Combating Noisy Labels with Sample Selection by Mining High-Discrepancy Examples

Xiaobo Xia[1], Bo Han[2], Yibing Zhan[3], Jun Yu[4]*, Mingming Gong[5], Chen Gong[6], Tongliang Liu[1]
[1]The University of Sydney [2]Hong Kong Baptist University
[3]JD Explore Academy [4]University of Science and Technology of China
[5]The University of Melbourne [6]Nanjing University of Science and Technology

Abstract

The sample selection approach is popular in learning with noisy labels. The state-of-the-art methods train two deep networks simultaneously for sample selection, which aims to employ their different learning abilities. To prevent two networks from converging to a consensus, their divergence should be maintained. Prior work presents that the divergence can be kept by locating the disagreement data on which the prediction labels of the two networks are different. However, this procedure is sample-inefficient for generalization, which means that only a few clean examples can be utilized in training. In this paper, to address the issue, we propose a simple yet effective method called CoDis. In particular, we select possibly clean data that simultaneously have high-discrepancy prediction probabilities between two networks. As selected data have high discrepancies in probabilities, the divergence of two networks can be maintained by training on such data. In addition, the condition of high discrepancies is milder than disagreement, which allows more data to be considered for training, and makes our method more sample-efficient. Moreover, we show that the proposed method enables to mine hard clean examples to help generalization. Empirical results show that CoDis is superior to multiple baselines in the robustness of trained models.

1. Introduction

Learning with noisy labels can be dated back to more than three decades ago [1], and still is one of the hottest problems in weakly supervised learning. The reason is that, in our daily life, noisy labels are *unavoidable* such as crowd sourcing [67, 32] and web queries [39, 58]. However, the combination of noisy labels and deep networks is *rather pessimistic*, since deep networks have strong learning capacities and can fully memorize given noisy labels, leading to poor generalization [88, 60, 9, 24, 35, 89, 11, 65, 78, 71, 33]. General-purpose regularization such as *dropout* and *weight decay* cannot address this issue well [80].

Fortunately, even though deep networks can fit anything given for training eventually, they *learn patterns first* [2]: this suggests that deep networks can *gradually memorize the data*, moving from clean data to mislabeled data. The *sample selection* approach therefore was proposed to handle noisy labels [21, 16, 47, 76], which is also *our focus* in this paper. The works on sample selection try to select possibly clean data out of noisy ones, and then use them to update the deep networks. Intuitively, if the training data can become less noisy, better generalization can be achieved.

As the idea of self-teaching sample selection is argued to have the inferiority of *accumulated errors* caused by the sample-selection bias [16], some advanced algorithms were proposed, which maintain two deep networks, working in a cooperative manner [76, 40, 31, 63]. The key component making the cooperative sample selection works better than the self-teaching one, is that two different networks have *different learning abilities* and can filter different types of errors introduced by noisy labels. That is to say, when each network selects clean data for its peer network for updates, the error flows coming from the biased selection, can be reduced by peer networks mutually [16].

To keep the different learning abilities of two networks, prior work [79] utilizes a simple strategy called "Update by Disagreement". In more detail, two networks feed forward and predict all data first, and only keep *prediction disagreement data*, *i.e.*, the data with *different prediction labels* from two networks. Then, each network selects its clean data from such disagreement data to the peer network. At first glance, this method can use less noisy data and meanwhile maintain the different learning abilities of two networks. However, its sample selection procedure is *sample-inefficient* for network weight updates. It is because the condition of disagreement is somewhat strong in sample selection, which makes that the sample size of prediction disagreement data is often small, especially when the label

*Corresponding author (harryjun@ustc.edu.cn).

noise rate is large [63]. When we tend to select clean data out of them, the sample size of available data for network weight updates will be further reduced. The issue causes that *a few clean examples* can be utilized in training, which impairs generalization severely [63].

In this paper, to handle the above problem, a robust learning paradigm called CoDis is proposed. Specifically, we inherit the property that deep networks learn patterns first for sample selection, as did in [21, 16, 40, 69]. Meanwhile, the training examples with *high discrepancies* between two networks are encouraged to be involved in training. The network divergence can be maintained by training on such examples. In this work, for a training example, we measure the discrepancy by using the *distance of prediction probabilities* between two networks, which is *continuously* valued. As the measurement of whether an example can be clean (*e.g.*, the cross-entropy loss), is also continuous, it is convenient to make a great *trade-off* that considers the examples which are likely to be clean (with small cross-entropy losses) and simultaneously can maintain the two networks diverged (with high discrepancies). Additionally, the condition of high discrepancies in sample selection is *milder* than the condition of disagreement. In other words, the prediction disagreement data must have high-discrepancy prediction probabilities, but the data with high discrepancies can have different prediction probabilities but the same prediction labels from two networks. The milder condition allows us to consider more data for training. Therefore, compared with the prior mentioned procedure of sample selection [79], our procedure is more *sample-efficient*, which improves generalization.

Furthermore, the examples with high discrepancies in training are probable to be hard examples [12], which play an important role in shaping the decision boundary. Shared with a similar philosophy, the proposed method emphasizes high-discrepancy examples and enables to mine hard clean examples that are critical for generalization. Benefiting from maintaining two networks simultaneously, the discrepancy measurement in our work can be conducted on-the-fly, and without the need to carefully determine that useful information on how many training iterations is introduced.

The main contributions of this paper are summarized as three aspects: (1). We provide a simple but effective method to tackle noisy labels, which is more sample-efficient to help generalization. (2). The proposed method can maintain the network divergence meanwhile enable to mine hard clean examples that are significant for generalization. We also provide theoretical insights into the divergence applied in sample selection. (3). We conduct a series of experiments on both simulated noisy datasets including class-balanced and imbalanced noisy datasets, and real-world noisy datasets. Extensive results demonstrate that the robustness of deep models trained by CoDis can well combat noisy labels. Particularly, on class-imbalanced noisy datasets, our method can outperform comparison methods by more than 5% of test accuracy.

2. Background

Notations. In the sequel, we use $\|\cdot\|_p$ as the ℓ_p norm of vectors or matrices and $\mathrm{KL}(\cdot\|\cdot)$ as the Kullback-Leibler (KL) divergence [44] between two probability distributions. We use $|\cdot|$ to denote the number of elements in a set. For a function g, we use ∇g to denote its gradient. For a vector $\mathbf{z}$, $\mathbf{z}^j$ denotes the j-th component of $\mathbf{z}$. We use $\mathbf{e}_i$ to denote the *one-hot* encoding, with $\mathbf{e}_i = (0, \ldots, 0, 1, \ldots, 0)$ (the i-th coordinate being 1). Let [n] = $\{1, 2, \ldots, n\}$.

