# Co-variance: Tackling Noisy Labels with Sample Selection by Emphasizing High-variance Examples

**Anonymous authors**
Paper under double-blind review

## Abstract

The *sample selection* approach is popular in learning with noisy labels, which tends to select potentially clean data out of noisy data for robust training. 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 during training. Typically, the divergence is kept by first locating the *disagreement data* on which the prediction labels of two networks are different, and then selecting clean data out of such data. However, this procedure is *sample-inefficient* for network weight updates, which means that a few clean examples can be utilized in training. In this paper, to address the issues, we propose a simple yet effective method called Co-variance. In particular, we select possibly clean data that simultaneously have *high-variance prediction probabilities* between two networks. As selected data have high variances, the divergence of two networks can be maintained by training on such data. Additionally, the condition of high variances is *milder* than the condition of disagreement in sample selection, 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 enough hard clean examples to help generalization. A series of empirical results show that Co-variance is superior to multiple baselines in the robustness of trained models, especially on class-imbalanced and real-world noisy datasets.

## 1 Introduction

Learning with noisy labels can be dated back to more than three decades ago (Angluin & Laird, 1988), and still is one of the hottest problems in weakly supervised learning (Northcutt et al., 2021). The reason is that, in our daily life, noisy labels are *unavoidable* such as crowdsourcing (Welinder & Perona, 2010) and web queries (Liu et al., 2011). 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 (Yao et al., 2020a). Unfortunately, general-purpose regularization such as *dropout* and *weight decay* cannot address this issue well (Zhang et al., 2017).

Fortunately, even though deep networks can fit anything given for training eventually, they *learn patterns first* (Arpit et al., 2017): for learning with noisy labels, this suggests that deep networks can *gradually memorize the data*, moving from clean data to mislabeled data. Besides, this phenomenon does not change with the choice of training optimizations or network architectures (Zhang et al., 2017). The *sample selection* approach therefore was proposed to handle noisy labels (Jiang et al., 2018; Han et al., 2018; Wang et al., 2018), 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.

The sample selection procedure can be executed in a *self-teaching* manner (Jiang et al., 2018). By using a predefined curriculum, *e.g.*, regarding training data with small losses to be clean, the deep network can select such data by itself and then use them for network weights updates. Nevertheless, the idea of self-teaching sample selection is argued to have the inferiority of *accumulated errors* caused by the sample-selection bias (Han et al., 2018). To relieve this issue, some advanced algorithms were proposed, which maintain two deep networks, working in a cooperative manner (Yao et al., 2020a). 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 to its peer network for updates, the error flows coming from the biased selection, can be reduced by peer networks mutually (Han et al., 2018).

To keep such different learning abilities of two networks for handling noisy labels, prior work (Yu et al., 2019) 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 noise rate is large (Wei et al., 2020). 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.

In this paper, to handle the above problem, a robust learning paradigm called Co-variance is proposed. Specifically, we inherit the property that deep networks learn patterns first for sample selection, as did in (Han et al., 2018). Meanwhile, the training examples with *high variances* 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 variance 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 good *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 variances). Additionally, the condition of high variances in sample selection is *milder* than the condition of disagreement. In other words, the prediction disagreement data must have high-variance prediction probabilities, but the data with high-variances 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 mentioned procedure for sample selection (Yu et al., 2019), our procedure is more *sample-efficient* for network weight updates. Furthermore, the examples with high variances in training are probable to be hard examples (Gao et al., 2015), which play an important role in shaping the decision boundary. Shared with a similar philosophy, the proposed method emphasizes high-variance examples and therefore enables to mine hard clean examples that are critical for generalization. Benefiting from maintaining two networks simultaneously, the variance 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.

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 the Co-variance method can well combat noisy labels. Specifically, on class-balanced noisy datasets, our method is superior to many state-of-the-art methods. On class-imbalanced noisy datasets, our method can outperform comparison methods by more than 5% of test accuracy. Lastly, on real-world noisy datasets, Co-variance also achieves the best performance and can be exploited to improve the cutting edge performance of state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we set up the problem and review the sample selection approach in learning with noisy labels. In Section 3, we present our method and discuss how to select clean examples with an emphasis on high-variance examples. Experimental results are provided in Section 4. Conclusions are given in Section 5.

## 2 Preliminaries

### 2.1 Notations and Problem Statement

In the sequel, scalars are in lowercase letters, vectors are in lowercase boldface letters, and matrices are in uppercase boldface letters. We use $\|\cdot\|_p$ as the $\ell_p$ norm of vectors or matrices and $\mathrm{KL}(\cdot||\cdot)$ as the Kullback-Leibler (KL) divergence (Mohri et al., 2018) between two probability distributions. For a function $g$, we use $\nabla g$ 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] = $\{1, 2, \ldots, n\}$.

We consider a multi-classification problem with $c$ classes in total. 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 i.i.d. 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 exploiting a noisy training sample $\tilde{\mathcal{D}}$.

### 2.2 Learning with Sample Selection

In this subsection, we formally introduce the *sample selection* approach applied in learning with noisy labels. Specifically, with the assumption that clean labels are of the majority in a noisy class, 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 (Han et al., 2018; Yu et al., 2019; Huang et al., 2019; Lyu & Tsang, 2020; Wei et al., 2020). Also, the examples that have large classification margins (Pleiss et al., 2020), minimize the determinant value of the corresponding sample covariance matrix (Lee et al., 2019), or minimize the average gradient dissimilarity to all the other examples (Mirzasoleiman et al., 2020), can be seen as clean examples and then be used for network parameter updates.

