PMNAR: Positive and Multi-Negative Learning from Noisy Labels with Adaptive Reweighting

Anonymous CVPR submission

Paper ID *****

Abstract

*Loss correction is a typical approach to learning from noisy labels. These methods usually design robust loss functions or reweight the losses of training examples to alleviate the harmfulness of noisy labels. Unfortunately, the former often obtains unfavorable performance due to under-fitting on clean examples, while the latter requires extra knowledge, e.g., additional clean validation examples. In this paper, we propose a novel framework of **P**ositive and **M**ulti-**N**egative learning with **A**daptive **R**eweighting (PMNAR) for tackling noisy labels. On one hand, we adaptively reweight the losses by jointly considering the prediction confidence on a specific example and the overall distribution of predictions. By reversing the implicit weights of the gradient for different examples, the generated weights can effectively prevent the model from over-fitting noisy labels. On the other hand, we design the positive and multi-negative learning loss to precisely estimate the weight for each training example by preventing the model from being overconfident on its predictions. By benefiting from each other, these two components can effectively exploit the clean examples and filter out mislabeled examples. Extensive experiments on multiple benchmarks and realistic datasets validate that the proposed method can achieve state-of-the-art performance.*

1. Introduction

The great success of supervised learning highly depends on large-scale and carefully labeled data. However, precisely labeling a large number of examples is difficult and costly. To save the labeling expense, one can adopt cost-effective strategies, *e.g.*, crowdsourcing labeling [12,20,43], to annotate training examples. Although these methods can significantly improve labeling efficiency, they inevitably introduce label noise, which makes the model suffer from over-fitting issue and obtain unfavorable performance [41].

Recently, numerous methods have been developed to improve the robustness of deep neural networks (DNNs) against label noise. Among them, loss correction that modifies the loss function to train DNNs robustly, has become the most popular method. Existing methods can be roughly divided into two groups. One kind of methods designs robust losses that have immunity to label noise. For example, a modified version of Cross Entropy (CE) called Generalized Cross Entropy (GCE) [44] has been developed to balance the powerful fitting ability of CE and strong robustness of MAE. Symmetric Cross Entropy (SCE) [35] combines the CE loss with a Reverse Cross Entropy (RCE) term to achieve the symmetry loss property, which has been theoretical proven to be robust to label noise. Unfortunately, it has been shown empirically that robust loss functions usually suffer from the under-fitting issue on clean examples, making it difficult to obtain desirable performance [25]. Another kind of methods reweights losses of training examples to alleviate the harmfulness of mislabeled examples. For example, L2RW [29] and its variants [32,45] adaptively estimate the weight of each training example with a meta objective on the validation set. However, in many real-world scenarios, it is difficult and costly to obtain a clean validation set.

In this paper, we propose a novel framework of **P**ositive and **M**ulti-**N**egative learning with **A**daptive **R**eweighting (PMNAR) for combating noisy labels. Specifically, we first drive an **A**daptive **R**eweighting (AR) strategy by assigning each training example with a weight according to its predicted confidence and subtracting a global adaptive factor on it in each training epoch. By reversing implicit weights of CE's gradient for different examples, the AR strategy can effectively avoid CE over-fitting mislabeled examples. Meanwhile, benefiting from DNNs' memorization effect, the proposed AR strategy can guarantee that all the potentially clean examples will gradually receive large weights, while possibly mislabeled examples receive small weights. To further ensure a precise estimation of the weights, we improve the loss used in [14] by incorporating multiple supervised signals and then design the noise-robust **P**ositive and **M**ulti-negative learning (PM) loss. The proposed loss can effectively reduce the risk of incorrect updating and restrain the model from outputting over-confident predictions. These components would help each other and achieve a balance between the powerful learning ability and the strong robustness of the model. Furthermore, we simultaneously train two networks and let them provide weights for each other, which avoids the risk of accumulating errors in a single network. Experiments are performed on multiple benchmarks and real-world datasets. The experimental results demonstrate that our method can achieve strong robustness to label noise. Our main contributions can be summarized as follows:

- Based on the analysis of the gradient of CE loss, an AR strategy is proposed based on the overall distribution of predicted confidences with a global adaptive factor. The proposed AR strategy can effectively avoid CE over-fitting noisy labels by providing potential clean examples with large weights, while assigning potential mislabeled examples with weights approaching zero.
- The proposed PM loss effectively reduces the risk of incorrect updating to ensure the model convergence, and precisely estimates the weights by restraining the model from outputting over-confident predictions.
- Comprehensive experiments on multiple benchmarks and real-world datasets validate that the proposed method can achieve state-of-the-art performance.

2. Related Work

Robust architecture The robust structure aims to improve generalization by modifying the DNNs' output based on the estimated label transition probability, mainly implemented by adding a noise adaptation layer on top of the softmax layer [5, 33] or designing a new dedicated architecture [18,38,39]. Webly learning [5] selects easy examples to initialize the weights of the noise adaptation layer and finetunes the entire model on hard examples in an end-to-end manner. Probabilistic noise modeling [38] trains two specialized networks and makes them predict noise type and label transition probability for each other. In [33], the authors directly initialize the weights by an identity matrix and add a regularization term to diffuse the weights. Authors in [39] proposed a contrastive additive noise network that adjusts for incorrectly estimated label transition probabilities by modeling the trustworthiness of noisy labels. A robust generative classifier is proposed in [18] which is applicable on top of hidden feature spaces of any discriminative neural classifier pre-trained on noisy datasets.