Problem statement. We consider a c-class ($c \geq 2$) classification problem. Let $\mathcal{X}$ and $\mathcal{Y}$ be the instance/feature space and label space respectively, with $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^c$, where d is the dimensionality of the feature space. Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ be an i.i.d. training sample lying in the joint distribution $\mathcal{X} \times \mathcal{Y}$, where n denotes the sample size. In supervised learning, the aim is to learn a precise classifier that can assign labels for given instances with the sample $\mathcal{D}$. However, before being observed, true labels of examples in $\mathcal{D}$ are independently flipped and what we can obtain is a noisy training sample $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^n$, where $\tilde{\mathbf{y}}$ denotes the one-hot noisy label. The aim is changed to learn a robust classifier that can assign clean labels to test data by only exploiting a noisy training sample $\tilde{\mathcal{D}}$.

2.1. Handling Noisy Labels with Sample Selection

We formally introduce the sample selection approach applied in learning with noisy labels. Specifically, with the assumption that clean labels are the majority in a noisy class [41], we can select possibly clean examples from noisy examples based on some criteria. For example, the *small-loss* examples can be approximately seen as clean examples [16, 79, 19, 40, 63]. In addition, the examples that have large classification margins [52], minimize the determinant value of the corresponding sample covariance matrix [29], or minimize the average gradient dissimilarity to all the other examples [43], can be seen as clean examples and then be used for network parameter updates.

In this paper, we target the procedure of using the *small-loss* criterion for sample selection, which is most commonly used. It is straightforward for using a *single network* to select clean examples for robust training [21]. However, this paradigm inherited the inferiority of *accumulated errors* caused by the sample-selection bias. More specifically, at the stage when the network begins to fit training examples, the losses are not very informative. Therefore, we may select mislabeled examples mistakenly for updates. This issue causes the network to memorize incorrect information which greatly affects the selection of examples in subse-

quent iterations. Although Co-teaching [16] trains two networks and makes them select clean examples for its peer network, it still cannot address the issue of accumulated errors well, because two networks will *converge to a consensus* with the increase of training epochs.

To address the issue of accumulated errors, some works follow the idea of "Update by Disagreement". The core components of this idea are to employ two networks and keep divergence among them. For example, Decoupling [42] conducts updates only on selected data with prediction disagreement between two networks. Co-teaching+ [79] concerns that the disagreement area of two networks is noisy and further selects small-loss examples within the area for updates. However, in the manner of Co-teaching+, a few clean examples can be used to help generalization, due to the strict disagreement measurement.

Recently, JoCor [63] starts with a new perspective named "Update by Agreement", which is motivated by Co-training [4] for multi-view learning and semi-supervised learning. Still using two networks, JoCor uses a joint cross-entropy loss for sample selection but exploits the KL divergence to constrain the outputs of two networks, which makes predictions of each network closer to ground true labels and peer network's. JoCor can achieve promising performance on balanced noisy datasets. Unfortunately, for more practical tasks, *e.g.*, training on imbalanced noisy datasets, the mechanism of JoCor will *accelerate the degradation* of deep learning capabilities of two networks, which severely hinders the use of hard clean examples. Nevertheless, this type of examples is always the key to generalization [6]. Results in Section 4.3 will highlight the vulnerability of JoCor.

2.2. Other Methods for Learning with Noisy Labels

In addition to sample selection, we briefly review other kinds of methods for handling noisy labels. There is a large body of works proposed various methods for coping with noisy labels, which include but are not limited to, learning with a label noise transition matrix [17, 77, 90, 22, 70], reweighting examples [38, 54, 10, 55], recalibrating labels [59, 86, 84], using graph models [73, 34, 62], designing robust loss functions [85, 74, 41, 66], exploiting (implicit) regularization [82, 23, 18, 68, 7, 64, 61], and combining semi-supervised learning [46, 31, 37, 87, 20], etc. Readers can refer [57, 15] for more details of learning with noisy labels.

3. CoDis Meets Noisy Labels

3.1. Method Description

Given a training example $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$, we formulate the proposed method with two deep neural networks denoted by $f(\mathbf{x}_i; \mathbf{w}_1)$ and $f(\mathbf{x}_i; \mathbf{w}_2)$, where $\mathbf{w}_1$ and $\mathbf{w}_2$ are weights of two deep neural networks. While, $\mathbf{p}_1(\mathbf{x}_i) = [p_1^1(\mathbf{x}_i), p_1^2(\mathbf{x}_i), \ldots, p_1^c(\mathbf{x}_i)]$ and $\mathbf{p}_2(\mathbf{x}_i) = [p_2^1(\mathbf{x}_i), p_2^2(\mathbf{x}_i), \ldots, p_2^c(\mathbf{x}_i)]$ denote their *prediction probabilities* for the instance $\mathbf{x}_i$ respectively, which are the outputs of the *softmax* layer in two networks. That is to say, denoted the softmax activation function [13] by $S(\cdot)$, we have $\mathbf{p}_1(\mathbf{x}_i) = S(f(\mathbf{x}_i; \mathbf{w}_1))$ and $\mathbf{p}_2(\mathbf{x}_i) = S(f(\mathbf{x}_i; \mathbf{w}_2))$. In the following, we introduce two losses in CoDis, *i.e.*, the classification loss and discrepancy loss.

Classification loss. For the classification task, we exploit the *cross-entropy* loss ℓ_{CE} to minimize the distance between predictions and given labels. Specifically, for the training example $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$, the classification loss on it with each network (*e.g.*, the network with weights $\mathbf{w}_1$) is defined as

$$\mathcal{L}_{\text{C}} = \ell_{\text{CE}}\left(\mathbf{p}_1(\mathbf{x}_i), \tilde{\mathbf{y}}_i\right) = -\sum_{j=1}^{c} \tilde{\mathbf{y}}_i^j \log \mathbf{p}_1^j(\mathbf{x}_i). \quad (1)$$

As deep networks learn patterns first [2], they would first memorize training data of clean labels with the assumption that clean labels are of the majority in a noisy class. Small-loss training examples can thus be regarded as clean examples with high probability. Based on this, we can employ the loss (1) for sample selection as did in [16, 79, 69].

