An Ensemble Model for Combating Label Noise

Yangdi Lu
McMaster University
Hamilton, Ontario, Canada
luy100@mcmaster.ca

Yang Bo
McMaster University
Hamilton, Ontario, Canada
boy2@mcmaster.ca

Wenbo He
McMaster University
Hamilton, Ontario, Canada
hew11@mcmaster.ca

ABSTRACT

The labels crawled from web services (e.g. querying images from search engines and collecting tags from social media images) are often prone to noise, and the presence of such label noise degrades the classification performance of the resulting deep neural network (DNN) models. In this paper, we propose an ensemble model consisting of two networks to prevent the model from memorizing noisy labels. Within our model, we have one network generate an anchoring label from its prediction on a weakly-augmented image. Meanwhile, we force its peer network, taking the strongly-augmented version of the same image as input, to generate prediction close to the anchoring label for knowledge distillation. By observing the loss distribution, we use a mixture model to dynamically estimate the clean probability of each training sample and generate a confidence clean set. Then we train both networks simultaneously by the clean set to minimize our loss function which contains unsupervised matching loss (i.e., measure the consistency of the two networks) and supervised classification loss (i.e. measure the classification performance). We theoretically analyze the gradient of our loss function to show that it implicitly prevents memorization of the wrong labels. Experiments on two simulated benchmarks and one real-world dataset demonstrate that our approach achieves substantial improvements over the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → *Ensemble methods*; *Supervised learning by classification*; *Neural networks*; *Mixture modeling*; Uncertainty quantification.

KEYWORDS

noisy labels, image classification, ensemble learning, weakly supervised learning

ACM Reference Format:
Yangdi Lu, Yang Bo, and Wenbo He. 2022. An Ensemble Model for Combating Label Noise. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3488560.3498376

WSDM '22, February 21–25, 2022, Tempe, AZ, USA.

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498376

1 INTRODUCTION

Deep Neural Networks (DNNs) have become the par excellence approach to deal with a variety of computer vision tasks [19, 24]. However, the superior performance comes with the cost of requiring a large-scale training dataset with high-quality annotations. It is difficult to attain such strong supervision information due to the high cost of the manually labeling process. Hence, we turn to web services to obtain large-scale training data with labels, such as querying online search engines [22], collecting online websites images with surrounding texts [42], downloading social media images with tags [29] or crowdsourcing [45]. However, these approaches usually inevitably introduce label noise. For example, [42] collects 1 million training images from online shopping websites and generates their labels from surrounding texts with 38.5% estimated label noise. Previous studies [2, 46] demonstrate DNN can memorize noisy labels easily and generalize poorly on clean test data. Therefore, mitigating the effects of noisy labels has attracted considerable attention especially when the training data are from web resource.

To handle noisy labels, most approaches focus on estimating the noise transition matrix [11, 31, 41] and correcting the label according to model prediction [28, 33, 36, 43]. Another promising direction of study is based on sample selection, which trains two networks simultaneously by using small-loss instances [5, 13, 40, 44]. For instance, Decoupling [30] and Co-teaching+ [44] introduce the "Disagreement" strategy to keep the two networks diverged to achieve better ensemble effects. However, the samples selected by "Disagreement" strategy are not guaranteed to have correct labels [40], resulting in only a small portion of clean samples being utilized in the training process. Co-teaching [13] and JoCoR [40] aim to reduce the divergence between two different networks so that the number of clean labels utilized in each mini-batch increases. In the beginning, two networks with different learning abilities filter out different types of error. However, with the increasing training epoch number, two networks gradually converge to a consensus and even make the wrong predictions consistently. Besides, these methods rely on a known noise rate to accurately select the small-loss samples in each mini-batch, which is usually impractical.

To address the above concerns, it is crucial to keep the balance between divergence and consistency of two networks through the whole training procedure. In this paper, we propose an ensemble model for robust learning with noisy labels. Specifically, we use weak (e.g. using only crop-and-flip) and strong (e.g. using RandAugment [8]) augmentations for two networks respectively to avoid the consensus of their predictions. The stronger augmentation results in disparate prediction compared to the weak one, which guarantees the better ensemble effect due to the divergence between two networks. Aside from the supervised classification loss, we propose an unsupervised matching loss to keep the consistency of predictions from two networks without using noisy labels. According

to the observation of loss distribution in early-learning stage, we propose a method based on a mixture model to dynamically select the confident clean samples without the need to estimate noise rates. In this way, the influence of mislabeled samples in the process of model learning is mitigated, and our framework is inherently noise-tolerant. The main contributions are summarized as follows:

- We propose an framework with two networks fed with different augmented inputs to achieve the better ensemble effect, wherein an unsupervised matching loss is used to mitigate the influence of noisy labels and improve the generalization.
- We analyze the gradient of our loss function and show that the gradient term derived from unsupervised matching loss corrects the gradient of the cross-entropy loss.
- Different from the existing approaches that require estimating noise rates, we fit a mixture model to loss distribution to select the high-confident clean samples for network parameter update, making our method applicable in real life.
- We conduct extensive experiments on both simulated and real-world noisy datasets. Experiments show that our approach significantly advances state-of-the-art results. We study the effect of data augmentations and provide an ablation study to examine the influence of different components.

2 RELATED WORK

Numerous methods have been proposed for robust classification with noisy labels. Herein, we briefly review the relevant existing approaches.

Curriculum learning. Inspired from human cognition, Curriculum learning (CL) [3] proposes to start from easy samples and go through harder samples to improve convergence and generalization. In the noisy label scenario, easy (hard) concepts are associated with clean (noisy) samples. Based on CL, [34] leverages an additional validation set to adaptively assign weights to noisy samples for less loss contribution in every iteration.

Sample selection. Another set of emerging methods aim to select the clean labels out of the noisy ones to guide the training. Previous work [2] empirically demonstrates the *early-learning* phenomenon that DNNs tend to learn clean labels before memorizing noisy labels during training, which justifies that instances with small-loss values are more likely to be clean instances. Based on this observation, [26] proposes a curriculum loss that chooses samples with small-loss values for loss calculation. MentorNet [17] pre-trains a mentor network for selecting small-loss instances to guide the training of the student network. Nevertheless, similar to Self-learning approach, MentorNet inherits the same inferiority of accumulated error caused by the sample-selection bias.

Two classifiers with Disagreement and Agreement. Inspired by Co-training [4], Co-teaching [13] symmetrically trains two networks by selecting small-loss instances in a mini-batch for updating the parameters. These two networks could filter different types of errors brought by noisy labels since they have different learning abilities. When the error from noisy data flows into the peer network, it will attenuate this error due to its robustness [13]. However, two networks converge to a consensus gradually with the increase of epochs. To tackle this issue, Decoupling [30] and Co-teaching+ [44] introduce the "Update by Disagreement" strategy which conducts updates only on selected instances, where there is a prediction disagreement between two classifiers. Through this, the decision of "when to update" depends on a disagreement between two networks instead of depending on the noisy labels. As a result, it would reduce the dependency on noisy labels as well as keep two networks divergent. However, as noisy labels are spread across the whole space of examples, there may be very few clean labels in the disagreement area. Thus, JoCoR [40] suggests jointly training two networks with the instances that have prediction agreement between two networks. However, the two networks in JoCoR are also prone to converge to a consensus and even make the same wrong predictions when datasets are under high noise ratio.

