previous work Li et al. (2020), 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.

B Additional Experimental Results

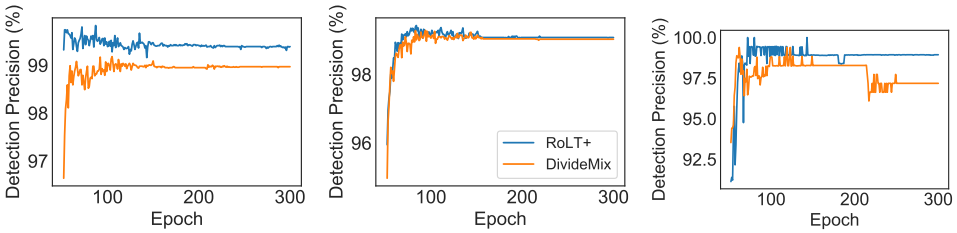
B.1 Comparison with DivideMix with respect to Noise Detection

To further demonstrate our proposed noise detection that is tailored for long-tailed learning, we compare it with DivideMix and the results are shown in Figure 5$\sim$8. This experiment is conducted under imbalance ratio $\rho = 100$ and noise level $\gamma = 0.2$. 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-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.



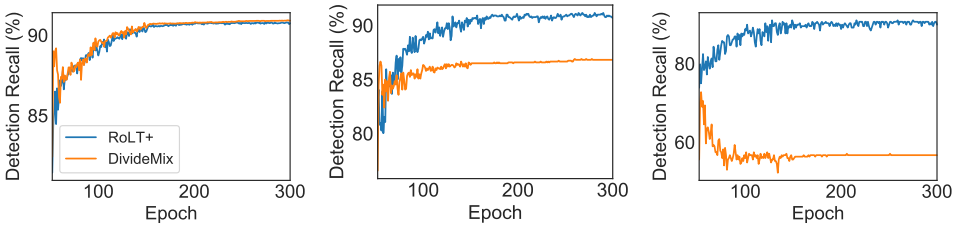
(a) CIFAR-10 Many Shot (b) CIFAR-10 Medium Shot (c) CIFAR-10 Few Shot

Figure 5: Comparison of detection recall between RoLT+ and DivideMix on CIFAR-10.



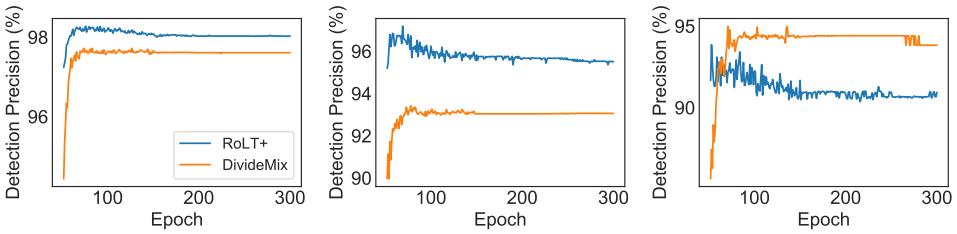
(a) CIFAR-10 Many Shot (b) CIFAR-10 Medium Shot (c) CIFAR-10 Few Shot

Figure 6: Comparison of detection precision between RoLT+ and DivideMix on CIFAR-10.



(a) CIFAR-100 Many Shot (b) CIFAR-100 Medium Shot (c) CIFAR-100 Few Shot

Figure 7: Comparison of detection recall between RoLT+ and DivideMix on CIFAR-100.



(a) CIFAR-100 Many Shot (b) CIFAR-100 Medium Shot (c) CIFAR-100 Few Shot

Figure 8: Comparison of detection precision between RoLT+ and DivideMix on CIFAR-100.

B.2 Comparison with DivideMix on Balanced Datasets

We compare the performance of our method with DivideMix on balanced datasets with noise level $\rho \in \{0.2, 0.5\}$. The results are reported in Table 7 and our method is comparable with DivideMix. This shows that the proposed prototypical noise detector also works well on balanced datasets.

