Manifold DivideMix: A Semi-Supervised Contrastive Learning Framework for Severe Label Noise

Fahimeh Fooladgar[1] Minh Nguyen Nhat To[1] Parvin Mousavi[2] Purang Abolmaesumi[1]
[1] Department of Electrical and Computer Engineering, University of British Columbia, CA
[2] School of Computing, Queen's University, CA

fahimeh.fooladgar@ubc.ca, mtrcl@student.ubc.ca, mousavi@queensu.ca, purang@ece.ubc.ca

Abstract

Deep neural networks have proven to be highly effective when large amounts of data with clean labels are available. However, their performance degrades when training data contains noisy labels, leading to poor generalization on the test set. Real-world datasets contain noisy label samples that either have similar visual semantics to other classes (in-distribution) or have no semantic relevance to any class (out-of-distribution) in the dataset. Most state-of-the-art methods leverage ID labeled noisy samples as unlabeled data for semi-supervised learning, but OOD labeled noisy samples cannot be used in this way because they do not belong to any class within the dataset. Hence, in this paper, we propose incorporating the information from all the training data by leveraging the benefits of self-supervised training. Our method aims to extract a meaningful and generalizable embedding space for each sample regardless of its label. Then, we employ a simple yet effective K-nearest neighbor method to remove portions of out-of-distribution samples. By discarding these samples, we propose an iterative "Manifold DivideMix" algorithm to find clean and noisy samples, and train our model in a semi-supervised way. In addition, we propose "MixEMatch", a new algorithm for the semi-supervised step that involves mixup augmentation at the input and final hidden representations of the model. This will extract better representations by interpolating both in the input and manifold spaces. Extensive experiments on multiple synthetic-noise image benchmarks and real-world web-crawled datasets demonstrate the effectiveness of our proposed framework. Code is available at https://github.com/Fahim-F/ManifoldDivideMix.

1. Introduction

The empirical observation of deep neural networks (DNNs) applied to computer vision problems reveals that they produce superior performance when trained with a significant amount of clean, annotated data [13]. However, their primary weakness is also the vast number of clean labeled examples needed for training. Collecting and manually annotating such data could be complex, time-consuming, and costly, particularly in certain fields. Meanwhile, open-source online data that can be automatically annotated using search engine queries and user tags is the backbone of most large-scale data collection methods [22, 37]. However, this annotation approach will certainly result in label noise. Therefore, it is challenging to train DNNs using such data, as they can effectively memorize noisy random labels during training [4, 23].

Several methods have been developed to deal with label noise in automatically annotated datasets, such as semi-supervised learning [5, 31], self-supervised learning [6], and robust training [19]. These methods can be classified into two primary categories. The first category assumes that the true labels of noisy samples are included in the label set (i.e. in-distribution labeled noisy samples (ID samples). The community has invested a lot of effort in designing robust methods to train DNNs in the presence of ID label noise [11, 19]. The second category arises from the observation that techniques that are robust to ID noise tend to perform poorly when applied in more realistic settings (with real-world label noise). In fact, the authors of [2] suggest that most of the label noise in web-crawled datasets is out-of-distribution (OOD) label noise, which means that the true labels for noisy samples cannot be inferred from the distribution (we call them OOD samples). To evaluate this, they randomly collected three small but representative sample sets from the WebVision 1.0 dataset [21] to determine the typical level of noise present in web-crawled, automatically annotated datasets. They reported that approximately 70% of the data is properly labeled, 5% have in-distribution noisy labels, and 25% have out-of-distribution noisy labels.

Given the observations above, we need approaches to first separate potentially clean, ID noise and OOD noise instances, then decide how to incorporate those noisy data in the training. However, while ID noisy samples may be

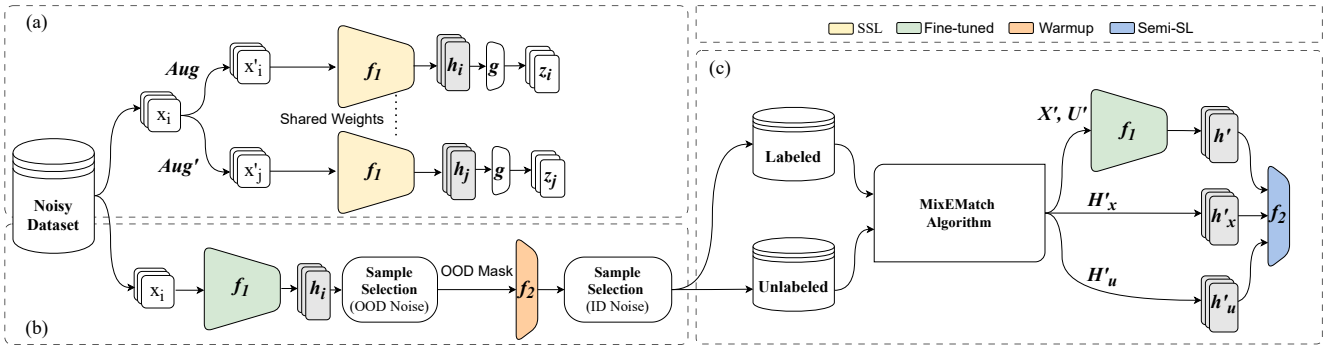

Figure 1. An overview of the proposed method. (a) Step (1): Learn an embedding space of each image regardless of its label in the self-supervised way using contrastive loss (Eq. 1) [17]. (b) Once, the backbone model trained, in Step (2), Select OOD samples based on the distances in the embedding space by using KNN and create OOD mask to remove OOD from supervised learning. Then, train a linear classifier (f_2) on top of the trained backbone (f_1) and fine-tune it for a few epochs. (c) Step (3): Select ID noisy samples based on the distribution of classification loss and find clean/noisy samples to form labeled/unlabeled sets. Next, continue training classifier (f_2) and fine-tuning backbone model (f_1) in the Semi-supervised way by using MixEMatch (Algorithm 1). Step 2 and 3 repeated until convergence. The color legend shows how each component in each step has been trained.

