How does Disagreement Help Generalization against Label Corruption?

Xingrui Yu [1] Bo Han [2] Jiangchao Yao [3] Gang Niu [2] Ivor W. Tsang [1] Masashi Sugiyama [2 4]

Abstract

Learning with noisy labels is one of the hottest problems in weakly-supervised learning. Based on memorization effects of deep neural networks, training on small-loss instances becomes very promising for handling noisy labels. This fosters the state-of-the-art approach "Co-teaching" that cross-trains two deep neural networks using the small-loss trick. However, with the increase of epochs, two networks converge to a consensus and Co-teaching reduces to the self-training MentorNet. To tackle this issue, we propose a robust learning paradigm called Co-teaching+, which bridges the "Update by Disagreement" strategy with the original Co-teaching. First, two networks feed forward and predict all data, but keep prediction disagreement data only. Then, among such disagreement data, each network selects its small-loss data, but back propagates the small-loss data from its peer network and updates its own parameters. Empirical results on benchmark datasets demonstrate that Co-teaching+ is much superior to many state-of-the-art methods in the robustness of trained models.

1. Introduction

In weakly-supervised learning, learning with noisy labels is one of the most challenging questions, since noisy labels are ubiquitous in our daily life, such as web queries (Liu et al., 2011), crowdsourcing (Welinder et al., 2010), medical images (Dgani et al., 2018), and financial analysis (Aït-Sahalia et al., 2010). Essentially, noisy labels are systematically corrupted from ground-truth labels, which inevitably degenerates the accuracy of classifiers. Such degeneration becomes even more prominent for deep learning models (e.g., convolutional and recurrent neural networks), since these complex models can fully memorize noisy labels (Zhang et al., 2017; Arpit et al., 2017).

To handle noisy labels, classical approaches focus on either adding regularization (Miyato et al., 2016) or estimating the label transition matrix (Patrini et al., 2017). Specifically, both explicit and implicit regularizations leverage the regularization bias to overcome the label noise issue. Nevertheless, they introduced a permanent regularization bias, and the learned classier barely reaches the optimal performance. Meanwhile, estimating the label transition matrix does not introduce the regularization bias, and the accuracy of classifiers can be improved by such accurate estimation. However, the label transition matrix is hard to be estimated, when the number of classes is large.

Recently, a promising way of handling noisy labels is to train on small-loss instances (Jiang et al., 2018; Ren et al., 2018). These works try to select small-loss instances, and then use them to update the network robustly. Among those works, the representative methods are MentorNet (Jiang et al., 2018) and Co-teaching (Han et al., 2018b). For example, MentorNet pre-trains an extra network, and then it uses the extra network for selecting clean instances to guide the training of the main network. When the clean validation data is not available, self-paced MentorNet has to use a predefined curriculum (e.g., small-loss instances). Nevertheless, the idea of self-paced MentorNet is similar to the self-training approach, and it inherits the same inferiority of accumulated error.

To solve the accumulated error issue in MentorNet, Co-teaching has been developed, which simultaneously trains two networks in a symmetric way (Han et al., 2018b). First, in each mini-batch data, each network filters noisy (i.e., big-loss) samples based on the memorization effects. Then, it teaches the remaining small-loss samples to its peer network for updating the parameters, since the error from noisy labels can be reduced by peer networks mutually. From the initial training epoch, two networks having different learning abilities can filter different types of error. However, with the increase of training epochs, two networks will converge to a consensus gradually and Co-teaching reduces to the self-training MentorNet in function.

To address the consensus issue in Co-teaching, we should consider how to always keep two networks diverged within

[1]CAI, University of Technology Sydney [2]RIKEN-AIP [3]Alibaba Damo Academy [4]University of Tokyo. Correspondence to: Xingrui Yu <xingrui.yu@student.uts.edu.au>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

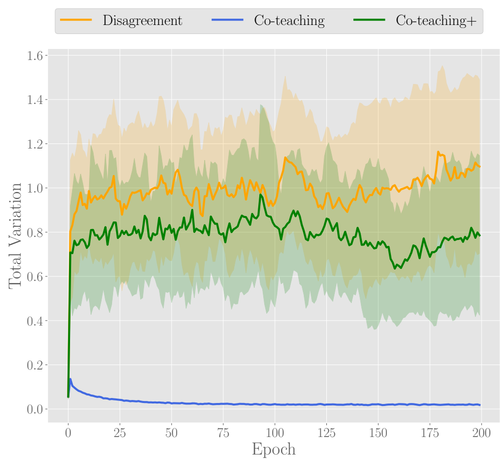


Figure 1. Comparison of divergence (evaluated by Total Variation) between two networks trained by the “Disagreement” strategy, Co-teaching and Co-teaching+, respectively. Co-teaching+ naturally bridges the “Disagreement” strategy with Co-teaching.

the training epochs, or how to slow down the speed that two networks will reach a consensus with the increase of epochs. Fortunately, we find that a simple strategy called “Update by Disagreement” (Malach & Shalev-Shwartz, 2017) may help us to achieve the above target. This strategy conducts updates only on selected data, where there is a prediction disagreement between two classifiers.

To demonstrate that the “Disagreement” strategy can keep two networks diverged during training, we train two 3-layer MLPs (Goodfellow et al., 2016) on *MNIST* simultaneously for 10 trials, and report total variations of Softmax outputs between two networks in Figure 1. We can clearly observe that two networks trained by Co-teaching (blue in Figure 1) converge to a consensus gradually, while two networks trained by the “Disagreement” strategy (orange in Figure 1) often keep diverged.

