future internet

MDPI

Article

JoSDW: Combating Noisy Labels by Dynamic Weight

Yaojie Zhang [1,*], Huahu Xu [1,2], Junsheng Xiao [1] and Minjie Bian [1,2]

[1] School of Computer Science and Engineering, Shanghai University, Shanghai 200444, China; huahuxu@staff.shu.edu.cn (H.X.); jsxiao@shu.edu.cn (J.X.); bianmj0302@aliyun.com (M.B.)
[2] Shanghai Shangda Hairun Information System Co., Ltd., Shanghai 200444, China
[*] Correspondence: 19721561@shu.edu.cn

Abstract: The real world is full of noisy labels that lead neural networks to perform poorly because deep neural networks (DNNs) are prone to overfitting label noise. Noise label training is a challenging problem relating to weakly supervised learning. The most advanced existing methods mainly adopt a small loss sample selection strategy, such as selecting the small loss part of the sample for network model training. However, the previous literature stopped here, neglecting the performance of the small loss sample selection strategy while training the DNNs, as well as the performance of different stages, and the performance of the collaborative learning of the two networks from disagreement to an agreement, and making a second classification based on this. We train the network using a comparative learning method. Specifically, a small loss sample selection strategy with dynamic weight is designed. This strategy increases the proportion of agreement based on network predictions, gradually reduces the weight of the complex sample, and increases the weight of the pure sample at the same time. A large number of experiments verify the superiority of our method.

Keywords: small loss selection; noise label; dynamic weight; disagreement strategy; agreement maximization principle

check for updates

Citation: Zhang, Y.; Xu, H.; Xiao, J.; Bian, M. JoSDW: Combating Noisy Labels by Dynamic Weight. *Future Internet* **2022**, *14*, 50. https://doi.org/10.3390/fi14020050

Academic Editors: Vijayakumar Varadarajan, Rajanikanth Aluvalu and Ketan Kotecha

Received: 31 December 2021
Accepted: 31 January 2022
Published: 2 February 2022

1. Introduction

Deep neural networks have accomplished outstanding achievements in various artificial intelligence tasks, which is mainly due to the massive high-quality datasets with labels at this stage. Large amounts of data, on the other hand, need a lot of money and time to annotate with high-quality annotations. However, accurately labeling large amounts of data is a very time-consuming and labor-intensive task. To resolve this problem, data labeling companies have begun to seek alternative and inexpensive methods, such as searching commercial search engines [1], gathering web label information [2], employing machine-generated labels [3], or having a single annotator mark each sample [4]. Although cheap and efficient, these alternative methods are always accompanied by samples with noisy labels. Even in industry, noise labels are generated due to errors in prop models or subtle differences in the cutting process [5]. Existing studies have shown that deep networks are liable to overfitting label noise during the training process, and their generalizability is low, resulting in a substantial drop in DNN performance [6]. With the advent of 5G, the amount of data is increasing rapidly, and it is more necessary to study noise label learning [7].

Because the existence of noisy tags severely restricts the implementation and development of neural network models in the industry. Initially, the method of data preprocessing is used to deal with noisy labels, but it is inefficient. On the one hand, research attempts to improve the quality of the data at the modeling stage, such as by boosting ensembles [8]. On the other hand, a large number of weakly supervised learning algorithms for learning with noise have been developed. The existing noise label learning methods mainly use sample selection methods and loss correction methods. The loss correction method estimates the noise transfer matrix. Then, the loss function is corrected with the noise transfer matrix. However, it is challenging to correctly estimate the noise transfer matrix. Some methods use

