PropMix: Hard Sample Filtering and Proportional MixUp for Learning with Noisy Labels

Filipe R. Cordeiro[1]
filipe.rolim@ufrpe.br

Vasileios Belagiannis[2]
vasileios.belagiannis@uni-ulm.de

Ian Reid[3]
ian.reid@adelaide.edu.au

Gustavo Carneiro[3]
gustavo.carneiro@adelaide.edu.au

[1] Universidade Federal Rural de Pernambuco
Recife, Brazil

[2] Universität Ulm
Ulm, Germany

[3] University of Adelaide
Adelaide, Australia

Abstract

The most competitive noisy label learning methods rely on an unsupervised classification of clean and noisy samples, where samples classified as noisy are re-labelled and "MixMatched" with the clean samples. These methods have two issues in large noise rate problems: 1) the noisy set is more likely to contain hard samples that are incorrectly re-labelled, and 2) the number of samples produced by MixMatch tends to be reduced because it is constrained by the small clean set size. In this paper, we introduce the learning algorithm PropMix to handle the issues above. PropMix filters out hard noisy samples, with the goal of increasing the likelihood of correctly re-labelling the easy noisy samples. Also, PropMix places clean and re-labelled easy noisy samples in a training set that is augmented with MixUp, removing the clean set size constraint and including a large proportion of correctly re-labelled easy noisy samples. We also include self-supervised pre-training to improve robustness to high noisy label scenarios. Our experiments show that PropMix has state-of-the-art (SOTA) results on CIFAR-10/-100 (with symmetric, asymmetric and semantic label noise), Red Mini-ImageNet (from the Controlled Noisy Web Labels), Clothing1M and WebVision. In severe label noise benchmarks, our results are substantially better than other methods. The code is available at https://github.com/filipe-research/PropMix.

1 Introduction

Deep neural network models reached promising results in several computer vision applications recently [1, 14, 39]. Nevertheless, the superior performance depends on the availability of well-curated large-scale training sets with clean annotations [32]. The labeling process can produce noisy labels due to human failure, low-quality data, and challenging labelling tasks [11]. The problem is that noisy labels in the training set harms the learning process by reducing model generalization [56]. Developing methods that are robust to label noise is

important to deal with real-world applications, where noisy annotations is often part of the training set.

A common approach to address this challenge is based on semi-supervised learning (SSL) methods [10, 30, 41, 42], based on a 2-stage process formed by an unsupervised learning method to classify training samples as clean or noisy, followed by SSL that "Mix-Matches" [4] the labelled set formed by the samples classified as clean, and the re-labelled samples from the noisy set. There are two issues with such approach. First, samples classified as noisy can be easy or hard to be re-labelled, where the hard samples are unlikely to be correctly re-labelled, which can bias the training process. Second, MixMatch [4] relies on a one-to-one sampling between the clean and noisy sets, where the total number of samples is constrained by the clean set size. However, the clean set tends to be smaller and the noisy set tends to have more incorrectly re-labelled samples for larger noise rates–such fact impairs the robustness of SSL methods for large noise rate problems. We hypothesise that by filtering out hard noisy samples, we can reduce the risk of over-fitting those samples. Furthermore, by re-labelling the easy noisy samples and mixing them with the clean set with MixUp [57] for training a classifier, we not only remove the constraint on the clean set size, but also use a noisy set with better chances to have correctly re-labelled samples, allowing the method to be more robust to large noise rate problems.

In this paper, we propose a new learning algorithm called PropMix that addresses the two points above. PropMix filters out hard noisy samples via a two-stage process, where the first stage classifies samples as clean or noisy using the loss values, and the second stage eliminates hard noisy samples using their classification confidence. Then, by re-labelling the easy noisy samples with the model output, adding these samples to the training set, and running a regular classification training with MixUp [57], we show that PropMix is robust to a wide range of noise rates. To improve the feature representation and model confidence in high noise scenarios, we also add a self-supervised pre-training stage [6, 7, 12, 19]. Empirical results on CIFAR-10/-100 [25] under symmetric, asymmetric and semantic noise, show that PropMix outperforms previous approaches. For high-noise rate problems in CIFAR-10/-100 [30] and Red Mini-ImageNet from the Controlled Noisy Web Labels [52], PropMix presents the best results in the field by a substantial margin.

2 Prior Work

Recently proposed noisy label learning methods rely on the following strategies: *robust loss* functions [36, 48, 50], *sample selection* [16], *label cleansing* [21, 55], *sample weighting* [45], *meta-learning* [15], *ensemble learning* [38] and *semi-supervised learning* (SSL) [4]. The most successful approaches use SSL, combined with other methods [22, 30, 33].

SSL methods [30, 41] first classify samples as clean or noisy, where the noisy samples are re-labelled by the model, and these clean and noisy sets are combined with MixMatch [4]. As mentioned before, SSL methods have two issues: 1) the training set size for the MixMatch stage is limited by the clean set size that reduces with increasing label noise, and 2) the noisy sample re-labelling accuracy also reduces with increasing label noise. Although the first point is not addressed by SSL methods, the second point is mitigated by replacing the cross entropy (CE) loss by the Mean Absolute Error (MAE) loss to fit the noisy samples. Although MAE has been shown to be robust to label noise [13], it tends to underfit the training set [36].

