Learning from Noisy Data with Robust Representation Learning

Junnan Li Caiming Xiong Steven C.H. Hoi
Salesforce Research
{junnan.li,cxiong,shoi}@salesforce.com

Abstract

Learning from noisy data has attracted much attention, where most methods focus on label noise. In this work, we propose a new learning framework which simultaneously addresses three types of noise commonly seen in real-world data: label noise, out-of-distribution input, and input corruption. In contrast to most existing methods, we combat noise by learning robust representation. Specifically, we embed images into a low-dimensional subspace, and regularize the geometric structure of the subspace with robust contrastive learning, which includes an unsupervised consistency loss and a supervised mixup prototypical loss. We also propose a new noise cleaning method which leverages the learned representation to enforce a smoothness constraint on neighboring samples. Experiments on multiple benchmarks demonstrate state-of-the-art performance of our method and robustness of the learned representation. Code is available at https://github.com/salesforce/RRL/.

1. Introduction

Data in real life is *noisy*. However, deep models with remarkable performance are mostly trained on clean datasets with high-quality human annotations. Manual data cleaning and labeling is an expensive process that is difficult to scale. On the other hand, there exists almost infinite amount of noisy data online. It is crucial that deep neural networks (DNNs) could harvest noisy training data. However, it has been shown that DNNs are susceptible to overfitting to noise [43].

As shown in Figure 1, a real-world noisy image dataset often consists of multiple types of noise. *Label noise* refers to samples that are wrongly labeled as another class (*e.g.* flower labeled as orange). *Out-of-distribution input* refers to samples that do not belong to any known classes. *Input corruption* refers to image-level distortion (*e.g.* low brightness) that causes data shift between training and test.

Most of the methods in literature focus on addressing the more detrimental label noise. Two dominant approaches include: (1) find clean samples as those with smaller loss and assign larger weights to them [6, 42, 32, 1]; (2) relabel noisy samples using model's predictions [31, 25, 34, 41, 23, 18]. Previous methods that focus on addressing label noise do not consider out-of-distribution input or input corruption, which limits their performance in real-world scenarios. Furthermore, using a model's own prediction to relabel samples could cause confirmation bias, where the prediction error accumulates and harms performance.

Figure 1. Google search images from WebVision [22] dataset with keyword "orange".

We propose a new direction for effective learning from noisy data. Different from existing methods, our method learns noise-robust low-dimensional representations, and performs noise cleaning by enforcing a smoothness constraint on neighboring samples. Specifically, our **algorithmic contributions** include:

- We propose noise-robust contrastive learning, which introduces two contrastive losses. The first is an unsupervised consistency contrastive loss. It enforces inputs with perturbations to have similar normalized embeddings, which helps learn robust and discriminative representation.
- Our second contrastive loss is a weakly-supervised prototypical contrastive loss. We compute class prototypes as normalized mean embeddings, and enforces each sample's embedding to be closer to its class prototype. Inspired by Mixup [44], we construct virtual training samples as linear interpolation of inputs, and encourage the same linear relationship *w.r.t* the class prototypes.
- We propose a new noise cleaning method which leverages the learned representations to enforce a smoothness constraint on neighboring samples. For each sample, we aggregate information from its top-k neighbors to create a pseudo-label. A subset of training samples with confi-

dent pseudo-labels are selected to compute the weakly-supervised loss. This process can effectively clean both label noise and out-of-distribution (OOD) noise.

Our **experimental contributions** include:

- We experimentally show that our method achieves **state-of-the-art** performance on multiple datasets with controlled noise and real-world noise.
- We demonstrate that the proposed noise cleaning method can effectively clean a majority of label noise. It also learns a curriculum that gradually leverages more samples to compute the weakly-supervised loss as the pseudo-labels become more accurate.
- We validate the robustness of the learned low-dimensional representation by showing (1) k-nearest neighbor classification outperforms the softmax classifier. (2) OOD samples can be separated from in-distribution samples.

2. Related work

2.1. Label noise learning

Learning from noisy labels have been extensively studied in the literature. While some methods require access to a small set of clean samples [40, 35, 36, 17, 11], most methods focus on the more challenging scenario where no clean labels are available. These methods can be categorized into two major types. The first type performs label correction using predictions from the network [31, 25, 34, 41, 23]. The second type tries to separate clean samples from corrupted samples, and trains the model on clean samples [6, 1, 14, 13, 38, 3, 24, 20]. DivideMix [18] combines label correction and sample selection with the Mixup [44] data augmentation under a co-training framework, but cost $2\times$ the computational resource of our method.

Different from existing methods, our method addresses label noise learning by learning robust representations. We propose a more effective noise cleaning method by leveraging the structure of the learned representations. Furthermore, our model is robust not only to label noise, but also to out-of-distribution and corrupted input. A previous work has studied open-set noisy labels [38], but their method does not enjoy the same level of robustness as ours.

2.2. Contrastive learning

Contrastive learning is at the core of recent self-supervised representation learning methods [4, 8, 29, 39]. In self-supervised contrastive learning, two randomly augmented images are generated for each input image. Then a contrastive loss is applied to pull embeddings from the same source image closer, while pushing embeddings from different source images apart. Recently, prototypical contrastive learning (PCL) [21] has been proposed, which uses cluster centroids as prototypes, and trains the network by pulling an image embedding closer to its assigned prototypes.