In this paper, we focus on the procedure of using the *small-loss* criterion for sample selection, which is most commonly used in learning with noisy labels and shared by the state-of-the-art methods. We first present the procedure that only uses a *single* network/classifier, *i.e.,* the self-teaching MentorNet (Jiang et al., 2018), which is shown in Algorithm 1. The procedure is straightforward. Let $f$ be the classifier with learnable parameters. At the $i$-th iteration, when a mini-batch data $\bar{\mathcal{D}}$ is formed (Step 2), we select a subset of small-loss examples $\bar{\mathcal{D}}_f$ (Step 3) for classifier parameter updates (Step 4). To the end of the training, we can obtain a robust classifier since we select less noisy examples for updates. It is intuitive for using a *single network* to select clean examples for robust training. 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 subsequent iterations. Although Co-teaching (Han et al., 2018) trains two networks simultaneously 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.

**Algorithm 1** Sample selection with the small-loss criterion in learning with noisy labels.

```
1: Input: initialized classifier $f$ and the maximum number of iterations $i_{\max}$.
for $t = 1, \ldots, t_{\max}$ do
    2: Draw a mini-batch $\bar{\mathcal{D}}$ from $\tilde{\mathcal{D}}$;
    3: Select small-loss examples $\bar{\mathcal{D}}_f$ from $\bar{\mathcal{D}}$;
    4: Update classifier parameters using $\bar{\mathcal{D}}_f$;
end
5: Output: trained classifier $f$.
```

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 (Malach & Shalev-Shwartz, 2017) conducts updates only on selected data with prediction disagreement between two networks. Co-teaching+ (Yu et al., 2019) 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 (Wei et al., 2020) starts with a new perspective named "Update by Agreement", which is motivated by Co-training (Blum & Mitchell, 1998) 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

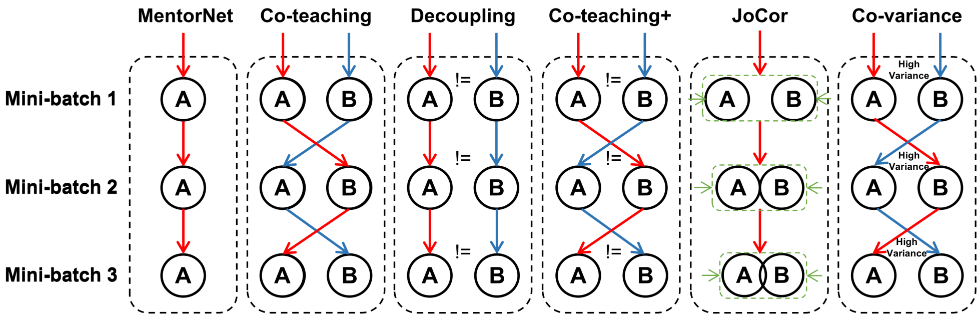


Figure 1: Comparison of error flows among MentorNet (the self-teaching version), Co-teaching, Decoupling, Co-teaching+, JoCor, and Co-variance (ours). Assume that error flows comes from the biased selection of training examples, and the error flow from network A or B is denoted by red arrows or blue arrows, respectively. **The 1st panel**: MentorNet maintains only one network A. **The 2nd panel**: Co-teaching maintains two networks (A & B). In each mini-batch data, each network selects its small-loss data to teach its peer network for robust training. **The 3rd panel**: Decoupling updates the two networks with *prediction-disagreed* (**!=**) examples from a mini-batch. **The 4th panel**: In Co-teaching+, each network selects its small-loss instances *within prediction disagreement* (**!=**) to teach its peer network. **The 5th panel**: JoCor trains two networks as a whole with a joint loss, which makes predictions of each network closer to peer network's. **The 6th panel**: Co-variance also maintains two networks (A & B). In each mini-batch data, each network selects its small-loss data that meanwhile have *high variances* among two networks, to teach its peer network.

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 (Chang et al., 2017). The experimental results in Section 4.3 will highlight the vulnerability of JoCor. A comparison between the related methods on sample selection and our method Co-variance is illustrated in Figure 1. Additionally, other deep methods for learning with noisy labels are summarized in Appendix B.

## 3 Method

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 (Goodfellow et al., 2016) by $S(\cdot)$, we have $\mathbf{p}_1(\mathbf{x}_i) = S(f(\mathbf{x}_i; \mathbf{w}_1))$ and $\mathbf{p}_2(\mathbf{x}_i) = S(f(\mathbf{x}_i; \mathbf{w}_2))$.

**Classification loss.** For multi-class classification, we exploit the *cross-entropy* loss $\ell_{\text{CE}}$ to minimize the distance between predictions and given labels. For a 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

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

As deep networks learn patterns first (Arpit et al., 2017), 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 (Han et al., 2018; Yu et al., 2019).