An alternative way to reduce the risk of overfitting incorrectly re-labelled noisy samples is by rejecting them altogether [16, 44], but such rejection ignores the information present in noisy samples, which can decrease the effectiveness of the approach. Moreover, in high

noise rate scenarios, the filtered clean set can be too small to train the model. The noisy set can be used after re-labelling the noisy samples [30, 41, 49], which works well for low noise rate problems, but as the noise rate increases, this approach is not effective given that the incorrectly re-labelled noisy samples tend to bias the training. The use of clean validation sets in a meta-learning approach [2] can reduce this issue, but the existence of a clean validation set may be infeasible in real world applications. SELFIE [47] proposes label correction to a subset of samples that present consistent prediction, while discarding samples that are less consistent. The main issue with SELFIE is that the classification of clean samples is based on a loss threshold that might include some relabeling in the predicted clean set, producing false positives. Our method proposes a hybrid approach. We claim that hard noisy samples are unlikely to have their label corrected, mainly in a high noise scenario. On the other hand, we can find easy noisy samples that are likely to be correctly relabelled and used in a supervised training. The main difference of existing filtering methods and our approach is that we filter out hard noisy samples, while keeping easy noisy samples to be relabelled and included in the training process. We show in the experiments that easy noisy samples can be relabelled correctly with high accuracy, whereas hard noisy samples are unlikely to be correctly relabelled and therefore should be removed from training.

3 Method

3.1 Problem Definition

Consider the training set be denoted by $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}|}$, with $\mathbf{x}_i \in \mathcal{S} \subset \mathbb{R}^{H \times W \times 3}$ being the ith RGB image of size $H \times W$, and $\mathbf{y}_i \in \{0,1\}^{|\mathcal{Y}|}$ denoting a one-hot vector representing the given label, with $\mathcal{Y} \in \{1, ..., |\mathcal{Y}|\}$ denoting the set of labels, and $\sum_{c \in \mathcal{Y}} \mathbf{y}_i(c) = 1$. The hidden true label $\hat{\mathbf{y}}_i$ can differ from the given label $\mathbf{y}_i$ that is a result of noise process, represented by $\mathbf{y}_i \sim p(\mathbf{y}|\mathbf{x}_i, \mathcal{Y}, \hat{\mathbf{y}}_i)$, with $p(\mathbf{y}(j)|\mathbf{x}_i, \mathcal{Y}, \hat{\mathbf{y}}_i(c)) = \eta_{jc}(\mathbf{x}_i)$, where the $j, c \in \mathcal{Y}$ are the classes, $\eta_{jc}(\mathbf{x}_i) \in [0,1]$ the probability of flipping the class c to j, and $\sum_{j \in \mathcal{Y}} \eta_{jc}(\mathbf{x}_i) = 1$. There are three common types of noise in the literature: symmetric [24], asymmetric [43], and semantic [27]. The symmetric noise is a noisy type where the hidden label are flipped to a random class with a fixed probability η, where the true label is included into the label flipping options, which means that $\eta_{jc}(\mathbf{x}_i) = \frac{\eta}{|\mathcal{Y}|-1}, \forall j \in \mathcal{Y}$, such that $j \neq c$, and $\eta_{cc}(\mathbf{x}_i) = 1 - \eta$. The asymmetric noise has its labels flipped between similar-looking object categories [43], where $\eta_{jc}(\mathbf{x}_i)$ depends only on the classes $j, c \in \mathcal{Y}$, but not on $\mathbf{x}_i$. Finally, the semantic noise [27] is the noisy type where the label flipping depends both on the classes $j, c \in \mathcal{Y}$ and image $\mathbf{x}_i$.

3.2 PropMix

The proposed PropMix algorithm (Fig. 1) starts with a self-supervised pre-training [5, 7, 12, 19]. Then, we perform a supervised training, with a new filtering step to identify clean samples, easy noisy samples, and hard noisy samples, which are removed from training. The easy noisy samples are re-labelled and proportionally combined with clean samples using MixUp [52] for supervised training.

The self-supervised pre-training estimates ϕ of the feature representation $f_\phi : \mathcal{S} \to \mathcal{Z} \subset \mathbb{R}^d$, by minimizing the contrastive loss [5, 7, 19]:

$$\ell_{i,j} = -log \frac{exp(sim(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} exp(sim(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \tag{1}$$
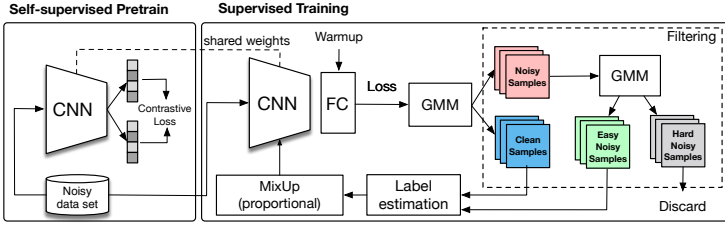
Figure 1: Our proposed PropMix has a self-supervised pre-training stage [6, 7, 12, 19], followed by a supervised training stage, where we first warm-up the classifier with a classification loss, using the pre-trained weights. Then, using the classification loss, we train a GMM to separate the samples into clean and noisy. Next, using classification confidence for the noisy set, we train a second GMM to separate the easy and hard noisy samples. The clean and easy noisy samples are proportionally combined in the MixUp for training.