directly fed to the network as unlabeled data using common semi-supervised learning (Semi-SL) algorithms, OOD label noise samples cannot be assigned to any category or used to train the network. Therefore, a simple solution is to remove them, once detected as OOD samples. The author of [2] propose a method to detect OOD labeled noisy samples and enforce their prediction to converge to a uniform distribution, rather than deleting the samples. Leveraging the OOD samples during unsupervised training improves the generality as these samples still include useful information for learning low-level features. As a result, they may be used to improve the representations that are acquired [38]. Although having both in- and out-of-distribution labeled noisy data presents challenges for supervised training of a model, these can be mitigated, and in some cases the data can even be advantageous, in a framework for self-supervised training.

In this paper, we propose "Manifold DivideMix", which addresses learning with ID and OOD label noise in a semi-supervised way. Based on the success of Self-Supervised Learning (SSL) and different from most existing methods, we consider one model instead of two or ensemble model, and train it using unsupervised contrastive learning on the noisy dataset to learn meaningful representations of data without explicit labels. Next, we add a linear classifier on top of the SSL model and incorporate two sample selection methods to detect OOD and ID samples gradually during training. As a results, we iteratively remove OOD samples and leverage the clean and ID samples in the Semi-SL training (see Figure 1). In summary, our contributions are as follows.

- Based on the assumption that measuring distances between images is more meaningful in the embedding space, we take the average distance between each embedding space and each of its K-Nearest Neighbours (KNN) as the OOD score to detect OOD labeled noisy samples;
- To learn better representation and improve the generalisation performance, we propose MixEMatch, an algorithm to apply mixup augmentation both on the input and representation spaces of the Semi-SL step. These augmentations boost the overall quality of pseudo-labels and, as a result, substantially improve the effectiveness of subsequent Semi-SL training;
- We demonstrate experimentally that unsupervised feature learning reduces the effect of overfitting to label noise and significantly outperforms baseline approaches by a large margin, particularly when applied on high label noise.

2. Related Work

Song et al. [32] categorize the current research trends to address noisy label problem through: 1) Robust regularisation [33, 41], which involves explicitly or implicitly forcing a DNN to overfit less to falsely labeled examples; 2) Robust loss functions [34, 45] that can handle label noise; 3) Robust architectures [10, 36] that have added noise adaptation layers to a DNN to learn a label transition process or a dedicated architecture to reliably support more diverse types of label noise; 4) Loss correction [26] and loss reweighting [24], which modify the loss value based on the certainty of a particular loss or label; and 5) Sample selection, which extracts correctly labeled instances from noisy training data using multiple networks or iterative learning [11, 39]. Recently, based on the assumption that all noise in the data are either ID or a mix of ID and OOD label noise, there are two categories of research that address the following questions: 1) how can we detect noisy samples?, and 2) how do we leverage them during training once they are detected?

2.1. ID Sample Selection Methods

Many noise-robust methods assume that all the noise is ID. On the presumption that clean labels represent the majority in a noisy label dataset, deep networks initially remember training data with clean labels before moving on to data with noisy labels or complex samples. Hence, low-loss instances might be considered clean (highly probable occurrences). This approach has been successful in identifying potentially clean instances in a wide range of situations. To obtain the clean/noisy separation using unsupervised methods, a mixture model is fitted to the loss distribution of training samples. In DivideMix [19], a Gaussian Mixture Mode (GMM) is used to detect high-loss samples. This concept is further developed in PropMix [8] by applying another GMM on the noisy samples to separate simple and hard noisy samples. Nevertheless, clean samples from the hard classes may generate high-loss values, hence the selection procedure is often skewed towards easy classes. This is especially noticeable at the beginning of training, and it might cause class-disparity among the chosen clean samples. Karim et al. [16] propose an efficient technique based on Jensen-Shannon divergence for separating samples, and choose an equal number of high-quality examples from each class to solve the problem of the over-selection of easy and complex samples. In [12], the trusted small clean set was considered to estimate the uncertainty of prediction and detect noisy labeled samples.

2.2. ID-OOD Sample Selection Methods

A new category of noise-resistant algorithms has recently emerged to deal with noisy datasets that include both in- and out-of-distribution label noise. In [38], the authors utilized the Jensen-Shannon divergence between a predicted label and the original label to identify ID labeled noisy samples, and then samples with poor agreement across different views were considered OOD. Albert et al. [1] proposed a method called SNCF that identifies clean, ID, and OOD samples by applying outlier sensitive clustering at the class-level. In [20], the authors computed a pseudo-label by aggregating information from the top K neighbors, and then selected a subset of training samples with reliable pseudo-labels as clean samples, assuming that images with several neighbors from the same class were less likely to be noisy. Meanwhile, in [2], the authors proposed a method called Dynamic Softening for Out-of-distribution Samples (DSOS) to distinguish between ID and OOD samples by computing the collision entropy of the interpolation between the original label and network prediction. Another method, EvidentialMix [27], improved upon loss values clustering by using evidential loss [29] to indicate the presence of two separate noisy modes, one for OOD and one for ID.