Discrepancy loss. Given a training example $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$, to measure the difference of the two networks' predictions $\mathbf{p}_1(\mathbf{x}_i)$ and $\mathbf{p}_2(\mathbf{x}_i)$, we adopt the Jensen-Shannon (JS) divergence [44], which is continuous like the cross entropy loss. We formulate the discrepancy loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{D}} &= \text{JS}(\mathbf{p}_1(\mathbf{x}_i)||\mathbf{p}_2(\mathbf{x}_i)) \\ &= \frac{1}{2}\text{KL}\left(\mathbf{p}_1(\mathbf{x}_i)||\frac{\mathbf{p}_1(\mathbf{x}_i)+\mathbf{p}_2(\mathbf{x}_i)}{2}\right) \\ &+ \frac{1}{2}\text{KL}\left(\mathbf{p}_2(\mathbf{x}_i)||\frac{\mathbf{p}_1(\mathbf{x}_i)+\mathbf{p}_2(\mathbf{x}_i)}{2}\right).\end{aligned} \quad (2)$$

Intuitively, the discrepancy loss (2) can quantify the output difference of two networks. For a training example, a large discrepancy loss means that the two networks have a high discrepancy on it.

Sample selection criterion. As discussed, we tend to select possibly clean examples based on the small-loss criterion and involve high-discrepancy examples in training at the same time. Therefore, the losses (1) and (2) should have a confrontation state. We define the *joint loss* for sample selection during training as follows:

$$\mathcal{L}_{\text{J}} = \mathcal{L}_{\text{C}} - \alpha * \mathcal{L}_{\text{D}}, \quad (3)$$

where $\alpha > 0$ is a hyper-parameter to balance the above two terms. We select the examples with smaller joint losses. More specifically, the example with a smaller classification loss can be seen as clean as mentioned [2, 80, 16, 21]. A larger discrepancy loss means that we select the possibly

Algorithm 1: CoDis Algorithm.

1: **Input:** two networks with initialized weights $\mathbf{w}_1$ and $\mathbf{w}_2$, learning rate η, fixed τ, epoch T_k and $T_{\max}$, iteration $t_{\max}$;
for $T = 1, 2, \ldots, T_{\max}$ **do**
  2: **Shuffle** training dataset $\tilde{\mathcal{D}}$;
  for $t = 1, \ldots, t_{\max}$ **do**
    3: **Fetch** mini-batch $\bar{\mathcal{D}}$ from $\tilde{\mathcal{D}}$;
    4: **Obtain** $\bar{\mathcal{D}}_1 = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}_{\mathrm{J}}(\mathbf{w}_1, \mathcal{D}')$;
    5: **Obtain** $\bar{\mathcal{D}}_2 = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}_{\mathrm{J}}(\mathbf{w}_2, \mathcal{D}')$;
    6: **Update** $\mathbf{w}_1 \leftarrow \mathbf{w}_1 - \eta\nabla\mathcal{L}_{\mathrm{C}}(\mathbf{w}_1, \bar{\mathcal{D}}_2)$;
    7: **Update** $\mathbf{w}_2 \leftarrow \mathbf{w}_2 - \eta\nabla\mathcal{L}_{\mathrm{C}}(\mathbf{w}_2, \bar{\mathcal{D}}_1)$;
  8: **Update** $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
9: **Output:** two trained networks with $\mathbf{w}_1$ and $\mathbf{w}_2$.

clean examples but with a high divergence between two networks, which could be hard clean examples for generalization. Then selected examples are used for robust training.

To determine the value of α, if we have a small trusted and unbiased dataset, we can choose a suitable α with meta learning [55, 56]. However, it may be somewhat strong to have such a small dataset in practice. Therefore, we choose α with a noisy validation set as did in [49, 8, 46]. In fact, the proposed sample selection criterion is stable with the change of α. We present detailed analyses and discussions for algorithm stability. More details are presented in Section 4. Besides, prior methods [25] design a low bound for the loss function to keep its value from going negative, which relieves the overfitting issue. As a contrast, our method can achieve great robustness, but does not rely on this operation.

Network weight updates. We maintain two networks simultaneously. The *cross-update* strategy is used. Specifically, each network selects training examples for its peer network based on the loss (3). Then each network employs the selected examples from the peer network for updates. Note that the joint loss consists of two terms, which controls the memorization of clean examples and enforces the divergence of two networks, respectively. To avoid the explicit enforcement hurting clean example memorization and impairing generalization [42], we only use the classification loss for weight updates. The divergence of the two networks can be maintained implicitly because of the proposed sample selection criterion. The overall procedure of CoDis is shown in Algorithm 1.

3.2. Theoretical Analysis

Our method inherits the paradigm of cross updates [4, 16], where two deep networks are exploited. The philosophy is, even though two networks have the same structures, with different initialization, they have different outputs during training, *i.e.*, $\mathcal{L}_{\mathrm{D}} > 0$. Note that the outputs of two networks cannot be *totally different*. A network can provide a part of information that the other network itself does not have. For example, for the network with weights $\mathbf{w}_1$ (denoted by f_1), on some instances, the network with weights $\mathbf{w}_2$ (denoted by f_2) has large discrepancy losses with f_1. While, for the other instances, f_2 has small discrepancy losses with f_1. Therefore, for any network, if the selected examples $\mathcal{S}$ have a size n_t, we can set a threshold for discrepancy losses to divide $\mathcal{S}$ into two sets σ_s and σ_l. The set σ_s includes the examples with smaller discrepancy losses with $|\sigma_s| = n_s$. While, the set σ_l includes the examples with larger discrepancy losses with $|\sigma_l| = n_l$. We provide the following theorem to show how the divergence between two networks in selected examples influences the classification on them.