DNN predictions to correct the label and adjust the loss accordingly [9,10]. These methods perform poorly under high noise ratios. The reason is that DNN predictions dominate the training process leading to overfitting. The sample selection rule is to screen clean samples or reweight the samples to reduce the contribution of noise samples to the loss. Designing a reliable algorithm and criteria to select clean samples is a challenging problem. Research shows that before fitting label noise, the DNN tends to fit simple samples first [11]. Therefore, samples with small loss are considered clean samples and are used by many methods [12,13]. Among them, co-teaching [14] and co-teaching+ [15] train two networks. Each network selects the samples with a small loss in mini-batch to train peer-to-peer networks. It is worth mentioning that decoupling [16] and co-teaching+ [15] adopt the "disagreement" strategy, and the network itself decides when to update. JoCOR [17] set up a joint loss between the two networks of co-teaching to encourage the model to reach an "agreement". Chen [18] started to combat label noise by compressing regularized neural networks, which they called Nested Dropout. This method does not innovate the structure of the dual network but performs compression and regularization training in the dataset in advance to provide a more reliable basic network for the subsequent network. This method can be combined with methods such as co-teaching.

The essence of the co-teaching type of algorithm design is to seek "agreement" in "disagreement". Inspired by this, this article is different from the previous "Disagreement" strategies of decoupling [16] and co-teaching+ [15]. It turns its attention to the "Agreement" part and combines small loss samples with the "Agreement" and "Disagreement" strategies. The intention is to mine clean samples with higher purity and discard high noise rate noise. However, focusing too much on the "Agreement" part also causes the model to overfit, which reduces the accuracy of the model. Therefore, we perform dynamic weight distribution on the agreement sample and the "disagreement" sample. Specifically, this article inputs an image into two neural networks with the same structure and different initializations, generates prediction probability labels correspondingly, and divides the samples into the "Agreement group" and the "Disagreement group" according to whether the sample predicts disagreement or not. At the same time, a joint loss is set between the networks, which includes the traditional cross-entropy loss and the comparison loss between the two networks. Based on these two loss functions, small loss samples are taken as clean samples to enter the DNN training. According to two strategies, the samples are processed for differences. The main contributions of this paper are as follows:

- This article selects reliable samples through a small loss sample strategy by using relative loss and multi-class loss, subdivides it, and proposes a distinction between pure samples and complex samples based on the prediction consistency in the multi-view of the sample.
- In this paper, a dynamic weight is set between the pure sample and the complex sample to reduce the weight of the noise sample. While deepening the neural network, the complex sample weight is gradually reduced. The dynamic weight is determined based on the results of the previous round of iterative training.
- By providing comprehensive experimental results, we show that our method outperforms the most advanced methods on noisy datasets. In addition, extensive ablation studies are conducted to verify the effectiveness of our method.

The remainder of the paper is structured as follows: Section 2 provides an overview of agreement and disagreement. Section 3 describes the framework of JoSDW. The experimental results of our method will be demonstrated in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work

2.1. Co-Teaching

Co-training is designed for "multi-view" data. In each cycle, two different classifiers are trained from the labeled data, and then the two classifiers are used to classify the unlabeled data, and then the unlabeled data with the highest confidence is classified. Add to the labeled data and continue to loop until there is no data in the unlabeled data or the maximum number of loops is reached.

Co-teaching refers to the idea of training two classifiers. Co-teaching [14] is motivated by co-training for learning using noisy labels. The idea is to train two identical deep networks at the same time, but these two networks have different initialization parameters, and both networks adopt a small loss sample selection strategy, that is, samples with small losses are regarded as clean samples. In each mini-batch, each network selects instances of its small loss as valuable information. These valuable instances are impacted by its peer-to-peer network for the following training. The two networks can filter different types of errors introduced by noisy labels because they have different learning capabilities. Finally, each network backpropagates the data selected by another and updates itself. Table 1 shows the reference baseline and its description, and Table 2 shows the comparison of baseline.

Table 1. Reference baseline and its description.

Baselines	Description
F-correction	F-correction corrects the label predictions by the label transition matrix.
Decoupling	Decoupling only uses instances with different predictions from the two classifiers to update the parameters.
Co-teaching	Co-teaching trains two networks simultaneously and let them cross-update.
Co-teaching+	Co-teaching+ trains two networks simultaneously and lets them cross-update when the two networks predict disagreement.
JoCOR	JoCOR trains two networks at the same time and sets a contrast loss between the two networks to facilitate the two grids to reach an “Agreement”.

Table 2. Comparison of baseline.