Motivated by this phenomenon, in this paper, we propose a robust learning paradigm called Co-teaching+ (Figure 2), which naturally bridges the “Disagreement” strategy with Co-teaching. Co-teaching+ trains two deep neural networks similarly to the original Co-teaching, but it consists of the disagreement-update step (data update) and the cross-update step (parameters update). Initially, in the disagreement-update step, two networks feed forward and predict all data first, and only keep prediction disagreement data. This step indeed keeps two networks (trained by Co-teaching+) diverged (green in Figure 1). Then, in the cross-update step, each network selects its small-loss data from such disagreement data, but back propagates the small-loss data from its peer network and updates its own parameters. Intuitively, the idea of disagreement-update comes from

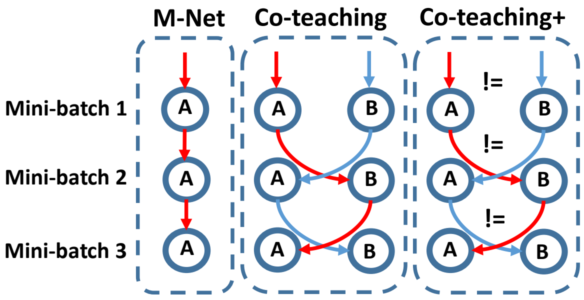


Figure 2. Comparison of error flow among MentorNet (M-Net), Co-teaching and Co-teaching+. Assume that the error flow comes from the selection of training instances, and the error flow from network A or B is denoted by red arrows or blue arrows, respectively. **Left panel:** M-Net maintains only one network (A). **Middle panel:** Co-teaching maintains two networks (A & B) simultaneously. In each mini-batch data, each network selects its small-loss data to teach its peer network for the further training. **Right panel:** Co-teaching+ also maintains two networks (A & B). However, two networks feed forward and predict each mini-batch data first, and keep prediction disagreement data (!=) only. Based on such disagreement data, each network selects its small-loss data to teach its peer network for the further training.

Co-training (Blum & Mitchell, 1998), where two classifiers should keep diverged to achieve the better ensemble effects. The intuition of cross-update comes from culture evolving hypothesis (Bengio, 2014), where a human brain can learn better if guided by the signals produced by other humans.

We conduct experiments on both simulated and real-world noisy datasets, including noisy *MNIST*, *CIFAR-10*, *CIFAR-100*, *NEWS*, *T-ImageNet* and three *Open-sets* (Wang et al., 2018). Empirical results demonstrate that the robustness of deep models trained by the Co-teaching+ approach is superior to many state-of-the-art methods, including Co-teaching, MentorNet and F-correction (Patrini et al., 2017). Before delving into details, we clearly emphasize our contribution as follows.

- We denote that “Update by Disagreement” (i.e., the Decoupling algorithm) itself *cannot* handle noisy labels, which has been empirically justified in Section 3.
- We realize that the “Disagreement” strategy *can* keep two networks diverged, which significantly boosts the performance of Co-teaching.
- We summarize *three* key factors towards training robust deep networks with noisy labels: (1) using the small-loss trick; (2) cross-updating parameters of two networks; and (3) keeping two networks diverged.

The rest of this paper is organized as follows. In Section 2, we propose our robust learning paradigm Co-teaching+. Experimental results are discussed in Sections 3 and 4. Conclusions are given in Section 5.

2. Co-teaching+: Towards Training of Robust Deep Networks with Noisy Labels

Similar to Co-teaching, we also train two deep neural networks. As in Figure 2, in each mini-batch data, each network conducts its own prediction, then selects instances for which there is a prediction disagreement between two networks. Based on such disagreement data, each network further selects its small-loss data, but back propagates the small-loss data selected by its peer network and updates itself parameters. We call such algorithm as Co-teaching+ (Algorithm 1), which consists of disagreement-update step and cross-update step. This brings the question as follows.

How does disagreement benefit Co-teaching? To answer this question, we should first understand the main drawback of Co-teaching. In the early stage of training, the divergence of two networks mainly comes from different (random) parameter initialization. Intuitively, this divergence between two networks pushes Co-teaching to become more robust than self-paced MentorNet, since two diverged networks have different abilities to filter different types of error. However, with the increase of training epochs, two networks will gradually converge to be close to each other (blue in Figure 1). Thus, Co-teaching degenerates to self-paced MentorNet, and will not promote the learning ability to select clean data any more. To overcome this issue, we need to keep the constant divergence between two networks or slow down the speed that two networks reach a consensus. This intuition comes from Co-training algorithm, where in semi-supervised learning (Chapelle et al., 2009), the better ensemble effects require to keep diverged more between two classifiers.

Fortunately, the "Disagreement" strategy (Malach & Shalev-Shwartz, 2017) can help us to keep two networks diverged (orange in Figure 1), since this strategy conducts algorithm updates only on selected data, where there is a prediction disagreement between the two classifiers. Therefore, within the whole training epochs, if two networks always select the disagreement data for further training, the divergence of two networks will be always maintained. Specifically, during the training procedure of Co-teaching, if we use the "Disagreement" strategy to keep two networks diverged, then we can prevent Co-teaching reducing to self-training MentorNet in function. This brings us the new robust training paradigm Co-teaching+ (Algorithm 1, green in Figure 1).

Take "complementary peer learning" as an illustrative example for Co-teaching+. When students prepare for their exams, the peer learning will normally more boost their review efficiency than the solo learning. However, if two students are identically good at math but not good at literature, their review process in literature will have no any progress. Thus, the optimal peer should be complementary, which means that a student who is good at math should best review with another student who is good at literature. This point also explains why the diverged peer has more powerful learning ability than the identical peer.