Robust regularization Regularization can effectively prevent the model from over-fitting noisy data. The existing regularization methods can be divided into two main categories: explicit regularization [22, 23, 26] (e.g. weight decay) and implicit regularization [24, 42] (e.g. data augmentation). For explicit regularization, PHuber [26] proposes a composite loss-based gradient clipping for label noise robustness. ELR [22] incorporates target probabilities estimated from the model outputs to construct the regularization term. SOP [23] models the label noise through a sparse over-parameterization term and exploits implicit algorithmic regularizations to recover and separate the underlying corruptions. For implicit regularization, mixup [42] regularizes the DNNs to favor simple linear behavior in-between training examples. Label smoothing [24] reduces overfitting by preventing the DNN from assigning a full probability to noisy training examples.

Sample selection Sample selection attempts to directly identify potentially noisy examples and then learning only based on clean examples or learning in a semi-supervised manner. MentorNet [11] firstly trains a teacher network and then uses it to select small loss examples as clean examples for guiding the training of the student network. Co-teaching [8] and Co-teaching+ [40] maintain two networks simultaneously and let them select training examples for each other. DivideMix [19] leverages Gaussian mixture model to distinguish clean and noisy data, and use a semi-supervised technique called MixMatch [2] in the Co-teaching framework. In contrast, O2U-net [10] is a straightforward noisy label detection approach, which only requires adjusting the learning rate to keep the trained network transferring from overfitting to underfitting cyclically.

Loss modification One type of loss modification is to directly design a robust loss function [6, 25, 35, 44]. Given that losses like Mean Absolute Error (MAE) loss satisfied the symmetric condition are robust to label noise while CE loss is not [6]. GCE [44] is proposed to balance the MAE and CE. SCE [35] designed a loss term named RCE to satisfy the symmetric condition. In [14], the authors proposed a Negative Learning (NL) loss based on CE, which uses complementary label for training and has been proven to be robust to noise. However, [25] shows empirically that the robust loss functions usually suffer from an underfitting problem and is difficult to obtain the desired model performance. Another type of loss modification is to reduce the negative impact of noisy labels by multiplying sample level weights [3, 9, 34, 37] or label transition matrix [28]. Active bias [3] uses the sample prediction variance as its weight to emphasize examples with inconsistent predictions. In [28], the authors proposed a loss correction method based on the pre-calculated backward or forward noise transition matrix, which is obtained by exploiting anchor points.

3. Methodology

In this section, we first give necessary preliminaries, and then introduce our proposed method consisting of two components, including adaptive reweighting and positive and multi-negative learning.

3.1. Preliminaries

We consider the problem of ordinary K-class classification. Let $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ be an input feature vector, $y, y^* \in \mathcal{Y} = \{1, ..., K\}$ be its annotated label (possibly incorrect) and true label, respectively. Suppose a DNN $f(\boldsymbol{x}, \theta)$ with the softmax output layer maps the input space to the label space $f : \mathcal{X} \rightarrow \mathcal{Y}$, where θ represents the network parameters. Our goal is to learn a classifier f by minimizing the classification risk: $\mathcal{R}(f) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\ell(f(\boldsymbol{x}, \theta), y)$ over the noisy distribution $p(\boldsymbol{x}, \boldsymbol{y})$, where ℓ represents a loss function used to measure the performance of learned classifier.

3.2. Adaptive Reweighting

In multi-class classification, the Cross Entropy (CE) loss is the most commonly used loss function, which can be defined as:

$$\ell_{\text{CE}}\left(f(\boldsymbol{x}), y\right) = -\boldsymbol{e}_y \log f(\boldsymbol{x}) = -\log f_y(x), \qquad (1)$$

where $f_y(x)$ represents the y-th element of $f(\boldsymbol{x})$ and $\boldsymbol{e}_y$ is a one-hot vector with value 1 at the y-th element while 0, otherwise. Previous works [6, 35, 44] have shown that trained with standard CE loss, DNNs are more prone to suffer from the over-fitting issue. One reasonable explanation behind this phenomenon can be given in terms of the gradient of CE loss. Specifically, its gradient with respect to network parameters θ can be formulated as follows:

$$\frac{\partial \ell_{\text{CE}}(f(\boldsymbol{x}), y)}{\partial \theta} = -\frac{1}{f_y(\boldsymbol{x})}\nabla_\theta f_y(\boldsymbol{x}). \qquad (2)$$

For notational simplicity, we express $\frac{\partial \ell(f(x),y)}{\partial \theta}$ by $\frac{\partial \ell}{\partial \theta}$ in the following content. From Eq.(2), it can be observed that when updating the network parameters, the examples with more confident predictions will receive larger weights than those with less confident predictions. Due to the memorization effect, at the early stage of training, the model tends to fit the clean examples and thus predicts small confidences for mislabeled examples. This indicates at the subsequent training, the model would pay more attention to fit the mislabeled examples, which degrades its performance significantly.

