ProMix: Combating Label Noise via Maximizing Clean Sample Utility

Ruixuan Xiao¹, Yiwen Dong¹, Haobo Wang¹*, Lei Feng², Runze Wu³, Gang Chen¹ and Junbo Zhao¹
¹Zhejiang University, Hangzhou, China
²Nanyang Technological University, Singapore
³NetEase Fuxi AI Lab, Hangzhou, China
{xiaoruixuan, dyw424, wanghaobo, cg, j.zhao}@zju.edu.cn, lfengqaq@gmail.com, wurunze1@corp.netease.com

1 Introduction

The great success of deep neural networks (DNNs) attributes to massive and accurately-labeled training data, which is notoriously expensive and time-consuming to obtain. As an alternative, online queries [Blum et al., 2003] and crowdsourcing [Yan et al., 2014] have been widely used to enhance the labeling efficiency. However, due to the lack of expertise, such annotation processes inevitably introduce wrong labels to models and result in performance degradation. To this end, learning with noisy labels (LNL) [Zhang et al., 2017], which aims to mitigate the side-effect of label noise, has attracted huge attention from the community.

*Corresponding Author.
great potential for performance enhancement.

Motivated by this, we present a novel framework ProMix which comprehensively investigates the opportunities and challenges of maximizing the utility of clean samples in LNL. To achieve this, we first propose a progressive selection mechanism that dynamically expands a base class-wise selection set by collecting those having high prediction scores on the observed labels. Empirically, such a simple modification allows us to reach an extraordinary trade-off between the quality and the quantity of the clean set. As shown in Figure 1a, compared with DivideMix [Li et al., 2020a], our progressive selection strategy selects much more clean samples while rarely introducing detested noise samples. Moreover, a simple baseline that is trained merely on such selected clean samples establishes the state-of-the-art performance on the real-world noisy CIFAR-N [Wei et al., 2022b] benchmark, and incorporating semi-supervised learning can further boost the performance. These results clearly validate the importance of well-utilizing clean samples.

Despite the promising results, this maximal selection procedure inevitably causes side effects for the SSL as well. First, the clean samples may not be uniformly distributed across different labels, leading to a biased label distribution. Second, the selection and joint semi-supervised learning procedures are tightly interdependent, which may suffer from the confirmation bias stemming from its own hard-to-correct errors. To mitigate these problems, we develop a debiased semi-supervised training framework that comprises two key components. The first one is an auxiliary pseudo head that decouples the generation and utilization of unreliable pseudo-labels so as to alleviate the model from confirming its error recursively. Second, we incorporate a calibration algorithm that simultaneously rectifies the skewed pseudo-labels and the biased cross-entropy loss. As depicted in Figure 1b, integrated with debiased SSL, ProMix is able to generate pseudo-labels with higher quality on the unselected samples, thus facilitating the model training. By integrating the two key components into a whole framework, our proposed ProMix framework establishes state-of-the-art performance over various LNL benchmarks, even in the imbalanced LNL scenario (see Appendix B'). For example, on the CIFAR-10N datasets with real-world noisy labels, ProMix outperforms the best baseline by 2.04%, 2.11%, and 3.10% under Aggregate, Rand1, and Worst label annotations respectively.

2 Related Work

Learning with Noisy Labels. Existing methods for LNL can be roughly categorized into two types. Earlier LNL approaches focus on loss correction. Since the vanilla cross-entropy (CE) loss has been proved to easily overfit corrupted data [Zhang et al., 2017], a vast amount of robust losses has been designed to achieve a small and unbiased risk for unobservable clean data, including robust variants of the CE loss [Wang et al., 2019; Xu et al., 2019; Ma et al., 2020] or sample reweighted losses [Shu et al., 2019; Zheng et al., 2021]. Some other methods [Patrini et al., 2017; Xia et al., 2019; Zhu et al., 2022] consider the noise transition matrix and depend on the estimated noise transition matrix to explicitly correct the loss function, but the transition matrix could be hard to estimate in the presence of heavy noise and a large number of classes.