Other methods. Some approaches focus on creating noise-tolerant loss functions [10, 39, 48]. Other methods attempt to correct the loss [1, 15, 28, 31, 33, 35, 36, 38]. Many approaches [9, 20, 36] have been proposed to combat noisy labels through semi-supervised learning. These approaches [16, 23, 25] introduce the regularization term to avoid memorization of noisy labels.

Our proposed approach is related to sample selection using two networks. However, it is also fundamentally different from existing methods. Instead of only using classification loss, we propose an extra matching loss to reduce the effect of noisy labels. We feed different augmented images to two networks respectively, yielding a stronger generalization ability. Beyond heuristic design, we provide the gradient analysis to explain how our loss function avoids memorization of the mislabeled samples. Different from selecting small-loss samples by using a fixed ratio, we use a mixture model to dynamically select the high-confident clean samples.

3 METHODOLOGY

3.1 Background

Our work aims to develop an algorithm to learn a classifier that achieves robust performance on the test set even the provided training data contains noisy labels. Consider the C-class classification problem in noisy label scenario, we have a training set $D = \{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is an input and $\hat{\boldsymbol{y}}_i \in \{0,1\}^C$ is one-hot vector corresponding to $\boldsymbol{x}_i$. In the noisy label scenario, true label $\boldsymbol{y}_i$ is not observable in practice. The classification model maps each input $\boldsymbol{x}_i$ to a C-dimensional logits using a DNN model $\mathcal{M}_\Theta$ and then feeds the logits into a softmax function to produce $\boldsymbol{p}_i$ of the conditional probability of each class.

$$\boldsymbol{p}_i = \texttt{softmax}(\mathcal{M}_\Theta(\boldsymbol{x}_i)) = \frac{e^{\mathcal{M}_\Theta(\boldsymbol{x}_i)}}{\sum_{c=1}^{C} e^{(\mathcal{M}_\Theta(\boldsymbol{x}_i))_c}}. \tag{1}$$

Θ denotes the parameters of the DNN and $(\mathcal{M}_\Theta(\boldsymbol{x}_i))_c$ denotes the c-th entry of logits $\mathcal{M}_\Theta(\boldsymbol{x}_i)$. Then the DNN is trained via the cross-entropy loss to measure how the model fits the training set D.

$$\ell_{ce}(D, \Theta) = -\frac{1}{N}\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i^T \log(\boldsymbol{p}_i). \tag{2}$$

However, as noisy label $\hat{\boldsymbol{y}}_i$ is likely to be wrong, the model gradually memorizes the training samples with wrong labels when optimizing ℓ_{ce}. Existing studies [2, 46] have observed that DNNs fully overfit

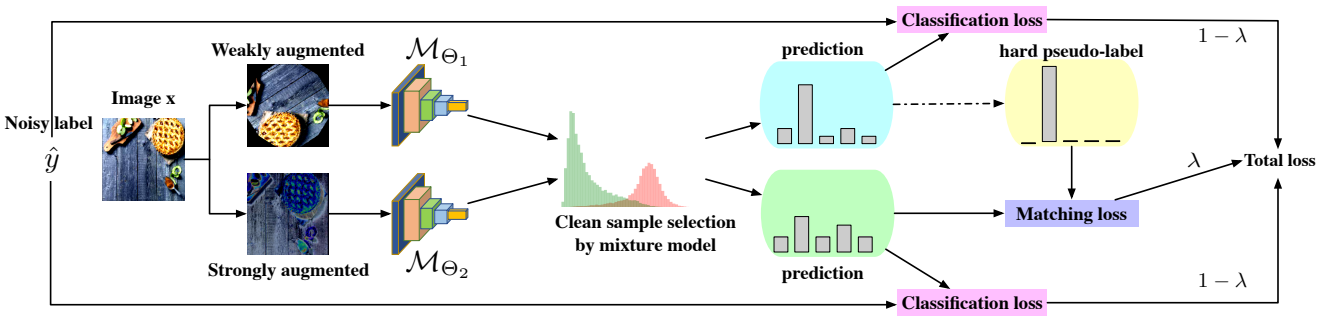

Figure 1: Our approach trains two networks ($\mathcal{M}_{\Theta_1}$ and $\mathcal{M}_{\Theta_2}$) simultaneously. A weakly-augmented version of an image x (top) is fed into the model $\mathcal{M}_{\Theta_1}$ to obtain its prediction (blue box). We convert the prediction to a one-hot hard pseudo-label as an anchoring label (yellow box). Then, we compute second model's prediction (green box) for a strongly-augmented version of the same image (bottom). The models are trained on total loss (i.e. the linear convex combination of the unsupervised matching loss and classification loss) to make the prediction on the strongly-augmented version match the anchoring label. For parameter update, we fit the Gaussian Mixture Model (GMM) to loss distribution for distinguishing the clean samples, which ensuring error caused by noisy labels would not be accumulated.

to noisy labels during training, causing the classification performance degradation. In addition, [21, 23] have observed and also theoretically proved that when trained on noisy labels, DNNs first fit the training data with clean labels during an *early learning* phase, before eventually *memorizing* the training data with wrong labels. This *early learning* phenomenon also reflects on the loss distribution which motivates us to develop an approach to select clean samples in Section 3.4.

3.2 Our approach: Co-matching

A diagram of our approach is shown in Figure 1. We name our approach *Co-matching* as it **Co**-trains two deep networks by minimizing the loss contains an unsupervised **matching** loss term. We denote the two deep networks in our model as $\mathcal{M}_{\Theta_1}$ and $\mathcal{M}_{\Theta_2}$ with parameters Θ_1 and Θ_2 respectively. Thus $\texttt{softmax}(\mathcal{M}_{\Theta_1}(\boldsymbol{x}_i))$ and $\texttt{softmax}(\mathcal{M}_{\Theta_2}(\boldsymbol{x}_i))$ are the softmax probabilities for input $\boldsymbol{x}_i$ produced by $\mathcal{M}_{\Theta_1}$ and $\mathcal{M}_{\Theta_2}$. For the model inputs, we perform two types of augmentation for each network: weak and strong, denoted by $\alpha(\cdot)$ and $\mathcal{A}(\cdot)$ respectively. We will describe the forms of augmentation used for $\mathcal{A}(\cdot)$ and $\alpha(\cdot)$ in section 3.5. For notation simplicity, we denote $\boldsymbol{p}_i^{\Theta_1}$ and $\boldsymbol{p}_i^{\Theta_2}$ as abbreviations for predictions, i.e., $\texttt{softmax}(\mathcal{M}_{\Theta_1}(\alpha(\boldsymbol{x}_i)))$ and $\texttt{softmax}(\mathcal{M}_{\Theta_2}(\mathcal{A}(\boldsymbol{x}_i)))$ respectively.