To cope with this problem, a straightforward method is to assign a weight $f_y(\boldsymbol{x})$ for each example, whose gradient can be reformulated as follows:

$$f_y(\boldsymbol{x})\frac{\partial \ell_{\text{CE}}}{\partial \theta} = -f_y(\boldsymbol{x})\frac{1}{f_y(\boldsymbol{x})}\nabla_\theta f_y(\boldsymbol{x}) = -\nabla_\theta f_y(\boldsymbol{x}). \quad (3)$$

In contrast to Eq.(2), Eq.(3) allows the model to treat all examples equally, which is similar to MAE loss. Although MAE loss has been proven to achieve strong robustness against noisy labels [6], it often obtains unfavorable performance due to the under-fitting issue caused by gradient saturation [25]. Obviously, treating all examples equally is unreasonable for model training, since there exist many mislabeled examples hidden in the training set. To alleviate the harmfulness of mislabeled examples, we propose a method called adaptive reweighting (AR), which modifies the weight $f_y(\boldsymbol{x})$ by subtracting it by an adaptive factor f_{min}, where $f_{\text{min}} = \min_{\boldsymbol{x}} f_y(\boldsymbol{x})$ is the minimal confidence among all training example on its annotated label. For notational simplicity, denoting by $w_y(\boldsymbol{x})$ the modified weight for instance $\boldsymbol{x}$, we have the following gradient:

$$\begin{aligned} w_y(\boldsymbol{x})\frac{\partial \ell_{\text{CE}}}{\partial \theta} &= -\left(f_y(\boldsymbol{x}) - f_{\text{min}}\right)\frac{1}{f_y(\boldsymbol{x})}\nabla_\theta f_y(\boldsymbol{x}) \\ &= -\left(1 - \frac{f_{\text{min}}}{f_y(\boldsymbol{x})}\right)\nabla_\theta f_y(\boldsymbol{x}). \end{aligned} \qquad (4)$$

The intuition behind Eq.(4) is that if an mislabeled example $\boldsymbol{x}$ with a low predicted confidence $f_y(\boldsymbol{x}) \rightarrow f_{\text{min}}$, the implicit weight of $\nabla_\theta f_y(x)$ will approach 0, and thus alleviate over-fitting on noisy labels. Conversely, the example with a high predicted confidence $f_y(\boldsymbol{x}) \rightarrow f_{\text{max}}$, where $f_{\text{max}} = \max_{\boldsymbol{x}} f_y(\boldsymbol{x})$ is the maximal predicted confidence of the current model, is more likely to be a clean example and will be given a large implicit weight. The factor f_{min} can be regarded as a threshold, which aims to discard possible mislabeled examples with very low predicted confidences. Since the weight $w_y(\boldsymbol{x})$ would be updated adaptively according to the changing factor f_{min} with the increase of epochs, we call this strategy adaptive reweighing (AR).

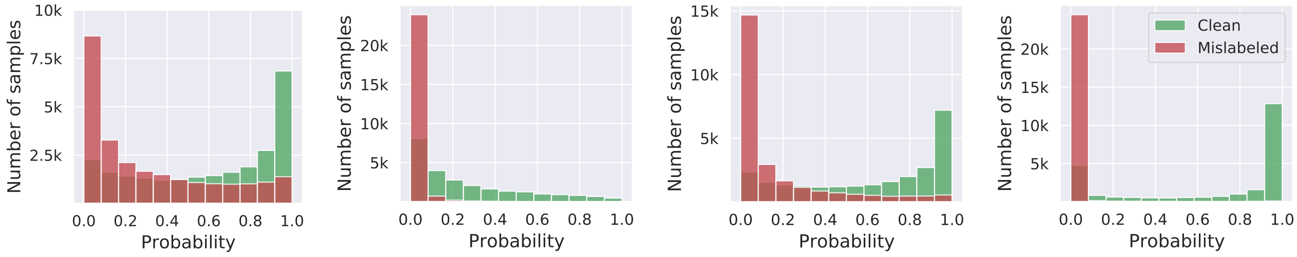
3.3. Positive and Multi-Negative Learning

The proposed AR method highly depends on precise estimation of the confidence. Unfortunately, as the model training proceeds, the model easily suffers from the over-fitting issue on clean examples, leading to over-confident predictions [7]. In the extreme case, when the model has a powerful capacity to fit all the examples, *i.e.*, $\forall f_y(\boldsymbol{x}) \rightarrow 1$, it would destroy the updating of network parameters, since the weight for each example converges to 0.

Inspired by recent works [14, 15], we propose to alleviate the issue of over-confident predictions by incorporating negative learning (NL) [14] into CE loss. Specifically, NL adopts an indirect learning method that explores the usefulness of complementary labels, *i.e.*, the classes that the instance does not belong to, which has been proven to be effective for alleviating over-fitting on noisy labels. Formally, the commonly used NL loss can be defined as follows:

$$\ell_{\text{NL}}(f(\boldsymbol{x}), \bar{y}) = -\boldsymbol{e}_{\bar{y}} \log\left(1 - f(\boldsymbol{x})\right) = -\log\left(1 - f_{\bar{y}}(\boldsymbol{x})\right), \qquad (5)$$

where $\bar{y}$ is a complementary label that is randomly selected from all classes other than the given label y at every iteration during model training. Considering that training with



(a) CE

(b) CE w/ AR

(c) PM

(d) PM w/ AR

Figure 1. The distribution of clean and mislabeled examples achieve by four strategies in terms of their predicted probabilities on CIFAR-100 at epoch 50. From the figures, compared with the other three strategies, PM w/ AR achieves an almost perfect separation of clean examples and mislabeled ones, which validates that these two components make significant contributions to the final performance.

the CE loss, the mislabeled examples have a risk of probability 1 being trained in a reverse way, *i.e.*, maximize the probability of a complementary label. When training with the NL loss, for each mislabeled example, the probability of a random complementary label to be its true label is $P(\bar{y} = y^*) = \frac{1}{K-1}$, which indicates that the mislabeled examples only have a risk of probability $\frac{1}{K-1}$ being trained in a reverse way, *i.e.*, minimize the probability of a true label. This means that NL loss has a much smaller probability to introduce label noise than CE loss, and thus often becomes more robust.