Different from these methods, our method performs contrastive learning in the principal subspace by training a linear autoencoder. Our weakly-supervised contrastive loss improves PCL [21] by using pseudo-labels to compute class-prototypes, and augments the input with Mixup [44]. Different from the original Mixup where learning happens at the classification layer, our learning takes places in the low-dimensional subspace to learn robust representation.

3. Method

Given a noisy training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i$ is an image and $y_i \in \{1, ..., C\}$ is its class label. We aim to train a network that is robust to the noise in training data and achieves high accuracy on a clean test set. The proposed method performs two steps iteratively: (1) noise-robust contrastive learning, which trains the network to learn robust representations; (2) noise cleaning with smooth neighbors, which aims to correct label noise and remove OOD samples. A pseudo-code is given in Algorithm 1. Next, we delineate each step in details.

3.1. Noise-robust contrastive learning

As shown in Figure 2, the network consists of three components: (1) a deep encoder (a convolutional neural network) that encodes an image $\boldsymbol{x}_i$ to a high-dimensional feature $\boldsymbol{v}_i$; (2) a classifier (a fully-connected layer followed by softmax) that receives $\boldsymbol{v}_i$ as input and outputs class predictions; (3) a linear autoencoder that projects $\boldsymbol{v}_i$ into a low-dimensional embedding $\boldsymbol{z}_i \in \mathbb{R}^d$. We aim to learn robust embeddings with two contrastive losses: unsupervised consistency loss and weakly-supervised mixup prototypical loss.

Unsupervised consistency contrastive loss. Following the NT-Xent [4] loss for self-supervised representation learning, our consistency contrastive loss enforces images with semantic-preserving perturbations to have similar embeddings. Specifically, given a minibatch of b images, we apply weak-augmentation and strong-augmentation to each image, and obtain $2b$ inputs $\{\boldsymbol{x}_i\}_{i=1}^{2b}$. Weak augmentation is a standard flip-and-shift augmentation strategy, while strong augmentation consists of color and brightness changes with details given in Section 4.1.

We project the inputs into the low-dimensional space and obtain their normalized embeddings $\{\hat{\boldsymbol{z}}_i\}_{i=1}^{2b}$. Let $i \in \{1, ..., b\}$ be the index of a weakly-augmented input, and $j(i)$ be the index of the strong-augmented input from the same source image, the consistency contrastive loss is defined as:

$$\mathcal{L}_{\text{cc}} = \sum_{i=1}^{b} -\log \frac{\exp(\hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_{j(i)}/\tau)}{\sum_{k=1}^{2b} \mathbb{1}_{i \neq k} \exp(\hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{z}}_k/\tau)}, \quad (1)$$

where τ is a scalar temperature parameter. The consistency

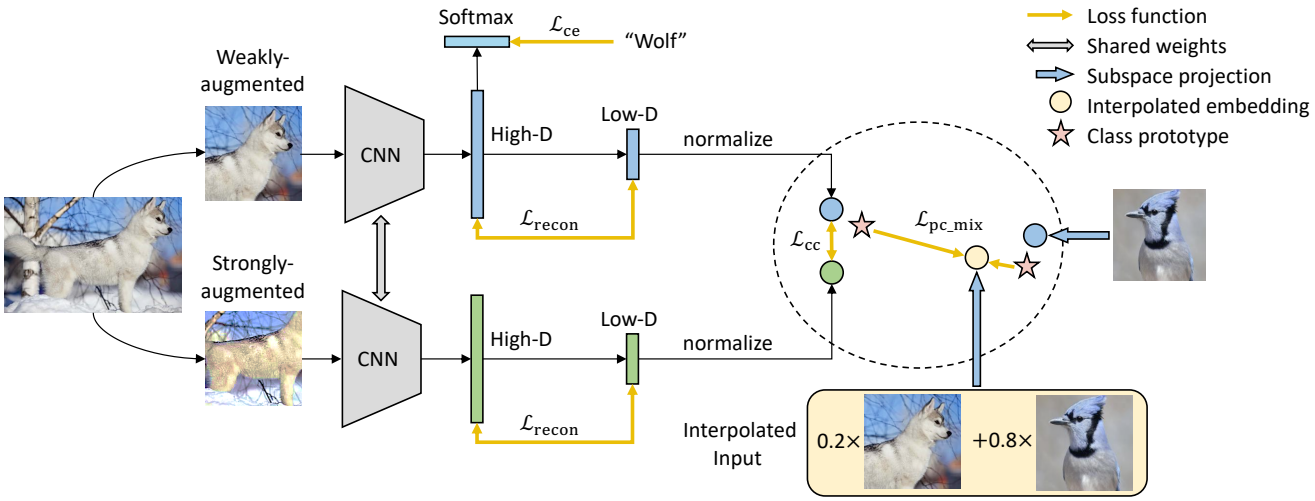


Figure 2. Our proposed framework for noise-robust contrastive learning. We project images into a low-dimensional subspace, and regularize the geometric structure of the subspace with (1)$\mathcal{L}_{\text{cc}}$: a consistency contrastive loss which enforces images with perturbations to have similar embeddings; (2)$\mathcal{L}_{\text{pc\_mix}}$: a prototypical contrastive loss augmented with mixup, which encourages the embedding for a linearly-interpolated input to have the same linear relationship *w.r.t* the class prototypes. The low-dimensional embeddings are also trained to reconstruct the high-dimensional features, which preserves the learned information and regularizes the classifier.

contrastive loss maximizes the inner product between the pair of positive embeddings $\hat{\boldsymbol{z}}_i$ and $\hat{\boldsymbol{z}}_{j(i)}$, while minimizing the inner product between $2(b-1)$ pairs of negative embeddings. By mapping different views (augmentations) of the same image to neighboring embeddings, the consistency contrastive loss encourages the network to learn discriminative representation that is robust to low-level image corruption.