	Decoupling	Co-Teaching	Co-Teaching+	JoCOR	JoSDW
Small loss	X	√	√	√	√
Cross update	X	√	√	X	X
Disagreement	√	X	√	X	√
Agreement	X	X	X	√	√
Dynamic weight	X	X	X	X	√

Samples with small loss are more likely to be clean [10,12,13,19], we can train them on these instances to obtain a classifier that is resistant to noise labels. However, after conducting experiments that combined the “Agreement” strategy with small loss samples, the cleanliness of the small loss samples proves that they can be further improved.

2.2. Disagreement and Agreement

The agreement strategy is inspired by semi-supervised learning, and JoCOR proposes the agreement maximization principle, which encourages two different classifiers to make closer predictions through an explicit regularization method. The samples that are considered to be consistent in the predictions of the two networks are more credible, and the consistency of the network predictions should be promoted as much as possible. The method is to add js divergence between the two network prediction results to judge the distance between the two network prediction results.

Disagreement was proposed by Decoupling, and its key idea is to decouple “when to update” from “how to update”. In previous studies, the network was updated all the time. For the “Disagreement” strategy, “when to update” depends on the Disagreement between the two networks rather than all the time. Co-teaching+ combines the “Disagreement” strategy with the cross-update on Co-teaching and proposes that the prediction divergence of the two networks can better help network training and improve the robustness of the network. Therefore, Co-teaching only exchanges and updates the parts with inconsistent predictions in the small loss samples. In this study, we combine "Agreement" and "Disagreement", the "Agreement" part is called the pure sample, and the "Disagreement" part is called the complex sample, and dynamic weights are set for the complex sample.

We strongly agree with maximizing the agreement, which combined with small loss samples results in pure samples that are cleaner than small loss samples. For the complex sample in the early stage of network model training, it can help the network to train better. This solves the problem of "when to update". However, with the training of the network model, the complex samples may no longer be a good guide for model updating.

2.3. Contrastive Learning

In a range of tasks, contrastive learning, a recently suggested unsupervised learning paradigm [20–24], has achieved state-of-the-art achievement. The key difference between these approaches is the data augmentation approach and the contrastive loss they use. In summary, most contrastive learning approaches start by constructing positive and negative pairs at the instance level via a sequence of data augmentations. Following that, the contrastive loss can be used to optimize positive pair similarity while minimizing negative pair similarity. Such as NT-Xent [20], Triplet [25] and NCE [26].

The key idea of contrastive learning is to bring similar samples closer and dissimilar samples farther away. Given data x, the goal of contrastive learning is to learn an encoder f such that:

$$\text{score}\big(f(x), f\big(x^{+}\big)\big) \gg \text{score}\big(f(x), f\big(x^{-}\big)\big) \tag{1}$$

where x is called pinpoint data, x^{+} is a positive sample similar to x, x^{-} is a negative sample that is not similar to x, and score() is a measurement function to measure positive and negative samples' similarity. The score() function often takes Euclidean distance, cosine similarity, etc.

3. The Proposed Method

As mentioned before, the essence of co-teaching is to seek the "Agreement" of the two networks in the "Disagreement" of the two networks. Therefore, this article claims that as the training of the co-teaching network deepens, clean labels tend to concentrate on the label noise rate of the "Agreement" part, which gradually decreases. In the experiment in this article, two neural networks with the same network framework but with distinct initializations were used as classifiers, and the two classifiers defined training samples for prediction. According to the small loss sample strategy and the results of the classifier, the samples were divided into three types of samples: pure samples, complex samples, and dirty samples. We processed these samples differently, which can be seen as follows:

For multi-class classification with M classes. There are N samples in the dataset $D = \{x_i,\ y_i\}_{i=1}^{N}$, where x_i. represents the i-th instance and its given label as $y_i \epsilon \{1, \ldots, M\}$. We formulate the proposed JoSDW approach with two deep neural networks denoted by $f(x{,}\Theta_1)$ and $f(x{,}\Theta_2)$, while $p_1 = [p_1^1,\ p_1^2, \ldots, p_1^M]$ and $p_2 = [p_2^1,\ p_2^2, \ldots, p_2^M]$ denote the prediction probabilities of instance x_i, while p_1 and p_2 represent the outputs of the "softmax" layer in Θ_1 and Θ_2.