Algorithm 1 Co-teaching+. Step 4: disagreement-update; Step 5-8: cross-update.

```
1: **Input** $w^{(1)}$ and $w^{(2)}$, training set $\mathcal{D}$, batch size $B$, learning rate $\eta$, estimated noise rate $\tau$, epoch $E_k$ and $E_{\max}$;
**for** $e = 1, 2, \ldots, E_{\max}$ **do**
    2: **Shuffle** $\mathcal{D}$ into $\frac{|\mathcal{D}|}{B}$ mini-batches;    //noisy dataset
    **for** $n = 1, \ldots, \frac{|\mathcal{D}|}{B}$ **do**
        3: **Fetch** $n$-th mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
        4: **Select** prediction disagreement $\bar{\mathcal{D}}'$ by Eq. (1);
        5: **Get** $\bar{\mathcal{D}}'^{(1)} = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq \lambda(e)|\bar{\mathcal{D}}'|} \ell(\mathcal{D}'; w^{(1)})$; //sample $\lambda(e)\%$ small-loss instances
        6: **Get** $\bar{\mathcal{D}}'^{(2)} = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq \lambda(e)|\bar{\mathcal{D}}'|} \ell(\mathcal{D}'; w^{(2)})$; //sample $\lambda(e)\%$ small-loss instances
        7: **Update** $w^{(1)} = w^{(1)} - \eta\nabla\ell(\bar{\mathcal{D}}'^{(2)}; w^{(1)})$;    //update $w^{(1)}$ by $\bar{\mathcal{D}}'^{(2)}$;
        8: **Update** $w^{(2)} = w^{(2)} - \eta\nabla\ell(\bar{\mathcal{D}}'^{(1)}; w^{(2)})$;    //update $w^{(2)}$ by $\bar{\mathcal{D}}'^{(1)}$;
    **end**
    9: **Update** $\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, \tau\}$ or $1 - \min\{\frac{e}{E_k}\tau, (1 + \frac{e-E_k}{E_{\max}-E_k})\tau\}$;
**end**
10: **Output** $w^{(1)}$ **and** $w^{(2)}$.
```

Algorithm description. Algorithm 1 consists of the disagreement-update step (step 4) and the cross-update step (step 5-8), where we train two deep neural networks in a mini-batch manner.

In step 4, two networks feed forward and predict the same mini-bach of data $\bar{\mathcal{D}}=\{(x_1, y_1), (x_2, y_2), \cdots, (x_B, y_B)\}$ first, where the batch size is B. Then, they keep prediction disagreement data $\bar{\mathcal{D}}'$ (Eq. (1)) according to their predictions $\{\bar{y}_1^{(1)}, \bar{y}_2^{(1)}, \ldots, \bar{y}_B^{(1)}\}$ (predicted by $w^{(1)}$) and $\{\bar{y}_1^{(2)}, \bar{y}_2^{(2)}, \ldots, \bar{y}_B^{(2)}\}$ (predicted by $w^{(2)}$):

$$\bar{\mathcal{D}}' = \{(x_i, y_i) : \bar{y}_i^{(1)} \neq \bar{y}_i^{(2)}\}, \tag{1}$$

where $i \in \{1, \ldots, B\}$. The intuition of this step comes from Co-training, where two classifiers should keep diverged to achieve the better ensemble effects.

In step 5-8, from the disagreement data $\bar{\mathcal{D}}'$, each network $w^{(1)}$ (resp. $w^{(2)}$) selects its own small-loss data $\bar{\mathcal{D}}'^{(1)}$ (resp. $\bar{\mathcal{D}}'^{(2)}$), but back propagates the small-loss data $\bar{\mathcal{D}}'^{(1)}$ (resp. $\bar{\mathcal{D}}'^{(2)}$) to its peer network $w^{(2)}$ (resp. $w^{(1)}$) and updates parameters. The intuition of step 5-8 comes from the aforementioned culture evolving hypothesis (Bengio, 2014), where a human brain can learn better if guided by the signals produced by other humans.

In step 9, we update $\lambda(e)$, which controls how many small-loss data should be selected in each training epoch. Due to the memorization effects, deep networks will fit clean data first and then gradually over-fit noisy data.

Thus, at the beginning of training, we keep more small-loss data (with a large $\lambda(e)$) in each mini-batch, which is equivalent to dropping less data. Since deep networks will fit clean data first, noisy data do not matter at the initial training epochs. With the increase of epochs, we keep less small-loss data (with a small $\lambda(e)$) in each mini-batch. As deep networks will over-fit noisy data gradually, we should drop more data. The gradual decrease of $\lambda(e)$ prevents deep networks over-fitting noisy data to some degree.

Similar to Co-teaching, we decrease $\lambda(e)$ quickly at the first E_k epochs to stop networks over-fitting to the noisy data, namely $\lambda(e) = 1 - \frac{e}{E_k}\tau$. However, after E_k epochs, Co-teaching+ has two types of $\lambda(e)$. The first type keeps a constant $\lambda(e)$, where $\lambda(e) = 1 - \tau$; while the second type further decreases $\lambda(e)$ slowly, where $\lambda(e) = 1 - (1 + \frac{e - E_k}{E_{\max} - E_k})\tau$. We take an example to explain the difference.

Assume that the estimated noise rate τ is 30%. It means that, after E_k epochs, the first type will constantly fetch 70% small-loss data in each mini-batch as "clean" data. However, the τ estimation tends to be inaccurate in practice. Therefore, given the estimated τ, we should fetch less data, e.g., 60% small-loss data, to keep remained data more clean. This explains why, in real-world noisy datasets, Co-teaching+ chooses the second type to further decrease $\lambda(e)$ slowly after E_k epochs (Section 4).

Relations to other approaches. We compare our Co-teaching+ with related approaches in Table 1. We try to find the connections among them, and pinpoint the key factors that can handle noisy labels. First, self-paced MentorNet (Jiang et al., 2018) employs the small-loss trick to handle noisy labels. However, this idea is similar to the self-training approach, and it inherits the same inferiority of accumulated error caused by the sample-selection bias. Inspired by Co-training (Blum & Mitchell, 1998) that trains double classifiers and cross updates parameters, Co-teaching (Han et al., 2018b) has been developed to cross train two deep networks, which addresses the accumulated error issue in MentorNet. Note that, Co-training does not exploit the memorization in deep neural networks, while Co-teaching does (i.e., leveraging small-loss trick).