2.3. Dealing with Label Noise

Various methods have been developed to deal with noisy data once they are detected. The simplest way is to discard the potential either ID noise samples [11, 14, 30] or OOD noise samples [2, 27] during training. This leads to sub-optimal results as those samples still have helpful information to enhance generalization of the model. Most other methods handle the noisy samples as unlabeled data and continue training using common semi-supervised learning algorithms [5, 31]. In DivideMix [19] and UNICON [16], two models were trained with a semi-supervised consistency regularization algorithm to incorporate noisy samples as unlabeled samples and correct their labels. This concept was further developed in PropMix [8], where self-supervised initialization was employed and only simple noisy samples were corrected, while the hardest ones were discarded. ScanMix [28] was an advancement on DivideMix that used a semi-supervised contrastive method to fix the label and semantic clustering in a self-supervised feature space. The method in [20] encouraged clean samples to be similar to their class prototypes by training on a weakly supervised prototype. In order to overcome the challenges of label noise, the authors of [25] presented a Multi-Objective Interpolation Training (MOIT) framework, where supervised contrastive learning and semi-supervised learning complemented one another. The authors of [40] propose Gradient Switching Strategy (GSS) to prevent damages caused by mislabeled samples during training. They aim to eliminate the impact of misleading gradient directions of noisy labeled samples by assigning a random gradient direction for them.

3. Proposed Method

Our proposed Manifold DivideMix algorithm (Figure 1) begins with a self-supervised phase. We next use filtering procedures in a semi-supervised training setup to first separate and eliminate the simple out-of-distribution samples. Afterwards, during semi-supervised training phase, the remaining samples are determined iteratively and proportionately as clean, in- and out-of-distribution label noise samples. The following subsections provide a detailed explanation of each step.

3.1. Background

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ represent the training set, where $\mathbf{x}_i \in \mathcal{X}$ is the i^{th} sample, and y_i denotes the given label associated with one of the C classes, and N is the total number of training samples. Specifically, we address the situation when the dataset contains $\mathcal{D}_c = \{(x_i, y_i)\}_{i=1}^{N_c}$, a properly labeled set, $\mathcal{D}_{in} = \{(x_i, y_i)\}_{i=1}^{N_{in}}$ as an incorrectly labeled in-distribution noisy set, and $\mathcal{D}_{out} = \{(x_i, y_i)\}_{i=1}^{N_{out}}$ as an out-of-distribution noisy set. Here, $N = N_c + N_{in} + N_{out}$ represents the total number of samples available. We

assume that the sample distribution among $\mathcal{D}_c$, $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ is unknown. As a result of noise, the true label, $\hat{\mathbf{y}}_i$, may differ from the given label $\mathbf{y}_i$. We design the DNN model with an encoder base model, $f_1(.;\theta)$, a classification layer, $f_2(.;\phi)$, and a projection head, $g(.;\psi)$, with parameters θ, ϕ, and ψ, respectively. We present an algorithm effective for training such a model on the corrupted annotated dataset $\mathcal{X}$ without overfitting to the noise with reliable classification of samples within the class distribution.

3.2. Self-supervised Learning

Initially, regardless of the noisy labels, the model is trained in a self-supervised way to learn representations from images. The goal is to extract rich feature representations not only to mitigate the effect of ID noise but also to capture clusters of similar images to identify OOD samples in early stages. To achieve this, our algorithm learns representations by maximizing the agreement between two distinct augmented versions of the same image using a contrastive loss in the latent space. This approach is inspired by recent contrastive learning models [6, 17].

To train the model, two random data augmentations are applied to each training image x_i to provide two related views of the same example. These views are considered as a positive pair in the contrastive learning algorithm and are denoted as $\boldsymbol{x}'_i$ and $\boldsymbol{x}'_j$. A neural network encoder, $f_1(\cdot;\theta)$, and a small neural network projection head, $g(\cdot;\psi)$, are constructed to map $\boldsymbol{x}$ to the representation space $\boldsymbol{h}_i = f_1(\boldsymbol{x}'_i;\theta) \in \mathbb{R}^d$, and h_i to the $\boldsymbol{z}_i$ space, where the contrastive loss is applied (Figure 1-(b)). The projection head is removed at inference time. Both augmented samples feed separately into the same encoder, resulting in a pair of representation vectors $(\boldsymbol{h}_i, \boldsymbol{h}_j)$ and projection heads $(\boldsymbol{z}_i, \boldsymbol{z}_j)$. The loss function for each pair of examples (i, j) is defined as self-supervised contrastive learning (e.g., [17]), as follows:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2M} \mathbb{1}_{k\neq i} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \quad (1)$$

where $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ denotes the cosine similarity between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, $\mathbb{1}_{k\neq i} \in \{0, 1\}$ is an indicator function evaluating 1 if $k \neq i$, and τ denotes a temperature parameter. The final loss, $\mathcal{L}^{self} = \sum_{i\in I} \mathcal{L}_i^{self}$, is computed across all augmented pairs in a mini-batch I, where $\mathcal{L}_i^{self}$ defined in Eq. (1).

3.3. Sample Selection

Following self-supervised training, we first detect potential OOD samples based on the trained embedding space and remove them from the training set (Figure 1-(b)). Additionally, we add a linear classifier, $f_2(.;\phi)$, on top of the learned representations for classification. We train the classifier for a few epochs on the whole training set with a symmetric cross-entropy loss [34] to warm it up and make it robust to a high level of label noise, even at the beginning of training. Next, we divide remaining samples into clean and noisy in-distribution sets to start the Semi-SL algorithm. We iteratively apply filtering to identify noisy samples (ID and OOD) during training explained in the following subsections.

OOD Label Noise Detection: Rather than defining a new metric based on the predictions of the model and given noisy labels [2], Generative Adversarial Networks [42], or autoencoder architecture [35], we utilize a simple algorithm to detect OOD samples by leveraging the K nearest neighbours of any example x and their distance to x as a primitive estimate of the local density around x [18]. To determine the degree of OOD in a given image x, we calculate the average distance between x and each of its K nearest neighbours from the training data, which has demonstrated acceptable OOD detection results.