As the DNN model can easily overfit the noisy labels when trained with standard cross-entropy loss, resulting in poor classification performance. Our basic idea is to reduce the dependence of loss function on noisy labels. Therefore, the loss function in Co-matching exclusively consists of two loss terms: a supervised loss ℓ_c for classification task and an unsupervised matching loss ℓ_m for augmentation anchoring (i.e. encourage the two models to output the consistent predictions on different augmented data). So our total loss on dataset D is calculated as follows:

$$\mathcal{L}(D,\Theta_1,\Theta_2) = (1-\lambda)\ell_c(D,\Theta_1,\Theta_2) + \lambda\ell_m(\boldsymbol{x},\Theta_1,\Theta_2), \quad (3)$$

where $\lambda \in [0,1]$ is a fixed scalar hyperparameter controlling the importance weight of the two loss terms. Since there are correctly-labeled samples remaining in noisy training data, our classification loss ℓ_c is the traditional cross-entropy loss over two models.

$$\begin{aligned}\ell_c(D,\Theta_1,\Theta_2) &= \ell_{ce}(D,\Theta_1) + \ell_{ce}(D,\Theta_2)\\ &= -\frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i^T\log(\boldsymbol{p}_i^{\Theta_1}) - \frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i^T\log(\boldsymbol{p}_i^{\Theta_2})\\ &= -\frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i^T\log(\boldsymbol{p}_i^{\Theta_1}\odot\boldsymbol{p}_i^{\Theta_2}).\end{aligned} \quad (4)$$

Compared to cross-entropy loss in Eq. (2), our classification loss term ℓ_c is more resistant to noisy labels. On the one hand, assume $\hat{\boldsymbol{y}}_i$ is a correct label for $\boldsymbol{x}_i$, when minimizing ℓ_c, both $\boldsymbol{p}_i^{\Theta_1}$ and $\boldsymbol{p}_i^{\Theta_2}$ are updated toward $\hat{\boldsymbol{y}}_i$. To gain the optimal result of Hadamard product between $\boldsymbol{p}_i^{\Theta_1}$ and $\boldsymbol{p}_i^{\Theta_2}$, two models are required to produce more confident and consistent predictions close to $\hat{\boldsymbol{y}}_i$. On the other hand, assume $\hat{\boldsymbol{y}}_i$ is a wrong label for $\boldsymbol{x}_i$. Weakly augmented input can result in prediction $\boldsymbol{p}_i^{\Theta_1}$ close to $\hat{\boldsymbol{y}}_i$, while strongly augmented input generates disparate prediction $\boldsymbol{p}_i^{\Theta_2}$ compare to the weak one. It makes the result of Hadamard product between $\boldsymbol{p}_i^{\Theta_1}$ and $\boldsymbol{p}_i^{\Theta_2}$ more difficult to reach the wrong label $\hat{\boldsymbol{y}}_i$, resulting in mitigating the effect of overfitting noisy labels in both networks. Besides, it also explains that Co-matching can always keep two networks diverged throughout the whole training to achieve better ensemble effects. Nevertheless, solely keeping the divergence of two networks may not promote the learning ability to select clean samples, which is the main drawback of Co-teaching+ [44]. In addition, under high level of label noise (i.e. most of the nois labels $\hat{\boldsymbol{y}}_i$ are wrong), it is difficult for the supervised loss ℓ_c to learn a robust classifier. This motivates us to develop an extra loss term that does not require using noisy label $\hat{\boldsymbol{y}}_i$ but still improves the generalization ability.

Our basic idea is to use the model's prediction for a weakly augmented input as the target label for the strongly augmented version of the same image. This maximizes the consistency of the two networks resulting in helping the model find a wider minimum and provides better generalization performance. In Co-matching,

we compute an *anchor* (or anchoring label) for each sample by the prediction of model $\mathcal{M}_{\Theta_1}$. To obtain an anchoring label of given image $\boldsymbol{x}_i$, we get the predicted class distribution from model $\mathcal{M}_{\Theta_1}$ given a weakly-augmented version of the image: $\boldsymbol{p}_i^{\Theta_1}$ and $\boldsymbol{p}_i^{\Theta_1} = [(p_i^{\Theta_1})_1, (p_i^{\Theta_1})_2, \cdots, (p_i^{\Theta_1})_C]$. Then, we use hard pseudo-labeling way to get $\boldsymbol{t}_i^{\Theta_1}$ as the anchoring label.

$$(t_i^{\Theta_1})_j = \begin{cases} 1 & \text{if } j = \arg\max_c (p_i^{\Theta_1})_c \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The use of hard pseudo-labeling has the similar function to entropy maximization [12], where the model's predictions are encouraged to be low-entropy (i.e., high-confidence). Besides, the hard pseudo-labeling is likely to reduce the negative effect of knowledge distillation caused by noisy labels in early learning stage. The anchoring label is used as the target probability for the prediction of a strongly-augmented version of same image in $\mathcal{M}_{\Theta_2}$. However, not all samples are suitable to generate anchoring labels, as hard pseudo-label of low-confident noisy samples may introduce unstable inconsistency. Thus, we develop an approach to select high-confident clean samples in Section 3.4.

We use the standard cross-entropy loss rather than mean squared error or Jensen-Shannon Divergence as it maintains stability and simplifies implementation. Thus, the unsupervised matching loss is

$$\ell_m(\boldsymbol{x}, \Theta_1, \Theta_2) = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{t}_i^{\Theta_1} \log(\boldsymbol{p}_i^{\Theta_2}). \tag{6}$$

Therefore, by minimizing the total loss in Eq. (3), the model consistently improves the generalization performance under different levels of label noise. Under low-level label noise (i.e., 20%), supervised classification loss $\ell_c(D, \Theta_1, \Theta_2)$ takes the lead. Co-matching tends to learn from the most correctly-labeled samples. Under high-level label noise (i.e., 80%), unsupervised matching loss $\ell_m(\boldsymbol{x}, \Theta_1, \Theta_2)$ takes the lead such that Co-matching inclines to maximize the consistency of the networks to improves the generalization without requiring noisy labels.

3.3 Theoretical Analysis on Loss Function

We explain how our loss function can effectively prevent the model memorizing the mislabeled samples by analyzing the gradient. We first explain how the standard cross-entropy loss in Eq. (2) memorizes samples with wrong labels. The gradient of cross-entropy loss with respect to Θ equals

$$\nabla \ell_{ce}(D, \Theta) = -\frac{1}{N}\sum_{i=1}^{N} \nabla \mathcal{M}_{\Theta}(\boldsymbol{x}_i)(\boldsymbol{p}_i - \hat{\boldsymbol{y}}_i), \tag{7}$$

where $\nabla \mathcal{M}_{\Theta}(\boldsymbol{x}_i)$ is the Jacobian matrix of the DNN logits for the i-th input with respect to Θ. In clean training data scenario, $\boldsymbol{p}_i - \hat{\boldsymbol{y}}_i$ of true class entry will always be negative and rest entries are positive. Therefore, performing stochastic gradient descent increases the probability of true class and reduces the residual probabilities at other entries. However, in noisy-label scenario, if c is the true class, but c-th entry of noisy label $(\hat{y}_i)_c = 0$, then the contribution of the i-th sample to $\nabla \ell_{ce}(D, \Theta)$ is reversed (i.e. $(p_i - \hat{y}_i)_c$ should be negative but get positive instead). In the meanwhile, the entry corresponding to the impostor class c', is also reversed because

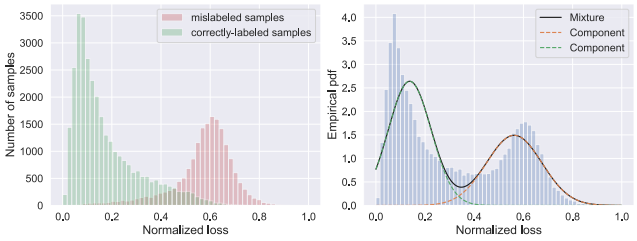

Figure 2: Train on CIFAR-10 with 40% label noise after 10 epochs with cross-entropy loss. Left: The ground truth normalized loss distribution. Right: The pdf of mixture model and two components after fitting a two component GMM to loss distribution.