Compared with CE loss, although NL loss shows superiority against label noise, it often suffers from the issue of slow convergence, also leading to undesirable performance. To mitigate this dilemma, we modify the original NL loss by performing negative learning on multiple complementary labels, which introduce much more supervised signals, making it converge fastly. Formally, we define the following multi-negative learning (ML) loss:

$$\begin{aligned}\ell_{\text{ML}}(f(\boldsymbol{x}), \bar{\boldsymbol{y}}) &= -\bar{\boldsymbol{e}}_y \log\left(1 - f(\boldsymbol{x})\right) \\ &= -\sum_{\bar{y} \in \bar{\boldsymbol{y}}} \log\left(1 - f_{\bar{y}}(\boldsymbol{x})\right),\end{aligned} \tag{6}$$

where $\bar{\boldsymbol{y}} = \mathcal{Y} \backslash y$ represents the complementary label set consisting of all labels other than the given label y, and $\bar{\boldsymbol{e}}_y = 1 - \boldsymbol{e}_y$ is a vector with value 0 at the y-th element while 1, otherwise.

Furthermore, by combining it with CE loss, we propose a new loss function called Positive and Multi-negative learning (PM) loss, which can be defined as follows:

$$\ell_{\text{PM}}(f(\boldsymbol{x}), y, \bar{\boldsymbol{y}}) = -\log f_y(\boldsymbol{x}) - \sum_{\bar{y} \in \bar{\boldsymbol{y}}} \log\left(1 - f_{\bar{y}}(\boldsymbol{x})\right). \tag{7}$$

Roughly speaking, the PM loss trains the model on a totally K possible labels, and suffers from a risk of incorrect updating on at most two labels. This yields that it has a probability lower than $\frac{2}{K}$ of being trained in a reverse way, which is still more robust than CE loss. As mentioned above, in the case with the true label y^* included in the complementary label set $\bar{\boldsymbol{y}}$, the model will suffer from a risk of misclassifying the corresponding examples. To alleviate the harmfulness of the false negative label, we propose to identify the potential true label from the complementary label set $\bar{\boldsymbol{y}}$ based on historical predictions of each training example. Specifically, if an instance has ever been classified as the label $y' = \arg\max f(\boldsymbol{x})$ during the whole training process, then the label y' can be regarded as a potential true label and would be omitted from the complementary label set $\bar{\boldsymbol{y}}$.

Finally, we can improve AR strategy by directly replacing CE loss with PM loss. To empirically study how these components benefit from each other, Figure 1 illustrates the performance of noise detection achieved by four strategies, including CE, CE w/ AR, PM, and PM w/ AR. From the figures, it can be observed that: 1) CE suffers from severe over-fitting on noisy labels, which can be reflected by an entire overlap between the distributions of clean and mislabeled ones. 2) By adopting the AR strategy, CE w/ AR alleviate the over-fitting issue and obtains better performance. However, due to a greater risk of incorrect training, CE w/ AR fail to achieve a perfect separation between clean examples and mislabeled ones. 3) Although PM becomes more robust than CE to label noise, it cannot achieves a favorable separation due to the lack of AR strategy. 4) PM w/ AR achieves an almost perfect separation, which convincingly validates the effectiveness of PM and AR, and more importantly, these two components can benefit from each other.

3.4. Overall Training

Although the proposed PM loss and AR training strategy can significantly lower the risk of over-fitting on noisy labels, it is hard to identify examples that have been over-fitted, leading to increasing accumulated error. To solve this problem, inspired by previous works [8, 19, 36], we adopt the dual networks to alternatively update weights for each other. Generally, the dual-network strategy has two

Algorithm 1 The PMNAR algorithm.

```
Input: Training examples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, two networks $f(\boldsymbol{x}, \theta)$ and $f'(x, \theta')$, $\beta$, warm-up epochs $T_{warm}$, and the maximal epoch $T$
Output: Learned model parameters $\theta$.
Initialize $w_y(x)$ of all examples $x$ to 1.
Generate $\bar{\boldsymbol{y}}$ for each sample $x$ based on given label $y$.
Initialize $S$ by $\{\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_2, \ldots, \bar{\boldsymbol{y}}_n\}$.
for $t = 1 : T$ do
    if $t > T_{warm}$ then
        for each example $\boldsymbol{x}$ do
            Obtain $f_y(\boldsymbol{x})$ and $f'_y(\boldsymbol{x})$.
            Obtain $y' = \arg\max f(\boldsymbol{x})$ and update $S$ by discarding $y'$ from $\bar{\boldsymbol{y}}$ for $\boldsymbol{x}$.
        end for
        Calculate $w_y(\boldsymbol{x}) = f'_y(\boldsymbol{x}) - f'_{\text{min}}$ and $w'_y(\boldsymbol{x}) = f_y(\boldsymbol{x}) - f_{\text{min}}$.
    end if
    Update $\theta$ and $\theta'$ by minimizing $\sum_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y, \bar{\boldsymbol{y}})$ and $\sum_{\boldsymbol{x}} \ell(f'(\boldsymbol{x}), y, \bar{\boldsymbol{y}})$, respectively.
end for
```