The latter strand of LNL methods tries to filter out a clean subset for robust training [Li et al., 2020a; Karim et al., 2022; Cordeiro et al., 2023]. Early methods [Han et al., 2018; Jiang et al., 2018; Wei et al., 2020] typically adopt a peer network and perform small loss selection that is motivated by the memorization effect of DNNs [Zhang et al., 2017]. The seminal work DivideMix [Li et al., 2020a] further incorporates MixMatch [Berthelot et al., 2019] to leverage the unselected examples and obtain extraordinary performance. This paradigm is then followed by the most recent LNL approaches [Li et al., 2020b; Liu et al., 2020; Bai et al., 2021; Karim et al., 2022]. Recent LNL works also incorporate contrastive learning methods [Ortego et al., 2021; Yao et al., 2021]. Despite the promising, existing selection-based methods mostly select a mediocre clean set with medium size and restricted budget, which overlooks a flurry of clean samples and naturally exhibits a quality-quantity trade-off. A too-small budget leads to inadequate clean sample utility, while a too-large budget enrolls many unexpected noisy examples. To cope with this, our work targets to maximize the utility of clean samples to unleash their great potential.

Debiased Learning. Debiased learning has gained huge attention since the source dataset and algorithms can be naturally biased [Kim et al., 2019; Nam et al., 2020]. For example, in long-tailed learning, models can be easily biased towards dominant classes due to the distribution bias in source data [Zhang et al., 2021a]. As for semi-supervised learning, besides the distribution bias in source data and pseudo-labels [Wang et al., 2022], confirmation bias caused by inappropriate training with unreliable pseudo-labels may jeopardize generalization performance [Chen et al., 2022]. Debiased learning has also been a prevailing topic discussed in some other research topics like contrastive sampling [Chuang et al., 2020] and OOD detection [Ming et al., 2022]. Despite the surge in other areas, the bias in LNL still remains underexplored. In this work, we attempt to alleviate the biases in our selection and pseudo-labeling procedures. We hope our work will inspire future works to integrate this important component in advanced LNL frameworks.

3 Proposed Method

Problem Formulation. For a multi-class classification task with C classes, we denote the noisy training dataset with n examples by \( D = \{(x_i, \tilde{y}_i)\}_{i=1}^n \), where each tuple comprises of an image \( x_i \in \mathcal{X} \subset \mathbb{R}^d \) and the corresponding noisy label \( \tilde{y}_i \in \mathcal{Y} = \{1,...,C\} \). In contrast to the supervised learning setup, the ground-truth labels are not observable. The goal of LNL is to obtain a functional mapping \( f_{\theta,\phi} : \mathcal{X} \to \tilde{\mathcal{Y}} \) that takes an example \( x_i \) as the input and outputs the logits \( f_{\theta,\phi}(x_i) \) for predicting the true labels. This classification model \( f_{\theta,\phi} = h_{\phi} \circ g_{\theta} \) consists of a feature extractor \( g_{\theta} \) and a classification head \( h_{\phi} \), which are parameterized by \( \theta \) and \( \phi \) respectively. In our work, we adopt the cross-entropy loss \( \mathcal{L} = -\sum_{c=1}^{C} \tilde{y}_ic \log p_{ic} \) where \( p_{ic} = \text{softmax}(f(x_i)) \) is the
probabilistic output of \( f(x_i) \). \( y^j_i \) is the \( j \)-th entry of \( p_i \) and \( \hat{y}_i^j \) is the \( j \)-th entry of one-hot encoded label vector \( \hat{y}_i \).

Similar to previous works, our framework also divides the samples into a clean set \( D_1 \) and a noisy set \( D_u \), which are then applied for semi-supervised learning. The vanilla classification losses for these two parts are formulated:

\[
\begin{align*}
L_x(D_1) &= \frac{1}{|D_1|} \sum_{x_i \in D_1} \mathcal{H}(\hat{y}_i, p_i) \\
L_u(D_u) &= \frac{1}{|D_u|} \sum_{x_i \in D_u} \mathcal{H}(\text{Sharpen}(p_i, T), p_i)
\end{align*}
\]

(1)

where \( \text{Sharpen}(p_i, T) = \frac{(p_i^c)^{2}}{\sum_{c=1}^{C}(p_i^c)^{2}} \) for \( c = 1, 2, \ldots, C \). An overview of the proposed method is illustrated in Figure 2.

### 3.1 Progressive Sample Selection

As aforementioned, we would like to excessively collect clean samples. To achieve this, we first draw inspiration from the classical learning theory, Theorem 1. [Shalev-Shwartz and Ben-David, 2014] Let \( \mathcal{F} \) be the hypothesis set of \( f \). Denote the population risk by \( R(f) = \mathbb{E}[\mathcal{H}] \) and the corresponding empirical risk by \( \hat{R}(f) \). We define the true hypothesis by \( f^* = \arg \min \hat{R}(f) \) and empirical risk minimizer by \( \hat{f} = \arg \min \hat{R}(f) \). Given a dataset containing \( n \) examples, with probability at least \( 1 - \delta \), the following inequality holds:

\[
R(\hat{f}) \leq R(f^*) + 2\hat{R}(\mathcal{H}) + 5\sqrt{\frac{2\log(8/\delta)}{n}}
\]

(2)

where \( \hat{R}(\mathcal{F}) \) is the empirical Rademacher of \( \mathcal{F} \).