ID Label Noise Detection: In the presence of ID label noise and prior to overfitting, we may anticipate that the properly classified samples are those with clean labels, while the remaining samples have noisy labels or are complicated ones. Let $p\left(x_i = \text{clean}|\ell_i^{sup}, \gamma\right)$ be a function that estimates the probability that $(\mathbf{x}_i, \mathbf{y}_i)$ is a clean label sample based on $\ell_i^{sup} = -\mathbf{y}_i \log p_\phi(y_i|f_\theta(\mathbf{x}_i))$ which is the simple cross-entropy loss function. The probability $p\left(x_i = \text{clean}|\ell_i^{sup}, \gamma\right)$ can be estimated using a bimodal Gaussian mixture model (GMM) [19]. γ represents the GMM parameters, where the component with the larger mean value is considered to be the noisy component, while the component with the lower mean value is considered to be the clean component. As a result, clean ($\mathcal{X} \subseteq \mathcal{D}$) and possibly ID labeled noisy samples ($\mathcal{U} \subseteq \mathcal{D}$) are formed as follows:

$$\begin{aligned}\mathcal{X} &= \left\{(\mathbf{x}_i, \mathbf{y}_i, w_i) : w_i = p\left(x_i = \text{clean}|\ell^{sup}, \gamma\right) \geq \tau_2\right\}, \\ \mathcal{U} &= \left\{(\mathbf{x}_i, \mathbf{y}_i, w_i) : w_i = p\left(x_i = \text{clean}|\ell^{sup}, \gamma\right) < \tau_2\right\},\end{aligned} \quad (2)$$

where τ_2 denotes a clustering threshold.

3.4. Semi-supervised Learning

The presence of some noisy ID and OOD data in the fraction of clean samples results in noisy Semi-SL training. To address this issue, we propose to incorporate three components in our Semi-SL training pipeline: 1) we use the symmetric cross entropy as a robust-to-noise loss function for the supervised loss to reduce the risk of noisy label memorization; 2) we include contrastive learning to enable feature learning without labels/pseudo-labels; and 3) we use the manifold mixup as the augmentation of the feature space to enable feature learning for iterative OOD detection. This unsupervised feature learning approach significantly reduces

the risk of noisy label memorization since it does not rely on the incorrect split of clean and noisy samples or on the incorrect pseudo-labels generated during Semi-SL training. Figure 1-(c) depicts the specifics of our SSL model with semi-supervised learning on input and embedding spaces. We employ Semi-SL by combining input mixup augmentation as well as manifold augmentation on top of the basic ideas of FixMatch algorithm, which we call "MixEMatch".

To achieve this, we create four copies of each sample, two with weak and two with strong random augmentations. We use the weakly augmented samples of labeled data, denoted as x, and unlabeled data, denoted as u, to create the smoothing labels and estimate the pseudo-labels for the labeled and unlabeled sets, respectively. We use the weakly augmented images to create pseudo-labels for the strongly augmented images and train the model in a semi-supervised manner on the strongly augmented images. We incorporate mixup into our pipeline similar to the MixMatch algorithm with two differences: 1) we mix strongly augmented labeled and unlabeled images with their labels, which are smoothed and guessed based on the weakly augmented labeled and unlabeled samples, respectively; and 2) we apply mixup to both the input and embedding spaces. As a result, the outputs of our MixEMatch algorithm are:

$$\mathcal{X}', \mathcal{U}', \mathcal{H}'_x, \mathcal{H}'_u = \text{MixEMatch}(\mathcal{X}, \mathcal{U}, \mathcal{H}_x, \mathcal{H}_u, T, \alpha) \quad (3)$$

where $\mathcal{X}, \mathcal{H}_x$ are input and embedding spaces of strongly augmented labeled images and $\mathcal{U}, \mathcal{H}_u$, are the ones for unlabeled images. Then $\mathcal{X}', \mathcal{U}'$ are the mixed up augmented input space of labeled and unlabeled sets while $\mathcal{H}_x, \mathcal{H}_u$ are the mixed up augmented embedding space of labeled and unlabeled sets, respectively. The full MixEMatch algorithm is provided in Algorithm 1. In the following subsections, these augmentations and objective functions are explained.

Mixup Augmentations: Applying mixup on the embedding space allows us to use semantic interpolations of the deepest layer as an additional training signal, which encourages our classifier to generate less confident predictions for interpolations of representations. Although high-level representations are often low-dimensional and useful for linear classifiers, linear interpolations of hidden representations should effectively explore significant sections of the feature space. To employ combinations of hidden data representations as a new training signal, we perform the same linear interpolation on the corresponding pair of one-hot labels, resulting in mixed samples with soft targets.

Let x_1^s, x_2^s be strongly augmented images of samples x_1, x_2 with their corresponding pseudo-labels probabilities p_1, p_2, and h_1^s, h_2^s be the embedding spaces of samples x_1^s and x_2^s. The mixed input-label pair (x', p') and mixed embedding-label pair (h', p') are computed as follows:

$$\lambda \sim \text{Beta}(\alpha, \alpha), \quad (4)$$

$$\lambda' = \max(\lambda, 1-\lambda), , \quad (5)$$

$$x' = \lambda' x_1^s + (1-\lambda') x_2^s, \quad (6)$$

$$h' = \lambda' h_1^s + (1-\lambda') h_2^s, \quad (7)$$

$$p' = \lambda' p_1 + (1-\lambda') p_2, \quad (8)$$

where α is a hyperparameter. Consequently, $\mathcal{X}$, $\mathcal{U}$, $\mathcal{H}_x$, $\mathcal{H}_u$ are transformed into $\mathcal{X}'$, $\mathcal{U}'$ and $\mathcal{H}'_x$, $\mathcal{H}'_u$, which are collections of multiple mixed-up augmentations of each example with the corresponding labels.