advantages. On one hand, it can avoid accumulating errors by transferring mistakes back to itself. On the other hand, different networks are able to learn distinct patterns from training examples and correct diverse label noise. In the following content, we conduct an informal analysis to demonstrate the effectiveness of dual networks. By using f and f' to denote two networks, for instance $\boldsymbol{x}$, suppose $f'(\boldsymbol{x}) = f(\boldsymbol{x}) + \sigma$, where σ capture the difference between f and f' on $\boldsymbol{x}$, we can derive the gradient as follows:

$$w_y(\boldsymbol{x}) \frac{\partial \ell_{CE}}{\partial \theta} = -\left(1 - \frac{f_{\text{min}}}{f_y(\boldsymbol{x})} + \frac{\sigma}{f_y(\boldsymbol{x})}\right) \nabla_\theta f_y(\boldsymbol{x}), \quad (8)$$

In an ideal case, when f makes mistakes on $\boldsymbol{x}$, *i.e.*, it fits a mislabeled examples $\boldsymbol{x}$, $\frac{\sigma}{f_y(\boldsymbol{x})}$ can be used as a correction factor to correct the weight of $\boldsymbol{x}$ such that lower its contribution to model updating.

In our experiments, the commonly used consistency regularization is adopted to further improve the model performance by aligning the output distribution of examples and their corresponding augmented ones. Use $\text{Aug}(\boldsymbol{x})$ to denote a random augmentation of the original example, we implement the consistency regularization by minimizing the Kullback-Leibler (KL) divergence between the predicted predictions of $\boldsymbol{x}$ and $\text{Aug}(\boldsymbol{x})$ as:

$$\ell_{\text{Reg}}(\boldsymbol{x}) = D_{\text{KL}}(\frac{f(\boldsymbol{x}) + f'(\boldsymbol{x})}{2} \parallel f(\text{Aug}(\boldsymbol{x}))). \quad (9)$$

Here, we use the ensemble output to guide the consistency training instead of using a single model.

Formally, we have the final training objective function as follows:

$$\ell(f(\boldsymbol{x}), y, \bar{\boldsymbol{y}}) = w_y \ell_{\text{PM}}(f(\boldsymbol{x}), y, \bar{\boldsymbol{y}}) + \beta \ell_{\text{Reg}}(\boldsymbol{x}), \quad (10)$$

where β is a balancing parameter to control the strength of regularization. The pseudo codes of our proposed method is summarized in Algorithm 1.

4. Experiments

4.1. Experiment Setup

Datasets. We conduct experiments on four benchmark datasets, including CIFAR-10 [16], CIFAR-100 [16], Tiny-ImageNet [17], as well as real-world dataset mini-WebVision [21]. CIFAR-10 and CIFAR-100 both contain 50k training images and 10k test images, where each image is of size 32 × 32. To generate noisy labels, we consider symmetric noise rates of $\{20\%, 40\%, 50\%\}$, asymmetric noise rates of $\{10\%, 30\%, 40\%\}$. Tiny-ImageNet is a subset of ImageNet [30] and contains 200 classes with 500 training images per class, where each image is of size 84 × 84. For this dataset, we generate noisy labels by using symmetric noise rates of $\{20\%, 50\%\}$. Mini-WebVision is a subset of WebVision that contains top 50 classes from the original dataset. We resize the image to 84 × 84.

Implementation Details. For four benchmark datasets, we use PreAct ResNet18 as the base model. The model is updated by using the stochastic gradient descent (SGD) optimizer for 300 epochs. The learning rate is initialized as 0.02 and is divided by 10 per 120 epochs. The parameter of weight decay is set as 5e-4. The batch size is set as 128. For mini-WebVision, we train the model for 120 epochs with an initial learning rate of 0.02. The learning rate is divided by 10 at the 60th epoch and 90th epoch.

Additional experimental details can be found in the supplementary material.

4.2. Classification Performance

Table 1 reports comparison results between the proposed method and comparing methods on CIFAR-10 and CIFAR-100 with different levels of symmetric or asymmetric label noise in terms of accuracy. From the table, it can be observed that our method consistently achieves the best performance in all cases. Especially for asymmetric noise, the proposed method outperforms the comparing methods with a significant performance gap. In particular, on CIFAR-100 with 30% asymmetric noise, our method achieves an increment of 3.4% compared to the state-of-the-art performance.