**Contrastive 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, which is *continuous* and *bounded* like the cross entropy loss. We formulate contrastive loss in the following:

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

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

**Sample selection criterion.** As discussed, we tend to select possibly clean examples based on the small-loss criterion and involve high-variance 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:

$$\mathcal{L}_{\text{JOINT}} = \mathcal{L}_{\text{CLASS}} + \alpha * \mathcal{L}_{\text{CONTRAST}}, \tag{3}$$

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 (Arpit et al., 2017; Zhang et al., 2017; Han et al., 2018; Jiang et al., 2018). A larger contrastive loss means that we select the possibly clean examples but with a high divergence between two networks, which could be hard clean examples for generalization. Then the selected examples are used for robust training. To determine the value of $\alpha$, if we have a small trusted and unbiased dataset, we can choose a suitable $\alpha$ with meta learning (Shu et al., 2019; 2020). However, it may be somewhat strong to have such a small dataset in practice. Therefore, we choose $\alpha$ with a noisy validation set as did in (Patrini et al., 2017; Chen et al., 2019; Nguyen et al., 2020). In fact, the proposed sample selection criterion is stable with the change of $\alpha$. We present detailed analyses and discussions for algorithm stability. More details are presented in Section 4.

**Algorithm 2** Co-variance Algorithm.

```
1: Input: two networks with weights $\mathbf{w}_1$ and $\mathbf{w}_2$, learning rate $\eta$,
fixed $\tau$, 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 $\bar{\mathcal{D}}_1 = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}_{\text{JOINT}}(\mathbf{w}_1, \mathcal{D}')$;
        5: Obtain $\bar{\mathcal{D}}_2 = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}_{\text{JOINT}}(\mathbf{w}_2, \mathcal{D}')$;
        6: Update $\mathbf{w}_1 = \mathbf{w}_1 - \eta\nabla\mathcal{L}_{\text{CLASS}}(\mathbf{w}_1, \bar{\mathcal{D}}_2)$;
        7: Update $\mathbf{w}_2 = \mathbf{w}_2 - \eta\nabla\mathcal{L}_{\text{CLASS}}(\mathbf{w}_2, \bar{\mathcal{D}}_1)$;
    end
    8: Update $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
9: Output: $\mathbf{w}_1$ and $\mathbf{w}_2$.
```

**Network weight updates.** We maintain two networks simultaneously. The *cross-update* strategy is used, in which the intuition comes from culture evolving hypothesis (Bengio, 2014). 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, we only use the classification loss for weight updates. The divergence of the two networks can be maintained implicitly because of our sample selection criterion.

The overall procedure of the proposed method is shown in Algorithm 2, which works in a mini-batch manner. After fetching a mini-batch training data (Step 3), each network selects its clean examples with the joint loss (Step 4 and 5). Then the selected examples are used for weight updates of peer network (Step 6 and 7). Following the setting of prior methods (Yu et al., 2019; Wei et al., 2020), we update $R(T)$ (Step 8), which controls how many data should be selected in each training epoch. The value of $R(T)$ should be larger at the beginning of training, and be smaller until $1 - \tau$, when the number of epochs goes large, which aims to prevent deep networks from overfitting noisy labels.

## 4 Experiments

In this section, we first introduce the comparison methods (Section 4.1). The experiments on balanced noisy datasets (Section 4.2) and imbalanced noisy datasets (Section 4.3) are then presented respectively. Finally, the experiments on the real-world noisy datasets are provided (Section 4.4).

### 4.1 Comparison Methods

We compare the proposed method with the state-of-art methods on sample selection: (1). MentorNet (Jiang et al., 2018). We use self-teaching MentorNet in this paper. (2). SIGUA (Han et al., 2020),

| | MentorNet | SIGUA | Co-teaching | Decoupling | Co-teaching+ | JoCor | Co-variance |
|---|---|---|---|---|---|---|---|
| small loss | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| double classifiers | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| cross update | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |
| disagreement | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ |
| agreement | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |

Table 1: Comparison of state-of-the-art and related techniques with our Co-variance method. In the first column, "small loss": regarding small-loss samples as "clean" samples; "double networks": training two networks simultaneously; "cross update": updating parameters in a cross manner instead of a parallel manner; "disagreement": restricting two networks to diverge during training; "agreement": restricting two networks to be converged during training.

which exploits stochastic integrated gradient under-weighted ascent to handle noisy labels. We use self-teaching SIGUA in this paper. (3). Co-teaching (Han et al., 2018). (4). Decoupling (Malach & Shalev-Shwartz, 2017). (5). Co-teaching+ (Yu et al., 2019). (6). JoCor (Wei et al., 2020). The above methods are systematically compared in Table 1. 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 (Ma et al., 2020) and CDR (Xia et al., 2021). APL combines two mutually reinforcing robust loss functions. While, CDR employs unstructured network pruning to enhance the robustness of deep networks.