Theorem 1 *The hypothesis spaces of f_1 and f_2 are denoted by $\mathcal{F}_1$ and $\mathcal{F}_2$. Suppose that by only minimizing the empirical risk on σ_s, we can train two networks f_1^0 and f_2^0, with $\mathcal{L}_{C1}^0$ and $\mathcal{L}_{C2}^0$. Assume that $n_s \geq \max\{\frac{2}{(\mathcal{L}_{Cl}^0)^2}\log\frac{2}{|\mathcal{F}_1|}, \frac{2}{(\mathcal{L}_{C2}^0)^2}\log\frac{2}{|\mathcal{F}_2|}\}$. Let $\Psi = \mathcal{L}_D(f_1, f_2) - \mathcal{L}_{C2}$, $\Phi = \mathcal{L}_D(f_2, f_1) - \mathcal{L}_{Cl}$, $\zeta_1 = \frac{\mathcal{L}_{Cl}^0\sqrt{n_s^2+n_s n_l}}{n_s} - \frac{n_l\Psi}{n_s}$, and $\zeta_2 = \frac{\mathcal{L}_{C2}^0\sqrt{n_s^2+n_s n_l}}{n_s} - \frac{n_l\Phi}{n_s}$. If $\Psi > \frac{\mathcal{L}_{Cl}^0}{2}$ and $\Phi > \frac{\mathcal{L}_{C2}^0}{2}$, then $\zeta_1 < \mathcal{L}_{Cl}^0$, $\zeta_2 < \mathcal{L}_{C2}^0$, and the following bounds on the classification losses of f_1 and f_2 hold for any $\delta > 0$:*

$$p(\mathcal{L}_{Cl} < \zeta_1) \geq 1 - \delta \quad \textit{and} \quad p(\mathcal{L}_{C2} < \zeta_2) \geq 1 - \delta. \quad (4)$$

The proof is provided in Appendix A. In Theorem 1, we claim that for any δ, the bounds in Eq. (4) hold with probability of at least $1 - \delta$. This probabilistic expression is widely used to analyze the generalization of an algorithm (*c.f.*, [44]). The above theorem provides theoretical insights to understand what factors influence the classification of a network on selected examples. Note that selected examples have high label precision [16, 63]. Our analysis can provide insights to the use of discrepancy losses in sample selection.

4. Experiments

4.1. Comparison Methods

We compare the proposed method with the state-of-art methods on sample selection: (1). MentorNet [21]. We use self-teaching MentorNet in this paper. (2). SIGUA [14], which exploits stochastic integrated gradient underweighted ascent to handle noisy labels. We use self-teaching SIGUA in this paper. (3). Co-teaching [16]. (4). Decoupling [42]. (5). Co-teaching+ [79]. (6). JoCor [63]. Although we focus on the sample selection approach for

combating noisy labels, to make this work more convincing, we also compare our method with other types of advanced methods. We employ the methods belonging to designing robust loss functions and exploiting (implicit) regularization, *i.e.*, APL [41] and CDR [68]. APL combines two mutually reinforcing robust loss functions. While, CDR employs unstructured network pruning to enhance the robustness of deep networks.

Note that we do not directly compare the proposed method with some state-of-the-art methods, *e.g.*, DivideMix [31]. It is because DivideMix is an aggregation of multiple techniques, *e.g.*, Mixup [81], soft labels [53], and semi-supervised learning [3]. We mainly focus on sample selection in learning with noisy labels. The direct comparison is not fair. Therefore, in this paper, to compare with it fairly, we follow the paradigm of DivideMix to boost our method. The enhanced method is named DivideMix+, where we replace the sample selection procedure [51] in DivideMix by CoDis. In this way, we show that our method can be exploited to improve the cutting-edge performance of state-of-the-art methods effectively.

4.2. Experiments on Balanced Noisy Datasets

Datasets. We verify the effectiveness of our method on the manually corrupted version of the following datasets: *MNIST* [28], *F-MNIST* [72], *SVHN* [45], *CIFAR-10* [26], *CIFAR-100* [26], and *NEWS* [27]. The six datasets are popularly used in prior works. Note that for *NEWS*, we borrowed the pre-trained word embeddings from GloVe [50]. Important statistics of used datasets are provided in Appendix B.1.

Generating noisy labels. We consider broad types of noisy labels: Symmetric noise (abbreviated as Sym.), Pair-flip noise (abbreviated as Pair.), Tridiagonal noise (abbreviated as Trid.), and Instance-dependent noise (abbreviated as Ins.). The noise rates are set to 20% and 40% consistently, which aim to ensure that clean labels in noisy classes are *diagonally dominant* [41]. More details about generating noisy labels are provided in Appendix B.2 . We leave 10% of noisy training examples as a validation set. Note that the clean labels are dominating in noisy classes and that noisy labels are random, the accuracy on the noisy validation set and the accuracy on the clean test data set are *positively correlated*. The noisy validation set can thus be used.

Implementation. For a fair comparison, we implement all methods with default parameters by PyTorch, and conduct all the experiments on NVIDIA TITAN XP GPUs. For *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*, we employ a 9-layer CNN structure from [16], which is a standard testbed for weakly supervised learning. For *CIFAR-100*, we use a 7-layer CNN structure from [79]. For *NEWS*, we use a 3-layer MLP with the Softsign active function. Adam optimizer is with an initial learning rate of 0.001, and the batch size is set to 128 and we run 200 epochs. The learning rate is linearly decayed to zero from 80 to 200 epochs. Note that deep networks are highly non-convex, even with the same network and optimization method, different initializations can lead to different local optimal [42]. Thus, following [16, 79], we also take two networks with the same architecture but different initializations as two classifiers. Here, we assume the noise level τ is known and set $R(T) = 1 - \min\{\frac{T}{T_k}\tau, \tau\}$ with T_k=10. If τ is not known in advance, it can be inferred using validation sets [38].

Measurement. To measure performance, we use test accuracy, *i.e.*, *test accuracy = (# of correct predictions) / (# of testing)*. Intuitively, a higher test accuracy means that a method is more robust to noisy labels. Besides, we use the selected ratio, *i.e.*, *selected ratio = (# of selected training examples) / (# of all training examples)*. The higher selected ratio means that a method is more sample-efficient. Note that due to the limited page, in Appendix C.4, we compare the label precision of sample selection, *i.e.*, *label precision = (# of clean labels) / (# of all selected labels)*.

Experimental results. For test accuracy, the results of experiments on balanced noisy datasets are provided in Table 1. In general, the proposed method achieves superior robustness compared with multiple baselines. More specifically, for each dataset, our method can achieve the best performance in most cases. In some cases, although it cannot surpass all baselines, it often obtains the second-best performance. Therefore, the performance is still competitive. In addition, for selected ratios, we compare CoDis with Co-teaching+. The results are shown in Table 2. We can see that CoDis is more sample-efficient than Co-teaching+. We may notice that, no matter on which dataset, the selected ratio of CoDis is 60.90%. It is because $R(T) = 1-\min\{T/T_k\tau, \tau\}$ controls the numbers of selected examples. If the parameters T_k and τ are fixed to 10 and 40% respectively, the average selected ratio must be 60.90% over 200 epochs.