with N denoting mini-batch size, $\mathbf{z} = f_\phi(\mathbf{x})$ represents the features extracted from input $\mathbf{x}$, with $\mathbf{z_i}$ and $\mathbf{z_j}$ being the feature vectors from two views of the same image (these views are obtained via different data augmentations of the same image), $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ being an indicator function, τ denoting the temperature parameter, and $sim(.)$ representing the cosine similarity. Following [12], we then learn a clustering classifier $p_\theta(.|f_\phi(\mathbf{x}))$. More precisely, we form an initial set of K nearest neighbours (KNN) in $\mathcal{Z}$ for each training sample, producing the set $\mathcal{N}_{\mathbf{x}_i} = \{\mathbf{x}_j\}_{j=1}^K$ (for $\mathbf{x}_j \in \mathcal{D}$) for each sample $\mathbf{x}_i \in \mathcal{D}$. We train $p_\theta(.|f_\phi(\mathbf{x}))$ with [12]:

$$\ell_{CLU} = \ell_{\mathcal{N}} + \lambda_e \ell_e, \tag{2}$$

with $\ell_{\mathcal{N}} = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} q(z_{ji}) \log \left(p_\theta(.|f_\phi(\mathbf{x}_i))^\top p_\theta(.|f_\phi(\mathbf{x}_j))\right)$, $q(z_{ji}) = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same classification result (i.e., $\arg\max_{c \in \mathcal{Y}} p_\theta(c|f_\phi(\mathbf{x}_i)) = \arg\max_{c \in \mathcal{Y}} p_\theta(c|f_\phi(\mathbf{x}_j)))$, and $\ell_e = \frac{1}{|\mathcal{D}|} \sum_{c \in \mathcal{Y}} \sum_{\mathbf{x} \in \mathcal{D}} p_\theta(c|f_\phi(\mathbf{x})) \log p_\theta(c|f_\phi(\mathbf{x}))$ that maximises the entropy of the average classification and is weighted by λ_e. The weights from the feature map $f_\phi(\mathbf{x})$ are also updated in the clustering process using backpropagation.

After the pre-training, we warm-up the classifier by training it for a few epochs on the (noisy) training data set with the cross-entropy (CE) loss. The clean and noisy sets, $\mathcal{X}, \mathcal{U} \subseteq \mathcal{D}$, are formed with [3, 23, 27, 30]:

$$\mathcal{X} = \{(\mathbf{x}_i, \mathbf{y}_i, w_i) : w_i = p(\text{clean}|\ell_i, \gamma) \geq \tau\}, \quad \mathcal{U} = \{(\mathbf{x}_i, \mathbf{y}_i, w_i) : w_i = p(\text{clean}|\ell_i, \gamma) < \tau\}, \tag{3}$$

with τ denoting a classification threshold, $\ell_i = -\mathbf{y}_i^\top \log p_\theta(.|f_\phi(\mathbf{x}_i))$, and $p(\text{clean}|\ell_i, \gamma)$ being a function that estimates the probability that $(\mathbf{x}_i, \mathbf{y}_i)$ is a clean label sample. The function $p(\text{clean}|\ell_i, \gamma)$ in Eq.3 is a bi-modal Gaussian mixture model (GMM) [30], where γ denotes the GMM parameters and the larger mean component is the noisy component whereas the smaller mean component is the clean component. Next, we obtain the sets of easy and hard noisy samples $\mathcal{U}_E, \mathcal{U}_H \subseteq \mathcal{U}$, as follows:

$$\mathcal{U}_E = \left\{(\mathbf{x}_i, \mathbf{y}_i) : p\left(\text{hard}|p_\theta(c_i^*|f_\phi(\mathbf{x})_i), \gamma\right) < \tau'\right\},$$
$$\mathcal{U}_H = \left\{(\mathbf{x}_i, \mathbf{y}_i) : p\left(\text{hard}|p_\theta(c_i^*|f_\phi(\mathbf{x}_i)), \gamma\right) \geq \tau'\right\}, \tag{4}$$

with $c_i^* = \arg\max_{c_i \in \mathcal{Y}} p_\theta(c_i|f_\phi(\mathbf{x}_i))$, $p\left(\text{hard}|p_\theta(c_i^*|f_\phi(\mathbf{x}_i)), \gamma\right)$ being a function that estimates the probability that $(\mathbf{x}_i, \mathbf{y}_i)$ is a hard noisy label sample, and τ' denoting the hard

noisy sample threshold. The function $p\left(\text{hard}|p_\theta(c_i^*|f_\phi(\mathbf{x}_i)),\gamma\right)$ in Eq. 4 is a GMM, where γ denotes the GMM parameters and the smaller mean component is the hard noise component whereas the larger mean component is the easy noise component. We assume hard noisy samples have wrong label and wrong prediction, so we remove them from training.