Loss Functions: The Semi-SL loss function is computed based on the strongly augmented copies as well as the mixUp augmentations of the input and embedding spaces of samples as follows:

$$\mathcal{L}^{semi} = \mathcal{L}^{sup} + \lambda_u \mathcal{L}^{unsup} + \lambda_c \mathcal{L}^{self}, \quad (9)$$

The supervised loss function ($\mathcal{L}^{sup}$) consists of two symmetric cross-entropy loss terms on the mixed-up augmented input ($\mathcal{X}'$) and the mixed-up augmented embedding spaces ($\mathcal{H}'_x$) of the labeled set. The unsupervised loss function ($\mathcal{L}^{unsup}$) consists of two Mean Squared loss terms on the mixed-up augmented input ($\mathcal{U}'$) and the mixed-up augmented embedding spaces ($\mathcal{H}'_u$) of the unlabeled set. The contrastive loss function ($\mathcal{L}^{self}$), defined in Eq. (1), is applied to the projection head of the embedding spaces of $\mathcal{H}_x$ and $\mathcal{H}_u$.

4. Experiments and Results

We evaluate our model in terms of classification accuracy on a variety of standard benchmarks explained below.

CIFAR-10/100: The dataset used for our experiments consists of 50K and 10K training and test images with 10 and 100 image classes, respectively. Synthetic ID noise (symmetric and asymmetric) is often used to assess noise-robust algorithms on these two datasets. Our first set of experiments was conducted on a corrupted CIFAR-100, where controlled ID and OOD noise was added. We followed the same configuration proposed in [1]. ID noise was introduced by randomly switching the labels of $r_{in}\%$ of the training dataset to a random one. Additionally, $r_{out}\%$ of the training images were replaced with images from another dataset, either ImageNet32 [7] or Places365 [44], as OOD samples. By adding noise, the dataset still had the same

Algorithm 1 MixEMatch

Input: $\mathcal{X} = \big((x_b, y_b, w_b); b \in (1, \dots, B)\big)$ ▷ batch of labeled examples, their labels and probability of being clean sample
Input: $\mathcal{U} = \big(u_b, w_b; b \in (1, \dots, B)\big)$ ▷ batch of unlabeled examples and probability of being clean sample
Input: T, α ▷ Sharpening temperature, Parameter of Beta distribution for Mixup

for $b \leftarrow 1$ **to** B **do**
  $x^w_{b,1}, x^w_{b,2} \leftarrow \text{Aug}^W(x_b, 2)$ ▷ apply two weak augmentations on x_b
  $x^s_{b,1}, x^s_{b,2} \leftarrow \text{Aug}^S(x_b, 2)$ ▷ apply two strong augmentations on x_b
  $u^w_{b,1}, u^w_{b,2} \leftarrow \text{Aug}^W(u_b, 2)$ ▷ apply two weak augmentations on u_b
  $u^s_{b,1}, u^s_{b,2} \leftarrow \text{Aug}^S(u_b, 2)$ ▷ apply two strong augmentations on u_b
  $p_b \leftarrow \frac{1}{2}\big(\text{p}(x^w_{b,1}; \theta, \phi) + \text{p}(x^w_{b,2}; \theta, \phi)\big)$ ▷ average the predictions across augmentations of x_b
  $\bar{y}_b \leftarrow w_b y_b + (1 - w_b) p_b$ ▷ refine ground-truth label guided by the clean probability produced by the GMM
  $y_b \leftarrow \text{Sharpen}(\bar{y}_b, T)$ ▷ apply temperature sharpening to the refined label
  $\bar{q}_b \leftarrow \frac{1}{2}\big(\text{p}(u^w_{b,1}; \theta, \phi) + \text{p}(u^w_{b,2}; \theta, \phi)\big)$ ▷ guessing the label for unlabeled sample by averaging the predictions
  $q_b \leftarrow \text{Sharpen}(\bar{q}_b, T)$ ▷ apply temperature sharpening to the guessed label
  $h_{x_{b,1}} \leftarrow \text{f}_1(x^s_{b,1}; \theta)$ ▷ extract embedding of x^s_b
  $h_{x_{b,2}} \leftarrow \text{f}_1(x^s_{b,2}; \theta)$ ▷ extract embedding of x^s_b
  $h_{u_{b,1}} \leftarrow \text{f}_1(u^s_{b,1}; \theta)$ ▷ extract embedding of u^s_b
  $h_{u_{b,2}} \leftarrow \text{f}_1(u^s_{b,2}; \theta)$ ▷ extract embedding of u^s_b

  $\mathcal{X} \leftarrow \{(x^s_{b,1}, y_b, w_b), (x^s_{b,2}, y_b, w_b); b \in (1, ..., B)\}$ ▷ augmented labeled mini-batch
  $\mathcal{U} \leftarrow \{(u^s_{b,1}, q_b, w_b), (u^s_{b,2}, q_b, w_b); b \in (1, ..., B)\}$ ▷ augmented unlabeled mini-batch
  $\mathcal{X}', \mathcal{U}' \leftarrow \text{Mixup}(\mathcal{X}, \mathcal{U}, \alpha)$ ▷ Apply mixup on input data

  $\mathcal{H}_x \leftarrow \{(h_{x_{b,1}}, y_b), (h_{x_{b,2}}, y_b); b \in (1, ..., B)\}$ ▷ augmented labeled mini-batch
  $\mathcal{H}_u \leftarrow \{(h_{u_{b,1}}, q_b), (h_{u_{b,2}}, q_b); b \in (1, ..., B)\}$ ▷ augmented unlabeled mini-batch
  $\mathcal{H}'_x, \mathcal{H}'_u \leftarrow \text{Mixup}(\mathcal{H}_x, \mathcal{H}_u, \alpha)$ ▷ Apply mixup on embedding space of input data

return $\mathcal{X}', \mathcal{U}', \mathcal{H}'_x, \mathcal{H}'_u$

number of images as the original CIFAR-100 dataset.