Experiments with higher noise levels. Before this, for the symmetric noise, we set the noise rate to 20% and 40% respectively to verify the effectiveness of our method. Here, for symmetric noise, we increase the noise levels to 50%, 60%, and 70% to further support our claims. Experiments are conducted on *MNIST* and *F-MNIST*. Due to the limited page, experimental results and discussions are provided in Appendix C.1.

Comparison with DivideMix. We compare DivideMix[1] with DivideMix+ on *CIFAR-10* and *CIFAR-100*. Experimental settings, *e.g.*, the network structure and optimizer, follow those of settings in DivideMix. The results in Table 3 show that DivideMix+ can outperform DivideMix in all cases, which mean that our method can be used to improve the cutting-edge performance of state-of-the-arts.

[1] Official code: https://github.com/LiJunnan1992/DivideMix

Table 1. Mean and standard deviations of test accuracy (%) on two balanced noisy datasets with different noise levels. The test accuracy is calculated over the last ten epochs. The results are reported over five trials. The best result and second best result in each case are highlighted in red and blue respectively. Results achieved on the other four balanced noisy datasets are provided in Appendix C.1.

	Noise type	Sym.		Pair.		Trid.		Ins.	
	Setting	20%	40%	20%	40%	20%	40%	20%	40%
CIFAR-10	APL	76.20±1.07	67.20±0.89	77.74±0.98	62.05±0.96	79.05±0.61	70.88±1.04	78.32±0.52	66.25±1.92
	CDR	69.74±0.92	50.86±0.74	72.07±0.19	52.01±0.59	71.11±0.84	53.59±0.76	71.55±0.32	52.18±1.50
	MentorNet	80.92±0.48	74.67±1.17	77.98±0.31	69.39±1.73	78.02±0.29	71.56±0.93	77.02±0.71	68.17±2.52
	SIGUA	78.19±0.22	77.67±0.41	74.41±0.81	61.91±5.27	75.75±0.53	74.05±0.41	74.34±0.39	67.98±1.34
	Co-teaching	82.35±0.16	77.96±0.39	80.94±0.46	72.81±0.92	81.17±0.60	74.37±0.64	79.92±0.57	73.29±1.62
	Decoupling	74.05±0.38	55.62±0.61	74.62±0.48	53.34±0.71	75.00±0.50	56.93±0.65	74.16±0.25	54.71±0.95
	Co-teaching+	75.88±0.32	62.93±0.70	75.86±0.33	54.38±0.82	76.31±0.52	59.54±0.77	75.11±0.78	57.30±1.53
	JoCor	80.96±0.25	76.65±0.43	80.33±0.20	71.62±1.05	79.03±0.13	74.33±1.09	78.21±0.34	71.46±1.27
	CoDis	82.30±0.29	77.61±0.28	81.60±0.18	73.12±1.18	81.83±0.24	74.44±1.01	82.17±0.99	74.31±1.26
NEWS	APL	49.63±2.33	46.81±0.48	46.82±0.90	35.48±1.12	48.62±0.80	37.79±0.82	48.90±0.75	39.88±1.25
	CDR	45.07±0.81	32.54±0.88	46.78±0.83	35.29±0.63	46.52±0.76	35.76±0.74	45.75±0.85	34.69±0.79
	MentorNet	56.69±0.37	54.29±0.29	55.60±0.42	47.42±1.07	55.00±0.47	50.57±0.52	56.50±0.46	50.86±0.36
	SIGUA	54.44±0.75	53.22±0.73	48.13±0.39	43.73±0.32	49.51±0.52	49.74±1.50	53.22±0.44	50.02±0.28
	Co-teaching	56.99±0.28	54.85±0.53	55.61±0.20	46.29±1.07	56.40±0.73	51.63±0.33	56.61±0.36	51.37±0.32
	Decoupling	50.74±0.20	39.78±0.14	51.36±0.54	38.69±1.03	51.44±0.73	39.98±1.12	50.47±0.52	37.92±0.98
	Co-teaching+	50.84±0.40	44.81±1.01	51.12±0.62	39.34±0.99	51.68±1.09	43.08±1.65	50.71±0.86	42.77±0.93
	JoCor	57.15±0.33	55.48±0.29	55.96±0.26	47.23±1.57	56.55±0.89	52.40±0.65	56.88±0.45	51.32±0.46
	CoDis	57.15±0.20	54.93±0.21	55.52±0.35	47.45±1.05	56.07±0.79	52.28±0.47	56.92±0.47	52.24±0.31

Table 2. The average selected ratio (%) on *MNIST* and *F-MNIST* with different noise settings. The noise rate is set to 40%. The ratio is calculated over 200 epochs.

MNIST				
Method / Noise	Sym.	Pair.	Trid.	Ins.
Co-teaching+ (on *MNIST*)	9.04	5.77	10.13	7.90
Co-teaching+ (on *F-MNIST*)	12.37	10.56	15.21	13.60
CoDis	**60.90**			

Table 3. Mean and standard deviations of test accuracy (%) on *CIFAR-10* and *CIFAR-100* compared DivideMix with DivideMix+. The best mean results in each case are in **bold**.

CIFAR-10				
Method/Noise		Sym. 20%	Sym. 50%	Sym. 80%
DivideMix	Best	95.92±0.15	94.47±0.22	92.29±0.45
	Last	95.55±0.06	94.25±0.18	92.12±0.51
DivideMix+ (ours)	Best	**96.21±0.13**	**94.77±0.21**	**93.05±0.63**
	Last	**95.83±0.17**	**94.53±0.15**	**92.88±0.41**
CIFAR-100				
DivideMix	Best	77.25±0.27	74.25±0.38	59.93±0.75
	Last	76.78±0.29	73.95±0.33	59.33±1.09
DivideMix+ (ours)	Best	**77.45±0.20**	**74.67±0.40**	**60.36±0.84**
	Last	**77.02±0.32**	**74.32±0.30**	**60.12±0.59**

4.3. Experiments on Imbalanced Noisy Datasets

Datasets and implementation. We consider two kinds of experimental settings for imbalanced noisy cases. As discussed, CoDis emphasizes the data with high discrepancies between two networks, which are probably hard examples. Therefore, we exploit imbalanced noisy cases to verify the effectiveness of the proposed method, and show that it can better mine hard clean examples than baselines, following superior robustness. In more detail, the first one is asymmetric noise (abbreviated as Asym.), which considers the *visual similarity* in the flip process and is closer to instance noise [49]. This type of noise always makes noisy datasets *imbalanced*. We inject asymmetric noise on the image datasets, *i.e.*, *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*. The noise rate is set to 20%, 30%, 40%, and 45% respectively. More details are provided in Appendix B.2.