Next, we perform M data augmentations based on geometrical and visual transformations to increase the number of samples in $\mathcal{X}$ and $\mathcal{U}_E$, generating the augmented sets $\mathcal{X}' = \{(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)\}_{i=1}^{|\mathcal{X}'|}$ and $\mathcal{U}'_E = \{(\hat{\mathbf{u}}_i, \hat{\mathbf{q}}_i)\}_{i=1}^{|\mathcal{U}'_E|}$, with $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{u}}_i$ being the augmented samples from $\mathcal{X}$ and $\mathcal{U}_E$, respectively, and $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{q}}_i$ the label estimations defined as follows:

$$\hat{\mathbf{y}}_i = \text{TempShrp}(w_i\mathbf{y}_i + (1 - w_i)\mathbf{p}_b; T), \quad \hat{\mathbf{q}}_b = \text{TempShrp}(\mathbf{q}_b; T) \tag{5}$$

with $\mathbf{p}_b = \frac{1}{M}\sum_{m=1}^{M} p_\theta(.|f_\phi(\hat{\mathbf{x}}_{i,m}))$, $\mathbf{q}_b = \frac{1}{M}\sum_{m=1}^{M} p_\theta(.|f_\phi(\hat{\mathbf{u}}_i))$, and TempShrp(.) being a temperature sharpening [30].

The linear combination between the clean and easy noisy samples rely on MixUp [57] data augmentation, which is different from the SOTA SSL approaches [30] that use Mix-Match [4] which has a training set size restricted by the clean set size. In particular, MixMatch-based methods select $B_{\mathcal{U}'} = |\mathcal{X}'|$ samples from the noisy set and $B_{\mathcal{X}'} = |\mathcal{X}'|$ samples from the clean set, so the MixUp proportion of samples from the noisy set is $\zeta = \frac{B_{\mathcal{U}'}}{B_{\mathcal{X}'}+B_{\mathcal{U}'}} = 0.5$, and the size of the MixMatch training set is $|\mathcal{X}'|$. In our approach, we re-label the easy noisy samples in $\mathcal{U}'_E$ and place them together with the clean samples in $\mathcal{X}'$ for the MixUp data augmentation with $\mathcal{D}' = \mathcal{X}' \cup \mathcal{U}'_E$. This operation enables us to obtain the proportionally mixed set $\hat{\mathcal{D}} = MixUp(\mathcal{D}', \text{shuffled}(\mathcal{D}'))$ where the proportion of samples from the noisy set, denoted by $\zeta = \frac{|\mathcal{U}'_E|}{|\mathcal{X}'|+|\mathcal{U}'_E|}$, will be larger than 0.5 for high noise rate problems.

To optimise the classification term we rely on the regularised CE loss [30]:

$$\mathcal{L} = \ell_{CE} + \lambda_r\ell_r, \tag{6}$$

with $\ell_{CE} = -\frac{1}{|\hat{\mathcal{D}}|}\sum_{(\tilde{\mathbf{x}},\tilde{\mathbf{y}})\in\hat{\mathcal{D}}} \tilde{\mathbf{y}}^\top \log p_\theta(.|f_\phi(\tilde{\mathbf{x}}))$ and $\ell_r = KL\left[\pi_{|\mathcal{Y}|} \middle|\middle| \frac{1}{|\hat{\mathcal{D}}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}'} p_\theta(.|f_\phi(\mathbf{x}))\right]$,

where λ_r weights the regularisation loss, and $\pi_{|\mathcal{Y}|}$ denotes a vector of $|\mathcal{Y}|$ dimensions with values equal to $1/|\mathcal{Y}|$. Different from SOTA semi-supervised methods [30, 40], we do not have an additional loss for the noisy set because we assume that most of the samples in the noisy set have correct predictions (this is empirically shown in Sec. 4.3). A positive outcome from not using such additional loss for the noisy set is that we no longer need to manually set hyper-parameter values that depend on the noise rate of the problem, as is done by DivideMix [30]. The pseudo-code for the training of PropMix is shown in Algorithm 1 in the supplementary material.

4 Experiments

4.1 Data sets

We conduct our experiments on the data sets CIFAR-10, CIFAR-100 [25], Controlled Noisy Web Labels (CNWL) [23], Clothing1M [51] and WebVision [31]. CIFAR-10 and CIFAR-100 have 50k training and 10k testing images of size 32×32 pixels, where CIFAR-10 has 10 classes and CIFAR-100 has 100 classes and all training and testing sets have equal number of images per classes. As CIFAR-10 and CIFAR-100 data sets originally do not contain label

noise, we follow the literature [30] and add the following synthetic noise types (see Sec. 3.1): symmetric (with noise rate $\eta \in \{0.2, 0.5, 0.8, 0.9\}$, as defined in Sec. 3.1), asymmetric (using the mapping in [30, 43], with $\eta_{jc} = 0.4$), and semantic [27] (with noisy labels based on a trained VGG [46], DenseNet (DN), and ResNet (RN)).

The CNWL dataset [23] is a benchmark to study real-world web label noise in a controlled setting. Both images and labels are crawled from the web and the noisy labels are determined by matching images. The controlled setting provide different magnitudes of label corruption in real applications, varying from 0% to 80%. We study the red Mini-ImageNet that consists of 50k training images and 5k test images, with 100 classes. The original 84×84-pixel images are resized to 32×32 pixels. The noise rates are 20%, 60% and 80%, as used in [52].

Clothing1M consists of 1 million training images acquired from online shopping websites and it has 14 classes. As the images from the data set vary in size, we resized the images to 256×256 for training, as used in [17, 30]. The data set provide additional clean sets for training, validation, and testing of 50k, 14k and 10k images, respectively. For our experiments we do not use any of the clean training or validation sets, but we use the clean test set for evaluation.

WebVision contains 2.4 million images collected from the internet, with the same 1000 classes from ILSVRC12 [9] and images resized to 256×256 pixels. It provides a clean test set of 50k images, with 50 images per class. We compare our model using the first 50 classes of the Google image subset, as used in [5, 30].