Mini-ImageNet: It consists of 50k and 10K training and test images with 100 classes, respectively. To study real-world web label noise in a controlled setting, we use Web-corrupted Mini-ImageNet from the Controlled Noisy Web Labels (CNWL) dataset [15] with different noise levels (20%, 40%, 60%, 80%). We train our model on the red Mini-ImageNet images which are resized to 32×32 pixels, similar to the recent works [8, 28].

(mini)Webvision: It is constructed by using first 50 classes of a real-world noisy Webvision dataset [21], consisting of 66k and 2.5k training and test images, respectively. We train our model on a 227×227 image. We report the accuracy on the mini-WebVision validation.

Clothing1M: It consists of one million training images gathered from online shopping websites. These images are classified into 14 different classes. There are three additional sets of clean images: 50K for training, 14K for validation, and 10K for testing. The clean training and validation sets are not utilized in our experiments, but the clean test set is used for evaluation purposes.

4.1. Training Details

We leverage PreActResNet-18 for the CIFAR-10, CIFAR-100 and mini-ImageNet datasets and PreActResNet-50 for two others as an encoder model. Then, a multi-layer perceptron with a single hidden layer used as a projection head to project the representation onto a 128-dimensional latent space. The normalized activations of the final pooling layer 512 and 2048 are used as the representation vector in PreActResNet-18 and PreActResNet-50, respectively. The self-supervised model is trained for 1000 epochs with a batch size of 1024 on the corrupted datasets. Supervised training starts by filtering 10% of samples detected as OOD. Then the Semi-SL step is done for 300 epochs, where the model is trained in a supervised manner for 20 epochs as a warmup, and training is continued in a semi-supervised way. The learning rate of the linear classifier is set at 0.1, while the backbone is tuned with a learning rate of 0.001

Corruption	ImageNet32				Places365			
Noise Ratio (r_{out},r_{in})	20%, 20%	40%, 20%	60%, 20%	40%, 40%	20%, 20%	40%, 20%	60%, 20%	40%, 40%
CE	55.5	44.3	26.0	18.5	53.6	42.5	21.4	13.9
Mixup [41]	62.5	53.2	40.4	33.9	59.7	48.6	33.7	27.6
JoSRC [38]	64.2	61.4	37.1	41.4	66.7	60.6	39.6	32.6
ELR [23]	68.5	63.0	44.6	34.2	68.5	62.3	36.5	33.9
EDM [27]	70.4	61.8	14.6	1.6	70.3	61.6	14.7	11.9
DSOS [2]	70.5	62.1	49.1	42.9	69.1	59.5	35.4	29.5
RRL [20]	72.3	65.4	24.5	30.6	72.5	65.8	49.3	24.3
SNCP [1]	72.7	67.1	51.3	52.7	71.1	63.5	49.8	47.6
Ours	**75.3**	**69.1**	**59.8**	**63.0**	**74.5**	**69.3**	**59.0**	**62.5**

Table 1. Comparison of classification accuracy with the state-of-the-art methods on CIFAR100 corrupted with ImageNet32 or Places365 images with ID and OOD noise of ratio of r_{in} and r_{out}, respectively. Results of other previous methods are from [1].

and the help of a cosine annealing scheduler. We use SGD with momentum of 0.9 and weight decay of 5×10^{-4}. For OOD sample selection with KNN and ID sample selection with GMM, we set $k = 100$ and $\tau_2 = 0.3$ for all of our experiments, respectively. More details are delineated in the Supplementary section.

Augmentations: We use AutoRandAugment [9] for SSL training. For the weak augmentation of Semi-SL step, we sequentially apply simple augmentations by random cropping followed by resizing back to the original size and random horizontal flip. For the strong augmentation, we follow the Auto-Augment policy explained in [9] based on CIFAR10 and ImageNet policy. The same CIFAR10-policy has been applied to both the CIFAR10/100 datasets, while the ImageNet-policy has been used for others.

4.2. Results

The average test accuracies for the CIFAR100 dataset, corrupted with ID symmetric label noise and OOD images from the ImageNet32 and Places365 datasets, are reported in Table 1. Similar to [1, 2], the focus of our experiments is on OOD labeled noise. Therefore, we investigate the effect of three different levels of OOD noise rates with $r_{out} \in [20\%, 40\%, 60\%]$ and two different ratios of ID noise $r_{in} \in [20\%, 40\%]$. We show the benefits of using Manifold DivideMix to detect OOD noise data and correct the labels of ID noise samples in Table 1. It is challenging to achieve reasonable performance at a high level of noise ratio by using a fully supervised learning model, as there are only a small number of available samples per class. However, by using SSL pre-training, Manifold DivideMix consistently shows performance improvement under different noise settings. For 40% ID and 40% OOD noise rates, our method achieves about 10% improvement over state-of-the-art methods. Our method shows a high level of resilience to ID and OOD noisy labels, proving to be superior to other methods.

In the second set of experiments, we evaluate our method on the corrupted miniImageNet as a controlled real-world label noise dataset [15]. Table 2 illustrates the performance improvements of our method compared to the state-of-the-art. Our method achieves more than a 1% improvement for all levels of noise rates. However, the SNCF method [1] and PLS [3], specifically designed for both OOD and ID noise, as well as ScanMix [28] and PropMix [8] using self-supervised initialization, under-perform compared to our method.