## 4.2 Experiments on Balanced Noisy Datasets

| | Noise type | Sym. | | Pair. | | Trid. | | Ins. | |
|---|---|---|---|---|---|---|---|---|---|
| | Setting | 20% | 40% | 20% | 40% | 20% | 40% | 20% | 40% |
| *MNIST* | APL | 98.76±0.06 | 94.92±0.31 | **98.66±0.10** | 68.44±2.95 | **98.93±0.04** | 76.44±3.04 | 97.63±0.73 | 87.90±1.94 |
| | CDR | 94.77±0.17 | 92.16±0.73 | 93.25±0.90 | 71.02±3.89 | 94.06±0.92 | 70.28±4.01 | 93.17±0.96 | 77.45±3.04 |
| | MentorNet | 95.04±0.03 | 92.08±0.42 | 93.19±0.17 | 90.93±1.54 | 96.42±0.09 | 93.28±1.37 | 94.65±0.73 | 90.11±1.26 |
| | SIGUA | 92.31±1.10 | 91.88±0.92 | 93.77±1.40 | 86.22±1.75 | 94.92±0.83 | 83.46±2.98 | 92.90±1.82 | 86.34±3.51 |
| | Co-teaching | 97.53±0.12 | 95.62±0.30 | 96.05±0.96 | 94.16±1.37 | 98.05±0.06 | 96.18±0.85 | 97.96±0.09 | 95.02±0.39 |
| | Decoupling | 98.39±0.08 | 81.56±0.72 | 97.82±0.31 | 66.48±0.78 | 98.33±0.11 | 74.55±0.97 | 98.05±0.30 | 71.87±1.24 |
| | Co-teaching+ | 98.25±0.13 | 92.63±0.34 | 97.30±0.16 | 92.00±0.31 | 98.00±0.16 | 93.06±0.24 | 96.83±0.28 | 89.99±0.37 |
| | JoCor | 98.42±0.14 | 98.04±0.07 | 98.01±0.19 | **96.85±0.43** | 98.45±0.17 | 96.98±0.25 | **98.62±0.06** | 96.07±0.31 |
| | Co-variance | **98.80±0.04** | **98.33±0.09** | 98.28±0.12 | 95.39±1.24 | **98.93±0.04** | **97.17±0.14** | 98.40±0.15 | **96.12±0.96** |
| *F-MNIST* | APL | 91.73±0.20 | 89.06±0.41 | 90.22±0.80 | 78.54±4.33 | 90.84±0.22 | 86.53±0.76 | 90.96±0.77 | 85.55±2.86 |
| | CDR | 85.62±0.96 | 71.83±1.37 | 85.72±0.65 | 69.07±2.31 | 86.75±1.19 | 73.63±2.82 | 85.92±1.43 | 73.14±3.12 |
| | MentorNet | 90.37±0.17 | 86.53±0.65 | 87.92±0.18 | 83.70±0.49 | 88.74±0.33 | 85.63±0.59 | 87.52±0.15 | 83.27±1.42 |
| | SIGUA | 87.64±1.29 | 87.23±0.32 | 69.59±5.75 | 68.93±2.80 | 79.97±3.23 | 76.14±4.24 | 79.97±3.23 | 76.14±4.24 |
| | Co-teaching | 91.48±0.10 | 88.80±0.29 | 90.77±0.23 | 86.91±0.71 | 91.24±0.11 | 89.18±0.36 | 90.60±0.12 | 87.90±0.45 |
| | Decoupling | 88.89±0.47 | 70.45±0.62 | 87.03±0.32 | 60.12±0.23 | 88.42±0.37 | 65.98±1.05 | 87.16±0.77 | 63.48±0.88 |
| | Co-teaching+ | 89.95±0.18 | 83.73±0.44 | 88.33±0.45 | 71.76±1.57 | 89.68±0.41 | 79.47±0.92 | 88.64±0.26 | 75.40±2.40 |
| | JoCor | 91.97±0.13 | 89.96±0.19 | 91.52±0.24 | **87.40±0.58** | 92.01±0.17 | **89.42±0.33** | 91.43±0.71 | 87.59±0.94 |
| | Co-variance | **92.21±0.17** | **90.49±0.24** | **91.66±0.31** | 87.07±0.51 | **92.19±0.30** | 88.70±0.94 | **91.48±0.52** | **88.04±0.58** |
| *SVHN* | APL | 89.05±0.43 | 83.51±3.03 | 89.29±1.23 | 68.07±4.98 | 90.88±1.31 | 80.86±2.28 | 90.21±0.52 | 72.75±4.25 |
| | CDR | 83.45±1.23 | 61.99±1.42 | 82.72±0.76 | 59.76±1.06 | 83.42±0.88 | 63.19±1.22 | 82.11±0.27 | 60.05±1.39 |
| | MentorNet | 93.18±0.26 | 92.02±0.24 | 92.78±0.25 | 81.05±0.37 | 92.99±0.16 | 90.16±0.16 | 92.21±0.27 | 87.60±0.79 |
| | SIGUA | 92.31±0.32 | 89.73±0.34 | 75.88±2.43 | 72.21±3.61 | 82.94±2.06 | 78.14±4.25 | 77.29±7.68 | 76.40±3.85 |
| | Co-teaching | 93.61±0.11 | 91.89±0.25 | 93.53±0.20 | 90.37±0.49 | 93.62±0.19 | 90.65±0.43 | 93.13±0.36 | 89.99±0.65 |
| | Decoupling | 88.46±0.19 | 65.22±3.74 | 87.80±0.83 | 63.02±3.28 | 89.04±0.61 | 66.73±0.64 | 87.25±0.93 | 62.06±1.34 |
| | Co-teaching+ | 90.31±0.30 | 87.60±0.54 | 89.85±0.37 | 69.17±1.58 | 90.31±0.31 | 80.15±0.92 | 88.43±0.55 | 70.16±3.00 |
| | JoCor | 93.70±0.20 | 92.16±0.26 | **93.54±0.43** | 90.73±0.17 | **93.74±0.12** | **90.97±0.39** | 93.32±0.42 | 89.37±0.56 |
| | Co-variance | **93.75±0.17** | **92.22±0.42** | **93.54±0.26** | **91.29±0.33** | 93.65±0.14 | 90.75±0.27 | **93.42±1.02** | **90.15±1.29** |
| *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 |
| | Co-variance | 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 |
| | Co-variance | **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: Mean and standard deviations of test accuracy (%) on five 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 mean results are in **bold**.