4.2 Implementation

For CIFAR-10 and CIFAR-100 we used a 18-layer PreaAct-ResNet-18 (PRN18) [18] as our backbone model [30]. The models are trained with stochastic gradient descent (SGD) with momentum of 0.9, weight decay of 0.0005 and batch size of 64. For the self-supervised pre-training learning task, we adopt SimCLR [6] with a batch size of 1024, SGD optimiser with a learning rate of 0.4, decay rate of 0.1, momentum of 0.9 and weight decay of 0.0001, and run it for 800 epochs. This pre-trained model produces feature representations of 128 dimensions. Using these representations we mine $K = 20$ nearest neighbours (as in [12]) for each sample to form the sets $\{\mathcal{N}_{\mathbf{x}_i}\}_{i=1}^{|\mathcal{D}|}$, defined in Sec. 3.2. In the supervised training stage, the model is trained with SGD with momentum of 0.9, weight decay of 0.0005 and batch size of 64. The learning rate is 0.02 which is reduced to 0.002 in the middle of the training. The WarmUp and total number of epochs is defined according to each data set, as defined in [30]. For CIFAR-10 and CIFAR-100, PRN18 is trained with a WarmUp stage of 30 epochs for CIFAR-10 and 10 epochs for CIFAR-100, and 300 epochs of final training. In our training, we also use a co-teaching approach, as in [16, 30, 33]. We estimate noise rate with $|\mathcal{U}|/|\mathcal{D}|$, defined in Eq. 3 – if this ratio is larger than 50%, we use strong data augmentation with cutout [8], otherwise, we use standard augmentation (i.e., crop and flip).

For the semantic noise from [27], we use DenseNet-100 [20] as backbone, following [27]. The pre-training stage uses the same parameters as in CIFAR, except that we use a batch size of 512. In the supervised stage, we use the same protocol as [27], which uses SGD with momentum 0.9, weight decay 10-4 and learning rate of 0.1 that is divided by 10 after epochs 40 and 80 for CIFAR-10 (which runs for 120 epochs in total), and after epochs 80, 120 and 160 for CIFAR-100 (runs for 200 epochs in total). The WarmUp epochs is the same as CIFAR-10/CIFAR-100.

For red Mini-Imagenet we use PRN18 as backbone, following [52]. For the self-supervised

pre-training task, we adopt SimCLR [6] with batch size 128. All other parameters for the self-supervised pre-training are the same as described for CIFAR. For the supervised stage, we adopt the implementation of [52], where we train for 300 epochs, relying on SGD with learning rate of 0.02 (decreased by a factor of ten at epochs 200 and 250), momentum of 0.9 and weight decay of 5e-4.

For Clothing1M, we use ResNet-50 as backbone, following [30]. In this protocol, a ResNet-50 with ImageNet [9] pre-trained weights is used and we decided to not use the self-supervised stage for this experiment because the model is already pre-trained. The ResNet-50 is trained for 80 epochs, including a WarmUp stage of 1 epoch, with a batch size of 32, SGD with a learning rate of 0.002 (divided by 10 at epoch 40), momentum of 0.9 and weight decay of 0.0001.

For Webvision, we use InceptionResNet-V2 as backbone, following [30]. For self-supervised pre-training, we adopt MoCo-v2 (4-GPU training) [7], trained with 100 epochs, with a batch size of 128, SGD with a learning rate of 0.015 (divided by 10 at epoch 50), momentum of 0.9 and weight decay of 0.0001, and run it for 100 epochs with a WarmUp stage of 1 epoch. The feature representations learned from this process have 128 dimensions. All the other parameters were the same as described above for CIFAR.

4.3 PropMix Analysis

Fig. 2 shows the PropMix hard noisy sample filtering process for CIFAR-100 with 80% symmetric noise, at epoch 150 (i.e., half of the training epochs). The left histogram shows the clean and noisy classification from Eq. 3, while the right one shows the easy and hard noisy sample classification from Eq. (4). As can be seen, the use of confidence in Eq. (4), instead of the loss from Eq. 3, is an effective way to remove noisy samples with wrong prediction. We also evaluated the quality of the filtering stage using PropMix for CIFAR-100 under different symmetric noise rates. Fig. 3(a) and (b) show the precision and recall of the hard noisy sample filtering as a function of training epochs. These graphs show that the larger the noise, the higher the precision. For example, for 90% noise, the hard noise classification reaches 80% at the end of training. Fig. 3(c) shows the size of the noisy sample set during training. Fig. 3(d) compares the easy noise set re-labelling accuracy by PropMix with a baseline that does not filter out hard noisy samples. We can see that PropMix is substantially more accurate, which enables a more successful training with the easy noisy samples. We also evaluated the impact of the parameters τ and τ' in the training. Fig. 4 shows the PropMix accuracy for CIFAR-10 (a-b) and CIFAR-100 (c-d) under different symmetric noise rates, varying the parameters τ, which is related to the clean/noisy filtering from Eq. 3, and τ', which related to hard noise filtering, from Eq. 4. Fig. 4(a,c) vary τ while fixing $\tau' = 0.5$, and Fig. 4(b,d) vary τ' while fixing $\tau = 0.5$. We can see that τ' does not have a strong impact on the accuracy, which motivated us to use $\tau' = 0.5$ independently of the noise rate and data set. The parameter τ is shown to decrease accuracy at higher values. In PropMix we use $\tau = 0.5$ as in other works [30].