Table 2 presents the comparison results of our method and comparing methods on Tiny-ImageNet with different levels of symmetric noise in terms of accuracy. From the table, we can see that our method still achieves favorable performance on this challenging dataset which con-

Method	CIFAR-10						CIFAR-100					
	Symmetric			Asymmetric			Symmetric			Asymmetric		
	20%	40%	50%	10%	30%	40%	20%	40%	50%	10%	30%	40%
CE	86.8	81.7	79.4	88.8	81.7	76.1	62.0	51.2	46.7	68.1	53.3	44.5
JNPL [15]	93.5	91.9	90.2	94.2	92.5	90.7	70.9	68.1	67.7	72.0	68.1	59.5
MOIT [27]	94.1	91.2	91.1	94.2	94.1	93.2	75.9	70.9	70.1	77.4	75.1	74.0
Divide-Mix [19]	96.1	94.9	94.6	93.8	92.5	91.7	77.3	75.9	74.6	71.6	69.5	55.1
ELR [22]	95.8	95.1	94.8	95.4	94.7	93.0	77.6	75.2	73.6	77.3	74.6	73.2
UniCon [13]	96.0	95.6	95.6	95.3	94.8	94.1	78.9	78.1	77.6	78.2	75.6	74.8
PMNAR	**96.5**	**96.4**	**96.0**	**96.3**	**96.0**	**95.3**	**79.9**	**78.6**	**78.1**	**79.4**	**79.0**	**76.3**

Table 1. Comparison results on CIFAR-10 and CIFAR-100 with symmetric and asymmetric noise. Results for previous techniques are copied from their respective papers, except for 40% symmetric noise. The best performance is highlighted in bold.

Noise rate	20%		50%	
Method	Best	Avg.	Best	Avg.
Standard CE	35.8	35.6	19.8	19.6
F-correction [28]	44.5	44.4	33.1	32.8
MentorNet [11]	45.7	45.5	35.8	35.5
Co-teaching+ [40]	48.2	47.7	41.8	41.2
M-correction [1]	57.2	56.6	51.6	51.3
NCT [31]	58.0	57.2	47.8	47.4
UniCon [31]	59.2	58.4	52.7	52.4
PMNAR	**60.3**	**59.5**	**55.1**	**53.9**

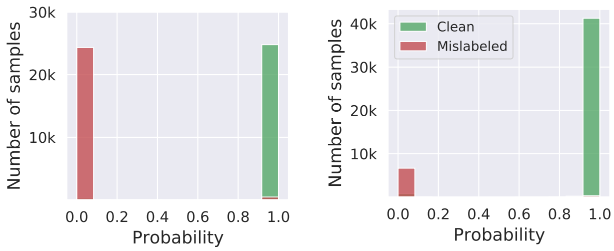
Table 2. Test performance on Tiny-ImageNet with symmetric noise. We report the results for other methods directly from [31] with the highest (Best) and the average (Avg.) test accuracy over the last 10 epochs.

Divide-Mix [19]	UNICON [31]	ELR [22]	PMNAR
69.36	68.77	70.96	**71.08**

Table 3. Test accuracy on mini-WebVision.

tains 200 classes. It is noteworthy that UNICON uses contrastive learning [4] to learn better representations, achieving state-of-the-art performance. Without contrastive learning, our method gains significant performance improvement over UNICON for all noise rates. The performance of our method will be enhanced hopefully by incorporating it into contrastive learning. An interesting phenomenon is that as the categories increase from 10 to 200, the superiority of our method becomes more significant. This is because the PM loss is class dependent. As the number of classes becomes larger, the risk of the model being trained in the wrong way gradually decreases, thus enhancing the robustness to noise.

Table 3 reports the test accuracy of each comparing method on mini-WebVision. Although mini-WebVision is a very challenging dataset that contains realistic label noise, our method still achieves competent performance and outperforms the state-of-the-art methods. These results convincingly validate the effectiveness of the proposed method.



(a) CIFAR-10 Sym-50%

(b) CIFAR-10 Asy-30%

Figure 2. The distribution of clean and mislabeled examples in terms of their predicted probabilities at final epoch on CIFAR-10.

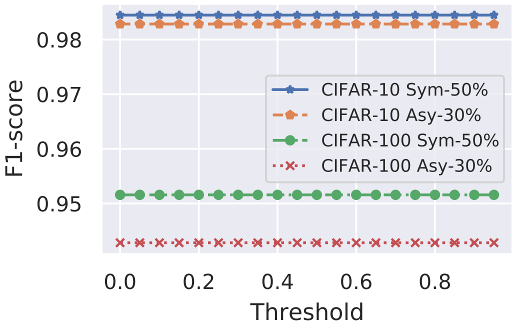


Figure 3. The performance curves with the increase of the threshold on CIFAR-10 and CIFAR-100.

4.3. Noise Detection

Figure 2 shows the distribution of clean and mislabeled examples in terms of their predicted probabilities on CIFAR-10 under 50% symmetric noise and 30% asymmetric noise. From the figures, it can be observed that the mislabeled and clean examples are completely separated on two

Metrics	CIFAR-10				CIFAR-100			
	Sym-50%		Asy-30%		Sym-50%		Asy-30%	
	Divide-Mix	PMNAR	Divide-Mix	PMNAR	Divide-Mix	PMNAR	Divide-Mix	PMNAR
Precision	95.92	98.80	84.57	97.73	87.66	92.25	84.71	90.00
Recall	98.84	98.09	97.46	98.81	97.17	98.26	93.43	99.03
F1-score	97.36	98.45	90.56	98.27	92.18	95.16	88.86	94.28
Noise rate	46.59	49.31	26.44	15.61	54.46	53.11	36.37	36.23

Table 4. The performance of noise detection of the proposed PMNAR and Divide-Mix. For PMNAR, we directly set the threshold as 0.5. For Divide-Mix, we report the results by ensembling the outputs of its two models.

sides with probabilities of 0 and 1, respectively, even without any overlap in the middle region. This indicates that by assigning the potential mislabeled examples with small weights, our method can achieve strong discrimination ability to distinguish between mislabeled and clean examples.