Weakly-supervised mixup prototypical contrastive loss. Our second contrastive loss injects structural knowledge of classes into the embedding space. Let $\mathcal{I}_c$ denote indices for the subset of images in $\mathcal{D}$ labeled with class c, we calculate the class prototype as the normalized mean embedding:

$$\boldsymbol{z}^c = \frac{1}{|\mathcal{I}_c|}\sum_{i\in\mathcal{I}_c}\hat{\boldsymbol{z}}_i, \quad \hat{\boldsymbol{z}}^c = \frac{\boldsymbol{z}^c}{\|\boldsymbol{z}^c\|_2}, \tag{2}$$

where $\hat{\boldsymbol{z}}_i$ is the embedding of a center-cropped image, and the class prototypes are calculated at the beginning of each epoch.

The prototypical contrastive loss enforces an image embedding $\hat{\boldsymbol{z}}_i$ to be more similar to its corresponding class prototype $\hat{\boldsymbol{z}}^{y_i}$, in contrast to other class prototypes:

$$\mathcal{L}_{\text{pc}}(\hat{\boldsymbol{z}}_i, y_i) = -\log\frac{\exp(\hat{\boldsymbol{z}}_i\cdot\hat{\boldsymbol{z}}^{y_i}/\tau)}{\sum_{c=1}^{C}\exp(\hat{\boldsymbol{z}}_i\cdot\hat{\boldsymbol{z}}^{c}/\tau)}. \tag{3}$$

Since the label y_i is noisy, we would like to regularize the encoder from memorizing training labels. Mixup [44] has been shown to be an effective method against label noise [1, 18]. Inspired by it, we create virtual training samples by linearly interpolating a sample (indexed by i) with another sample (indexed by $m(i)$) randomly chosen from the same minibatch:

$$\boldsymbol{x}_i^m = \lambda\boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_{m(i)}, \tag{4}$$

where $\lambda\sim\text{Beta}(\alpha,\alpha)$.

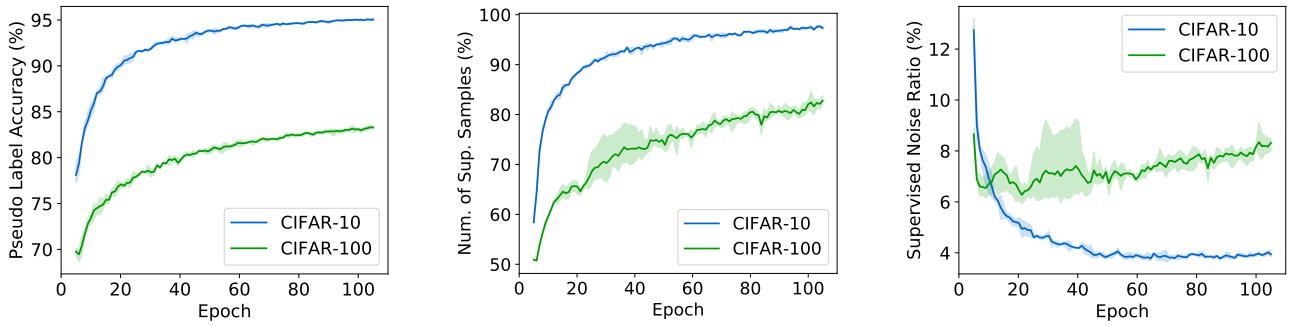
Let $\hat{\boldsymbol{z}}_i^m$ be the normalized embedding for $\boldsymbol{x}_i^m$, the mixup version of the prototypical contrastive loss is defined as a weighted combination of the two $\mathcal{L}_{\text{pc}}$ *w.r.t* class y_i and $y_{m(i)}$. It enforces the embedding for the interpolated input to have the same linear relationship *w.r.t.* the class prototypes.

$$\mathcal{L}_{\text{pc\_mix}} = \sum_{i=1}^{2b}\lambda\mathcal{L}_{\text{pc}}(\hat{\boldsymbol{z}}_i^m, y_i) + (1-\lambda)\mathcal{L}_{\text{pc}}(\hat{\boldsymbol{z}}_i^m, y_{m(i)}). \tag{5}$$

Reconstruction loss. We also train a linear decoder $\mathbf{W}_{\text{d}}$ to reconstruct the high-dimensional feature $\boldsymbol{v}_i$ based on $\boldsymbol{z}_i$. The reconstruction loss is defined as:

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{2b}\|\boldsymbol{v}_i - \mathbf{W}_{\text{d}}\boldsymbol{z}_i\|_2^2. \tag{6}$$

There are several benefits for training the autoencoder. First, an optimal linear autoencoder will project $\boldsymbol{v}_i$ into its low-dimensional principal subspace and can be understood as applying PCA [2]. Thus the low-dimensional representation $\boldsymbol{z}_i$ is intrinsically robust to input noise. Second, minimizing the reconstruction error is maximizing a lower bound of the mutual information between $\boldsymbol{v}_i$ and $\boldsymbol{z}_i$ [37]. Therefore, knowledge learned from the proposed contrastive losses can be maximally preserved in the high-dimensional representation, which helps regularize the classifier.