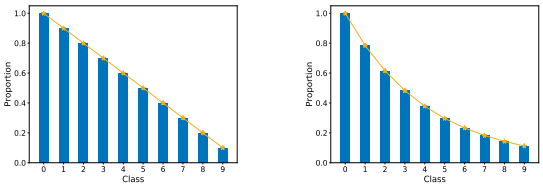

Figure 1. Illustrations for two types of long-tailed datasets.

The second one is *long-tailed* noise (abbreviated as L-Tailed.), where training data exhibit long-tailed distributions with class imbalance [75]. In this paper, we reduce the proportion of training examples with different classes to simulate long-tailed distributions. We use two simulation ways, which are shown in Figure 1. Taking *MNIST* as an example, the built datasets are called *L-MNIST-1* (Figure 1, Left) and *L-MNIST-2* (Figure 1, Right). Other used datasets are named in the same way. We employ *MNIST* and *SVHN* in this setting. Besides, asymmetric noise is further imposed on long-tailed datasets, which forms noisy long-tailed datasets. The implementation details are kept the same as the cases in experiments on balanced noisy datasets, including optimization and network structures.

Experimental results. The results of experiments only with asymmetric noise are presented in Table 4. Extensive results show that our method can achieve clear leads over all baselines. For the most challenging cases, *i.e.*, our method achieves more than 5% improvements on *MNIST*, *F-MNIST*, and *SVHN*. For *CIFAR-10*, our method also

Table 4. Mean and standard deviations of test accuracy (%) on class-imbalanced noisy datasets with different noise levels. The test accuracy is calculated over the last ten epochs. The results are reported over five trials. The best result and second best result in each case are highlighted in red and blue respectively.

	Noise type	Asym. 20%	Asym. 30%	Asym. 40%	Asym. 45%
MNIST	APL	98.63±0.05	98.03±0.38	88.65±1.72	90.82±2.04
	CDR	96.73±0.19	94.33±1.07	91.05±0.76	76.79±3.07
	MentorNet	96.32±0.17	93.75±3.91	90.96±0.97	67.91±5.44
	SIGUA	93.96±0.82	89.15±1.15	62.59±0.15	50.22±2.74
	Co-teaching	98.25±0.08	98.26±0.11	95.08±0.43	76.17±5.38
	Decoupling	98.71±0.06	95.02±0.23	86.72±0.41	83.29±0.55
	Co-teaching+	98.79±0.11	96.70±0.24	94.99±0.41	93.47±0.49
	JoCor	98.05±0.37	94.95±3.84	94.55±1.08	80.50±2.11
	CoDis	99.55±0.03	99.42±0.02	99.18±0.07	99.01±0.14
F-MNIST	APL	90.13±0.17	86.26±0.47	80.34±0.63	60.15±2.72
	CDR	89.78±0.41	85.17±1.04	79.05±1.39	52.75±2.44
	MentorNet	89.69±0.19	84.20±3.36	67.21±2.94	61.18±2.98
	SIGUA	76.97±2.59	63.64±7.36	45.96±3.40	43.52±2.37
	Co-teaching	91.03±0.14	88.67±0.60	68.07±4.58	64.87±4.88
	Decoupling	90.74±0.35	85.34±0.30	79.45±0.42	60.39±2.87
	Co-teaching+	91.66±0.34	89.38±0.39	82.33±0.64	68.29±3.14
	JoCor	90.95±0.21	85.59±3.91	79.79±2.39	62.53±2.33
	CoDis	93.12±0.15	92.11±0.28	84.10±2.93	74.30±3.92
SVHN	APL	92.57±0.44	89.22±0.46	84.00±1.07	79.52±1.18
	CDR	90.17±0.37	86.16±0.30	81.79±0.82	79.45±0.62
	MentorNet	92.63±0.32	89.31±0.41	83.02±2.06	71.68±3.27
	SIGUA	71.78±2.55	66.84±3.53	43.34±5.93	42.06±8.72
	Co-teaching	94.87±0.36	93.48±0.42	91.55±0.33	88.79±4.22
	Decoupling	92.77±0.61	86.33±1.23	82.60±0.85	80.38±0.84
	Co-teaching+	93.32±0.29	89.88±0.36	86.60±1.09	85.01±1.02
	JoCor	93.40±0.28	90.79±0.23	72.94±6.38	67.13±4.15
	CoDis	95.38±0.21	95.10±0.29	94.62±0.28	94.00±0.30
CIFAR-10	APL	79.98±0.31	76.32±1.16	70.72±0.98	67.01±0.53
	CDR	78.86±0.41	74.49±0.94	70.52±0.47	67.35±0.30
	MentorNet	77.98±0.31	78.81±0.56	69.39±1.73	53.11±1.15
	SIGUA	74.41±0.81	70.55±0.92	61.91±5.27	33.59±4.73
	Co-teaching	80.94±0.96	80.87±0.24	72.81±0.92	57.20±1.91
	Decoupling	79.18±0.42	74.56±0.54	69.56±0.52	63.11±3.56
	Co-teaching+	79.67±0.30	75.74±0.22	70.70±0.41	64.11±3.64
	JoCor	80.33±0.20	80.25±0.40	71.62±1.05	53.47±1.41
	CoDis	84.78±0.22	82.70±0.42	75.24±1.44	68.80±2.14

achieves superior robustness. The results on four noisy long-tailed datasets are shown in Figure 2. From all training curves, we can see that CoDis can achieve superior robustness on long-tailed noisy datasets.

It should be noted that, during training, Co-teaching selects the same number of examples as CoDis. As be seen in results, in many cases, CoDis outperforms Co-teaching with a large margin. The results demonstrate that, when learning with class-imbalanced noisy datasets, CoDis can successfully mine clean hard examples for training, which enhances generalization. Besides, the baseline JoCor is *weak* on imbalanced noisy datasets. Compared with its performance achieved on balanced noisy datasets, we can see that it cannot handle these realistic cases well. Moreover, JoCor is *unstable* during training, with *large error bars*. This issue is pessimistic, and could limit the method's practical applications largely.