As shown in Figure 1, in our method JoSDW, the dataset $D = \{x_i,\ y_{\mathrm{i}}\}_{i=1}^{N}$, is fed into two different networks ($f(x{,}\Theta_1)$ and $f(x{,}\Theta_2)$). The loss $L(x_i)$ is calculated based on p_1= $[p_1^1,\ p_1^2, \ldots, p_1^M]$ and p_2= $[p_2^1,\ p_2^2, \ldots, p_2^M]$. In each mini-batch, $L(x_i)$ are sorted from small to large. The large part called dirty sample are considered more likely to be noisy label instances, so dropout it. While the small part is divided into pure samples and complex samples by the consistency of the predicted labels.

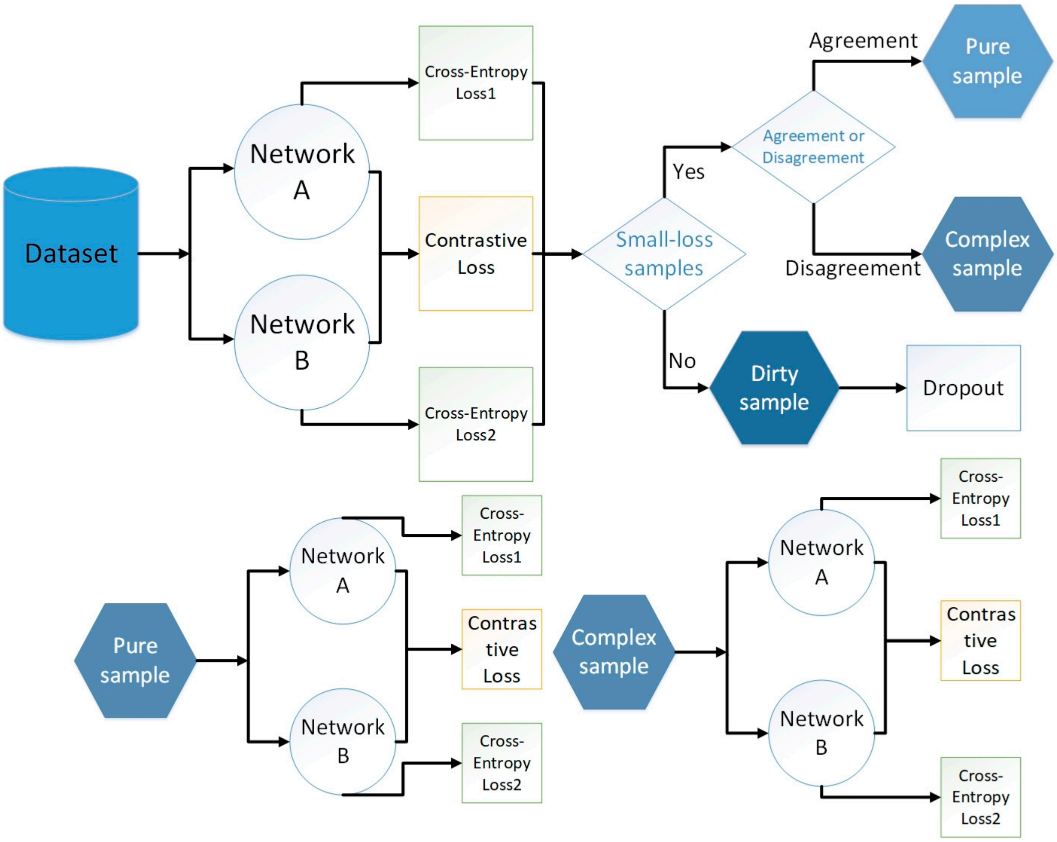

Figure 1. The framework of JoSDW. For JoSDW, each network can be used to predict labels on its own. We determine whether the sample is a small loss sample based on the respective cross-entropy loss of the two networks and the joint loss between the two networks. Then, the two networks are classified into pure samples and complex samples by the consistency of the predicted labels of the two networks.