4.4 Comparison with State-of-the-Art

For CIFAR-10 and CIFAR-100, we evaluate our model using symmetric label noise ranging from 20% to 90%, and 40% asymmetric noise. We report both the best test accuracy across all epochs and the averaged test accuracy over the last 10 epochs of training, similar to [30]. Tab. 1 shows that for CIFAR-10 and CIFAR-100 data sets, our method obtains the best results for almost all evaluated noisy rates.
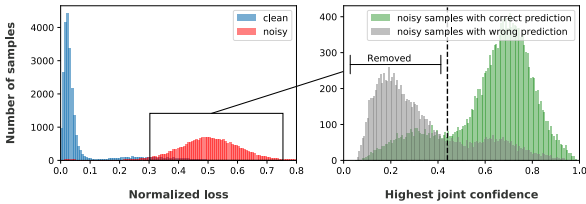
Figure 2: PropMix classification of easy and hard noisy samples. From the (normalized) loss of all samples, PropMix uses GMM to split clean and noisy samples, and then, using the largest classification confidence, it uses another GMM to split the hard and easy noisy samples. This example was run on CIFAR-100, 80% noise rate at epoch 150.
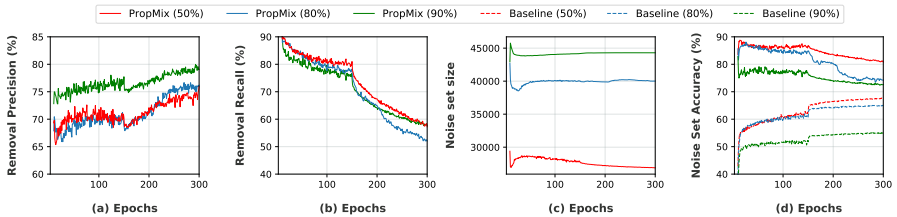

Figure 3: Evaluation of hard noisy sample filtering, for CIFAR-100, with 50% to 90% symmetric noise. Graphs in (a,b) show the precision and recall of the classification of hard noisy samples (i.e., noisy samples that are incorrectly labelled by the model), Graph (c) shows the estimated noise set size, and (d) shows the classification accuracy of the re-labelled easy noisy samples, compared to the baseline that does not filter out hard noisy samples.
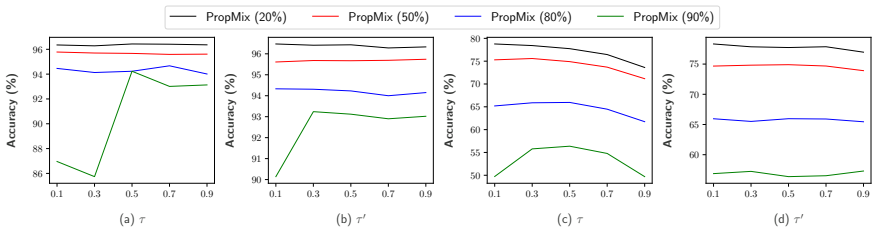

Figure 4: Evaluation of the parameters τ (clean sample threshold) and τ' (hard noisy sample threshold), for CIFAR-10 and CIFAR-100, with 20% to 90% symmetric noise. Graphs in (a,b) show the best accuracy when training using PropMix with CIFAR-10, and graphs (c,d) show results for CIFAR-100, at different values of τ and τ'. Graph (a,c) vary the parameter τ, while fixes τ' at 0.5. Graph (b,d) vary the parameter τ' while fixes τ at 0.5.

Tab. 2 shows the results for semantic noise [27], which is harder and more realistic than the synthetic noise, where PropMix shows significantly more accurate results than any of the methods in [27]. In Tab. 4, we show that PropMix improves the SOTA by a large margin on the Red Mini-ImageNet semantic noise experiment from [23].

We also evaluate PropMix on the noisy large-scale datasets WebVision [51] and Clothing1M [51]. Tab. 5 shows competitive results compared to SOTA. As Clothing1M experimental protocol uses ImageNet pre-trained weights, the training could not benefit from Prop-