(a) (b) (c)

Figure 3. Curriculum learned by the proposed label correction method for training on CIFAR datasets with 50% sym. noise. (a) Accuracy of pseudo-labels *w.r.t* to clean training labels. Our method effectively cleans a majority of the label noise. (b) Number of samples in the weakly-supervised subset $\mathcal{D}_{\text{ws}}^t$. As the pseudo-labels become more accurate, more samples are used to compute the supervised losses. (c) Label noise ratio in the weakly-supervised subset, which maintains at a low level even as the size of the subset grows.

Classification loss. Given the softmax output from the classifier, $\boldsymbol{p}(\boldsymbol{y};\boldsymbol{x}_i)$, we define the classification loss as the cross-entropy loss. Note that it is only applied to the weakly-augmented inputs.

$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{b} \log p(y_i;\boldsymbol{x}_i). \tag{7}$$

The overall training objective is to minimize a weighted sum of all losses:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \omega_{\text{cc}}\mathcal{L}_{\text{cc}} + \omega_{\text{pc}}\mathcal{L}_{\text{pc\_mix}} + \omega_{\text{recon}}\mathcal{L}_{\text{recon}} \tag{8}$$

For *all* experiments, we fix $\omega_{\text{cc}} = 1$, $\omega_{\text{recon}} = 1$, and change ω_{pc} only across datasets. Our method is in general net sensitive to the values of the weights. In our ablation study, we show that setting either $\omega_{\text{cc}} = 0$ or $\omega_{\text{recon}} = 0$ still yields performance competitive or better than the current SoTA.

3.2. Noise cleaning with smooth neighbors

After warming-up the model by training with the noisy labels $\{y_i\}_{i=1}^n$ for t_0 epochs, we aim to clean the noise by generating a soft pseudo-label $\boldsymbol{q}_i$ for each training sample. Different from previous methods that perform label correction purely using the model's softmax prediction, our method exploits the structure of the low-dimensional subspace by aggregating information from top-k neighboring samples, which helps alleviate the confirmation bias problem.

At the t-th epoch, for each sample $\boldsymbol{x}_i$, let $\boldsymbol{p}_i^t$ be the classifier's softmax prediction, let $\boldsymbol{q}_i^{t-1}$ be its soft label from the previous epoch, we calculate the soft label for the current epoch as:

$$\boldsymbol{q}_i^t = \frac{1}{2}\boldsymbol{p}_i^t + \frac{1}{2}\sum_{j=1}^{k} w_{ij}^t \boldsymbol{q}_j^{t-1}, \tag{9}$$

where w_{ij}^t represents the normalized affinity between a sample and its neighbor and is defined as $w_{ij}^t = \frac{\exp(\hat{\boldsymbol{z}}_i^t \cdot \hat{\boldsymbol{z}}_j^t/\tau)}{\sum_{j=1}^k \exp(\hat{\boldsymbol{z}}_i^t \cdot \hat{\boldsymbol{z}}_j^t/\tau)}$. We set $k = 200$ in all experiments.

The soft label defined by eqn.(9) is the minimizer of the following quadratic loss function:

$$J(\boldsymbol{q}_i^t) = \sum_{j=1}^{k} w_{ij}^t \left\|\boldsymbol{q}_i^t - \boldsymbol{q}_j^{t-1}\right\|_2^2 + \left\|\boldsymbol{q}_i^t - \boldsymbol{p}_i^t\right\|_2^2. \tag{10}$$

The first term is a smoothness constraint which encourages the soft label to take a similar value as its neighbors' labels, whereas the second term attempts to maintain the model's class prediction.

We construct a weakly-supervised subset which contains (1) *clean* sample whose soft label score for the original class y_i is higher than a threshold η_0, (2) *pseudo-labeled* sample whose maximum soft label score exceeds a threshold η_1. For pseudo-labeled samples, we convert their soft labels into hard labels by taking the class with the maximum score.

$$\mathcal{D}_{\text{ws}}^t = \{\boldsymbol{x}_i, y_i \,|\, q_i^t(y_i) > \eta_0\} \cup \{\boldsymbol{x}_i, \hat{y}_i^t = \arg\max_c q_i^t(c) \,|\, \forall \max_c q_i^t(c) > \eta_1, c \in \{1,..,C\}\} \tag{11}$$

Given the weakly-supervised subset, we modify the classification loss $\mathcal{L}_{\text{ce}}$, the mixup prototypical contrastive loss $\mathcal{L}_{\text{pc\_mix}}$, and the calculation of prototypes $\hat{\boldsymbol{z}}^c$, such that they only use samples from $\mathcal{D}_{\text{ws}}^t$. The unsupervised losses (*i.e.* $\mathcal{L}_{\text{cc}}$ and $\mathcal{L}_{\text{recon}}$) still operate on all training samples.

Learning curriculum. Our iterative noise cleaning method learns an effective training curriculum, which gradually increases the size of $\mathcal{D}_{\text{ws}}^t$ as the pseudo-labels become more accurate. To demonstrate such curriculum, we analyse the noise cleaning statistics for training our model on CIFAR-10 and CIFAR-100 datasets with 50% label noise (experimental

Algorithm 1: Pseudo-code for our method.