		CIFAR-10		CIFAR-100	
Noise Level		0.2	0.5	0.2	0.5
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 7: Test accuracy (%) on class-balanced CIFAR datasets with different noise level.

	CIFAR-10					CIFAR-100				
Noise Level	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
ERM	69.33	63.60	58.69	55.85	43.38	35.10	30.27	25.11	19.49	16.97
ERM-DRW	71.99	65.76	58.69	55.85	43.38	37.74	32.63	27.19	21.43	17.52
LDAM	75.06	71.34	64.71	54.42	42.95	39.94	33.43	30.01	23.30	17.51
LDAM-DRW	79.01	76.41	71.83	62.22	48.88	**42.88**	36.60	33.12	25.91	19.48
BBN	72.86	68.01	60.49	52.89	46.22	39.40	36.48	26.89	21.08	16.77
cRT	69.22	65.02	60.64	51.90	43.26	35.70	30.23	24.37	19.90	17.47
NCM	72.37	69.60	65.26	56.78	49.68	38.91	33.49	28.85	23.91	19.01
HAR-DRW	72.12	67.44	60.73	63.04	52.35	38.46	28.86	29.33	22.06	16.75
RoLT	77.51	75.80	71.74	63.07	55.38	40.52	36.28	31.58	28.54	24.25
RoLT-DRW	**80.16**	**77.86**	**74.47**	**67.83**	**60.15**	42.62	**38.94**	**33.57**	**30.78**	**25.51**

Table 8: Test accuracy (%) on CIFAR datasets with imbalance ratio $\rho = 50$ and different noise level.

B.3 Additional Results on CIFAR Datasets

We report the results on CIFAR-10 and CIFAR-100 with simulated imbalance ratio $\rho = 50$ with noise level $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ in Table 8. The performance of comparison methods is in line with that of $\rho = 10$ and $\rho = 100$ which are reported in the main text. This further justifies that our method can adapt to various class-imbalanced and noisy datasets.

B.4 Results on Clean CIFAR Datasets

Although our method is particularly designed for long-tailed learning with noisy labels, it is interesting to study its performance on clean datasets. We report the results in Table 9. Intriguingly, RoLT consistently outperforms vanilla ERM 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. This validates the robustness of our method, which does not hurt the performance in the corner case.

B.5 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 long-tailed learning, 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.

	CIFAR-10			CIFAR-100		
Imbalance Ratio	10	50	100	10	50	100
ERM	86.75	77.38	71.83	56.31	44.15	38.88
ERM-DRW	87.71	80.58	76.33	57.68	46.71	41.90
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
NCM	**88.14**	82.75	79.59	56.05	45.13	41.73
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

Table 9: Test accuracy (%) on clean CIFAR datasets with different imbalanced ratio.

DRW	Classifier	Pseudo-Label	$\gamma = 0.2$				$\gamma = 0.5$			
			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.

B.6 The Impact of Label Noise on Representation and Classifier Learning

In Table 11$\sim$14, we study the impact of label noise for two-stage long-tailed learning 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 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.

	$\rho = 1$						$\rho = 10$						$\rho = 100$					
	γ_c						γ_c						γ_c					
γ_r	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5
0	93.15	92.85	92.76	92.55	92.56	92.40	86.78	85.90	85.43	85.21	83.13	81.49	71.18	68.59	66.31	65.92	61.83	57.58
0.1	91.43	91.37	91.36	91.31	91.33	91.41	81.13	80.22	78.84	77.60	77.05	75.13	62.48	61.54	59.91	58.70	55.57	53.17
0.2	90.40	90.42	90.31	90.33	90.35	90.24	76.91	76.48	76.15	75.09	75.20	73.66	61.33	60.18	59.92	57.98	56.34	52.82
0.3	88.74	88.80	88.77	88.58	88.73	88.54	75.64	74.60	74.36	74.17	72.76	71.17	55.26	55.05	53.79	54.05	50.45	47.74
0.4	87.00	86.91	86.82	86.89	86.85	86.75	72.26	71.61	70.95	69.96	70.05	67.83	51.98	51.22	51.05	50.36	50.12	46.28
0.5	84.57	84.53	84.46	84.38	84.29	83.95	67.01	67.04	66.83	64.68	64.16	64.15	41.70	40.90	40.75	40.07	38.61	36.73