$(\hat{y}_i)_{c'} = 1$. Therefore, performing stochastic gradient descent eventually results in memorization of mislabeled samples.

However, our loss function can counteract this influence. Similarly, we derive the gradient of simplified $\mathcal{L}(D, \Theta_1, \Theta_2)$ ($\lambda = 0.5$) with respect to Θ_1 and Θ_2 equals

$$\begin{aligned} \nabla \mathcal{L}(D, \Theta_1, \Theta_2) &= -\frac{1}{N}\sum_{i=1}^{N} \nabla \mathcal{M}_{\Theta_1,\Theta_2}(\boldsymbol{x}_i)\Big(\frac{\boldsymbol{p}_i^{\Theta_1} + \boldsymbol{p}_i^{\Theta_2}}{2} - \hat{\boldsymbol{y}}_i + \\ &\qquad \frac{\boldsymbol{p}_i^{\Theta_2} - \boldsymbol{t}_i^{\Theta_1}}{2}\Big) \\ &= -\frac{1}{N}\sum_{i=1}^{N} \nabla \mathcal{M}_{\Theta_1,\Theta_2}(\boldsymbol{x}_i)\Big(\boldsymbol{p}_i^{\Theta_2} - \hat{\boldsymbol{y}}_i + \frac{\boldsymbol{p}_i^{\Theta_1} - \boldsymbol{t}_i^{\Theta_1}}{2}\Big) \end{aligned} \tag{8}$$

where $\nabla \mathcal{M}_{\Theta_1,\Theta_2}(\boldsymbol{x}_i)$ is the Jacobian matrix of the DNNs logits for the i-th input with respect to Θ_1 and Θ_2. If c is the true class, since $\boldsymbol{t}_i^{\Theta_1}$ is calculated by hard pseudo-labeling in Eq. (5), then the c-th entry of $\boldsymbol{p}_i^{\Theta_1} - \boldsymbol{t}_i^{\Theta_1}$ is negative. Compared to the gradient of cross-entropy loss, we have an additional negative term $(\boldsymbol{p}_i^{\Theta_1} - \boldsymbol{t}_i^{\Theta_1})/2$ to adjust the gradient coefficients, which is useful both for correctly-labeled and mislabeled samples. For correctly-labeled samples, the first term $\boldsymbol{p}_i^{\Theta_2} - \hat{\boldsymbol{y}}_i$ vanishes after the early-learning stage, allowing the mislabeled samples to dominate the gradient. Adding the negative term $(\boldsymbol{p}_i^{\Theta_1} - \boldsymbol{t}_i^{\Theta_1})/2$ counteracts this effect by ensuring that the magnitudes of the coefficients on correctly-labeled samples remains large. For mislabeled samples, the c-th entry of first term $(\boldsymbol{p}_i^{\Theta_2} - \hat{\boldsymbol{y}}_i)_c$ is positive because $(\hat{\boldsymbol{y}}_i)_c = 0$. Adding the negative term $((\boldsymbol{p}_i^{\Theta_1} - \boldsymbol{t}_i^{\Theta_1})/2)_c$ dampens the coefficients on these mislabeled samples, thereby diminishing their effect on the gradient. Thus, our loss function boosts the gradient of correctly-labeled samples and neutralizes the gradient of mislabeled samples, which prevents the second model memorizing the noisy labels.

3.4 Clean Sample Selection

Since DNNs learn clean patterns before memorizing noisy labels [2], small-loss instances are more likely to be the ones that are correctly labeled [13]. Training the model only using correctly-labeled instances in each mini-batch data would be resistant to

noisy labels. This approach is named as "small-loss" trick and is widely used in existing works [13, 40, 44]. However, it requires a given noise rate or estimating the noise rate using additional steps. In this paper, we observe that the correctly-labeled samples can be distinguished from the loss distribution alone. To estimate the probability of being correctly-labeled sample, we introduce a two component Gaussian Mixture Model (GMM) [32] to fit the normalized loss distribution as shown in Figure 2. The probability density function (pdf) of GMM with K components on the loss ℓ can be defined as

$$P(\ell) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\ell \mid \mu_k, \sigma_k^2), \quad \sum_{k=1}^{K} \pi_k = 1, \tag{9}$$

where π_k are the mixing coefficient for the linear convex combination of each individual pdf $\mathcal{N}(\ell \mid \mu_k, \sigma_k^2)$. In our case, we use an Expectation-Maximization (EM) algorithm to estimate the π_k, μ_k and σ_k^2. Therefore, we can obtain the probability of a sample being correctly-labeled or mislabeled through the posterior probability:

$$P(k \mid \ell) = \frac{P(k)P(\ell \mid k)}{P(\ell)} = \frac{\pi_k \mathcal{N}(\ell \mid \mu_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\ell \mid \mu_k, \sigma_k^2)} \tag{10}$$

where $k = 0(1)$ indicate correct (wrong) labels. Note that we always calculate the cross-entropy loss to estimate the clean probability for all samples after every epoch. But we use our loss defined in Eq. (3) for training the model which contains other loss term to deal with label noise. Then we select the clean samples in n-th mini-batch D_n for updating the network parameters as follows

$$\hat{D}_n = \{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i) \mid P(k = 0 \mid \ell_{ce}(\boldsymbol{x}_i)) > 0.5 \text{ and } (\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i) \in D_n\}. \tag{11}$$

After obtaining the confident clean samples set $\hat{D}_n$ in mini-batch n, we calculate the loss for these examples to further conduct back propagation.

$$\mathcal{L}(\hat{D}_n, \Theta_1, \Theta_2) = -\frac{1}{|\hat{D}_n|} \sum_{(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i) \in \hat{D}_n} \hat{\boldsymbol{y}}_i^T \log(\boldsymbol{p}_i^{\Theta_1} \odot \boldsymbol{p}_i^{\Theta_2}) + \boldsymbol{t}_i^{\Theta_1} \log(\boldsymbol{p}_i^{\Theta_2}). \tag{12}$$

Therefore, our approach effectively select the clean samples without a need to estimate the noise rate. Put all these together, our algorithm is described in Algorithm 1. We also compare Co-matching to other existing approaches in Appendix A.3.

3.5 Augmentations in Co-matching

Our framework leverages two kinds of augmentations: "weak" and "strong". In our experiments, weak augmentation is a standard crop-and-flip augmentation strategy. Specifically, we randomly crop the images and flip them horizontally with a probability of 50% on all datasets. As for "strong" augmentation, we adopt RandAugment [8], which is based on AutoAugment [7]. AutoAugment learns an augmentation strategy based on transformations from the Python Imaging Libraries [1] using reinforcement learning. Given a collection of transformations (e.g., color inversion, contrast adjustment, translation, etc.), RandAugment randomly selects M transformations for each sample in a mini-batch. As originally proposed, RandAugment uses a single fixed global magnitude that controls the severity of all distortions [8]. Instead of optimizing the hyperparameter magnitude by using grid search, we find that sampling a random magnitude from a pre-defined range at each training step (instead of using a fixed global value) works better for learning with noisy labels. The implementation details are in Section 4.3.