However, with the increase of training epochs, two networks trained by Co-teaching will converge to a consensus, and Co-teaching will reduce to the self-training MentorNet. This brings us to think how to address the consensus issue in Co-teaching. Although Decoupling algorithm (Malach & Shalev-Shwartz, 2017) (i.e., "Update by Disagreement") itself *cannot* combat with noisy labels effectively, which has been empirically justified in Section 3, we clearly realize that the "Disagreement" strategy can always keep two networks diverged. Such divergence effects can boost the performance of Co-teaching and bring us Co-teaching+, since the better ensemble effects require to keep diverged more between two classifiers due to Co-training.

To sum up, there are three key factors that can contribute to effectively handle noisy labels (first column of Table 1). First, we should leverage the memorization effects of deep networks (i.e., the small-loss trick). Second, we should train two deep networks simultaneously, and cross update their parameters. Last but not least, we should keep two deep networks diverged during the whole training epochs.

3. Experiments on Simulated Noisy Datasets

3.1. Experimental setup

Datasets. First, we verify the efficacy of our approach on four benchmark datasets (Table 2), including three vision datasets (i.e., *MNIST*, *CIFAR-10*, and *CIFAR-100*) and one text dataset (i.e., *NEWS*). Then, we verify our approach on a larger and harder dataset called *Tiny-ImageNet* (abbreviated as *T-ImageNet*) [1]. These datasets are popularly used for the evaluation of learning with noisy labels in the literature (Reed et al., 2015; Goldberger & Ben-Reuven, 2017; Kiryo et al., 2017).

Since all datasets are clean, following (Reed et al., 2015; Patrini et al., 2017), we need to corrupt these datasets manually by the label transition matrix Q, where $Q_{ij} = \Pr(\tilde{y} = j | y = i)$ given that noisy $\tilde{y}$ is flipped from clean y. Assume that the matrix Q has two representative structures: (1) Symmetry flipping (van Rooyen et al., 2015); (2) Pair flipping (Han et al., 2018b): a simulation of fine-grained classification with noisy labels, where labelers may make mistakes only within very similar classes.

Baselines. We compare Co-teaching+ (Algorithm 1) with the following state-of-art approaches, and implement all methods with default parameters by PyTorch, and conduct all the experiments on a NVIDIA Titan Xp GPU.

(i). MentorNet (Jiang et al., 2018). An extra teacher network is pre-trained and then used to filter out noisy instances for its student network to learn robustly under noisy labels. Then, student network is used for classification. We used self-paced MentorNet in this paper;

(ii). Co-teaching (Han et al., 2018b), which trains two networks simultaneously and cross-updates parameters of peer networks. This method can deal with a large number of classes and is more robust to extremely noisy labels;

(iii). Decoupling (Malach & Shalev-Shwartz, 2017), which

[1] https://tiny-imagenet.herokuapp.com/

Table 1. Comparison of state-of-the-art and related techniques with our Co-teaching+ approach. In the first column, "small loss": regarding small-loss samples as "clean" samples, which is based on the memorization effects of deep neural networks; "double classifiers": training two classifiers simultaneously; "cross update": updating parameters in a cross manner instead of a parallel manner; "divergence": keeping two classifiers diverged during the whole training epochs.

	MentorNet	Co-training	Co-teaching	Decoupling	Co-teaching+
small loss	✓	✗	✓	✗	✓
double classifiers	✗	✓	✓	✓	✓
cross update	✗	✓	✓	✗	✓
divergence	✗	✓	✗	✓	✓

Table 2. Summary of data sets used in the experiments.

	# of train	# of test	# of class	size
MNIST	60,000	10,000	10	28×28
CIFAR-10	50,000	10,000	10	32×32
CIFAR-100	50,000	10,000	100	32×32
NEWS	11,314	7,532	7	1000-D
T-ImageNet	100, 000	10, 000	200	64×64

updates the parameters only using the instances which have different prediction from two classifiers.

(iv). F-correction (Patrini et al., 2017), which corrects the prediction by the label transition matrix. As suggested by the authors, we first train a standard network to estimate the transition matrix Q.

(v). As a simple baseline, we compare Co-teaching+ with the standard deep network that directly trains on noisy datasets (abbreviated as Standard).

Network structure. For *MNIST*, we use a 2-layer MLP. For *CIFAR-10*, we use a network architecture with 2 convolutional layers and 3 fully connected layers. For *CIFAR-100*, the 7-layer network architecture in our paper follows (Wang et al., 2018). For *NEWS*, we borrowed the pre-trained word embeddings from GloVe (Pennington et al., 2014), and a 3-layer MLP is used with Softsign active function. For *T-ImageNet*, we use a 18-layer Preact ResNet (He et al., 2016). The network structure here is standard test bed for weakly-supervised learning, and the details are in Table 3.

Optimizer. Adam optimizer (momentum=0.9) 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. As deep networks are highly nonconvex, even with the same network and optimization method, different initializations can lead to different local optimal. Thus, following (Malach & Shalev-Shwartz, 2017), we also take two networks with the same architecture but different initializations as two classifiers.

Initialization. Assume that the noise rate τ is known. To conduct a fair comparison in benchmark datasets, we set the ratio of small-loss samples $\lambda(e)$ as identical as Co-teaching:

$$\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, \tau\}, \tag{2}$$

where $E_k = 10$.

If τ is not known in advanced, τ can be inferred using validation sets (Liu & Tao, 2016; Yu et al., 2018). Note that $\lambda(e)$ only depends on the memorization effect of deep networks but not any specific datasets.

Measurement. To measure the performance, we use the test accuracy, i.e., *test accuracy = (# of correct predictions) / (# of test dataset)*. Intuitively, higher test accuracy means that the algorithm is more robust to the label noise.