**Datasets.** We verify the effectiveness of the proposed method on the manually corrupted version of the following datasets: *MNIST* (LeCun et al., 1998), *F-MNIST* (Xiao et al., 2017), *SVHN* (Netzer et al., 2011), *CIFAR-10* (Krizhevsky, 2009), and *NEWS* (Lang, 1995). The five datasets are popularly used in prior works. Note that for *NEWS*, we borrowed the pre-trained word embeddings from GloVe (Pennington et al., 2014). The important statistics of the used synthetic datasets are summarized in Appendix A.1.

**Generating noisy labels.** We consider broad types of noisy labels: Symmetric noise (abbreviated as Sym.), Pairflip noise (abbreviated as Pair.), Tridiagonal noise (abbreviated as Trid.), and Instance-dependent noise (abbreviated as Ins.). For all types of noise, the noise rates are set to 20% and 40% consistently. which aim to ensure that clean labels in noisy classes are *diagonally dominant* (Ma et al., 2020). More details about generating noisy labels are provided in Appendix A.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 therefore be used.

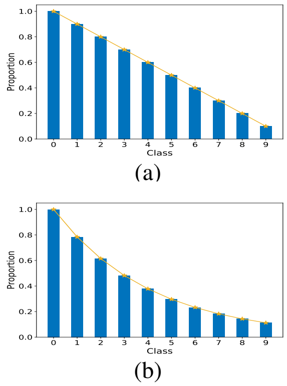
**Implementation and measurement.** For a fair comparison, we implement all methods with default parameters by PyTorch, and conduct all the experiments on a NVIDIA Titan Xp GPU Cluster. For *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*, we employ a 9-layer CNN structure from (Han et al., 2018), which is a standard testbed for weakly supervised learning. For *NEWS*, we use a 3-layer MLP with the Softsign active function. The details of network structures are presented in Appendix A.3. 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. Note that deep networks are highly non-convex, even with the same network and optimization method, different initializations can lead to different local optimal (Malach & Shalev-Shwartz, 2017). Thus, following (Han et al., 2018; Yu et al., 2019), we also take two networks with the same architecture but different initializations as two classifiers. Here, we assume the noise level $\tau$ is known and set $R(T) = 1 - \min\{\frac{T}{T_k}\tau, \tau\}$ with $T_k$=10. If $\tau$ is not known in advance, it can be inferred using validation sets (Liu & Tao, 2016; Yu et al., 2018). To measure the performance, we use the test accuracy, *i.e. test accuracy = (# of correct predictions) / (# of testing)*. Intuitively, the higher test accuracy means that a method is more robust to noisy labels.

**Experimental results.** The results of experiments on balanced noisy datasets are provided in Table 2. 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, *e.g.*, *MNIST* with pairflip noise. Thus, the performance is still competitive.

**Ablation study.** It is easy to analyze the role of the used divergence strategy by comparing our method with Co-teaching. As we employ $\alpha$ to keep divergence of two deep networks, we the algorithm stability with different values of $\alpha$. The experiments are conducted with five datasets with symmetric noise. Implementation details are kept the same as above. The results in Appendix A.5 demonstrate the stability of our method with different $\alpha$. As can be seen, our method is not sensitive to the change of $\alpha$, which is conducive for practical applications.

### 4.3 Experiments on Imbalanced Noisy Datasets

**Datasets and implementation.** We consider two kinds of experimental settings for imbalanced noisy cases, where the examples with *non-dominant* labels are hard examples and are critical for generalization. As discussed, Covariance emphasizes the data with high variances 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 (Patrini et al., 2017). 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%,



(a)

(b)

Figure 2: Illustrations for two types of long-tailed datasets.

| | 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 |
| | Co-variance | **99.55±0.03**<br>(**+0.76**) | **99.42±0.02**<br>(**+1.16**) | **99.18±0.07**<br>(**+4.10**) | **99.01±0.14**<br>(**+5.54**) |
| *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 |
| | Co-variance | **93.12±0.15**<br>(**+1.46**) | **92.11±0.28**<br>(**+2.73**) | **84.10±2.93**<br>(**+1.77**) | **74.30±3.92**<br>(**+6.01**) |
| *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 |
| | Co-variance | **95.38±0.21**<br>(**+0.51**) | **95.10±0.29**<br>(**+1.62**) | **94.62±0.28**<br>(**+3.07**) | **94.00±0.30**<br>(**+5.21**) |
| *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 |
| | Co-variance | **84.78±0.22**<br>(**+3.84**) | **82.70±0.42**<br>(**+1.83**) | **75.24±1.44**<br>(**+2.43**) | **68.80±2.14**<br>(**+1.45**) |

Figure 3: Mean and standard deviations of test accuracy (%) on two closed-set 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 mean results are in **bold**. The improvements over baselines are **highlighted**.