Considering a simple case in which \( f \) consists of multiple ReLU-activated fully-connected layers, we can apply the techniques outlined in [Allen-Zhu et al., 2019] to derive an upper bound of the Rademacher complexity \( O(\sqrt{1/n}) \). If we assume a perfectly clean set with \( n_c \) samples, the model has larger than a probability of \( 1 - O(e^{-n_c\epsilon^2/50}) \) to give an \( \epsilon \)-wrong prediction on the remaining dirty training data. Hence, this model is prone to produce a much larger probability on the observed labels for the remaining correctly-labeled data than noisy ones. Motivated by this, we present the following selection procedure for enlarging the base selection set.

**Class-wise Small-loss Selection (CSS).** Based on the small loss criterion, we aim to first filter out a base clean set as the cornerstone. In the training, the patterns of easy classes are usually fitted earlier and the loss values of examples with different observed labels may not be comparable [Gui et al., 2021]. So it would be more appropriate to select examples class by class with the same observed labels. At each epoch, we first split the whole training dataset \( D \) to \( C \) sets according to the noisy labels, i.e. \( S_j = \{(x_i, \hat{y}_i) \in D | \hat{y}_i = j\} \). Given the \( j \)-th set \( S_j \), we calculate the standard CE loss \( l_i \) for each example and select \( k = \min(|\frac{n}{C} R|, |S_j|) \) examples with smallest \( l_i \) to constitute the clean set \( C \), where \( R \) is the filter rate that is identical for all classes. Finally, the integral clean set is merged as \( D_{CSS} = \cup_{j=1}^{C} C_j \). Compared to the original small-loss selection, we relate \( k \) to the average number of examples \( n / C \) such that produce a roughly balanced base set.

**Matched High Confidence Selection (MHCS).** After getting the base clean set, we also introduce another selection strategy to make capital out of the potentially clean samples missed by CSS. Specifically, we calculate the confidence scores \( e_i = \max_{j \neq \hat{y}_i} p_i^j \) for remaining samples. Then, we select those examples whose predictions match the given labels, while their confidence is high enough:

\[
D_{MHCS} = \{(x_i, \hat{y}_i) \in D | e_i \geq \tau, y_i = \hat{y}_i\}
\]

(3)
where \( y_i' = \arg \max_j p_j^i \) is the predicted label of \( x_i \). In practice, we set a high threshold \( \tau \) such that the selected samples have a high probability of being clean. According to our theoretical motivation, the proposed MHCS mechanism is able to automatically explore new clean data to dynamically enlarge the clean set \( D_l = D_{CSS} \cup D_{MHCS} \) by trusting those examples with matched high confidence.

**Remark.** Our MHCS selection strategy draws inspiration from the pseudo-labeling procedure of FixMatch [Sohn et al., 2020]. But, in contrast to the SSL setup, our clean set is dynamically generated. With the naive high confidence selection, the DNNs can assign an arbitrary label with high confidence and select them as clean, which results in a vicious cycle. Thus, we choose those samples to have high confidence in their given labels, which tend to be originally clean.

### 3.2 Debiased Semi-Supervised Training

After sample selection, we would also like to integrate semi-supervised learning that utilizes the remaining noisy samples to boost performance. Nevertheless, our excessive selection procedure inevitably induces unexpected characteristics. First, the selected clean samples may inherently be imbalanced amongst different labels, since some labels are typically more ambiguous than others. Hence, the model may be biased towards the classes that possess more clean samples and in turn pollutes the quality of selection (as shown in Figure 1a). Second, the selection and pseudo-labeling are tightly coupled, which naturally brings about confirmation bias, so we should be particularly careful to train the model with the generated pseudo-labels. To escape such a dilemma, we introduce our debiased SSL framework in what follows.