Method	Noise Ratio			
	20%	40%	60%	80%
CE	47.4	42.7	37.3	29.8
Mixup [41]	49.1	46.4	40.6	33.6
DivideMix [19]	51.0	46.7	43.1	34.5
ScanMix [28]	59.1	54.5	52.4	40.0
PropMix [8]	61.2	56.2	52.8	43.4
SNCP [1]	61.6	59.9	54.9	45.6
PLS [3]	63.1	60.0	54.4	46.5
Ours	**64.4**	**61.4**	**56.2**	**47.8**

Table 2. Comparison of classification accuracy with the state-of-the-art methods on Web-corrupted miniImageNet from the CNWL [15] (32×32).

We further evaluate our proposed pipeline based on the top-1 and top-5 accuracy of (mini)WebVision and Clothing1M reported in Table 3. We believe that by using the self-supervised pre-training step and considering contrastive loss during the semi-supervised step, the model learns more generalizable features, which reduces the risk of overfitting to noisy samples as well as overconfident prediction on the semantically different class samples in the noisy real-world dataset. Our approach achieves superior results compared to the recent GSS-SSL method [40], establishing a new state-of-the-art top-1 accuracy for (mini)WebVision. However, it is worth noting that the GSS-SSL method outperforms ours on the Clothing1M dataset.

Method	Clothing1M	(mini)WebVision	
	Top1	Top1	Top5
GCE [43]	71.7	61.2	80.8
SL [34]	72.1	63.8	84.3
ELR+ [23]	71.5	63.6	83.5
Co-teaching [11]	72.5	64.1	85.0
JoCor [38]	71.7	60.8	82.5
DivideMix [19]	74.6	77.2	91.6
GSS-SSL [40]	**74.9**	77.4	**93.1**
Ours	73.1	**78.4**	92.0

Table 3. Comparison of classification accuracy with the state-of-the-art methods on Clothing1M and (mini)Webvision.

Method	Symmetric Noise				Asymmetric Noise		
	20%	50%	80%	90%	10%	30%	40%
CE	86.8	79.4	62.9	42.7	88.8	81.7	76.1
MixUp	95.6	87.1	71.6	52.2	93.3	83.3	77.7
DivideMix	95.0	93.7	92.4	74.2	93.8	92.5	91.4
ELR	95.8	94.8	93.3	78.7	95.5	94.8	93.0
UNICON	96.0	95.6	93.9	88.1	95.3	94.8	94.1
Propmix	**96.1**	95.5	93.7	93.2	-	-	**94.6**
GSS-SSL	94.3	-	91.6	-	-	92.4	91.8
Ours	96.0	**95.7**	**94.6**	**93.7**	**95.6**	**95.0**	93.8

Table 4. Comparison of classification accuracy on CIFAR10 with ID symmetric and asymmetric noise.

4.3. Ablation Studies

First, we analyze our method under the ID label noise scenario applied in CIFAR10 dataset under different symmetric noise rates as well as asymmetric noise rates in Table 4. Our method outperforms the baseline methods for the cases of 50% and 80% symmetric noise rates, and achieves a significantly better performance over the state-of-the-art for 90% noise rate. For high noise rates, loss-based selection procedures [19] often fail due to the selection of a large number of noisy samples as clean ones. As a result, SSL pre-training helps our method, as well as PropMix [8], to not struggle in the presence of sever label noise. Furthermore, we investigate the asymmetric noise scenario, where each class is not equally affected by label noise, leading to a more complicated selection process. Therefore, at high levels of label noise, the balancing approach in UNICON and PropMix achieves better performance. However, for low noise rates, our method performs slightly better.

In Table 5, we evaluate the impact of each component of our proposed method by investigating scenarios where we only apply sample selection for ID noise, as well as the combined impact of the ID and OOD sample selection (denoted as ID + OOD). In this experiment, we demonstrate the effectiveness of our MixEMatch algorithm by comparing it to the FixMatch algorithm in terms of model generalization

		FixMatch	MixEMatch	Best	Last
No noise corr	CE	✗	✗	41.4	18.5
	SSL+LC	✗	✗	47.5	19.9
Noise Robust	ID	✓	✗	58.6	58.3
	ID + OOD	✓	✗	58.0	58.0
	ID	✗	✓	61.6	61.4
	ID + OOD	✗	✓	**63.1**	**63.0**

Table 5. Ablation study on CIFAR-100 corrupted with ImageNet32 with $r_{out} = 40\%$ and $r_{in} = 40\%$.

during test time, specifically by incorporating both input and embedding mixup techniques. Unlike FixMatch, MixEMatch applies mixup to both input and embedding spaces, whereas FixMatch only applies mixup on the input space. In the Semi-SL step, the MixEMatch algorithm yields considerable gains (about 5% accuracy) for the Manifold DivideMix model. This implies that MixEMatch takes advantage of the Manifold mixup idea, which is useful for both ID and OOD sample selection algorithms. Additional experiments and ablation studies can be found in the supplementary material.

5. Conclusion

We have developed a semi-supervised framework that leverages self-supervised learning to address both in- and out-of-distribution label noise. To achieve this, we first train a deep neural network model using unsupervised contrastive learning on a noisy dataset to extract robust and flexible embedding representations of the input data that are not limited by explicit labels. Then, we use a basic K-Nearest Neighbors algorithm in the embedding space to identify likely out-of-distribution samples and exclude them from the subsequent semi-supervised learning stage. We also add a linear classifier on top of the self-supervised model to identify in- and out-of-distribution label noise, as well as clean sample sets, based on the clustering of loss values. To further improve the generalization performance and representation learning, we propose MixEMatch, an algorithm that applies mixup augmentation in both the input and representation spaces during the semi-supervised learning step. These augmentations enhance the overall quality of pseudo-labels and significantly improve the effectiveness of semi-supervised learning. Our experiments show that unsupervised feature learning reduces the effects of overfitting to label noise and improves noisy sample selection, particularly in the presence of severe noise. Our approach outperforms recent methods and offers more consistency across varying noise levels. For example, with 40% in- and 40% out-of-distribution label noise, our model improves the accuracy of corrupted CIFAR-100 over 10%. Additionally, our proposed approach achieves state-of-the-art top1 accuracy on the WebVision and mini-ImageNet datasets.