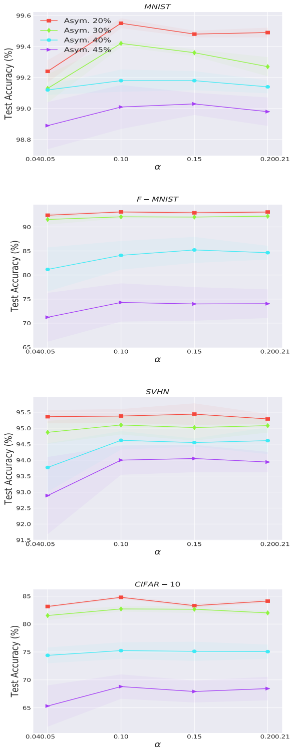


Figure 4: Illustrations of the hyperparameter sensitivity for the proposed method on four imbalanced noisy datasets. The error bar for standard deviation in each figure has been shaded.

40%, and 45% respectively. More details are provided in Appendix A.4. The second one is *long-tailed* noise (abbreviated as L-Tailed.), where training data exhibit long-tailed distributions with class imbalance (Yang & Xu, 2020). 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 2. Taking *MNIST* as an example, the built datasets are called *L-MNIST-1* (Figure 2(a)) and *L-MNIST-2* (Figure 2(b)). 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 3. 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 achieves superior robustness. In addition, the analyses of the hyperparameter sensitivity are provided in Figure 4. The curves demonstrate the importance of divergence in sample selection and the stability of our method. The results on four noisy long-tailed datasets are shown in Figure 5. From all training curves, we can see that Co-variance can achieve superior robustness on long-tailed noisy datasets.

It should be noted that the baseline JoCor is *weak* on imbalanced noisy datasets. Compared with its performance on balanced noisy datasets, we can see that it cannot handle these realistic cases well. Moreover, JoCor is rather *unstable* during training, with *large error bars*. This issue is pessimistic, and could limit practical applications largely.

### 4.4 Experiments on the Real-world Noisy Dataset

**Datasets and implementation.** *Clothing1M* (Xiao et al., 2015) is employed in this paper, which consists of 1M noisy training examples collected from online shopping websites. We follow previous

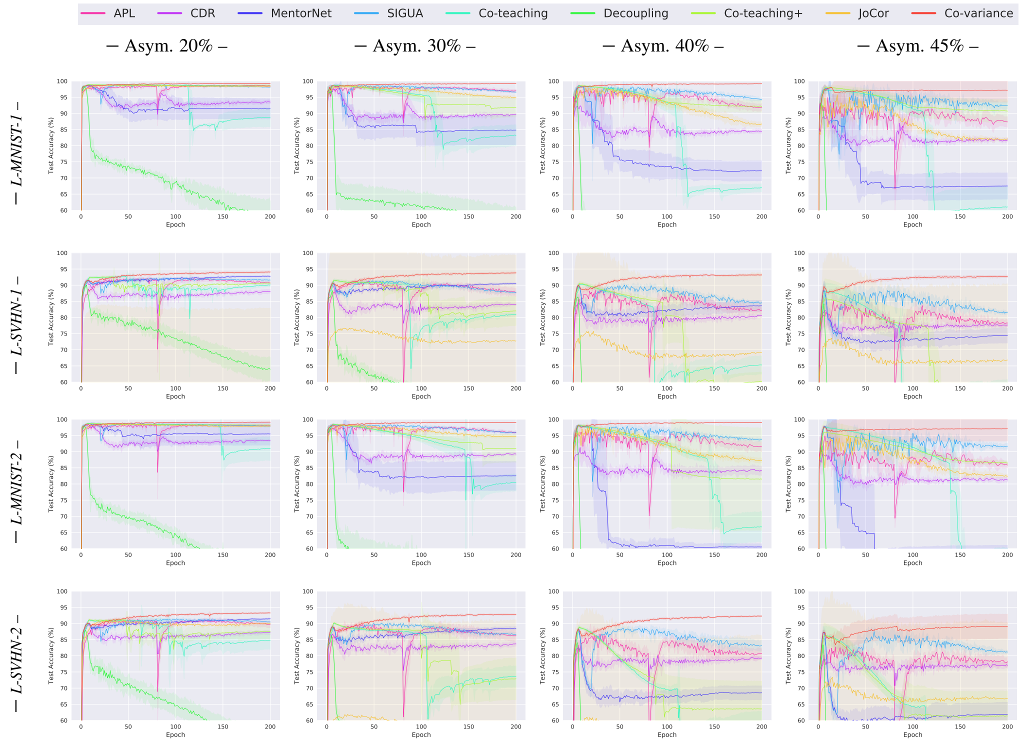


Figure 5: Test accuracy *vs.* the number of epochs on four long-tailed noisy datasets. The error bar for standard deviation in each figure has been shaded.