3.2. Comparison with the State-of-the-Arts

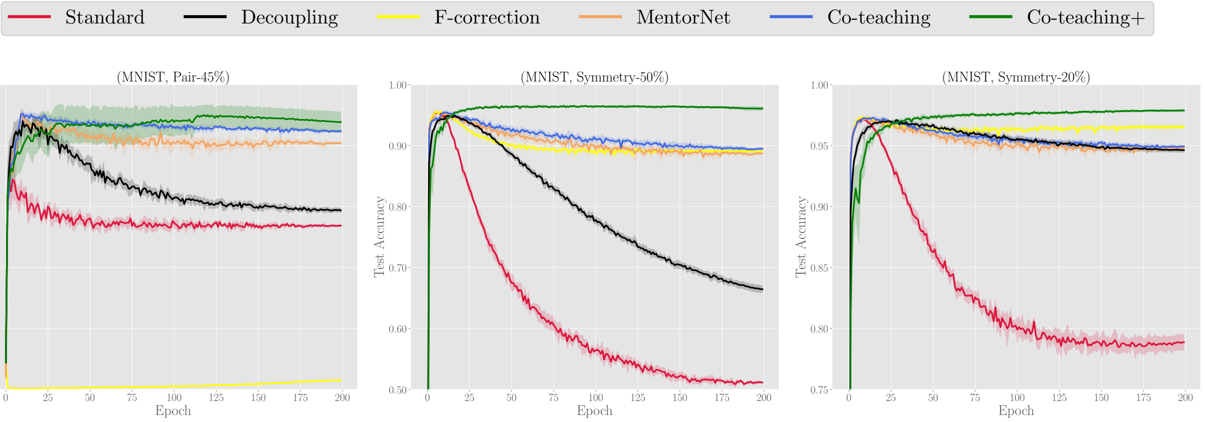
Results on *MNIST*. Figure 3 shows test accuracy vs. number of epochs on *MNIST*. In all three plots, we can clearly see the memorization effects of deep networks. For example, test accuracy of Standard first reaches a very high level since deep network will first fit clean labels. Over the increase of epochs, deep network will over-fit noisy labels gradually, which decreases its test accuracy accordingly. Thus, a robust training method should alleviate or even stop the decreasing trend in test accuracy.

In the easiest Symmetry-20% case, all new approaches work better than Standard obviously, which demonstrates their robustness. Co-teaching+ and F-correction work significantly better than Co-teaching, MentorNet and Decoupling. However, F-correction cannot combat with the other two harder cases, i.e., Pair-45% and Symmetry-50%. Especially in the hardest Pair-45% case, F-correction can learn nothing at all, which greatly restricts its practical usage in the wild. Besides, in two such cases, Co-teaching+ achieves higher accuracy than Co-teaching and MentorNet.

Results on *CIFAR-10*. Figure 4 shows test accuracy vs. number of epochs on *CIFAR-10*. Similarly, we can clearly see the memorization effects of deep networks, namely test accuracy of Standard first reaches a very high level then decreases gradually. In the easiest Symmetry-20% case, Co-teaching+ works much better than all other baselines,

Table 3. MLP and CNN models used in our experiments on *MNIST*, *CIFAR-10*, *CIFAR-100*/*Open-sets*, and *NEWS*.

MLP on *MNIST*	CNN on *CIFAR-10*	CNN on *CIFAR-100*/*Open-sets*	MLP on *NEWS*
28×28 Gray Image	32×32 RGB Image	32×32 RGB Image	1000-D Text
Dense 28×28 → 256, ReLU	5×5 Conv, 6 ReLU 2×2 Max-pool	3×3 Conv, 64 BN, ReLU 3×3 Conv, 64 BN, ReLU 2×2 Max-pool	300-D Embedding Flatten → 1000×300 Adaptive avg-pool → 16×300
	5×5 Conv, 16 ReLU 2×2 Max-pool	3×3 Conv, 128 BN, ReLU 3×3 Conv, 128 BN, ReLU 2×2 Max-pool	Dense 16×300 → 4×300 BN, Softsign
	Dense 16×5×5 → 120, ReLU Dense 120 → 84, ReLU	3×3 Conv, 196 BN, ReLU 3×3 Conv, 196 BN, ReLU 2×2 Max-pool	Dense 4×300 → 300 BN, Softsign
Dense 256 → 10	Dense 84 → 10	Dense 256 → 100/10	Dense 300 → 7



(a) Pair-45%.

(b) Symmetry-50%.

(c) Symmetry-20%.

Figure 3. Test accuracy vs. number of epochs on *MNIST* dataset.

where F-correction works similar to MentorNet but a bit worse than Co-teaching.

However, F-correction cannot combat with two harder cases easily, i.e., Pair-45% and Symmetry-50%. In the Symmetry-50% case, F-correction works better than Standard and Decoupling, but worse than Co-teaching and Co-teaching+. In the hardest Pair-45% case, F-correction almost learns nothing. In such two harder cases, our Co-teaching+ consistently achieves higher accuracy than Co-teaching and MentorNet.

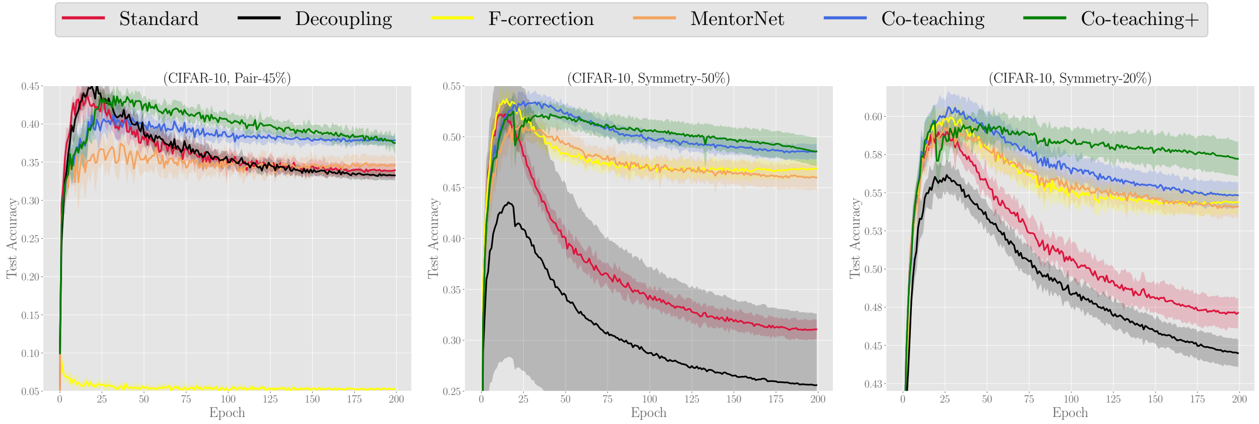
Results on *CIFAR-100*. Figure 5 shows test accuracy vs. number of epochs on *CIFAR-100*. Similarly, we can clearly see the memorization effects of deep networks, namely test accuracy of Standard first reaches a very high level then decreases gradually. In the easiest Symmetry-20% case, Co-teaching+ and F-correction work significantly better than Co-teaching, MentorNet and Decoupling.