[1] https://www.pythonware.com/products/pil/

Algorithm 1: Co-matching

Input: two networks $\mathcal{M}_{\Theta_1}$ and $\mathcal{M}_{\Theta_2}$ with parameters $\Theta = \{\Theta_1, \Theta_2\}$, weak augmentation $\alpha(\cdot)$, strong augmentation $\mathcal{A}(\cdot)$, importance weight λ, training set D, batch size B, learning rate η, number of training epochs T;

$\Theta_1, \Theta_2 =$ Warmup(D, Θ_1, Θ_1); `// train with cross-entropy loss 10 epochs for warmup.`

for $t = 1, 2, \ldots, T$ **do**

  $P(k = 0 \mid \ell_{ce}(\boldsymbol{x}_i)) =$ GMM $(D, \ell_{ce}, \Theta_1, \Theta_2)$; `// model loss distibution to obtain clean probability for each sample.`

  Shuffle D into $\frac{|D|}{B}$ mini-batches ;

  for $n = 1, 2, \ldots, \frac{|D|}{B}$ **do**

    Fetch n-th mini-batch D_n from D ;

    Calculate the prediction $\boldsymbol{p}^{\Theta_1} = \texttt{softmax}(\mathcal{M}_{\Theta_1}(\alpha(\boldsymbol{x})))$, $\forall \boldsymbol{x} \in D_n$;

    Calculate the prediction $\boldsymbol{p}^{\Theta_2} = \texttt{softmax}(\mathcal{M}_{\Theta_2}(\mathcal{A}(\boldsymbol{x})))$, $\forall \boldsymbol{x} \in D_n$;

    Calculate the anchoring label $\boldsymbol{t}^{\Theta_1}$ by Eq. (5) ;

    Obtain confident clean samples by Eq. (11) ;

    Calculate the loss by Eq. (12) ;

    Update $\Theta = \Theta - \eta \nabla \mathcal{L}(\hat{D}_n, \Theta_1, \Theta_2)$;

Output Θ_1 and Θ_2.

4 EXPERIMENTS

4.1 Experimental Settings

We evaluate our method on two benchmarks with simulated label noise, CIFAR-10 and CIFAR-100 [18], and one real-world dataset, Clothing1M [42]. Clothing1M consists of 1 million training images collected from online shopping websites with noisy labels generated from surrounding texts. CIFAR-10 and CIFAR-100 are initially clean. Following [31], we corrupt the datasets by label transition matrix Q, where $Q_{ij} = \Pr[\hat{y} = j | y = i]$ given that noisy label $\hat{y}$ is flipped from clean label y. The matrix Q has two representative label noise models: (1) Symmetric flipping [37] is generated by uniformly flipping the label to one of the other class label; (2) Asymmetric flipping [31] is a simulation of fine-grained classification with noisy labels in the real world, where the mistakes only occur within very similar classes. More details are described in Appendix A.1.

Networks and optimizer. For CIFAR-10 and CIFAR-100, we use a 7-layer network architecture for fair comparison with [40]. The Adam optimizer (momentum=0.9) is used with an initial learning rate of 0.001, and the batch size is set to 128. We run 200 epochs in total and linearly decay learning rate to zero from 80 to 200 epochs. As for Clothing1M, we accept ResNet18 [14] with ImageNet pretrained weights and use Adam optimizer (momentum=0.9) with a batch size of 64. We run 20 epochs in total and set learning rate to 8×10^{-4}, 5×10^{-4} and 5×10^{-5} for 5, 5 and 10 epochs respectively.

Table 1: Average test accuracy (%) on CIFAR-10 and CIFAR-100 over the last 10 epochs. The results (mean ± std) are reported over 5 random runs and best results are in bold.

			Same Category Methods				Other Category Methods			
	Noise ratio/Method	Standard	Decoupling	Co-teaching	Co-teaching+	JoCoR	GCE	SL	APL	Co-Matching (Ours)
CIFAR-10	Symmetric-20%	69.57 ± 0.20	69.55 ± 0.20	78.07 ± 0.24	78.66 ± 0.20	85.69 ± 0.06	89.93 ± 0.08	89.13 ± 0.16	85.54 ± 0.51	**90.07 ± 0.16**
	Symmetric-50%	42.48 ± 0.35	41.44 ± 0.46	71.54 ± 0.17	57.13 ± 0.46	79.32 ± 0.37	81.38 ± 0.06	79.76 ± 0.20	80.66 ± 0.13	**87.63 ± 0.22**
	Symmetric-80%	15.79 ± 0.37	15.64 ± 0.42	27.71 ± 4.39	24.13 ± 5.54	25.97 ± 3.11	44.33 ± 0.15	53.62 ± 0.37	44.19 ± 4.40	**58.86 ± 1.54**
	Asymmetric-40%	69.36 ± 0.23	69.46 ± 0.08	73.75 ± 0.34	69.03 ± 0.30	76.38 ± 0.32	74.17 ± 0.45	74.39 ± 0.75	78.37 ± 0.02	**82.32 ± 0.57**
CIFAR-100	Symmetric-20%	35.46 ± 0.25	33.21 ± 0.22	43.71 ± 0.20	49.15 ± 0.24	52.43 ± 0.20	57.83 ± 0.73	47.60 ± 0.34	59.92 ± 0.48	**60.47 ± 0.29**
	Symmetric-50%	16.87 ± 0.13	15.03 ± 0.33	34.30 ± 0.39	39.08 ± 0.73	42.73 ± 0.96	49.24 ± 0.30	31.66 ± 1.43	52.26 ± 0.37	**53.81 ± 0.56**
	Symmetric-80%	4.08 ± 0.21	3.80 ± 0.01	14.95 ± 0.15	15.00 ± 0.42	14.41 ± 0.60	32.18 ± 0.27	13.51 ± 0.82	26.28 ± 2.05	**35.23 ± 0.82**
	Asymmetric-40%	27.23 ± 0.45	26.25 ± 0.27	28.27 ± 0.22	30.45 ± 0.15	31.52 ± 0.31	40.02 ± 0.35	37.25 ± 0.16	**42.25 ± 0.30**	39.14 ± 0.37

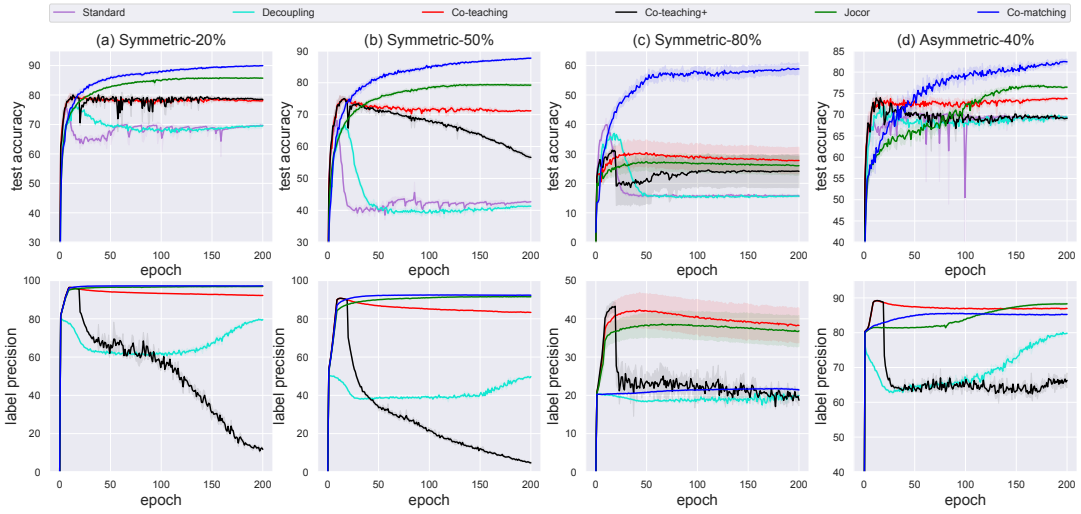

Figure 3: Results on CIFAR-10 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