We study the influence of the parameter threshold used for separating mislabeled and clean examples on the performance of noisy detection in terms of F1 score. Figure 3 illustrates the performance curves as the threshold changes. As shown in the figure, we can see that the F1 score remains stable over a very wide range of the threshold values. The results indicate that we do not need extra effort to tune the threshold for achieving a better performance of label noise detection. Here, we simply set the threshold to 0.5 and report the detection performance of Divide-Mix and our proposed method in terms of several commonly used metrics, including precision, recall, and F1-score, in Table 4. Obviously, our method achieves significant superiority to Divide-Mix in terms of all three metrics. In particular, our method also achieves a much smaller error on the estimation of noise rates. It is noteworthy that for CIFAR-10 the actual noise rate of Asy-30% is 15%, since for asymmetric label noise, the label flipping only occurs on "truck → automobile, bird → airplane, deer → horse, cat → dog" in CIFAR-10. The consistently high precision allows us to combine the proposed method with other advanced semi-supervised methods, *e.g.*, MixMatch [2], in order to obtain better results.

4.4. Ablation Studies

In this section, we first conduct ablation studies to analyze the contribution of each component of the proposed method. **CE** and **NL** indicate that only CE loss and NL loss are used, respectively. **ML** and **PM** represent the use of our proposed ML loss and PM loss, respectively. **PM+AR** means incorporating the proposed AR strategy into the PM loss. It is noteworthy that to make a valid comparison, we do not utilize consistency regularization in this experiment.

Figure 4 shows the test accuracy curves on CIFAR-100 under 50% symmetric noise. As shown in the figure, **NL** converges very slowly and achieves the worst test accuracy in a limited number of epochs. This is due to the fact that learning with complementary labels provides a weak supervised signal. **ML** has slightly improved performance compared to **NL**. Although **CE** achieves fast convergence compared to the ML loss, it often suffers from over-fitting to noisy labels, which can be reflected by the significant performance drop. **PM** can mitigate the slow convergence of ML and meanwhile achieves strong robustness to noisy labels. To further analyze the PM loss, we also illustrate the distribution of mislabeled and clean examples in terms of their predicted probabilities at epoch 40 and final epoch 160 in Figure 5. From the figure, it can be observed that in the early stages of training, the trained model can distinguish examples well, while in the later stages, it suffers from over-fitting issue, which can be reflected by the fact that the probabilities converge to 1 for all examples. Because PM loss is similar to CE in its gradient form, it also inevitably gives larger weights to examples with smaller prediction probabilities, making the model over-fit the mislabeled examples. By utilizing AR strategy, **PM+AR** solves the over-fitting issue and achieves better performance. The dual-networks strategy also obtains improvement due to the avoidance of accumulated error and increased learning capacity. We also obtain similar results on other settings with diverse types of noise and different noise rates in Table 5. These results con-

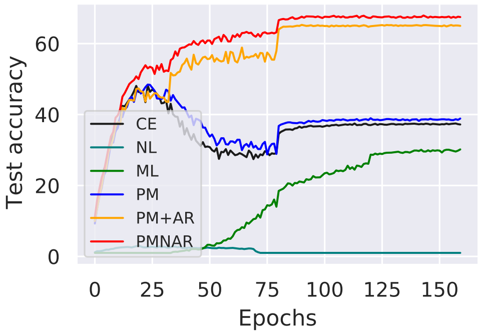
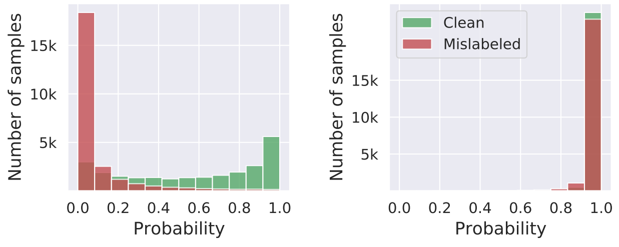


Figure 4. Ablation study of PMNAR on CIFAR-100 with 50% symmetric noise.



(a) The 40th epoch.

(b) The 160th epoch.

Figure 5. The distribution of clean and mislabeled examples in terms of the predicted probabilities on CIFAR-100. The results are achieved by using PM loss only. Figure 5a and Figure 5b denote the results of the 40th epoch and the 160th epoch, respectively.

vincingly validate that each component contributes to the final performance.

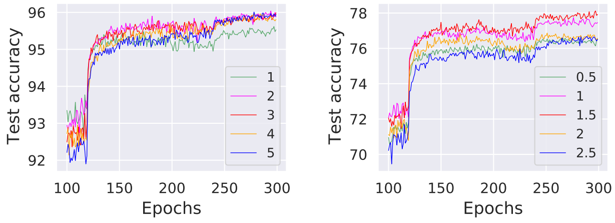
Methods	CIFAR-10		CIFAR-100	
	Sym-50%	Asy-30%	Sym-50%	Asy-30%
CE	51.83	81.74	37.32	54.74
ML	61.27	81.57	29.84	50.76
PM	55.71	82.65	38.61	55.45
PM+AR	87.13	89.54	65.04	66.73
PMNAR	**88.59**	**91.14**	**67.46**	**71.95**

Table 5. Ablation study of PMNAR on CIFAR-10 and CIFAR-100 under 50% symmetric and 30% asymmetric noise. We report the average test accuracy over the last 10 epochs.