However, F-correction cannot combat with two harder cases easily, i.e., Pair-45% and Symmetry-50%. In the Symmetry-50% case, F-correction works better than Standard and Decoupling, but worse than the other three approaches. In the hardest Pair-45% case, F-correction almost learns nothing. In such two harder cases, our Co-teaching+ consistently achieves higher accuracy than Co-teaching and MentorNet. An interesting phenomenon is, in the easiest case, Co-teaching+ not only fully stop the decreasing trend in test accuracy, but also performs better and better with the increase of epochs.

Results on *NEWS*. To verify Co-teaching+ comprehensively, we conduct experiments not only on vision datasets, but also on text dataset *NEWS*. Figure 6 shows test accuracy vs. number of epochs on *NEWS*.

Similar to results on vision datasets, we can still see the memorization effects of deep networks in all three plots, i.e., test accuracy of Standard first reaches a very high level and then gradually decreases. However, Co-teaching+ mitigates such memorization issue, and works much better than others across three cases. Note that F-correction cannot combat with all three cases, even in the easiest Symmetry-20% case. This interesting phenomenon in F-correction does not occur in vision datasets.

Results on *T-ImageNet*. To verify our approach on a complex scenario, Table 4 shows averaged/maximal test accuracy on *T-ImageNet* over last 10 epochs. As we can see, for both Symmetry cases, Co-teaching+ is the best. For the Pair case, Co-teaching and Co-teaching+ outperform

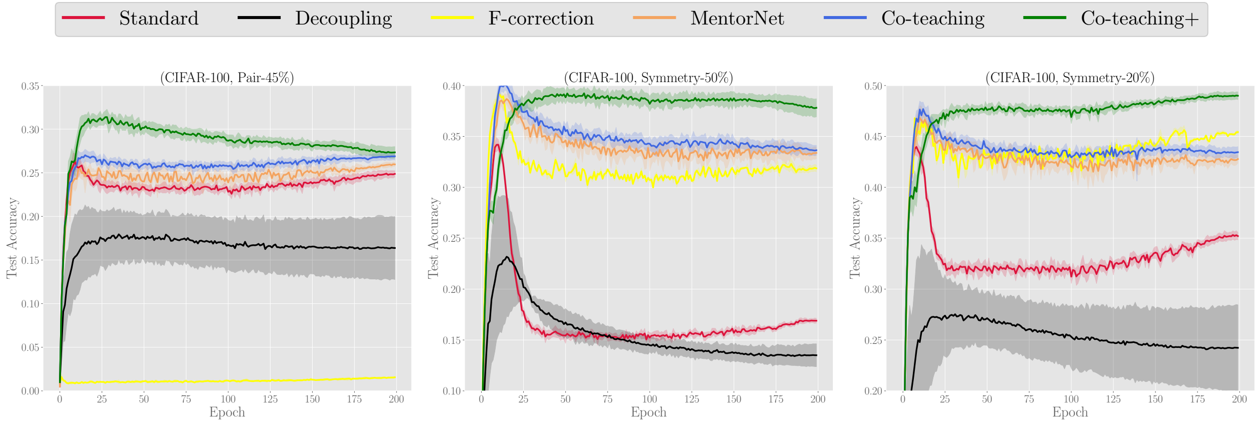


(a) Pair-45%.

(b) Symmetry-50%.

(c) Symmetry-20%.

Figure 4. Test accuracy vs. number of epochs on *CIFAR-10* dataset.



(a) Pair-45%.

(b) Symmetry-50%.

(c) Symmetry-20%.

Figure 5. Test accuracy vs. number of epochs on *CIFAR-100* dataset.

other four baselines.

4. Experiments on Real-world Noisy Datasets

4.1. Experimental setup

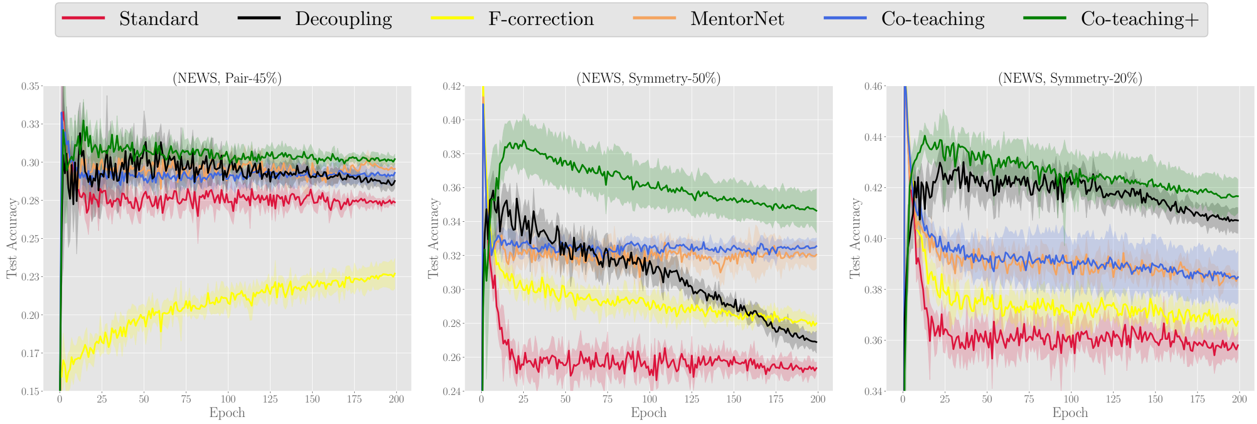
Dataset. To verify the efficacy of our approach in real-world scenario, we conduct experiments on open-set noisy datasets (abbreviated as *Open-sets*) (Wang et al., 2018). Specifically, *Open-sets* are built by replacing some training images in *CIFAR-10* by outside images, while keeping the labels and the number of images per class unchanged. The "mislabeled" images come from different outside datasets, including *CIFAR-100*, *ImageNet-32* (32 × 32 ImageNet images) and *SVHN*. Note that outside images whose labels exclude 10 classes in *CIFAR-10* are considered.