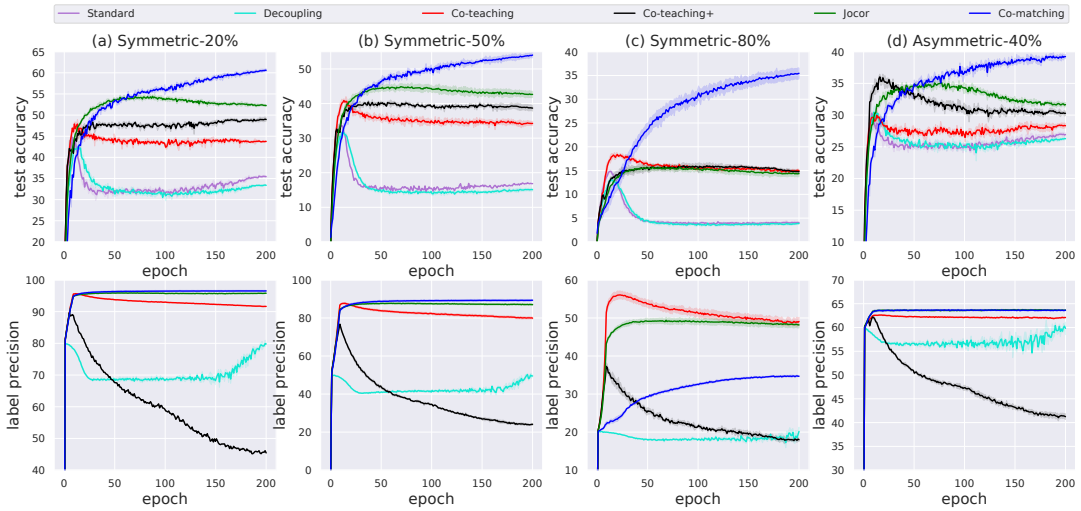

Figure 4: Results on CIFAR-100 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Metrics. To measure the performance, we use the test accuracy, i.e., *test accuracy* = (# *of correct predictions*) / (# *of test dataset*). Higher test accuracy means that the algorithm is more robust to the label noise. Following the [13, 40], we also calculate the label precision

in each mini-batch, i.e., *label precision* = (# *of clean labels*) / (# *of all selected labels*). Specifically, we attain the clean set by GMM in each mini-batch, and then calculate the ratio of clean samples in the clean set. Intuitively, an algorithm with higher label precision is also more robust to the label noise [13, 40]. However, we find that the higher label precision is not necessarily lead to higher test accuracy with extreme label noise in Co-matching, we will explore this phenomenon in Section 4.2.

Baselines. We compare Co-matching with four close-related methods, including Decoupling [30], Co-teaching [13], Co-teaching+ [44], JoCoR [40] and three methods from different category, including GCE [48], SL [39] and APL [27]. We implement all methods with same environment and default parameters by Pytorch. Note that all compared algorithms do not use extra techniques such as mixup [47] to improve the performance. All results are reported over five random runs. The error bar for standard deviation in each figure has been highlighted as shade.

4.2 Comparison with the State-of-the-Arts

Results on simulated datasets CIFAR-10 and CIFAR-100 We report the average test accuracy over the last 10 epochs of all methods in Table 1. Co-matching outperforms other methods by a large margin for almost all noise rates across all datasets. We also find that Co-matching is more effective when the noise rates are extremely high. For instance, on CIFAR-10 with Symmetric-80% label noise, Co-matching outperform the best baseline method by more than 5.24%. Within the methods from same category, JoCoR performs better than other baselines in all noise cases except the hardest Symmetric-80% case. It means JoCoR has reduced to Co-teaching in function and suffers the same problem which the two networks converge to a wrong consensus, resulting in making the wrong predictions consistently. Note that APL sometimes delivers a relatively good performance, as it has guaranteed robustness to asymmetric label noise.

The top rows of Figure 3 and Figure 4 show the test accuracy vs. epochs of close-related methods on CIFAR-10 and CIFAR-100. With different levels of symmetric and asymmetric label noise, we can clearly see the memorization effect of networks. i.e., test accuracy of Standard first reaches a very high level and then gradually decreases due to memorization of noisy labels. Thus, a robust training approach should alleviate or even stop the decreasing trend in test accuracy. On this point, the proposed approach Co-matching shows a clear advantage over other close-related methods, especially in the later stages of learning with noisy labels. The superior performance of Co-matching demonstrates that it prevents memorization of noisy labels throughout the whole training procedure, which consistently verifies our gradient analysis in Section 3.3.

We also plot label precision vs. epochs at the bottom row of Figure 3 and Figure 4. Only Decoupling, Co-teaching, Co-teaching+, JoCoR and Co-matching are considered here, as these methods include sample selection during training. First, we can see Co-matching, JoCoR and Co-teaching can successfully pick clean instances out in Symmetric-20 %, Symmetric-50% and Asymmetric-40% cases. Note that Co-matching not only reaches high label precision in these three cases but also performs better and better with the increase of epochs. Decoupling and Co-teaching+ fail in selecting clean samples, because “Disagreement" strategy does not guarantee to select clean samples, as mentioned in Section 2. However, an interesting phenomenon is that high label precision does not necessarily lead to high test accuracy under high-levels of label noise. For example, in Symmetric-80% case, the label precision of Co-matching is much lower than Co-teaching and JoCoR, while the test accuracy is higher than Co-teaching and JoCoR. Similarly, in all noise rates cases on CIFAR-100, Co-teaching has much higher label precision than Co-teaching+, while the test accuracy of Co-teaching is lower than Co-teaching+. The subset of clean samples selected by small-loss rule may not rich enough to generalize effectively to held-out data, we believe the samples near the margin with relatively larger loss contribute more towards improving the model's generalization.

Table 2: Test accuracy (%) on Clothing1M with ResNet18. Bold indicates best performance.

Methods	*best*	*last*
Standard	67.74	66.95
Decoupling	67.71	66.78
Co-teaching	69.05	68.99
Co-teaching+	67.84	67.68
JoCoR	70.30	69.79
SL	69.22	67.97
GCE	69.59	68.81
APL	69.92	68.84
Co-matching (Ours)	**71.16**	**70.78**

Results on real-world dataset Clothing1M. As shown in Table 2, *best* denotes the epoch where the validation accuracy is optimal, and *last* denotes the test accuracy at the end of training. Co-matching outperforms the state-of-the-art methods by a large margin on both *best* and *last*, e.g., improving the accuracy from 66.95% to 70.78% over Standard, better than best baseline JoCoR by 0.99%. This verifies the effectiveness of Co-matching against real-world label noise.