**Mitigating Confirmation Bias.** In most SSL pipelines, the generation and utilization of pseudo-labels are usually achieved by the same network blocks or between independent peer networks. As a result, the mistakes in pseudo-labels are nearly impossible to self-correct and can accumulate amplifying the existing confirmation bias. In order to alleviate this, we aim to disentangle the generation and utilization of pseudo-labels, which is achieved via introducing an extra classifier termed Auxiliary Pseudo Head (APH) \( h_{AP} \). The primary classifier \( h \) is the task-specific head that generates pseudo-labels for the unlabeled data but is merely optimized with labeled samples in \( D_l \). The newly devised classifier \( h_{AP} \) share the same representations with \( h \) but have independent parameters. It receives the pseudo-labels from \( h \) and is trained with both labeled set \( D_l \) and unlabeled set \( D_u \). Based on Eq. (1), the classification loss is formulated,

\[
\mathcal{L}_{cls} = \mathcal{L}_c(\{\tilde{y}_i, p_i\}) + \mathcal{L}_c(\{\tilde{y}_i, p_i'\}) + \lambda_u \mathcal{L}_u(\{\text{Sharpen}(p_i, T), p_i'\})
\]

where \( p_i \) is the softmax score from \( h \) and \( p_i' \) is that from \( h_{AP} \). \( \lambda_u \) controls the strength of the unsupervised loss. Through this procedure, the backbone benefits from the unlabeled data for enhanced representation via \( h_{AP} \). At the same time, the primary classifier \( h \) remains reliable without seeing unlabeled samples, which reduces the confirmation bias. It is worth noting that the auxiliary pseudo head only participates in training and is discarded at inference time.

**Mitigating Distribution Bias.** In addition to confirmation bias, there may still be distribution bias whose causes are two-fold. First, the selected samples can falsely calibrate the classifier due to skewed label distribution. Second, previous work [Wang et al., 2022] shows that pseudo-labels can naturally bias towards some easy classes even if the training data are curated and balanced, which can iteratively dominate the training process. To learn a well-calibrated classifier, we incorporate a Debiased Margin-based Loss (DML) \( l_{DML} \) to encourage a larger margin between the sample-rich classes and the sample-deficient classes. \( l_{DML} \) can be defined as:

\[
l_{DML} = - \sum_{j=1}^{C} \tilde{y}_i^j \log \frac{e^{f_j^l(x_i) + \alpha \log \pi_j}}{\sum_{k=1}^{C} e^{f_k^l(x_i) + \alpha \log \pi_k}}
\]

where \( \alpha \) is a tuning parameter that controls debiasing strength and \( \pi \) is the underlying class distribution \( \mathbb{P}(y | x) \) indicating which label possesses more samples on the training set. Here we assume \( \pi \) is known and will show how to estimate it later. The standard cross entropy loss \( \mathcal{H} \) is then replaced by \( l_{DML} \).

Similarly, we also resort to Debiased Pseudo Labeling (DPL) following [Menon et al., 2021] to calibrate the pseudo-labels. The refined logit \( \tilde{f}_i \) which is then fed into the softmax layer for subsequent pseudo-label generation is formulated:

\[
\tilde{f}_i = f(x_i) - \alpha \log \pi
\]

That is, we suppress the logits on those easy classes and ensure other classes are fairly learned. The refined pseudo-labels are then used for calculating unlabeled loss in Eq. (4).

Next, we elaborate on our dynamical estimation of \( \pi \). Notably, we employ two different class distribution vectors for the labeled and unlabeled data respectively. Formally, given a mini-batch of samples \( B \) from either \( D_l \) or \( D_u \), we update the estimation of class distribution \( \pi \) in a moving-average style:

\[
\pi = m \pi + (1 - m) \frac{1}{|B|} \sum_{x_i \in B} p_i
\]

where \( m \in [0, 1) \) is a momentum coefficient. When calculating the loss in Eq. (4), we use different estimation \( \pi_l \) and \( \pi_u \) for the two sets. For better understanding of the whole procedure, we refer the readers to Appendix A for more details.

To summarize the pipeline of debiased SSL, the labeled data are fed forward to both \( h \) and \( h_{AP} \) while the unlabeled data are provided with the debiased pseudo-labels by \( h \) as Eq. (6) and then propagated forward through \( h_{AP} \). Finally, the standard cross entropy in Eq. (4) is replaced by \( l_{DML} \). Through the above debiasing procedure, the refined pseudo-labels with higher quality and enhanced tolerance for incorrect pseudo-labels assist in milking unlabeled samples and in return enhance the purity of sample selection.