Ablation study. We conduct detailed ablation studies. Due to the limited page of the main paper, experiments are reported in Appendix. Specifically, we show that our method is robust to the choice of network architectures and the use of data augmentation technologies, which are presented in Appendix C.2. The results demonstrate the effectiveness of our method consistently. Moreover, we conduct sensitivity analyses of the hyperparameter α in Appendix C.3, which show that in the certain value range, our method is robust to the choice of α. We also provide detailed analysis on T_k and $T_{\max}$ there. The results mean that our method can be easy to apply, without sophisticated hyperparameter tuning.

Table 5. Test accuracy on *Food-101* and *Clothing1M*. The best result and second best result in each case are highlighted in red and blue respectively.

Dataset	*Food-101*	*Clothing1M*
Method	Accuracy (%)	Accuracy (%)
APL	82.17	54.46
CDR	86.36	66.59
MentorNet	81.25	67.25
SIGUA	79.68	65.37
Co-teaching	83.73	67.94
Decoupling	78.88	67.65
Co-teaching+	76.89	63.83
JoCor	84.04	69.06
CoDis	86.13	71.60
DivideMix	86.73	74.76
ELR+	85.77	74.81
DivideMix+ (ours)	86.88	74.92

4.4. Experiments on Real-world Noisy Datasets

Datasets. Three real-world noisy datasets are used in this paper, *i.e.*, *Food-101* [30], *Clothing1M* [73], and *Webvision* [34]. *Food-101* consists of 101 food categories, with 101,000 images. For each class, 250 manually reviewed clean test images are provided as well as 750 training images with real-world label noise. WebVision contains 2.4 million images crawled from the websites using the 1,000 concepts in ImageNet ILSVRC12. Following the "mini" setting in [41, 8], we take the first 50 classes of the Google resized image subset, and evaluate the trained networks on the same 50 classes of the WebVision and ILSVRC12 validation sets, which are exploited as test sets. *Clothing1M* consists of 1M noisy training examples collected from online shopping websites.

Implementation. For *Food-101* and *Clothing1M*, we use ResNet-50 with ImageNet pretrained weights. For preprocessing, we resize the image to 256×256, crop the middle 224×224 as input, and perform normalization. We use the Adam optimizer and set the batch size to 32/64 for *Food-101* and *Clothing1M* respectively. During training, we run 20 epochs in total and set the learning rate 8×10^{-4}, 5×10^{-4}, and 5×10^{-5} for 5 epochs each. For *Webvision*, we use Inception-ResNet V2. Note that, for DivideMix+, it follows the implementation details of [31], but with a different sample selection procedure.

Experimental results. The classification performance achieved on *Food-101* and *Clothing1M* is provided in Table 5. Specifically, compared with CoDis with the base-

Figure 2. Test accuracy *vs.* the number of epochs on four long-tailed noisy datasets. The error bar for standard deviation in each figure has been shaded.

Table 6. Comparison with state-of-the-art methods trained on (mini) WebVision dataset [8, 31]. Numbers denote top-1 (top-5) accuracy (%) on the WebVision validation set and the ImageNet ILSVRC12 validation set. The best result and second best result in each case are highlighted in red and blue respectively.

Test dataset	WebVision		ILSVRC12	
Method	top1	top5	top1	top5
APL	62.30	84.02	61.27	84.82
CDR	62.84	84.11	61.85	85.80
MentorNet	63.00	81.40	57.66	80.01
SIGUA	57.38	78.92	52.88	74.67
Co-teaching	63.58	85.20	61.22	84.78
Decoupling	62.54	84.74	57.26	80.50
Co-teaching+	61.18	83.30	58.74	82.72
JoCor	63.33	85.06	58.76	82.85
CoDis	63.80	85.54	62.29	85.39
DivideMix	77.32	91.64	75.20	90.84
ELR+	77.78	91.68	70.29	89.76
DivideMix+ (ours)	77.51	91.95	75.51	91.58

lines without the combination of multiple techniques, our method achieves the second best performance. CoDis is slightly lower than CDR, *i.e.*, 86.13% *vs.* 86.36%, but is clearly better than other baselines. When we combine other advanced methods to boost our method as did in DivideMix, DivideMix+ can outperform both DivideMix and ELR+. For results on *Clothing1M*, compared with CoDis with the baselines without the combination of multiple techniques, our method achieves an improvement of +2.54% over the best baseline JoCor. When we compared DivideMix+ with other baselines, DivideMix+ can outperform DivideMix and ELR+, and achieve the best performance.

The results on *WebVision* are provided in Table 6. In more detail, compared with CoDis with the baselines without multiple techniques, CoDis achieves the best performance in all cases. Moreover, compared with DivideMix and ELR+, DivideMix+ achieves the best performance in three cases. All results mean that our method can be exploited to improve the cutting-edge performance of state-of-the-art methods.

5. Conclusion

This paper presents a robust learning paradigm called CoDis, which trains deep neural networks robustly with noisy labels. CoDis maintains two networks simultaneously. The core idea is to make each network selects its clean data for peer network and tries to choose the data with high discrepancies between two networks at the same time. The proposed sample selection procedure is sample-efficient, and can ensure enough (hard) clean examples for

generalization. Comprehensive experiments with superior performance justify our claims well. In the future, we are interested in applying our method to data cleaning for large-scale pre-trained models [5, 36, 48, 83].

Acknowledgements

Yibing Zhan was partially supported by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project (No. 2021ZD0111700) and the National Natural Science Foundation of China (Grant No. 62002090). Bo Han was supported by NSFC Young Scientists Fund No. 62006202 and Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652. Jun Yu was supported by the Natural Science Foundation of China (62276242), National Aviation Science Foundation (2022Z071078001), CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-016B, CAAIXSJLJJ-2022-001A), Anhui Province Key Research and Development Program (202104a05020007), USTC-IAT Application Sci. & Tech. Achievement Cultivation Program (JL06521001Y), and Sci. & Tech. Innovation Special Zone (20-163-14-LZ-001-004-01). Mingming Gong was supported by ARC DE210101624. Chen Gong was supported by the NSF of China (No: 61973162), NSF of Jiangsu Province (No: BZ2021013), NSF for Distinguished Young Scholar of Jiangsu Province (No: BK20220080), and the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114). Tongliang Liu was partially supported by ARC projects: FT220100318, DP220102121, LP220100527, LP220200949, and IC190100031.