Input: noisy training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, model parameters θ.
for $t \leftarrow 0$ **to** $t_0 - 1$ **do** // learn from noisy labels for t_0 epochs (warm-up)
  $\{\hat{\boldsymbol{z}}_i\}_{i=1}^{n} = \{f_\theta(\boldsymbol{x}_i)\}_{i=1}^{n}$ // get normalized low-dimentional embeddings for all images
  $\{\hat{\boldsymbol{z}}^c\}_{c=1}^{C} = \text{Calculate-Prototype}(\{\hat{\boldsymbol{z}}_i, y_i\}_{i=1}^{n})$ // calculate class prototypes
  for $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{2b}$ **in** $\mathcal{D}$ **do** // load a minibatch
    $\hat{\boldsymbol{z}}_i = f_\theta(\boldsymbol{x}_i)$ // obtain normalized low-dimensional embeddings
    $\lambda \sim \text{Beta}(\alpha, \alpha)$ // sample a mixup weight from a beta distribution
    $\boldsymbol{x}_i^m = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_{m(i)}$ // generate virual training samples
    $\hat{\boldsymbol{z}}_i^m = f_\theta(\boldsymbol{x}_i^m)$ // obtain embeddings for virtual samples
    $\mathcal{L} = \sum_{i=1}^{b} \mathcal{L}_{\text{ce}}(\boldsymbol{x}_i, y_i) + \sum_{i=1}^{2b} \big(\omega_{\text{cc}}\mathcal{L}_{\text{cc}}(\hat{\boldsymbol{z}}_i) + \omega_{\text{pc}}\mathcal{L}_{\text{pc\_mix}}(\hat{\boldsymbol{z}}_i^m, y_i, \lambda) + \omega_{\text{recon}}\mathcal{L}_{\text{recon}}(\boldsymbol{x}_i, \hat{\boldsymbol{z}}_i)\big)$
    $\theta = \text{SGD}(\mathcal{L}, \theta)$ // compute loss and update model parameters
  end
end
for $t \leftarrow t_0$ **to** MaxEpoch **do** // learn from psuedo-labels
  $\{\hat{\boldsymbol{z}}_i^t, \boldsymbol{p}_i^t\}_{i=1}^{n} = \{f_\theta(\boldsymbol{x}_i)\}_{i=1}^{n}$ // get embeddings and softmax predictions for all images
  $\boldsymbol{q}_i^t = \frac{1}{2}\boldsymbol{p}_i^t + \frac{1}{2}\sum_{j=1}^{k} w_{ij}^t \boldsymbol{q}_j^{t-1}$, $\boldsymbol{q}_i^{t_0-1} = \boldsymbol{p}_i^{t_0}$ // aggregate information from top-k neighbors to generate soft labels
  $\mathcal{D}_{\text{ws}}^t = \{\boldsymbol{x}_i, y_i \mid q_i^t(y_i) > \eta_0\} \cup \{\boldsymbol{x}_i, \hat{y}_i^t = \arg\max_c q_i^t(c) \mid \forall \max_c q_i^t(c) > \eta_1, c \in \{1, .., C\}\}$ // construct a subset containing clean samples and pseudo-labeled samples
  Repeat line 4-12, but only use samples from $\mathcal{D}_{\text{ws}}^t$ to compute $\hat{\boldsymbol{z}}^c$, $\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{pc\_mix}}$.
end

details explained in the next section). In Figure 3 (a), we show the accuracy of the soft pseudo-labels *w.r.t* to clean training labels (only used for analysis purpose). Our method can significantly reduce the ratio of label noise from 50% to 5% (for CIFAR-10) and 17% (for CIFAR-100). Figure 3 (b) shows the size of $\mathcal{D}_{\text{ws}}^t$ as a percentage of the total number of training samples, and Figure 3 (c) shows the effective label noise ratio within the weakly-supervised subset $\mathcal{D}_{\text{ws}}^t$. Our method maintains a low noise ratio in the weakly-supervised subset, while gradually increasing its size to utilize more samples for the weakly-supervised losses.

4. Experiment

In this section, we validate the proposed method on multiple benchmarks with controlled noise and real-world noise. Our method achieves state-of-the-art performance across all benchmarks. For fair comparison, we compare with DivideMix [18] without ensemble. In Table 7, we report the result of our method with co-training and ensemble, which further improves performance.

4.1. Experiments on controlled noisy labels

Dataset. Following [34, 18], we corrupt the training data of CIFAR-10 and CIFAR-100 [16] with two types of label noise: *symmetric* and *asymmetric*. Symmetric noise is injected by randomly selecting a percentage of samples and changing their labels to random labels. Asymmetric noise is class-dependant, where labels are only changed to similar classes (*e.g.* dog↔cat, deer→horse). We experiment with multiple noise ratios: sym 20%, sym 50%, and asym 40% (see results for sym 80% and 90% in Table 7). Note that asymmetric noise ratio cannot exceed 50% because certain classes would become theoretically indistinguishable.

Implementation details. Same as previous works [1, 18], we use PreAct ResNet-18 [9] as our encoder model. We set the dimensionality of the bottleneck layer as $d = 50$. Our model is trained using SGD with a momentum of 0.9, a weight decay of 0.0005, and a batch size of 128. The network is trained for 200 epochs. We set the initial learning rate as 0.02 and use a cosine decay schedule. We apply standard crop and horizontal flip as the weak augmentation. For strong augmentation, we use AugMix [12], though other methods (*e.g.* SimAug [4]) work equally well. For all CIFAR experiments, we fix the hyper-parameters as $\omega_{\text{cc}} = 1, \omega_{\text{pc}} = 5, \omega_{\text{recon}} = 1, \tau = 0.3, \alpha = 8, \eta_1 = 0.9$. For CIFAR-10, we activate noise cleaning at epoch $t_0 = 5$, and set $\eta_0 = 0.1$ (sym.) or 0.4 (asym.). For CIFAR-100, we activate noise cleaning at epoch $t_0 = 15$, and set $\eta_0 = 0.02$. We use faiss-gpu [15] for efficient knn search in the low-dimensional subspace, which finishes within 1 second.