### 3.3 Practical Implementation

**Label Guessing by Agreement (LGA).** While we still anticipate selecting matched high confidence instances, there are no free labels to be matched on falsely-labeled data whose predictions mismatch the observed labels. To further distill potential “clean” samples from unlabeled data, we perform label guessing to correct noisy labels on the unlabeled set \( D_u \).
Table 1: Accuracy comparisons on CIFAR-10/100 with symmetric (20%-90%) and asymmetric noise (40%). We report both the averaged test accuracy over last 10 epochs and the best accuracy of ProMix. Results of previous methods are the best test accuracies cited from their original papers, where the blank ones indicate that the corresponding results are not provided. **Bold entries** indicate superior results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods \ Noise Ratio</td>
<td>20%</td>
<td>50%</td>
</tr>
<tr>
<td>CE</td>
<td>86.8</td>
<td>79.4</td>
</tr>
<tr>
<td>Co-Teaching+</td>
<td>89.5</td>
<td>85.7</td>
</tr>
<tr>
<td>IoCoR</td>
<td>85.7</td>
<td>79.4</td>
</tr>
<tr>
<td>M-correction</td>
<td>94.0</td>
<td>92.0</td>
</tr>
<tr>
<td>PENCIL</td>
<td>92.4</td>
<td>89.1</td>
</tr>
<tr>
<td>DivideMix</td>
<td>96.1</td>
<td>94.6</td>
</tr>
<tr>
<td>ELR+</td>
<td>95.8</td>
<td>94.8</td>
</tr>
<tr>
<td>LongReMix</td>
<td>96.2</td>
<td>95.0</td>
</tr>
<tr>
<td>MOIT</td>
<td>94.1</td>
<td>91.1</td>
</tr>
<tr>
<td>SOP+</td>
<td>96.3</td>
<td>95.5</td>
</tr>
<tr>
<td>PES(semi)</td>
<td>95.9</td>
<td>95.1</td>
</tr>
<tr>
<td>ULC</td>
<td>96.1</td>
<td>95.2</td>
</tr>
</tbody>
</table>

ProMix(last) 97.59 97.30 95.13 96.51 82.39 79.72 68.95 42.74

ProMix(best) 97.69 97.40 95.49 93.36 96.59 82.64 80.06 69.37 42.93

We train two peer networks $f_1$ and $f_2$ in parallel to revise the noisy labels by agreement. Then we select a new subset $D_c$ to enlarge the clean set $D_l = D_l \cup D_c$ by the condition,

$$\left( \max_j f_1^j(x) \geq \tau \right) \land \left( \max_j f_2^j(x) \geq \tau \right) \land \left( \arg \max_j f_1^j(x) = \arg \max_j f_2^j(x) \right)$$

In other words, the two networks should both hold high confidence on the same label. After that, we take the shared prediction $y' = \arg \max_j f_1^j(x)$ as the guessed labels for $D_c$. With such reliable refurbishment, more purified clean samples could be involved in $D_l$ to facilitate the SSL training.

**Training Objective.** To improve the representation ability of ProMix, we involve consistency training along with mixup augmentation on the reliable labeled set $D_l$. See Appendix A for more details. The losses for consistency regularization and mixup augmentation are denoted as $L_{cr}$ and $L_{mix}$ respectively. The overall training loss is given by,

$$L_{total} = L_{cls} + \gamma(L_{cr} + L_{mix})$$

where $\gamma$ is a weighting factor. Notably, we ensemble the outputs of the two peer networks at the inference time.

## 4 Experiments

In this section, we present the main results and part of the ablation results to demonstrate the effectiveness of the ProMix framework. We put more experimental results, including results on imbalanced dataset with synthetic noise, results on dataset with instance-dependent label noise and more results for ablation experiment in Appendix B.

### 4.1 Setup

**Datasets.** We first evaluate the performance of ProMix on CIFAR-10, CIFAR-100 [Krizhevsky et al., 2009], Clothing1M [Xiao et al., 2015] and ANIMAL-10N [Song et al., 2019] dataset. For CIFAR-10/100, we conduct experiments with synthetic symmetric and asymmetric label noise following the previous protocol [Tanaka et al., 2018]. Symmetric noise is introduced by randomly flipping each ground-truth label to all possible labels with probability $r$ and asymmetric noise mimics structure of realistic noise taking place in similar classes. We also verify the effectiveness of ProMix on the recently proposed real-world noisy CIFAR-N [Wei et al., 2022b] dataset. The CIFAR-N dataset is composed of CIFAR-10N and CIFAR-100N, which equips the training samples of CIFAR-10 and CIFAR100 with human-annotated real-world noisy labels collected from Amazon Mechanical Turk. We adopt three noise types, Aggregate, Rand1 and Worst with a noise rate of 9.03\%, 17.23\% and 40.21\% respectively for CIFAR-10N and one noise type Noisy-Fine whose noise rate is 40.20\% for CIFAR-100N. Clothing1M dataset is a large-scale real-world noisy dataset containing 1 million images crawled from online shopping websites whose labels are collected by extracting tags from the surrounding texts. ANIMAL-10N is also a real-world noisy dataset consisting of human-annotated online images of ten confusable animals.