work (Patrini et al., 2017) and use ResNet-50 with ImageNet pretrained weights. As we mainly focus on sample selection, it is not fair to compare our method with some state of art methods, *e.g.*, DivideMix (Li et al., 2020) and ELR+ (Liu et al., 2020), which combine multiple methods. Therefore, to compare with it, we follow the paradigm of DivideMix to boost our method. The enhanced method is named DivideMix+, where we replace the sample selection procedure (Permuter et al., 2006) in DivideMix by Co-variance. For preprocessing, we resize the image to 256×256, crop the middle 224×224 as input, and perform normalization. The experiments on *Clothing1M* are performed once due to the huge computational cost. We use a ResNet-50 pretrained on ImageNet as did in (Patrini et al., 2017). We also use the Adam optimizer and set the batch size to 64. During training, we run 20 epochs in total and set the learning rate $8 \times 10^{-4}$, $5 \times 10^{-4}$, and $5 \times 10^{-5}$ for 5 epochs each.

| Method | Accuracy (%) |
|---|---|
| APL | 54.46 |
| CDR | 66.59 |
| MentorNet | 67.25 |
| SIGUA | 65.37 |
| Co-teaching | 67.94 |
| Decoupling | 67.65 |
| Co-teaching+ | 63.83 |
| JoCor | 69.06 |
| Co-variance | **71.60** |
| DivideMix | 74.76 |
| ELR+ | 74.81 |
| DivideMix+ | **74.92** |

Table 3: Test accuracy (%) on *Clothing1M*. The best results are in **bold**.

**Experimental results.** The classification performance achieved on *Clothing1M* is shown in Table 3. Specifically, compared with Co-variance with the baselines without multiple techniques combination, our method achieves an improvement of +2.54% over the best baseline JoCor. When we combine other advanced methods to boost our method as did in DivideMix, DivideMix+ can outperform DivideMix and ELR+, which means 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 Co-variance, which trains deep neural networks robustly with noisy labels. Co-variance 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 variances 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.

## 6 Ethics Statement

This paper doesn't raise any ethics concerns. This study doesn't involve any human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues.

## 7 Reproducibility Statement

To ensure the reproducibility of experimental results, we provide source codes of this paper in the supplementary material.

## References

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

Devansh Arpit, Stanisław Jastrzębski, 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*, pp. 233–242, 2017.

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

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *ACCLT*, pp. 92–100, 1998.

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

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pp. 1062–1070, 2019.

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.

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

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.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

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*, pp. 8527–8537, 2018.

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*, pp. 4006–4016, 2020.

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.

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

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

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*, pp. 2309–2318, 2018.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings*, pp. 331–339. 1995.

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

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

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

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

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

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.

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

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

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pp. 6543–6553, 2020.

Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *NeurIPS*, pp. 960–970, 2017.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

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.

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.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021.

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*, pp. 1944–1952, 2017.

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

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.

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

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

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.

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.

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.

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

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, pp. 5596–5605, 2017.

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

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pp. 13726–13735, 2020.

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

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

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.

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

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

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*, pp. 6222–6233, 2019.

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

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*, pp. 10789–10798, 2020a.

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*, 2020b.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement benefit co-teaching? In *ICML*, 2019.

Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *CVPR*, pp. 4480–4489, 2018.

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

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

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

Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *ICML*, 2021b.

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

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

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

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

# A Complementary Experimental Settings

## A.1 The Details of Used Datasets

For the details of used datasets in experiments on closed-set noisy datasets, the important statistics of the used datasets are summarized in Table 4.

| | type | # of training | # of testing | # of class | size |
|---|---|---|---|---|---|
| *MNIST* | image | 60,000 | 10,000 | 10 | 28×28×1 |
| *F-MNIST* | image | 60,000 | 10,000 | 10 | 28×28×1 |
| *SVHN* | image | 73,257 | 26,032 | 10 | 32×32×3 |
| *CIFAR-10* | image | 50,000 | 10,000 | 10 | 32×32×3 |
| *NEWS* | text | 11,314 | 7,532 | 20 | 300-D |

Table 4: Summary of simulated closed-set noisy datasets used in the experiments.

## A.2 The Details of Generating Noisy Labels for Balanced Cases

$$\text{Sym. } \epsilon\text{:} \quad T = \begin{bmatrix} 1-\epsilon & \frac{\epsilon}{c-1} & \dots & \frac{\epsilon}{c-1} & \frac{\epsilon}{c-1} \\ \frac{\epsilon}{c-1} & 1-\epsilon & \frac{\epsilon}{c-1} & \dots & \frac{\epsilon}{c-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\epsilon}{c-1} & \dots & \frac{\epsilon}{c-1} & 1-\epsilon & \frac{\epsilon}{c-1} \\ \frac{\epsilon}{c-1} & \frac{\epsilon}{c-1} & \dots & \frac{\epsilon}{c-1} & 1-\epsilon \end{bmatrix}_{c\times c}. \tag{4}$$

$$\text{Pair. } \epsilon\text{:} \quad T = \begin{bmatrix} 1-\epsilon & \epsilon & \dots & 0 & 0 \\ 0 & 1-\epsilon & \epsilon & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1-\epsilon & \epsilon \\ \epsilon & 0 & \dots & 0 & 1-\epsilon \end{bmatrix}_{c\times c}. \tag{5}$$

$$\text{Trid. } \epsilon\text{:} \quad T = \begin{bmatrix} 1-\epsilon & \frac{\epsilon}{2} & \dots & 0 & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & 0 & \dots & \frac{\epsilon}{2} & 1-\epsilon \end{bmatrix}_{c\times c}. \tag{6}$$

Here, we introduce the details of generating different types of noisy labels. We mainly follow the settings in (Zhang et al., 2021b). The details are described as follows:

◇ Instance-independent noise

- Symmetric noise.: We flip clean labels in each class *uniformly* to incorrect labels of other classes.