dataset		CIFAR-10					CIFAR-100			
Noise type		sym.				asym.	sym.			
Method/ noise ratio		20%	50%	80%	90%	40%	20%	50%	80%	90%
Cross-Entropy [30]	Best	86.8	79.4	62.9	42.7	85.0	62.0	46.7	19.9	10.1
	Last	82.7	57.9	26.1	16.8	72.3	61.8	37.3	8.8	3.5
Coteaching+ [□]	Best	89.5	85.7	67.4	47.9	-	65.6	51.8	27.9	13.7
	Last	88.2	84.1	45.5	30.1	-	64.1	45.3	15.5	8.8
MixUp [□]	Best	95.6	87.1	71.6	52.2	-	67.8	57.3	30.8	14.6
	Last	92.3	77.3	46.7	43.9	-	66.0	46.6	17.6	8.1
PENCIL [□]	Best	92.4	89.1	77.5	58.9	88.5	69.4	57.5	31.1	15.3
	Last	92.0	88.7	76.1	58.2	88.1	68.1	56.4	20.7	8.8
Meta-Learning [□]	Best	92.9	89.3	77.4	58.7	89.2	68.5	59.2	42.4	19.5
	Last	92.0	88.8	76.1	58.3	88.6	67.7	58.0	40.1	14.3
M-correction [□]	Best	94.0	92.0	86.8	69.1	87.4	73.9	66.1	48.2	24.3
	Last	93.8	91.9	86.6	68.7	86.3	73.4	65.4	47.6	20.5
MentorMix [□]	Best	95.6	-	81.0	-	-	78.6	-	41.2	-
	Last	-	-	-	-	-	-	-	-	-
MOIT+ [□]	Best	94.1	-	75.8	-	93.3	75.9	-	51.4	-
	Last	-	-	-	-	-	-	-	-	-
DivideMix [□]	Best	96.1	94.6	93.2	76.0	93.4	77.3	74.6	60.2	31.5
	Last	95.7	94.4	92.9	75.4	92.1	76.9	74.2	59.6	31.0
ELR+ [□]	Best	95.8	94.8	93.3	78.7	93.0	77.6	73.6	60.8	33.4
	Last	-	-	-	-	-	-	-	-	-
PropMix (Ours)	Best	96.44	95.77	93.94	93.48	94.89	77.41	74.56	67.34	58.57
	Last	96.09	95.53	93.77	93.20	94.64	76.99	73.71	66.75	58.32

Table 1: Test accuracy (%) for all competing methods on CIFAR-10 and CIFAR-100 under symmetric and asymmetric noises. Results from related approaches are as presented in [30]. Top methods within 1% are in **bold**.

Data set	CIFAR-10			CIFAR-100		
Method/ noise ratio	DenseNet (32%)	ResNet (38%)	VGG (34%)	DenseNet (34%)	ResNet (37%)	VGG (37%)
D2L + RoG [□]	68.57	60.25	59.94	31.67	39.92	45.42
CE + RoG [□]	68.33	64.15	70.04	61.14	53.09	53.64
Bootstrap + RoG [□]	68.38	64.03	70.11	54.71	53.30	53.76
Forward + RoG [□]	68.20	64.24	70.09	53.91	53.36	53.63
Backward + RoG [□]	68.66	63.45	70.18	54.01	53.03	53.50
PropMix (Ours)	**84.25**	**82.51**	**85.74**	**60.98**	**58.44**	**60.01**

Table 2: Test accuracy (%) for the semantic noise benchmark [26], where baseline results are from [26]. The results with (*) were produced by locally running the published code provided by the authors. Top methods (±1%) are in **bold**.

Mix strategy. Tab. 3 shows the Top-1/-5 test accuracy using the WebVision and ILSVRC12 test sets. Results show that PropMix is slightly better than the SOTA for WebVision test set. This suggests that our approach is also effective in large-scale, low noise rate problems.

4.5 Ablation Study

We show the ablation study of PropMix with CIFAR-10 and CIFAR-100 under symmetric and asymmetric noises at several rates. In Tab. 6, we show *Self-superv. pre-train*, which has the result from self-supervised training (with SCAN [12]), where the result is the same across different noise rates because it never uses the noisy labels for training. *SSL (DivideMix)* [30]) is current SOTA in noisy label learning, but it has worse accuracy in high noise rate problems (> 80% noise) than SCAN. *Self-superv. pre-train+SSL (DivideMix)* pre-trains DivideMix with SCAN to improve accuracy in high-noise rate problems, without affecting low-noise rate results. Just removing the MSE loss in *Self-superv. pre-train+SSL (DivideMix) w/o MSE* does not help much because the noisy samples from DivideMix is not accurately re-labelled. Adding our hard noisy sample filtering in *Self-superv. pre-train+SSL (DivideMix) + filtering with MSE* does not improve accuracy because DivideMix still uses MSE loss to train the easy noisy samples, which is unnecessary given that these samples are accurately

Method/ noise ratio	20%	40%	60%	80%
Cross-entropy [□]	47.36	42.70	37.30	29.76
MixUp [□]	49.10	46.40	40.58	33.58
DivideMix [□]	50.96	46.72	43.14	34.50
MentorMix [□]	51.02	47.14	43.80	33.46
FaMUS [□]	51.42	48.06	45.10	35.50
PropMix (Ours)	**61.24**	**56.22**	**52.84**	**43.42**

Table 4: Test accuracy (%) for Red Mini-ImageNet [□]. Results from baseline methods are as presented in [□]. Top methods (±1%) are in **bold**.

Method	Test Accuracy
Cross-Entropy [□]	69.21
M-correction [□]	71.00
PENCIL[□]	73.49
DeepSelf [□]	**74.45**
CleanNet [□]	**74.69**
DivideMix [□]	**74.76**
PropMix (ours)	**74.30**

Table 5: Test accuracy (%) for Clothing1M [□] by methods trained with 80 epochs. Baselines come from [□]. Top methods within 1% are in **bold**.

re-labelled and can be trained with the CE loss. *PropMix* can achieve better results for high noise rates, using a simpler loss function (uses only CE loss) with less hyper-parameters, and removing hard noisy samples.