**Implementation details.** For CIFAR experiments, we use two ResNet-18 as the backbone of the peer networks. These networks are trained for 600 epochs, with a warm-up period of 10 epochs for CIFAR-10 and 30 epochs for CIFAR-100. We employ SGD as the optimizer with a momentum of 0.9 and weight decay of 5e-4. The batch size is fixed as 256 and the initial learning rate is 0.05, which decays by a cosine scheduler. The threshold $\tau$ is set as 0.99 and 0.95 for CIFAR-10 and CIFAR-100. The debiasing factor $\alpha$ is set as 0.8 and 0.5 for CIFAR-10 and CIFAR-100. $m$ is fixed as 0.9999. We leverage RandAugment to generate a strongly augmented view for consistency training. The filter rate $R$ is set as $\{0.7, 0.5, 0.2, 0.1\}$ for symmetric noise with noise ratio $r = \{0.2, 0.5, 0.8, 0.9\}$ respectively and 0.5 for asym-
metric noise and all settings On CIFAR-N. LGA is performed at the last $K = 250$ epochs such that the networks are accurate enough. We also set a maximum ratio threshold of total selected samples $\rho = 0.9$ for all settings to restrict excessive correction. We linearly ramp up $\gamma$ from 0 to 1 and $\lambda_\alpha$ from 0 to 0.1 to avoid overfitting false labels at the beginning.

For Clothong1M, following previous work [Li et al., 2020a], the backbone is ResNet-50 pre-trained on ImageNet and the training is conducted for 80 epochs. We set the learning rate as 0.002 and weight decay at 0.001. The filter rate $R$ is set as 0.7, $K$ is fixed as 40 and $\alpha$ is set at 0.8. We leverage AutoAugment [Cubuk et al., 2019] with ImageNet-Policy for augmentation. For ANIMAL-10N, we adopt VGG-19 [Simonyan and Zisserman, 2015] with batch-normalization as augmentation. For ANIMAL-10N, we adopt VGG-19 [Simonyan and Zisserman, 2015] with batch-normalization as the backbone following [Song et al., 2019]. We use SGD with the same settings as CIFAR for 300 epochs of training. The filter rate $R$ is set as 0.7. See Appendix for more details.

### 4.2 Main Results

We compare ProMix with multiple state-of-the-art methods. Specifically, they are Co-Teaching+ [Yu et al., 2019], M-correction [Arazo et al., 2019], PENCIL [Yi and Wu, 2019], SELFIE [Song et al., 2019], JoCoR [Wei et al., 2020], DivideMix [Li et al., 2020a], MOIT [Ortego et al., 2021], PLC [Zhang et al., 2021b], NCT [Chen et al., 2021], ELR+ [Liu et al., 2020], JoCoR [Wei et al., 2020], CORES* [Cheng et al., 2021], SOP+ [Bai et al., 2021], SOP+ [Liu et al., 2022], SELC [Lu and He, 2022], ULC [Huang et al., 2022], LongReMix [Cordeiro et al., 2023] and the naive cross-entropy (CE) method. For their performance, we directly adopt the reported results from their receptive papers. For fair comparisons, we follow the standard LNL evaluation protocols [Li et al., 2020a; Xia et al., 2019] and adopt the same backbone as in previous works for all benchmarks.

**Comparison with synthetic noisy labels.** Table 1 shows the results on CIFAR-10/100 dataset with different types and levels of synthetic label noise. ProMix significantly outperforms all the rivals by a notable margin with different settings. Specifically, on the CIFAR-10 dataset with symmetric noise, ProMix exceeds the best baseline by 1.39%, 1.90%, 1.49% and 6.96% under the noise ratio of 20%, 50%, 80%, 90% respectively. Moreover, most baselines exhibit significant performance degradation under heavier noise, whereas ProMix displays robustness and remains competitive. These observations clearly verify the effectiveness of ProMix.