Results. Table 1 shows the comparison with existing methods. Our method outperforms previous methods across all label noise settings. On the more challenging CIFAR-100, we achieve 3-4% accuracy improvement.

In order to demonstrate the advantage of the proposed noise-robust representation learning method, we perform k-nearest neighbor (knn) classification ($k = 200$), which projects training and test images into normalized low-dimensional embeddings. Compared to the trained classifier, knn achieves higher accuracy, which verifies the robustness of the learned representation.

Dataset	CIFAR-10			CIFAR-100	
Noise type	Sym 20%	Sym 50%	Asym 40%	Sym 20%	Sym 50%
Cross-Entropy [18]	82.7	57.9	72.3	61.8	37.3
Forward [30]	83.1	59.4	83.1	61.4	37.3
Co-teaching+ [42]	88.2	84.1	-	64.1	45.3
Mixup [44]	92.3	77.6	-	66.0	46.6
P-correction [41]	92.0	88.7	88.1	68.1	56.4
MLNT [19]	92.0	88.8	88.6	67.7	58.0
M-correction [1]	93.8	91.9	86.3	73.4	65.4
DivideMix [18]	95.0	93.7	91.4	74.8	72.1
ELR [23] (reproduced)	94.7±0.1	93.5±0.2	91.7±0.9	75.3±0.2	71.3±0.3
DivideMix (reproduced)	95.1±0.1	93.6±0.2	91.3±0.8	75.1±0.2	72.1±0.3
Ours (classifier)	95.8±0.1	94.3±0.2	91.9±0.8	79.1±0.1	74.8±0.4
Ours (knn)	**95.9**±0.1	**94.5**±0.1	**92.4**±0.9	**79.4**±0.1	**75.0**±0.4

Table 1. Comparison with state-of-the-art methods on CIFAR datasets with label noise. Numbers indicate average test accuracy (%) over last 10 epochs. We report results over 3 independent runs with randomly-generated label noise. Results for previous methods are copied from [1, 18]. We re-run DivideMix and ELR (without model ensemble) using the publicly available code on the same noisy data as ours.

Input noise	CE	Iterative [38]	GCE [45]	DivideMix [18]	Ours (cls.)	Ours (knn)
+ CIFAR-100 20k	53.6	87.2	87.3	89.0	91.5	**93.1**±0.3
+ SVHN 20k	58.1	88.6	88.8	91.9	93.3	**93.9**±0.2
Image corruption	53.8	87.7	87.9	89.8	91.4	**91.6**±0.2

Table 2. Comparison with state-of-the-art methods on CIFAR-10 dataset with label noise (50% symmetric) and input noise (OOD images or corrupted images). Numbers indicate average test accuracy (%) over last 10 epochs. We report results over 3 independent runs with randomly-generated noise. We re-run previous methods using publicly available code with the same data and model as ours.

4.2. Experiments on controlled noisy labels with noisy images

Dataset. We further corrupt a noisy-labeled (50% symmetric) CIFAR-10 dataset by injecting two types of input noise: out-of-distribution (OOD) images and input corruption. For OOD noise, we follow [38] and add $20k$ additional images from either one of the two other datasets: CIFAR-100 and SVHN [28], which enlarges the training set to $70k$. A random CIFAR-10 label is assigned to each OOD image. For input corruption, we follow [10] and corrupt each image in CIFAR-10 with a noise randomly chosen from the following four types: *Fog*, *Snow*, *Motion blur* and *Gaussian noise*. Examples of both types of input noise are shown in Figure 4. For training, we follow the same implementation details as the CIFAR-10 experiments described in Section 4.1.

Results. Table 2 shows the results. Our method consistently outperforms existing methods by a substantial margin. We observe that OOD images from a similar domain (CIFAR-100) are more harmful than OOD images from a more different domain (SVHN). This is because noisy images that are closer to the test data distribution are more likely to distort the decision boundary in a way that negatively affects test performance. Nevertheless, performing knn classification using the learned embeddings demonstrates high robustness to input noise.

Figure 4. Examples of input noise injected to CIFAR-10.

In Figure 5, we show the t-SNE [26] visualization of the low-dimensional embeddings for all training samples, including in-distribution CIFAR-10 images and out-distribution CIFAR-100 or SVHN images. As training progresses from epoch 10 to epoch 200, our model learns to separate OOD samples (represented as gray points) from in-distribution samples (represented as color points). It also learns to cluster CIFAR-10 images according to their true class, despite their noisy labels. Therefore, this visualization demonstrates that the proposed method learns representation that is robust to both label noise and OOD noise.