References

[1] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988. 1

[2] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pages 233–242, 2017. 1, 3

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 5

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *ACCLT*, pages 92–100, 1998. 3, 4

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901, 2020. 9

[6] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, pages 1002–1012, 2017. 3

[7] Pengfei Chen, Guangyong Chen, Junjie Ye, Pheng-Ann Heng, et al. Noise against noise: stochastic label noise helps combat inherent label noise. In *ICLR*, 2021. 3

[8] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pages 1062–1070, 2019. 4, 7, 8

[9] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated input-dependent label noise in large-scale image classification. In *CVPR*, pages 1551–1560, 2021. 1

[10] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. In *NeurIPS*, 2020. 3

[11] Dengpan Fu, Dongdong Chen, Hao Yang, Jianmin Bao, Lu Yuan, Lei Zhang, Houqiang Li, Fang Wen, and Dong Chen. Large-scale pre-training for person re-identification with noisy labels. In *CVPR*, pages 2476–2486, 2022. 1

[12] Jinyang Gao, HV Jagadish, and Beng Chin Ooi. Active sampler: Light-weight accelerator for complex data analytics at scale. *arXiv preprint arXiv:1512.03880*, 2015. 2

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 3

[14] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *ICML*, pages 4006–4016, 2020. 4

[15] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020. 3

[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 *NeurIPS*, pages 8527–8537, 2018. 1, 2, 3, 4, 5

[17] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018. 3

[18] Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2020. 3

[19] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *ICCV*, pages 3326–3334, 2019. 2

[20] Zhuo Huang, Chao Xue, Bo Han, Jian Yang, and Chen Gong. Universal semi-supervised learning. In *NeurIPS*, volume 34, pages 26714–26725, 2021. 3

[21] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318, 2018. 1, 2, 3, 4

[22] Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. An information fusion approach to learning with instance-dependent label noise. In *ICLR*, 2022. 3

[23] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *ICCV*, pages 101–110, 2019. 3

[24] Youngdong Kim, Juseung Yun, Hyounguk Shon, and Junmo Kim. Joint negative and positive learning for noisy labels. In *CVPR*, pages 9442–9451, 2021. 1

[25] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, 2017. 4

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[27] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings*, pages 331–339. 1995. 5

[28] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998. 5

[29] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *ICML*, pages 3763–3772, 2019. 2

[30] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 2018. 7

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

[32] Shikun Li, Xiaobo Xia, Jiankang Deng, Shiming Ge, and Tongliang Liu. Transferring annotator-and instance-dependent transition matrix for learning from crowds. *arXiv preprint arXiv:2306.03116*, 2023. 1

[33] Shikun Li, Xiaobo Xia, Hansong Zhang, Yibing Zhan, Shiming Ge, and Tongliang Liu. Estimating noise transition matrix with label correlations for noisy multi-label learning. In *NeurIPS*, 2022. 1

[34] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, pages 1910–1918, 2017. 3, 7

[35] Kevin J Liang, Samrudhdhi B Rangrej, Vladan Petrovic, and Tal Hassner. Few-shot learning with noisy labels. In *CVPR*, pages 9089–9098, 2022. 1

[36] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 9

[37] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 3

[38] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016. 3, 5

[39] Wei Liu, Yu-Gang Jiang, Jiebo Luo, and Shih-Fu Chang. Noise resistant graph ranking for improved web image search. In *CVPR*, pages 849–856, 2011. 1

[40] Yueming Lyu and Ivor W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *ICLR*, 2020. 1, 2

[41] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pages 6543–6553, 2020. 2, 3, 5, 7

[42] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *NeurIPS*, pages 960–970, 2017. 3, 4, 5

[43] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of neural networks against noisy labels. In *NeurIPS*, 2020. 2

[44] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. 2018. 2, 3, 4

[45] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y.Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 5

[46] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2020. 3, 4

[47] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. 1

[48] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. 9

[49] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017. 4, 6

[50] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 5

[51] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern recognition*, 39(4):695–706, 2006. 5

[52] Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020. 2

[53] Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. 5

[54] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340, 2018. 3

[55] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 3, 4

[56] Jun Shu, Qian Zhao, Zengben Xu, and Deyu Meng. Meta transition adaptation for robust deep learning with noisy labels. *arXiv preprint arXiv:2006.05697*, 2020. 4

[57] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020. 3

[58] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *ACM MM*, pages 1405–1413, 2021. 1

[59] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 3

[60] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *CVPR*, pages 11244–11253, 2019. 1

[61] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. In *ICML*, 2019. 3

[62] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, pages 5596–5605, 2017. 3

[63] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020. 1, 2, 3, 4

[64] Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *ICLR*, 2021. 3

[65] Jiaheng Wei, Harikrishna Narasimhan, Ehsan Amid, Wen-Sheng Chu, Yang Liu, and Abhishek Kumar. Distributionally robust post-hoc classifiers under prior shifts. In *ICLR*, 2023. 1

[66] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*, 2021. 3

[67] Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR-Workshop*, pages 25–32, 2010. 1

[68] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021. 3, 5

[69] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *ICLR*, 2022. 2, 3

[70] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*, 2020. 3

[71] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pages 6835–6846, 2019. 1

[72] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[73] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699, 2015. 3, 7

[74] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. $\mathcal{L}_{DMI}$: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, pages 6222–6233, 2019. 3

[75] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020. 6

[76] Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *ICML*, pages 10789–10798, 2020. 1

[77] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*, 2020. 3

[78] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, pages 7017–7025, 2019. 1

[79] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement benefit co-teaching? In *ICML*, 2019. 1, 2, 3, 4, 5

[80] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1, 3

[81] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5

[82] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3

[83] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023. 9

[84] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021. 3

[85] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8778–8788, 2018. 3

[86] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *ICML*, pages 11447–11457, 2020. 3

[87] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *ICLR*, 2021. 3

[88] Xiong Zhou, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Asymmetric loss functions for noise-tolerant learning: Theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[89] Yifan Zhou, Yifan Ge, and Jianxin Wu. Friends and foes in learning from noisy labels. *arXiv preprint arXiv:2103.15055*, 2021. 1

[90] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *CVPR*, pages 10113–10123, 2021. 3