4.3 Composition of Data Augmentation

We study the impact of data augmentation systematically by considering several common augmentations. One type of augmentation involves spatial/geometric transformation, such as cropping, flipping, rotation and cutout. The other type of augmentation involves appearance transformation, such as color distortion (e.g. brightness and contrast) and Gaussian blur. Since Clothing1M images are of different sizes, we always use cropping as a base transformation. We explore various “weak" augmentation by combining cropping with other augmentations. As for “strong" augmentation, we use RandAugment [8], which randomly select M transformations from a set $\mathcal{S}$ for each sample in a mini-batch. We denote RandAugment as $\mathcal{S}(M)$. In our experiment, $\mathcal{S}$ = {*Contrast*, *Equalize*, *Invert*, *Rotate*, *Posterize*, *Solarize*, *Color*, *Brightness*, *Sharpness*, *ShearX*, *ShearY*, *Cutout*, *TranslateX*, *TranslateY*, *Gaussian Blur*}.

Figure 5 shows the results under composition of transformations. We observe that using “weak" augmentation for both models

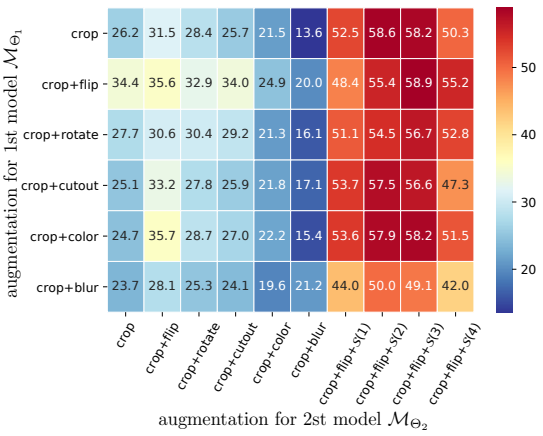

Figure 5: Test accuracy(%) over various combinations of augmentations on CIFAR-10 with Symmetric-80% label noise.

does not work much better than simple cropping. However, the performance of Co-matching benefits a lot by using stronger augmentations (e.g. add $\mathcal{S}(2)$ and $\mathcal{S}(3)$) on the second network $\mathcal{M}_{\Theta_2}$. We conclude that in the extreme label noise case, our loss function requires using stronger augmentation on model $\mathcal{M}_{\Theta_2}$ to constantly achieve its ensemble effect.

4.4 Ablation Study

In this section, we perform an ablation study to analyze the effect of each component in Co-matching, including the use of two networks, the use of joint update, the use of matching loss and the use of weak and strong augmentations. The experiments are conducted on CIFAR-10 with two cases: Symmetric-50% and Symmetric-80%. To verify the effect of using two networks and the use of joint update, we introduce Standard enhanced by "small-loss" selection (abbreviated as Standard+), Co-teaching and JoCoR to join the comparison. Besides, we simply set $\lambda = 0$ in Eq. (3) to see the influence of removing the matching loss (abbreviated as Co-matching-).

The results of their test accuracy vs. epochs are shown in Figure 6. In Symmetric-50% case, both Co-teaching and Standard+ keep a downward tendency after increasing to the highest point, which indicates they are still prone to memorizing noisy labels even with "small-loss" update. It also verifies the effect of using two networks as Co-teaching performs better than Standard+. JoCoR consistently outperforms Co-teaching, which verifies the conclusion in [40] that joint-update is more efficient than cross-update. However, things start to change in Symmetric-80% case. Co-teaching and Standard+ remain the same trend as these for Symmetric-50% case, but JoCoR performs unstable and even worse than Co-teaching and Standard+. This is likely because that, JoCoR uses the Jensen-Shannon (JS) Divergence to minimize the difference between two networks, resulting in the over consensus of two networks.

In both noise rates cases, Co-matching consistently outperforms Co-matching- and other methods, which validates that the use of

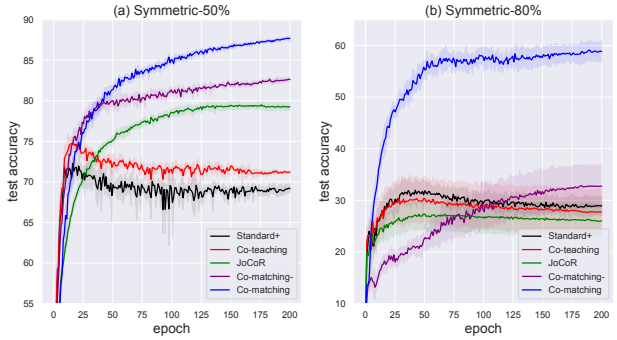

Figure 6: Results of ablation study on CIFAR-10

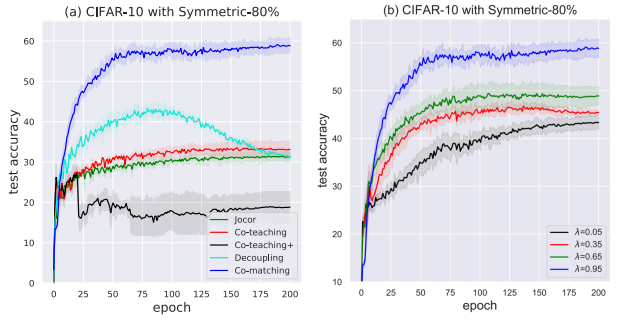

Figure 7: (a) Test accuracy of existing methods with same weak and strong augmentations. (b) Test accuracy of Co-matching with different hyperparameter λ.

matching loss can strongly prevent neural networks from memorizing noisy labels. To show the effect of weak and strong augmentations, we evaluate the state-of-art methods with the same augmentation strategy (i.e. weak and strong for each network respectively) as Co-matching. We conduct the experiments on CIFAR-10 with the hardest Symmetric-80% label noise. Figure 7 (a) shows the results. We observe that using weak and strong augmentations may not promise to improve performance for other methods. Co-teaching+ even performs worse. Figure 7 (b) shows the influence of λ. A larger λ gets a better accuracy in Symmetric-80% case. More results on hyperparameter sensitivity can be found in Appendix A.2.

5 CONCLUSION

In this paper, we identify the deficiencies of existing approaches and introduce a method for deep learning with noisy labels. Our method uses two networks with different strengths of augmented inputs to keep divergence and to achieve better ensemble effect. To avoid the influence of noisy labels, we introduce an unsupervised matching loss for knowledge distillation. In addition, we fit a mixture model to sample loss distribution to select the clean samples without the need of known noise rates. We provide the theoretical analysis on our loss functions and demonstrate the effectiveness of Co-matching on both benchmark and real-world datasets. We believe Co-matching is a promising framework for training robust DNNs against noisy labels from web services.

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. 2019. Unsupervised Label Noise Modeling and Loss Correction. In *International Conference on Machine Learning*. 312–321.

[2] Devansh Arpit, Stanislaw K Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron C Courville, Yoshua Bengio, et al. 2017. A Closer Look at Memorization in Deep Networks. In *ICML*.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[4] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*. 92–100.

[5] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels. In *International Conference on Machine Learning*. 1062–1070.

[6] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2021. Robustness of Accuracy Metric and its Inspirations in Learning with Noisy Labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11451–11461.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 113–123.

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.

[9] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. 2018. A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1215–1224.

[10] Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 1919–1925.

[11] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).