**Comparison with real-world noisy labels.** As shown in Table 3, on the CIFAR-N dataset with real-world noisy labels, our proposed ProMix achieves the best performance compared with other existing methods on all four settings. In specific, on the CIFAR-10N dataset, the absolute accuracy gains for ProMix are 2.04%, 2.11% and 3.10% under Aggregate, Rand1, and Worst label annotations. Additionally, ProMix further expand the lead and reach a performance gain at 2.66% on the even more challenging CIFAR-100N dataset. Table 2 and Table 4 show that ProMix surpasses all the previous methods on the large-scale Clothing1M and ANIMAL-10N dataset, particularly achieves a lead of 5.88% on ANIMAL-10N. These results clarify that our method can be well applied in real-world scenarios with large-scale data.

<table>
<thead>
<tr>
<th>Method</th>
<th>CE</th>
<th>PENCIL</th>
<th>JoCoR</th>
<th>DivideMix</th>
<th>ELR+</th>
<th>LongReMix</th>
<th>CORES*</th>
<th>SOP</th>
<th>PES</th>
<th>ULC</th>
<th>ProMix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>69.21</td>
<td>73.49</td>
<td>70.30</td>
<td>74.76</td>
<td>74.81</td>
<td>74.38</td>
<td>73.24</td>
<td>73.50</td>
<td>74.64</td>
<td>74.90</td>
<td><strong>74.94</strong></td>
</tr>
</tbody>
</table>

Table 2: Accuracy comparisons on CIFAR-10N and CIFAR-100N under different noise types. **Bold entries** indicate superior results.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CIFAR-10N</th>
<th>CIFAR-100N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Aggr</td>
<td>Rand1</td>
</tr>
<tr>
<td>CE</td>
<td>87.77</td>
<td>85.02</td>
</tr>
<tr>
<td>Co-Teaching+</td>
<td>91.20</td>
<td>90.33</td>
</tr>
<tr>
<td>JoCoR</td>
<td>91.44</td>
<td>90.30</td>
</tr>
<tr>
<td>DivideMix</td>
<td>95.01</td>
<td>95.16</td>
</tr>
<tr>
<td>ELR+</td>
<td>94.83</td>
<td>94.43</td>
</tr>
<tr>
<td>CORES*</td>
<td>95.25</td>
<td>94.45</td>
</tr>
<tr>
<td>SOP+</td>
<td>95.61</td>
<td>95.28</td>
</tr>
<tr>
<td>PES(Semi)</td>
<td>94.66</td>
<td>95.06</td>
</tr>
<tr>
<td>ProMix</td>
<td><strong>97.65</strong></td>
<td><strong>97.39</strong></td>
</tr>
</tbody>
</table>

Table 3: Accuracy comparisons on CIFAR-10N and CIFAR-100N under different noise types. **Bold entries** indicate superior results.

<table>
<thead>
<tr>
<th>Method</th>
<th>CE</th>
<th>SELFIE</th>
<th>PLC</th>
<th>NCT</th>
<th>SELC</th>
<th>ProMix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>79.4</td>
<td>81.8</td>
<td>83.4</td>
<td>84.1</td>
<td>83.7</td>
<td><strong>89.98</strong></td>
</tr>
</tbody>
</table>

Table 4: Accuracy comparisons on ANIMAL-10N dataset.

### 4.3 Ablation Study

**Effect of sample selection.** We first dismiss the label guessing in variant ProMix w/o LGA, where we note that LGA could boost performance, especially under severe noise. Based on this variant, we further equip ProMix with different sample selection strategies: 1) ProMix w/o MHCS which performs CSS and overlooks the remaining matched high-confidence samples; 2) ProMix w/o Base Selection which discards the base selection and only selects matched high-confidence samples. From Table 5, we can observe that ProMix w/o MHCS largely underperforms ProMix, which verifies the efficacy of MHCS mechanism. ProMix w/o Base Selection cannot establish a competitive performance because of inadequate labeled samples. This also proves that a clean-enough base selection set is the foundation of further progressive selection and debiased SSL. Moreover, We test a variant ProMix with Only Clean which merely employs the selected samples and disregards the unselected samples. This variant suffers from perceptible performance degradation, but still beats all the baselines on CIFAR-N by a large margin. Such result indicates that clean samples are the real treasures in LNL, well-utilizing which is the key to successful learning.

Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)

4447
Ablation LGA Selection Strategy CIFAR-10 CIFAR-100 CIFAR-10N CIFAR-100N

<table>
<thead>
<tr>
<th>ProMix</th>
<th>✓</th>
<th>CSS+MHCS</th>
<th>95.05</th>
<th>68.95</th>
<th>96.34</th>
<th>73.79</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o LGA</td>
<td>✗</td>
<td>CSS+MHCS</td>
<td>94.32</td>
<td>61.19</td>
<td>95.57</td>
<td>73.10</td>
</tr>
<tr>
<td>w/o MHCS</td>
<td>✗</td>
<td>CSS</td>
<td>93.17</td>
<td>60.35</td>
<td>95.11</td>
<td>70.40</td>
</tr>
<tr>
<td>w/o Base Selection</td>
<td>✗</td>
<td>MHCS</td>
<td>39.59</td>
<td>21.01</td>
<td>74.09</td>
<td>40.75</td>
</tr>
<tr>
<td>w/o CBR</td>
<td>✗</td>
<td>CSS+MHCS</td>
<td>93.96</td>
<td>60.72</td>
<td>95.26</td>
<td>72.61</td>
</tr>
<tr>
<td>w/o DBR</td>
<td>✗</td>
<td>CSS+MHCS</td>
<td>94.07</td>
<td>60.91</td>
<td>95.07</td>
<td>72.51</td>
</tr>
<tr>
<td>with Only Clean</td>
<td>✗</td>
<td>CSS+MHCS</td>
<td>93.82</td>
<td>60.35</td>
<td>95.18</td>
<td>72.06</td>
</tr>
</tbody>
</table>

Table 5: Ablation study of ProMix on CIFAR-10-Symmetric 80%, CIFAR-100-Symmetric 80%, CIFAR-10N-Worst and CIFAR-100N-Noisy. ProMix with Only Clean denotes training with merely selected samples. CBR/DBR indicates Confirmation/Distribution Bias Removal.

Effect of bias mitigation. Next, we ablate the contributions of the bias mitigation components in the debiased SSL. Specifically, we compare ProMix with two variants: 1) ProMix w/o Confirmation Bias Removal (CBR) which discards \( h_{AP} \), generating and utilizing the pseudo-labels on the same classification head; 2) ProMix w/o Distribution Bias Removal (DBR) which sticks to the vanilla pseudo-labeling and adopts the standard cross entropy loss. From Table 5, we can observe that the lack of any one of these two components will bring about performance degradation, which indicates the effectiveness of bias mitigation. The performance drop grows more severe in imbalance scenarios as shown in Appendix. Moreover, to reveal the impact of debiasing more intuitively, we compare the accuracy of the pseudo-labels of unlabeled data during training with and without incurring the bias mitigation strategies. As shown in Figure 4, we can see that integrated with the debiasing strategies, the pseudo-labels on unlabeled data grow more accurate, hence facilitating the training process and reciprocating the sample selection process.

ProMix improves the clean sample utility. To further prove that ProMix indeed improves the clean sample utility, we compare the precision and recall of filtered clean samples of ProMix and the other two baselines, DivideMix [Li et al., 2020a] and JoCoR [Wei et al., 2020]. From Figure 3, we observe that ProMix significantly outperforms its competitors in both precision and recall, that is, ProMix selects much more clean samples while introducing fewer noisy samples. In specific, ProMix achieves a precision of 96.59% and a recall of 93.78% on CIFAR-10-Sym 80%, with a lead of 2.23% in precision and 1.15% in recall compared to DivideMix. In more straightforward terms, ProMix picks 450 more clean samples and 891 fewer false positive noise samples than DivideMix, most of which are precious and meaningful hard samples, greatly facilitating training. Besides, on CIFAR-100-Sym 80% where performances of competitors decline, ProMix leads by an even wider margin, especially in precision. This indicates that ProMix is based on a pipeline that first selects a clean-enough set with high precision, and then tries to expand the clean set with MHCS while still ensuring high precision, gradually maximizing clean sample utility. See Appendix B for more results of clean sample utility.

5 Conclusion

In this work, we propose a novel framework called ProMix for learning with noisy labels with the principle of maximizing the utility of clean examples. ProMix dynamically expands a balanced base selection set by progressively collecting those samples with matched high confidence for the subsequent SSL. We also incorporate an auxiliary pseudo head and a calibration mechanism to mitigate the confirmation bias and the distribution bias in the process of training. The proposed framework is simple but achieves significantly superior performance to the state-of-the-art LNL techniques on different datasets with various kinds and levels of noise. We hope our work can inspire future research along this direction to promote the LNL methods by well utilizing the clean samples, which are the true treasures of the data.
Acknowledgements

Junbo Zhao wants to thank the support by the NSFC Grants (No. 62206247) and the Fundamental Research Funds for the Central Universities. Lei Feng is supported by the Joint NTU-WeBank Research Centre on Fintech (Award No: NWJ-2021-005), Nanyang Technological University, Singapore.

References


[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning - From Theory to Algorithms. Cambridge University Press, 2014.