Network & Optimizer & Initialization. We follow the experimental settings in (Wang et al., 2018). Specifically, we use a network architecture with 6 convolutional layers and 1 fully-connected layer, and its details can be found in the third column of Table 3. Batch normalization (BN) is applied in each convolutional layer before the ReLU activation, and a max-pooling layer is implemented every two convolutional layers. All networks are trained by Stochastic Gradient Descent (SGD) with learning rate 0.01, weight decay 10^{-4} and momentum 0.9, and the learning rate is divided by 10 after 40 and 80 epochs (100 in total).

Note that *Open-sets* are real-world noisy datasets. To handle these complex scenarios, we should set the ratio of small-loss samples $\lambda(e)$ as follows.

$$\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, (1 + \frac{e - E_k}{E_{\max} - E_k})\tau\}, \quad (3)$$

where $E_k = 10$ and $E_{\max} = 200$.



(a) Pair-45%.

(b) Symmetry-50%.

(c) Symmetry-20%.

Figure 6. Test accuracy vs. number of epochs on *NEWS* dataset.

Table 4. Averaged/maximal test accuracy (%) of different approaches on *T-ImageNet* over last 10 epochs. The best results are in bold.

Flipping-Rate(%)	Standard	Decoupling	F-correction	MentorNet	Co-teaching	Co-teaching+
Pair-45%	26.14/26.32	26.10/26.61	0.63/0.67	26.22/26.61	27.41/**27.82**	26.54/26.87
Symmetry-50%	19.58/19.77	22.61/22.81	32.84/33.12	35.47/35.76	37.09/37.60	41.19/**41.77**
Symmetry-20%	35.56/35.80	36.28/36.97	44.37/44.50	45.49/45.74	45.60/46.36	47.73/**48.20**

Table 5. Averaged/maximal test accuracy (%) of different approaches on *Open-sets* over last 10 epochs. The best results are in bold.

Open-set noise	Standard	MentorNet	Iterative (Wang et al., 2018)	Co-teaching	Co-teaching+
CIFAR-10+CIFAR-100	62.92	79.27/79.33	79.28	79.43/79.58	79.28/**79.74**
CIFAR-10+ImageNet-32	58.63	79.27/79.40	79.38	79.42/79.60	79.89/**80.52**
CIFAR-10+SVHN	56.44	79.72/79.81	77.73	80.12/80.33	80.62/**80.95**

4.2. Comparison with the State-of-the-Arts

Results on three *Open-sets*. Following (Wang et al., 2018), we report the classification accuracy on *CIFAR-10* noisy datasets with 40% open-set noise in Table 5. The Standard and Iterative results are borrowed from (Wang et al., 2018). For MentorNet, Co-teaching and Co-teaching+, we report the averaged/maximal test accuracy over the last 10 epochs. As can be seen, our approach outperforms other baselines on all three open-set noisy datasets. For *CIFAR-100* noise and *ImageNet-32* noise, both Co-teaching and Co-teaching+ are better than Iterative. For *SVHN* noise, Co-teaching+ is significantly better than Iterative; while MentorNet and Co-teaching also work better than Iterative.

Reflection of results. Different algorithm designs lead to different results. To sum up, self-paced MentorNet is concluded as training single deep network using the small-loss trick. Co-teaching moves further step, which is viewed as cross-training double deep networks using the small-loss trick. Based on Co-teaching, Co-teaching+ is regarded as cross-training double *diverged* deep networks using the small-loss trick. Thus, keeping two deep networks diverged is one of the key ingredients to train robust deep networks. This point has been empirically verified by the result difference between Co-teaching and Co-teaching+.

5. Conclusion

This paper presents a robust learning paradigm called Co-teaching+, which trains deep neural networks robustly under noisy supervision. Our key idea is to maintain two networks simultaneously that find the prediction disagreement data. Among such disagreement data, our method cross-trains on data screened by the “small loss” criteria. We conduct experiments to demonstrate that, our proposed Co-teaching+ can train deep models robustly with the extremely noisy supervision beyond Co-teaching and MentorNet. More importantly, we summarize three key points towards training robust deep networks with noisy labels: (1) using small-loss trick based on memorization effects of deep networks; (2) cross-updating parameters of two networks; and (3) keeping two deep networks diverged during the whole training epochs. In future, we will investigate the theory of Co-teaching+ from the view of disagreement-based algorithms (Wang & Zhou, 2017).

Acknowledgments

MS was supported by JST CREST JPMJCR18A2. IWT was supported by ARC FT130100746, DP180100106 and LP150100671. XRY was supported by China Scholarship Council No. 201806450045. We gratefully acknowledge the support of NVIDIA Corporation with the donation of Titan Xp GPU used for this research.

References

Aït-Sahalia, Y., Fan, J., and Xiu, D. High-frequency covariance estimates with noisy and asynchronous financial data. *Journal of the American Statistical Association*, 105(492):1504–1517, 2010.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M., Maharaj, T., Fischer, A., Courville, A., and Bengio, Y. A closer look at memorization in deep networks. In *ICML*, 2017.