4.5. Parameter Sensitivity Analyses

To further study the influence of the balancing parameter β, we conduct the parameter sensitivity analysis on CIFAR-10 and CIFAR-100 with 50% symmetric noise in Figure 6. From the figure, it can be observed that the performance is insensitive to the balancing parameter β on CIFAR-10. For CIFAR-100, choosing a small β often leads to better performance. Generally, the balancing parameter β is dependent to the number of classes. A large β would improve the robustness of the model, while a smaller β would strengthen the learning capacity of the model. Although PM Loss is more robust when the number of classes is large, its learning ability becomes weak accordingly. In practice, a smaller β should be chosen for CIFAR-100 compared to CIFAR-10. Furthermore, we conduct experiments to explore the effect of the number of epochs for warm-up and training batch size. Figure 7a and Figure 7b shows that our method is insensitive to the these two parameters.

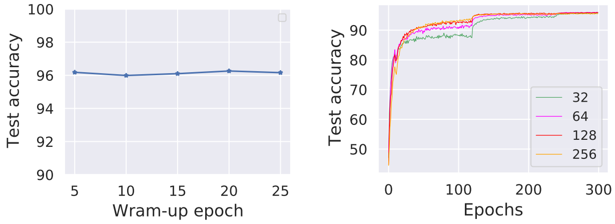
Finally, to examine the efficiency of the proposed method, we compare it with Divide-Mix in terms of training time. Specifically, for each method, we train a model for 300 epochs with a single Nvidia 3080Ti GPU. Table 6 shows the training time of these two methods on CIFAR-10. Compared with Divide-Mix, our method achieves higher simplicity and meanwhile obtains better performance as reported in the previous experiments



(a) CIFAR-10

(b) CIFAR-100

Figure 6. The parameter sensitivity analysis for balancing factor β on CIFAR-10 and CIFAR-100 under 50% symmetric noise.



(a) CIFAR-10

(b) CIFAR-100

Figure 7. The parameter sensitivity analysis for the number of epochs for warm-up (Figure 7a) and training batch size (Figure 7b) on CIFAR-10 under 50% symmetric noise.

Methods	CE	Divide-Mix	PMNAR
Training time	2.3h	10.9h	5.8h

Table 6. Comparison of total training time in hours on CIFAR-10 with 50% symmetric noise. We train all methods 300 epochs.

5. Conclusion

The paper studies the problem of learning with noisy labels. To achieve strong robustness to noisy labels, we propose an adaptive reweighting scheme that can neglect the likely mislabeled examples with very low confidences, while assign the potential clean examples with large weights. To precisely estimate the weights, we design the positive and multi-negative learning loss to restrain the model from outputting over-confident predictions. These two components complement each other, leading to achieve strong robustness and powerful learning ability of the model. Through extensive empirical analyses, we show that the proposed method can achieve superior performance compared with the state-of-the-art methods in terms of classification performance, noise detection, and training time.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR, 2019. 6

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2, 7

[3] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 6

[5] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439, 2015. 2

[6] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 2, 3

[7] Julia Grabinski, Paul Gavrikov, Janis Keuper, and Margret Keuper. Robust models are less over-confident. *arXiv preprint arXiv:2210.05938*, 2022. 3

[8] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018. 2, 4

[9] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018. 2

[10] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3326–3334, 2019. 2

[11] 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 *International conference on machine learning*, pages 2304–2313. PMLR, 2018. 2, 6

[12] David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, pages 81–92, 2013. 1

[13] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022. 6

[14] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 101–110, 2019. 1, 2, 3

[15] Youngdong Kim, Juseung Yun, Hyounguk Shon, and Junmo Kim. Joint negative and positive learning for noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2021. 3, 6

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[18] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019. 2

[19] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 2, 4, 6

[20] Shao-Yuan Li, Sheng-Jun Huang, and Songcan Chen. Crowdsourcing aggregation with deep bayesian learning. *Science China Information Sciences*, 64(3):1–11, 2021. 1

[21] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 5

[22] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020. 2, 6

[23] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. *arXiv preprint arXiv:2202.14026*, 2022. 2

[24] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 2

[25] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pages 6543–6553. PMLR, 2020. 1, 2, 3

[26] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2019. 2

[27] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6606–6615, 2021. 6

[28] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017. 2, 6

[29] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018. 1

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5

[31] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Noisy concurrent training for efficient learning under label noise. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3159–3168, 2021. 6

[32] 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. *Advances in neural information processing systems*, 32, 2019. 1

[33] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014. 2

[34] Ruxin Wang, Tongliang Liu, and Dacheng Tao. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems*, 29(6):2568–2580, 2017. 2

[35] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 1, 2, 3

[36] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020. 4

[37] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019. 2

[38] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 2

[39] Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*, 28(4):1909–1922, 2018. 2

[40] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? *arXiv preprint arXiv:1901.04215*, 2019. 2, 6

[41] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization (2016). *arXiv preprint arXiv:1611.03530*, 2017. 1

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[43] Jing Zhang, Xindong Wu, and Victor S Sheng. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review*, 46(4):543–576, 2016. 1

[44] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788, 2018. 1, 2, 3

[45] Qian Zhao, Jun Shu, Xiang Yuan, Ziming Liu, and Deyu Meng. A probabilistic formulation for meta-weight-net. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 1