dataset						CIFAR-10				CIFAR-100			
Noise type			sym.			asym.		sym.					
Method/ noise ratio	20%	50%	80%	90%	40%	20%	50%	80%	90%				
Self-superv. pre-train [□]	81.6	81.6	81.6	81.6	81.6	44.0	44.0	44.0	44.0				
SSL (DivideMix) [□]	**96.1**	94.6	**93.2**	76.0	93.4	**77.3**	**74.6**	60.2	31.5				
Self-superv. pre-train + SSL (DivideMix)*	**96.2**	94.8	**93.9**	92.2	93.3	**77.4**	**74.7**	66.7	56.2				
Self-superv. pre-train + SSL (DivideMix)* w/o MSE	**96.2**	**95.2**	91.4	85.1	93.3	**77.2**	73.9	62.6	52.1				
Self-superv. pre-train + SSL (DivideMix)* + filtering with MSE	**96.3**	**95.3**	**93.8**	91.6	93.0	**77.3**	**75.3**	66.7	55.7				
PropMix (Ours)	**96.4**	**95.8**	**93.9**	**93.5**	**94.9**	**77.4**	**74.6**	**67.3**	**58.6**				

Table 6: In this ablation study we show the test accuracy (%) on CIFAR-10 and CIFAR-100 under symmetric and asymmetric noise problems. First, we show the results of self-supervised pre-training with SCAN [□]. Then we show the current SOTA SSL learning for noisy label DivideMix [□]. Next, we show the results of DivideMix pre-trained with SCAN. Then, we remove the MSE loss to train DivideMix re-labelled noisy samples, and add the hard noisy filtering stage (but keeping the MSE loss) to the pre-trained DivideMix. The last row shows our PropMix . The top results within 1% are highlighted.

5 Conclusion and Future Work

We presented PropMix, a noisy label training algorithm to filter out hard noisy samples, remove samples with incorrect label and incorrect prediction from noisy set with the goal to keep the easy noisy samples that have a higher chance to be correctly re-labelled. PropMix reduces noisy-dependent parameters, while promoting the use of the entire filtered noisy set in a fully supervised training, with proportional MixUp data augmentation on the clean set. Our results on CIFAR-10/100, Red Mini-ImageNet, and WebVision outperform the SOTA methods, also demonstrating robustness to over-fitting in several noise rates with substantial improvement in high noise problems.

Method	Top-1	Top-5
F-correction [□]	61.12	82.68
Decoupling [□]	62.54	84.74
D2L [□]	62.68	84.00
MentorNet [□]	63.00	81.40
Co-teaching [□]	63.58	85.20
Iterative-CV [□]	65.24	85.34
MentorMix [□]	76.00	90.20
DivideMix [□]	77.32	**91.64**
ELR+ [□]	77.78	**91.68**
PropMix (Ours)	**78.84**	90.56

Table 3: Test accuracy (%) for WebVision [□] by methods trained with 100 epochs. Baselines come from [□]. Top methods within 1% in **bold**.

6 Acknowledgment

References

[1] Ravi Aggarwal, Viknesh Sounderajah, Guy Martin, Daniel SW Ting, Alan Karthike-salingam, Dominic King, Hutan Ashrafian, and Ara Darzi. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ digital medicine*, 4(1):1–23, 2021.

[2] Görkem Algan and Ilkay Ulusoy. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7142–7148. IEEE, 2021.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Un-supervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321, 2019.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. *arXiv e-prints*, art. arXiv:1905.02249, May 2019.

[5] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. Understand-ing and utilizing deep neural networks trained with noisy labels. *arXiv preprint arXiv:1905.05040*, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv e-prints*, art. arXiv:2002.05709, February 2020.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *arXiv e-prints*, art. arXiv:2003.04297, March 2020.

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

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[11] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[12] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels, 2020.

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

[14] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3): 362–386, 2020.

[15] Bo Han, Gang Niu, Jiangchao Yao, Xingrui Yu, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Pumpout: A meta approach for robustly training deep neural networks with noisy labels. 2018.

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

[17] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5138–5147, 2019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. *arXiv e-prints*, art. arXiv:1911.05722, November 2019.

[20] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[21] Lee Jaehwan, Yoo Donggeun, and Kim Hyo-Eun. Photometric transformer networks and label adjustment for breast density prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

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

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

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

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

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

[27] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. *arXiv preprint arXiv:1901.11300*, 2019.

[28] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[29] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.

[30] Junnan Li, Richard Socher, and Steven C. H. Hoi. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. *arXiv e-prints*, art. arXiv:2002.07394, February 2020.

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

[32] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[33] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

[34] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

[35] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364, 2018.

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

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

[38] Qiguang Miao, Ying Cao, Ge Xia, Maoguo Gong, Jiachen Liu, and Jianfeng Song. Rboost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *IEEE transactions on neural networks and learning systems*, 27(11):2216–2228, 2015.

[39] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[40] Kento Nishi, Yi Ding, Alex Rich, and Tobias Höllerer. Augmentation Strategies for Learning with Noisy Labels. *arXiv e-prints*, art. arXiv:2103.02130, March 2021.

[41] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Towards robust learning with different label noise distributions. *arXiv preprint arXiv:1912.08741*, 2019.

[42] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. *arXiv preprint arXiv:2012.04462*, 2020.

[43] 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.

[44] Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *arXiv preprint arXiv:2001.10528*, 2020.

[45] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343, 2018.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

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

[48] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*, 2019.

[49] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. *arXiv e-prints*, pages arXiv–2005, 2020.

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

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

[52] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. Faster meta update strategy for noise-robust deep learning. In *CVPR*, 2021.

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

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

[55] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. Iterative cross learning on noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 757–765. IEEE, 2018.

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

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