Bengio, Y. Evolving culture versus local minima. In *Growing Adaptive Machines*, pp. 109–138. 2014.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3): 542–542, 2009.

Dgani, Y., Greenspan, H., and Goldberger, J. Training a neural network based on unreliable human annotation of medical images. In *ISBI*, 2018.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018a.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *ECCV*, pp. 630–645. Springer, 2016.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018.

Jiang, L., Zhou, Z., Leung, T., Li, L., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

Kiryo, R., Niu, G., Du Plessis, M., and Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, 2017.

Lee, K., Yun, S., Lee, K., Lee, H., Li, B., and Shin, J. Robust inference via generative classifiers for handling noisy labels. In *ICML*, 2019.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, J. Learning from noisy labels with distillation. In *ICCV*, 2017.

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

Liu, W., Jiang, Y., Luo, J., and Chang, S. Noise resistant graph ranking for improved web image search. In *CVPR*, 2011.

Ma, X., Wang, Y., Houle, M., Zhou, S., Erfani, S., Xia, S., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *ICML*, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling" when to update" from" how to update". In *NeurIPS*, 2017.

Masnadi-Shirazi, H. and Vasconcelos, N. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *NeurIPS*, 2009.

Menon, A., Van Rooyen, B., Ong, C., and Williamson, B. Learning from corrupted binary labels via class-probability estimation. In *ICML*, 2015.

Miyato, T., Dai, A., and Goodfellow, I. Virtual adversarial training for semi-supervised text classification. In *ICLR*, 2016.

Natarajan, N., Dhillon, I., Ravikumar, P., and Tewari, A. Learning with noisy labels. In *NeurIPS*, 2013.

Patrini, G., Rozza, A., Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *EMNLP*, 2014.

Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Rodrigues, F. and Pereira, F. Deep learning from crowds. In *AAAI*, 2018.

Sanderson, T. and Scott, C. Class proportion estimation with application to multiclass anomaly rejection. In *AISTATS*, 2014.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

van Rooyen, B., Menon, A., and Williamson, B. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*, 2015.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 2017.

Wang, W. and Zhou, Z.-H. Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*, 2017.

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S. Iterative learning with open-set noisy labels. In *CVPR*, 2018.

Welinder, P., Branson, S., Perona, P., and Belongie, S. The multidimensional wisdom of crowds. In *NeurIPS*, 2010.

Yan, Y., Rosales, R., Fung, G., Subramanian, R., and Dy, J. Learning from multiple annotators with varying expertise. *Machine Learning*, 95(3):291–327, 2014.

Yu, X., Liu, T., Gong, M., Batmanghelich, K., and Tao, D. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *CVPR*, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

A. Related literature

Statistical learning methods. Statistical learning contributed a lot to the problem of learning with noisy labels, especially in theoretical aspects. Statistical learning approaches can be categorized into three strands: surrogate loss, noise rate estimation and probabilistic modeling. For example, in the surrogate losses category, Natarajan et al. (2013) proposed an unbiased estimator to provide the noise corrected loss approach. Masnadi-Shirazi & Vasconcelos (2009) presented a robust non-convex loss, which is the special case in a family of robust losses. In the noise rate estimation category, both Menon et al. (2015) and Liu & Tao (2016) proposed a class-probability estimator using order statistics on the range of scores. Sanderson & Scott (2014) presented the same estimator using the slope of the ROC curve. In the probabilistic modeling category, Raykar et al. (2010) proposed a two-coin model to handle noisy labels from multiple annotators. Yan et al. (2014) extended this two-coin model by setting the dynamic flipping probability associated with instances.

Deep learning approaches. Deep learning approaches are prevalent to handle noisy labels (Zhang & Sabuncu, 2018). Li et al. (2017) proposed a unified framework to distill the knowledge from clean labels and knowledge graph, which can be exploited to learn a better model from noisy labels. Veit et al. (2017) trained a label cleaning network by a small set of clean labels, and used this network to reduce the noise in large-scale noisy labels. Rodrigues & Pereira (2018) added a crowd layer after the output layer for noisy labels from multiple annotators. Tanaka et al. (2018) presented a joint optimization framework to learn parameters and estimate true labels simultaneously. Ren et al. (2018) leveraged an additional validation set to adaptively assign weights to training examples. Similarly, based on a small set of trusted data with clean labels, Hendrycks et al. (2018) proposed a loss correction approach to mitigate the effects of label noise on deep neural network classifiers. Ma et al. (2018) developed a new dimensionality-driven learning strategy, which monitors the dimensionality of deep representation subspaces during training and adapts the loss function accordingly. Wang et al. (2018) proposed an iterative learning framework for training CNNs on datasets with open-set noisy labels. Han et al. (2018a) proposed a human-assisted approach that conveys human cognition of invalid class transitions, and derived a structure-aware deep probabilistic model incorporating a speculated structure prior. Lee et al. (2019) proposed a novel inference method to obtain a robust decision boundary under any softmax neural classifier pre-trained on noisy datasets. Their idea is to induce a generative classifier on top of hidden feature spaces of the discriminative deep model.

B. Training details

For *MNIST* and *NEWS*, we train Co-teaching+ by default at the beginning of training. For other datasets, we use a warm-up strategy to achieve a higher test accuracy. Specifically, for *CIFAR-10*, we warm-up Co-teaching+ with training Co-teaching for the first 20 epochs (i.e., only conducting cross-update for the first 20 epochs). For *CIFAR-100*, we warm-up Co-teaching+ with training Co-teaching for the first 5 epochs. For *T-ImageNet*, we start disagreement-update in the middle of training, i.e., we warm-up Co-teaching+ with training Co-teaching for the first 100 epochs. For *Open-sets*, we warm-up Co-teaching+ with training two networks in parallel for the first 55 epochs, where both networks leverage the small-loss trick. Inevitably, there is few chance that we cannot find enough small-loss instances for cross-update. In that case, we only conduct disagreement-update in a mini-batch data during training.