- Pairflip noise: We flip clean labels in each class to its *adjacent* class.

- Tridiagonal noise: the noise corresponds to a spectral of classes where adjacent classes are easier to be mutually mislabeled, which can be implemented by *two consecutive pair flipping* transformations in the opposite direction.

We corrupt clean datasets manually by the label transition matrix $T$, where $T_{ij} = P(\tilde{\mathbf{y}} = \mathbf{e}_j | \mathbf{y} = \mathbf{e}_i)$, given that noisy $\tilde{\mathbf{y}}$ is flipped from clean $\mathbf{y}$. When the noise rate is set to $\epsilon$, the transition matrices for the above three types of label noise are shown in (4), (5), and (6).

◇ Instance-dependent noise

- Instance noise: We consider that the probability that an instance is mislabeled depends on its *features/instances*. The generation of such a kind of noise follows the procedure in (Zhu et al., 2021).

### A.3 The Details of Network Structures

We use a 9-layer CNN for *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*. The CNN is shown in Table 5. We use a 3-layer MLP for *NEWS*, which is shown in Table 6.

| Input |
|---|
| 3×3 Conv, 128 LReLU<br>3×3 Conv, 128 LReLU<br>3×3 Conv, 128 LReLU |
| 2×2 Max-pool, stride 2<br>Dropout, $p = 0.25$ |
| 3×3 Conv, 256 LReLU<br>3×3 Conv, 256 LReLU<br>3×3 Conv, 256 LReLU |
| 2×2 Max-pool, stride 2<br>Dropout, $p = 0.25$ |
| 3×3 Conv, 512 LReLU<br>3×3 Conv, 256 LReLU<br>3×3 Conv, 128 LReLU |
| Avg-pool |
| Dense 128→# of classes |

Table 5: The CNN used on *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*.

| Input |
|---|
| Dense 300→1280<br>BatchNorm1D |
| Softsign |
| Dense 1280→160<br>BatchNorm1D |
| Dense 160→ # of classes |

Table 6: The MLP used on *NEWS*.

### A.4 The Details of Generating Noisy Labels for Imbalanced Cases

In this paper, we consider two types of ways for building *imbalanced noisy* datasets. The first one is asymmetric noise, which is injected into four datasets, *i.e.*, *MNIST*, *F-MNIST*, *SVHN*, and *CIFAR-10*. For *MNIST*, flipping 2→7, 3→8, 5↔6. For *F-MNIST*, flipping TSHIRT→SHIRT, PULLOVER→COAT, SANDALS→SNEAKER. For *SVHN*, flipping 2→7, 3→8, 5↔6. For *CIFAR-10*, flipping TRUCK→AUTOMOBILE, BIRD→AIRPLANE, DEER→HORSE, CAT↔DOG. As some flip processes (*e.g.*, 2→7, but not 2↔7) are *not bidirectional*, the simulated noisy datasets are imbalanced accordingly.

### A.5 Supplementary Experimental Results

The results of ablation study on balanced closed-set noisy datasets are provided in Figure 6, which shows that Co-variance is stable with the change of $\alpha$.

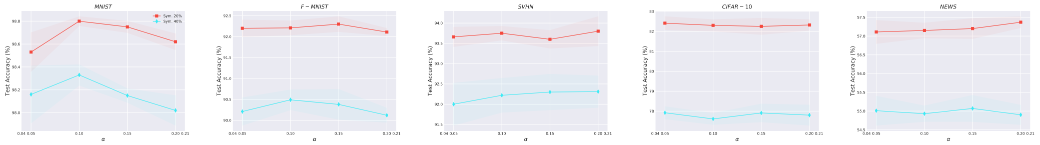


Figure 6: Illustrations of the hyperparameter sensitivity for the proposed method. The error bar for standard deviation in each figure has been shaded.

## B Other Deep Learning Methods for Learning with Noisy Labels

Expect for the introduced sample selection approach, a large body of work proposed various methods for coping with noisy labels, which include but are not limited to, learning a label noise transition matrix (Hendrycks et al., 2018; Yao et al., 2020b), reweighting examples (Liu & Tao, 2016; Ren et al., 2018; Fang et al., 2020), recalibrating labels (Tanaka et al., 2018; Zheng et al., 2020; Zhang et al., 2021a), using graph models (Xiao et al., 2015; Li et al., 2017; Vahdat, 2017), designing robust loss functions (Zhang & Sabuncu, 2018; Xu et al., 2019; Ma et al., 2020), exploiting (implicit) regularization (Zhang et al., 2018; Kim et al., 2019; Hu et al., 2020; Xia et al., 2021; Chen et al., 2021; Wei & Liu, 2021), and combining semi-supervised learning (Nguyen et al., 2020; Li et al., 2020; Liu et al., 2020; Zhou et al., 2021), *etc*. We suggest that readers can refer (Song et al., 2020) for more details of learning with noisy labels.