References

[1] Paul Albert, Eric Arazo, Noel E O'Connor, and Kevin McGuinness. Embedding contrastive unsupervised features to cluster in-and out-of-distribution noise in corrupted image datasets. In *Computer Vision–Eccv 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 402–419. Springer, 2022. 3, 5, 7

[2] Paul Albert, Diego Ortego, Eric Arazo, Noel O'Connor, and Kevin McGuinness. Addressing Out-of-Distribution Label Noise in Webly-Labelled Data. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. 1, 2, 3, 4, 7

[3] Paul Albert, Eric Arazo, Tarun Krishna, Noel E O'Connor, and Kevin McGuinness. Is your noise correction noisy? pls: Robustness to label noise with two stage detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 118–127, 2023. 7

[4] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A Closer Look at Memorization in Deep Networks. In *International Conference on Machine Learning (ICML)*, 2017. 1

[5] D. Berthelot, N. Carlini, I.J. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 3

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, 2020. 1, 4

[7] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A Downsampled Variant of Imagenet as an Alternative to the Cifar Datasets. *arXiv: 1707.08819*, 2017. 5

[8] Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. PropMix: Hard Sample Filtering and Proportional MixUp for Learning With Noisy Labels. *arXiv: 2110.11809*, 2021. 3, 6, 7, 8

[9] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 7

[10] J. Goldberger and E. Ben-Reuven. Training Deep Neural-Networks Using a Noise Adaptation Layer. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[11] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-Teaching: Robust Training of Deep Neural Networks With Extremely Noisy Labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 3, 8

[12] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[13] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv Preprint arXiv:1712.00409*, 2017. 1

[14] L. Jiang, Z. Zhou, T. Leung, L.J. Li, and L. Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *International Conference on Machine Learning (ICML)*, 2018. 3

[15] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond Synthetic Noise: Deep Learning on Controlled Noisy Labels. In *International Conference on Machine Learning (ICML)*, 2020. 6, 7

[16] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022. 3

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 2, 4

[18] Johnson Kuan and Jonas Mueller. Back to the basics: Revisiting out-of-distribution detection baselines. *arXiv Preprint arXiv:2207.03061*, 2022. 4

[19] Junnan Li, Hoi Socher, Richard, Steven CH, and Steven CH Hoi. DivideMix: Learning With Noisy Labels as Semi-Supervised Learning. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 3, 4, 7, 8

[20] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning From Noisy Data With Robust Representation Learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9485–9494, 2021. 3, 7

[21] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool. WebVision Database: Visual Learning and Understanding From Web Data. *arXiv: 1708.02862*, 2017. 1, 6

[22] Huafeng Liu, Chuanyi Zhang, Yazhou Yao, Xiu-Shen Wei, Fumin Shen, Zhenmin Tang, and Jian Zhang. Exploiting web images for fine-grained visual recognition by eliminating open-set noise and utilizing hard examples. *IEEE Transactions on Multimedia*, 24:546–557, 2021. 1

[23] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. Early-Learning Regularization Prevents Memorization of Noisy Labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 7, 8

[24] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015. 2

[25] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-Objective Interpolation Training for Robustness to Label Noise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

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

[27] Ragav Sachdeva, Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. EvidentialMix: Learning

With Combined Open-Set and Closed-Set Noisy Labels. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. 3, 7

[28] Ragav Sachdeva, Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. ScanMix: Learning From Severe Label Noise via Semantic Clustering and Semi-Supervised Learning. *arXiv: 2103.11395*, 2021. 3, 6, 7

[29] Murat Sensoy, Lance Kaplan, Melih Kandemir, and Melih Kandemir. Evidential Deep Learning to Quantify Classification Uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[30] Y. Shen and S. Sanghavi. Learning With Bad Training Data via Iterative Trimmed Loss Minimization. In *International Conference on Machine Learning (ICML)*, 2019. 3

[31] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. 1, 3

[32] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[33] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 2

[34] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. Symmetric Cross Entropy for Robust Learning With Noisy Labels. In *IEEE International Conference on Computer Vision (ECCV)*, 2019. 2, 4, 8

[35] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv Preprint arXiv:2110.11334*, 2021. 4

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

[37] Yazhou Yao, Jian Zhang, Fumin Shen, Xiansheng Hua, Jingsong Xu, and Zhenmin Tang. Exploiting web images for dataset construction: A domain robust approach. *IEEE Transactions on Multimedia*, 19(8):1771–1784, 2017. 1

[38] Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-Src: A Contrastive Approach for Combating Noisy Labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 7, 8

[39] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019. 2

[40] Xiaotian Yu, Yang Jiang, Tianqi Shi, Zunlei Feng, Yuexuan Wang, Mingli Song, and Li Sun. How to prevent the continuous damage of noises to model training? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12054–12063, 2023. 3, 7, 8

[41] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 2, 7

[42] Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, pages 12427–12436. PMLR, 2021. 4

[43] Z. Zhang and M. Sabuncu. Generalized Cross Entropy Loss for Training Deep Neural Networks With Noisy Labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 8

[44] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5

[45] Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *International Conference on Machine Learning*, pages 12846–12856. PMLR, 2021. 2