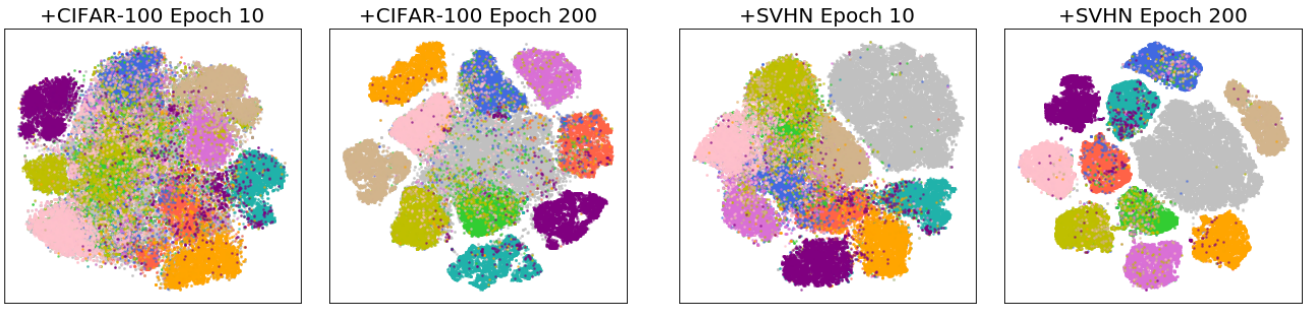


Figure 5. t-SNE visualization of low-dimensional embeddings for CIFAR-10 images (color represents the true class) + OOD images (gray points) from CIFAR-100 or SVHN. The model is trained on noisy CIFAR-10 ($50k$ images with 50% label noise) and $20k$ OOD images with random labels. Our method can effectively learn to (1) cluster CIFAR-10 images according to their true class, despite their noisy labels; (2) separate OOD samples from in-distribution samples, such that their harm is reduced.

	CIFAR-10 Sym 50%	+ CIFAR-100 20k	+ Image Corruption	CIFAR-100 Sym 50%
w/o $\mathcal{L}_{\text{pc\_mix}}$	85.9 (86.1)	79.7 (81.5)	81.6 (81.7)	65.6 (65.9)
w/o $\mathcal{L}_{\text{cc}}$	93.7 (93.8)	91.3 (91.5)	89.4 (89.5)	71.9 (71.8)
w/o $\mathcal{L}_{\text{recon}}$	93.3 (94.0)	90.7 (92.9)	90.2 (91.0)	73.2 (73.9)
w/o mixup	89.5 (89.9)	85.4 (87.0)	84.7 (84.9)	69.3 (69.7)
w/ standard aug.	94.1 (94.3)	90.8 (92.9)	90.5 (90.7)	74.5 (75.0)
DivideMix	93.6	89.0	89.8	72.1
Ours	94.3 (94.5)	91.5 (93.1)	91.4 (91.6)	74.8 (75.0)

Table 3. Effect of the proposed components. We show the accuracy of the classifier (knn) on four benchmarks with different noise. Note that DivideMix [18] also performs mixup.

bottleneck dimension	$d = 25$	$d = 50$	$d = 100$	$d = 200$
CIFAR-10 Sym 50%	93.4	94.3	94.2	93.7
CIFAR-100 Sym 50%	73.8	74.8	74.4	73.8

Table 4. Classifier's test accuracy (%) with different low-dimensions.

4.3. Ablation study

Effect of the proposed components. In Table 3, we study the effect of 5 components from the proposed method including (1) the weakly-supervised mixup prototypical contrastive loss $\mathcal{L}_{\text{pc\_mix}}$, (2) the unsupervised consistency contrastive loss $\mathcal{L}_{\text{cc}}$, (3) the reconstruction loss $\mathcal{L}_{\text{recon}}$, (4) mixup augmentation, and (5) strong data augmentation with AugMix. We remove each of these components and report the accuracy of the classifier and knn across four benchmarks. The result shows that $\mathcal{L}_{\text{pc\_mix}}$ is most crucial to the model's performance. $\mathcal{L}_{\text{cc}}$ has a stronger positive effect with image corruption or larger number of classes (CIFAR-100). Our method still achieves competitive performance when either the $\mathcal{L}_{\text{cc}}$ or $\mathcal{L}_{\text{recon}}$ is removed. When using standard data augmentation (random crop and horizontal flip) instead of AugMix, our method still achieves state-of-the-art result.

Effect of bottleneck dimension. We vary the dimensionality of the bottleneck layer, d, and examine the performance change in Table 4. Our model is in general not sensitive to the change of d.

4.4. Experiments on real-world noisy data

Dataset. Next, we verify our method on two real-word noisy datasets: WebVision [22] and Clothing1M [40]. Webvision contains images crawled from the web using the same concepts from ImageNet ILSVRC12 [5]. Following previous works [3, 18], we perform experiments on the first 50 classes of the Google image subset. Clothing1M consists of images collected from online shopping websites where labels were generated from surrounding texts. Note that we do not use the additional clean set for training.

Implementation details. For WebVision, we follow previous works [3, 18] and use inception-resnet v2 [33] as the encoder. We train the model using SGD with a weight decay of 0.0001 and a batch size of 64. We train for 40 epochs with an initial learning rate of 0.04. The hyper-parameters are set as $d = 50, \omega_{\text{cc}} = 1, \omega_{\text{pc}} = 2, \omega_{\text{recon}} = 1, \tau = 0.3, \alpha = 0.5, \eta_0 = 0.05, \eta_1 = 0.8, t_0 = 15$. For Clothing1M, we

Test dataset	WebVision		ILSVRC12	
Accuracy (%)	top1	top5	top1	top5
Forward [30]	61.1	82.7	57.4	82.4
Decoupling [27]	62.5	84.7	58.3	82.3
D2L [25]	62.7	84.0	57.8	81.4
MentorNet [14]	63.0	81.4	57.8	79.9
Co-teaching [6]	63.6	85.2	61.5	84.7
INCV [3]	65.2	85.3	61.0	85.0
ELR [23]	76.3	91.3	68.7	87.8
DivideMix [18]	75.9	90.1	73.3	89.2
Ours (w/o noise cleaning)	75.5	90.2	72.0	90.0
Ours (classifier)	76.3	**91.5**	73.3	**91.2**
Ours (knn)	**77.8**	91.3	**74.4**	90.9

Table 5. Comparison with state-of-the-art methods trained on WebVision (mini). Numbers denote accuracy (%) on the WebVision validation set and the ImageNet ILSVRC12 validation set. We report results for ELR and DivideMix without model ensemble.