3.1. Network

For each network, the predicted pseudo labels are generated separately, but during the training process, they are trained through the pseudo conjoined paradigm, which means that their parameters are different, but they are updated through joint loss.

3.2. Loss Function

Our small loss sample selection is as follows:

$$L(x_i) = (1-\gamma) * L_{con}(x_i) + \gamma * L_{sup}(x_i, y_i) \tag{2}$$

In the loss function, the first part L_{con} is the comparison loss between the predictions of the two networks to achieve common regularization, and the second part *Lsup* represents the traditional supervised learning loss of the two networks .

- Classification loss

For multi-classification tasks, we use cross-entropy loss as the supervision part to minimize the distance between the prediction and the label as follows:

$$L_{sup}(x_i, y_i) = L_{CE}(x_i, y_i) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \log\left(p_i^C\right) \tag{3}$$

- Contrastive loss

According to the agreement maximization principle [27,28], different networks under the label of most examples will agree, and they struggle to agree on incorrect labels. This can be calculated as follows:

$$L_{con}(x_i) \;=\; L_{KL} \;=\; -\frac{1}{N}\sum_{i=1}^{N} \rho_i(D_{KL}(p_i||p_i') + D_{KL}(p_i'||p_i)) \tag{4}$$

3.3. Sample Selection

- Small loss sample selection

Traditional sample selection uses a small loss sample selection strategy. Since the DNN tends to fit simple samples first [18], small loss samples are more likely to be clean samples. This standard method usually selects a predefined proportion of small loss samples in each small batch. Forget rate is an important parameter in the small loss sample strategy. According to [14,15,17], the forget rate takes the value of the noise rate assumed that is known, and there is an initialization process for the forget rate:

$$\varphi(e) = \min\left\{\frac{e}{k}\sigma, \sigma\right\} \tag{5}$$

Forget rate $\varphi(e)$ gradually increased to noise rate σ in the first k (=10) epochs e.

However, when training the DNN, the noise ratio in different small batches inevitably fluctuates. As shown in the figure above, and due to the deepening of the DNN training, the agreement part in the small sample selection quickly rises and then stabilizes, while the disagreement part in the small sample selection has a clean rate even lower than our 80% clean rate. Combined with the prediction accuracy map, this part of the noise samples began to affect the accuracy of the DNN model and reduce the prediction accuracy.

- Pure sample

The small loss samples are subdivided twice. Because "co-teaching" uses the comparative training of the two networks to promote the model from "Disagreement" to "Agreement" and because the DNN tends to fit simple samples first [18], we have reason to believe that a sample label that quickly reaches an "Agreement" is more likely to be credible. In the initial training stage of the "Agreement," although the label purity rate did not widen the gap from the disagreement part, with the deepening of the DNN training, the label purity rate opened a large gap.

- Complex sample

When choosing to use only pure samples for high purity samples, there are limitations. The limitations are as follows: Although the small loss sample strategy combined with the "Agreement" for secondary classification can help us quickly screen out high purity samples, this is under the premise of greatly reducing the total number of input samples, which causes the network model to overfit. Second, if only high clean rate samples are considered, the network model converges too slowly. Therefore, we also need to introduce the Disagreement part named the Complex sample.

3.4. Dynamic Weight

As the purity rate of the pure sample gradually increases during the experiment, the purity rate of the complex sample gradually decreases. Therefore, we apply dynamic weights to the samples between the pure sample and complex sample. As the network model fits deeper, the influence of the complex sample on the network parameters is gradually reduced as follows.

$$Loss_{selected} = Loss_{pure} + \lambda * Loss_{complex} \tag{6}$$

$$\lambda = \sqrt{\frac{N_{Complex}}{N_{disagreement}}} \tag{7}$$

where λ represents a dynamic weight parameter, $N_{disagreement}$ represents the number of disagreements samples in the previous round of training, and $N_{Complex}$ represents the number of complex samples in the previous round of training.

4. Experiments

In this section, a series of experimental results are presented.