Table 11: 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					
γ_r	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5
0	92.77	92.75	92.69	92.67	92.55	92.54	88.14	88.08	87.97	87.89	87.72	87.45	79.59	79.64	79.67	79.64	79.57	78.63
0.1	91.29	91.29	91.28	91.31	91.25	91.24	82.23	82.33	82.09	82.05	81.91	81.91	68.21	68.09	67.06	66.19	65.35	64.53
0.2	90.20	90.24	90.23	90.26	90.31	90.24	75.27	75.02	74.73	74.37	73.82	73.25	66.80	66.59	66.25	65.98	64.95	63.70
0.3	88.51	88.51	88.48	88.55	88.53	88.53	74.99	75.01	74.98	74.76	74.52	74.09	61.68	61.22	61.06	60.91	60.04	59.19
0.4	86.77	86.80	86.78	86.80	86.79	86.76	70.45	69.75	69.40	69.07	68.43	67.97	56.57	56.46	56.21	55.92	55.47	54.60
0.5	83.78	83.78	83.78	83.77	83.79	83.77	66.16	65.82	65.62	65.40	65.07	64.82	44.66	44.08	43.98	43.18	43.10	42.61

Table 12: 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					
γ_r	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5
0	69.73	68.90	68.09	67.49	66.61	65.70	56.62	53.55	52.21	50.52	49.09	47.34	39.44	35.43	33.93	32.34	31.32	29.68
0.1	68.55	68.03	67.38	66.58	66.48	65.87	50.55	49.13	47.89	46.23	44.55	43.36	33.19	32.25	30.69	28.76	27.61	25.97
0.2	65.51	65.21	64.86	64.45	64.46	63.96	45.31	44.22	42.56	41.73	40.27	38.61	27.77	27.02	26.31	24.57	23.79	22.82
0.3	63.01	62.74	62.32	62.02	61.65	60.96	41.72	40.82	39.77	37.80	37.84	36.38	24.91	23.83	23.61	21.48	21.28	19.61
0.4	60.78	60.42	60.30	59.73	59.19	58.93	37.33	36.76	35.31	34.46	32.18	32.68	23.02	22.38	22.04	21.49	20.62	19.48
0.5	57.88	57.51	56.98	56.83	55.97	55.14	32.07	31.09	30.29	29.90	28.58	25.55	19.05	18.60	17.93	17.67	16.89	16.01

Table 13: 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					
γ_r	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5
0	66.94	66.71	66.37	65.79	64.84	64.07	56.05	55.63	55.22	54.14	53.18	51.78	41.73	41.18	40.59	39.81	38.56	37.95
0.1	65.72	65.84	65.66	65.27	64.83	64.16	50.49	50.76	50.14	49.53	49.51	48.16	35.43	34.89	34.49	33.77	32.93	32.11
0.2	63.08	62.92	63.26	62.61	62.67	62.15	45.22	45.05	45.15	44.83	44.21	43.22	30.47	29.95	29.45	28.74	28.56	28.13
0.3	60.82	60.64	60.62	60.81	60.29	60.16	41.82	41.66	41.23	41.31	40.27	39.68	25.97	25.50	25.17	24.74	23.96	22.59
0.4	57.87	58.00	57.81	57.82	57.91	57.55	36.13	36.32	36.19	35.81	35.41	34.84	23.89	23.47	22.80	22.29	21.84	20.50
0.5	55.24	55.25	55.05	55.01	54.64	54.95	30.85	30.63	30.50	30.07	29.84	29.34	19.16	18.63	18.47	18.15	16.89	16.77

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