Method	CE	Forward	Joint-Opt	ELR	MLNT	MentorMix	SL	DivideMix	Ours (cls.)	Ours (knn)
Accuracy	69.21	69.84	72.16	72.87	73.47	74.30	74.45	74.48	74.84	**74.97**

Table 6. Comparison with state-of-the-art methods on Clothing1M dataset. Results for previous methods are directly copied from corresponding papers. We report results for ELR and DivideMix without model ensemble.

Dataset	CIFAR-10					CIFAR-100			
Noise type	Sym.				Asym.	Sym.			
Noise ratio	20%	50%	80%	90%	40%	20%	50%	80%	90%
DivideMix [18] w/o ensemble	95.0	93.7	92.4	74.2	91.4	74.8	72.1	57.6	29.2
DivideMix [18] w/ ensemble	95.7	94.4	92.9	75.4	92.1	76.9	74.2	59.6	31.0
ELR+ [23] w/ ensemble	95.6	94.6	93.1	76.1	92.0	77.1	74.0	59.9	31.3
Ours	95.8	94.3	92.4	75.0	91.9	79.1	74.8	57.7	29.3
Ours w/ co-training	96.1	94.8	92.8	76.3	92.4	79.8	75.3	58.9	31.5
Ours w/ co-training & ensemble	**96.4**	**95.3**	**93.3**	**77.4**	**92.6**	**80.3**	**76.0**	**61.1**	**33.1**

Table 7. Results of our proposed method with co-training and model ensemble. We report the average test accuracy over last 10 epochs.

follow previous works [7, 18] and use ResNet-50 with ImageNet pretrained weights. We sample 1000 mini-batches as one epoch, and train the model for 50 epochs with an initial learning rate of 0.01. The hyper-parameters are set as $d = 32, \omega_{\text{cc}} = 1, \omega_{\text{pc}} = 1, \omega_{\text{recon}} = 1, \tau = 0.3, \alpha = 0.5, \eta_0 = 0.4, \eta_1 = 0.9, t_0 = 1$. Most of hyper-parameters are kept to be the same across datasets.

Results. We report the results for WebVision in Table 5 and Clothing1M in Table 6. Our method achieves state-of-the-art performance on both datasets. The performance on WebVision is competitive even *without* noise cleaning, which shows the robustness of the learned representation.

4.5. Co-training and model ensemble

Co-training and model ensemble have been shown to be useful in combating label noise [6, 18, 23]. Therefore, we incorporate these two techniques by (1) simultaneously train two models that are randomly initialized and average their soft label $\boldsymbol{q}_i^t$ to produce a new soft label, (2) use their ensemble prediction during test. The results on CIFAR datasets are shown in Table 7.

5. Conclusion

In this paper, we propose a new method for learning from noisy data by learning robust representation. We propose a noise-robust contrasitve learning framework for representation learning, and a noise cleaning method based on nearest-neighbor constraints. Our method can address label noise, OOD noise, and image corruption. We demonstrate our model's state-of-the-art performance with extensive experiments on multiple noisy datasets. For future work, we plan to extend our framework to other domains.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, pages 312–321, 2019. 1, 2, 3, 5, 6

[2] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989. 3

[3] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pages 1062–1070, 2019. 2, 7, 8

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 7

[6] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018. 1, 2, 8

[7] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, pages 5137–5146, 2019. 8

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016. 5

[10] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 6

[11] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, pages 10477–10486, 2018. 2

[12] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 5

[13] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *ICML*, 2020. 2

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318, 2018. 2, 8

[15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 5

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Mater's thesis, University of Toronto, 2009. 5

[17] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, pages 5447–5456, 2018. 2

[18] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 1, 2, 3, 5, 6, 7, 8

[19] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, pages 5051–5059, 2019. 6

[20] Junnan Li, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021. 2

[21] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C.H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021. 2

[22] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv:1708.02862*, 2017. 1, 7

[23] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 1, 2, 6, 8

[24] Yueming Lyu and Ivor W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *ICLR*, 2020. 2

[25] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi N. R. Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pages 3361–3370, 2018. 1, 2, 8

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 6

[27] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NIPS*, pages 960–970, 2017. 8

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS-WS*, 2011. 6

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[30] 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 *CVPR*, pages 2233–2241, 2017. 6, 8

[31] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. 1, 2

[32] Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, pages 5739–5748, 2019. 1

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 7

[34] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018. 1, 2, 5

[35] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NIPS*, pages 5601–5610, 2017. 2

[36] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 6575–6583, 2017. 2

[37] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, 2010. 3

[38] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pages 8688–8696, 2018. 2, 6

[39] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2

[40] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699, 2015. 2, 7

[41] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, 2019. 1, 2, 6

[42] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pages 7164–7173, 2019. 1, 6

[43] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1

[44] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1, 2, 3, 6

[45] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8792–8802, 2018. 6