[12] Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*. 529–536.

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

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*. 10456–10465.

[16] Wei Hu, Zhiyuan Li, and Dingli Yu. 2019. Simple and Effective Regularization Methods for Training on Noisily Labeled Data with Generalization Guarantee. In *International Conference on Learning Representations*.

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

[18] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[20] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *International Conference on Learning Representations*.

[21] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*. PMLR, 4313–4324.

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

[23] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-Learning Regularization Prevents Memorization of Noisy Labels. *Advances in Neural Information Processing Systems* 33 (2020).

[24] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. 2019. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1950–1959.

[25] Yangdi Lu, Yang Bo, and Wenbo He. 2021. Confidence Adaptive Regularization for Deep Learning with Noisy Labels. *arXiv preprint arXiv:2108.08212* (2021).

[26] Yueming Lyu and Ivor W Tsang. 2019. Curriculum Loss: Robust Learning and Generalization against Label Corruption. In *International Conference on Learning Representations*.

[27] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*. PMLR, 6543–6553.

[28] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-Driven Learning with Noisy Labels. In *International Conference on Machine Learning*. 3355–3364.

[29] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*. 181–196.

[30] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*. 960–970.

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

[32] Haim Permuter, Joseph Francos, and Ian Jermyn. 2006. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern recognition* 39, 4 (2006), 695–706.

[33] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* (2014).

[34] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning*. 4334–4343.

[35] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*. 5907–5915.

[36] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5552–5560.

[37] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. 2015. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*. 10–18.

[38] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8688–8696.

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

[40] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. 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*. 13726–13735.

[41] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. 2019. Are Anchor Points Really Indispensable in Label-Noise Learning?. In *Advances in Neural Information Processing Systems*. 6838–6849.

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

[43] Kun Yi and Jianxin Wu. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7017–7025.

[44] X Yu, B Han, J Yao, G Niu, IW Tsang, and M Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *36th International Conference on Machine Learning, ICML 2019*.

[45] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. 2018. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 68–83.

[46] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (2021), 107–115.

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

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

A APPENDIX

A.1 Details of Datasets and Simulated Noise

The information of datasets are described in Table 3. For Clothing1M, it contains 1 million images of clothing obtained from online shopping websites with 14 classes: T-shirt, Shirt, Knitwear, Chiffon, Sweater, Hoodie, Windbreaker, Jacket, Down Coat, Suit, Shawl, Dress, Vest, and Underwear. The labels are generated by using the surrounding texts of the images that are provided by the sellers, and thus contain many wrong labels. The overall label noise level of Clothing1M is estimated at 38.46%, with some pairs of classes frequently confused with each other (e.g. Knitwear and Sweater). Note that we only use 14k and 10k clean data for validation and test. The 50k clean training data is not required during the training. As for simulating label noise, Figure 8 shows an example of noise transition matrix Q. As for simulated label noise, specifically, for CIFAR-10, the asymmetric noisy labels are generated by flipping *truck* $\rightarrow$ *automobile*, *bird* $\rightarrow$ *airplane*, *deer* $\rightarrow$ *horse* and *cat* $\leftrightarrow$ *dog*. For CIFAR-100, the noise flips each class into the next, circularly within super-classes.

Table 3: Summary of datasets used in the experiments.

	# of train	# of test	# of class	input size
CIFAR-10	50,000	10,000	10	32×32
CIFAR-100	50,000	10,000	100	32×32
Clothing1M	1,000,000	10,000	14	224×224

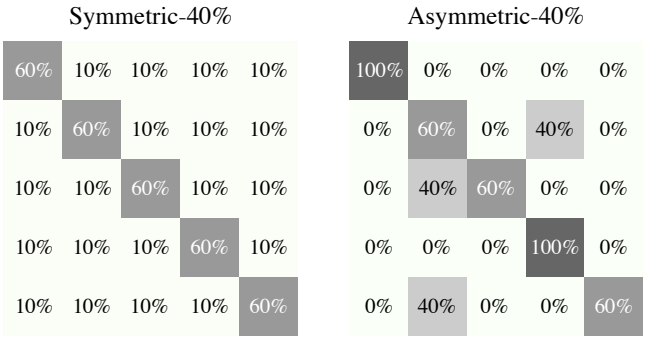

Figure 8: Example of noise transition matrix Q (taking 5 classes and noise ratio 0.4 as an example).

A.2 Hyperparameter Sensitivity

Co-matching only has one hyperparameter λ to control the importance weights of classification loss and matching loss. To tune the hyperparameter λ in our loss function Eq. (3), we search it in [0.05,0.35,0.65,0.95] with a noisy validation set for optimal performance due to the reliability of noisy validation set [6].

Figure 9 shows the influence of λ on CIFAR-10. In the CIFAR-10 with Symmetric-50% noise case, $\lambda = 0.35$ returns the best accuracy. Larger or smaller λ hurt the performance. In the CIFAR-10 with Symmetric-80% noise case, a larger λ gets a better accuracy, and

Table 4: Comparison of state-of-the-art and related techniques with our approach. In the first column, "cross update": updating parameters in a cross manner instead of a parallel manner; "joint update": updating the two networks parameters jointly. "divergence": keeping two classifiers diverged during the whole training procedure. "augmentation anchoring": encouraging the predictions of a strongly-augmented image to be close to the predictions from a weakly-augmented version of the same image. "noise rate": need a ground truth noise rate or an estimated noise rate.

	Decoupling	Co-teaching	Co-teaching+	JoCoR	Co-matching
cross update	×	√	√	×	×
joint update	×	×	×	√	√
divergence	√	×	√	×	√
augmentation anchoring	×	×	×	×	√
noise rate	×	√	√	√	×

$\lambda = 0.95$ achieves the best performance. It verifies the motivation of our loss function: under high-levels of label noise, the model is hard to get enough supervision from noisy labels if we only use the classification loss, more weights on matching loss is required to achieve good performance.

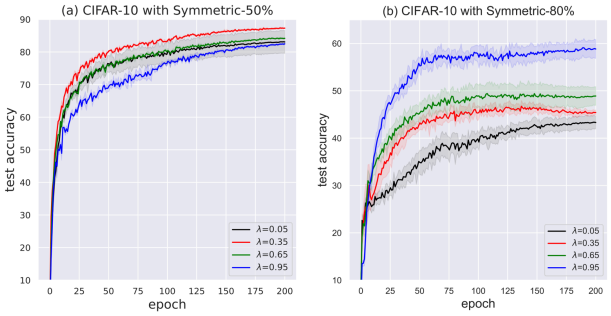

Figure 9: Results of Co-matching with different λ on CIFAR-10 with 50% and 80% symmetric label noise.

A.3 Comparision with Existing Methods

We compare Co-matching to other existing methods in Table 4.

Table 5: Average test accuracy (%) on CIFAR-10.

	Hard pseudo-labeling	Soft pseudo-labeling
Symmetric-20%	90.07 ± 0.16	89.81 ± 0.29
Symmetric-80%	58.86 ± 1.54	9.98 ± 0.00

A.4 Hard Pseudo-labeling in Matching Loss

Using hard pseudo-labeling for matching loss helps the model converge. We report the results in Table 5. We find that when the noise reaches to 80%, the Co-matching does not converge with soft pseudo-labeling. In order to improve the stability of our approach, we use hard pseudo-labeling for matching loss in Eq. (5).