We test the effectiveness of our proposed algorithm on three benchmark datasets, including MNIST [29], CIFAR-10, and CIFAR-100 [30]. The main information of these datasets is shown in Table 3. In past research, these datasets were commonly used to assess learning with noisy labels. [9,31,32]. We compare JoSDW with the following state-of-the-art algorithms, implement all methods with default parameters through PyTorch, and perform all experiments on NVIDIA 2080ti. The experimental data comes from the following three datasets: Mnist, CIFAR-10, CIFAR-100.

Table 3. Image resolution of common data sets for label noise.

Datasets	# of Class	# of Train	# of Test	Size
Mnist	10	60,000	10,000	28 × 28
CIFAR-10	10	50,000	10,000	32 × 32
CIFAR-100	100	50,000	10,000	32 × 32

In order to explore the performance in different noise cases, we use the following four noise ratios: Symmetry-20%, Symmetry-50%, Symmetry-80%, Asymmetry-40%. Figure 2 is an example of noise ratios.

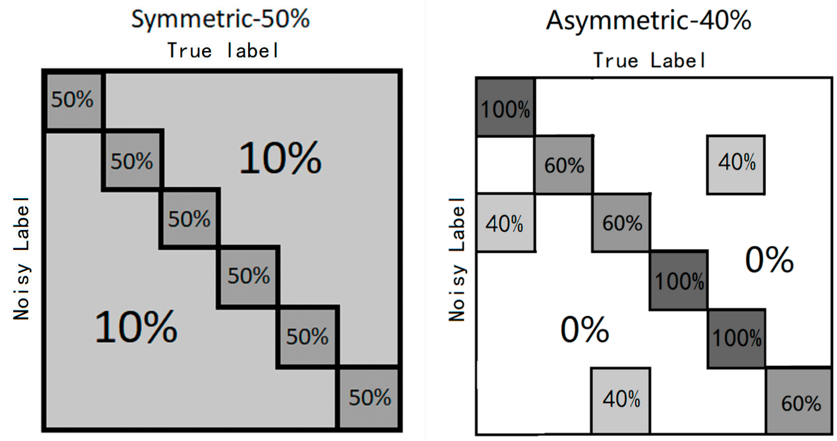

Figure 2. Example of noise transition matrices (taking 6 classes).

The symmetric noise, which is also called random or uniform noise, is a common setup of noisy label experiments. M represents the number of classes, and α represents the noise ratio. The probability of a true label is $P_{true} = 1 - \alpha$. The probability of a noisy label is $P_{noisy} = \alpha$. The probability of each noisy label is $P^{i}_{noise} = \frac{\alpha}{M-1}$.

The asymmetric noise is closer to a real-world label noise because of flipping. For example, on Mnist, the asymmetric noise maps 2→4, 3→1, 6→3.

We utilize a two-layer MLP for MNIST and a seven-layer CNN network architecture for CIFAR-10 and CIFAR-100 in terms of the network structure. The network architecture of MLP and CNN is shown in Table 4. Regarding the optimizer, we use an Adam optimizer with momentum = 0.9. The initial learning rate is 0.001. The batch size is set to 128. We run 200 epochs totally, with the learning rate gradually decaying to 0 from 80 to 200 epochs.

Table 4. Network architecture.

MLP	CNN
Gray Image 28 × 28	RGB Image 32 × 32
28 × 28→256, ReLU	3 × 3, 64 BN, ReLU 3 × 3, 64 BN, ReLU 2 × 2 Max-pool
	3 × 3, 128 BN, ReLU 3 × 3, 128 BN, ReLU 2 × 2 Max-pool
	3 × 3, 196 BN, ReLU 3 × 3, 196 BN, ReLU 2 × 2 Max-pool
256→10	256→100

4.1. Results on MNIST

On the left side of Figures 3–6, the comparison of precision accuracy on MNIST is shown. In these four pictures, we can see the memory effect of the network; the standard precision accuracy is first achieved at a high level and gradually decreases. Therefore, a solid, reliable training approach should be able to stop or slow down the decrease process. At this point, JoSDW regularly outperforms all other baselines in each of the four cases.

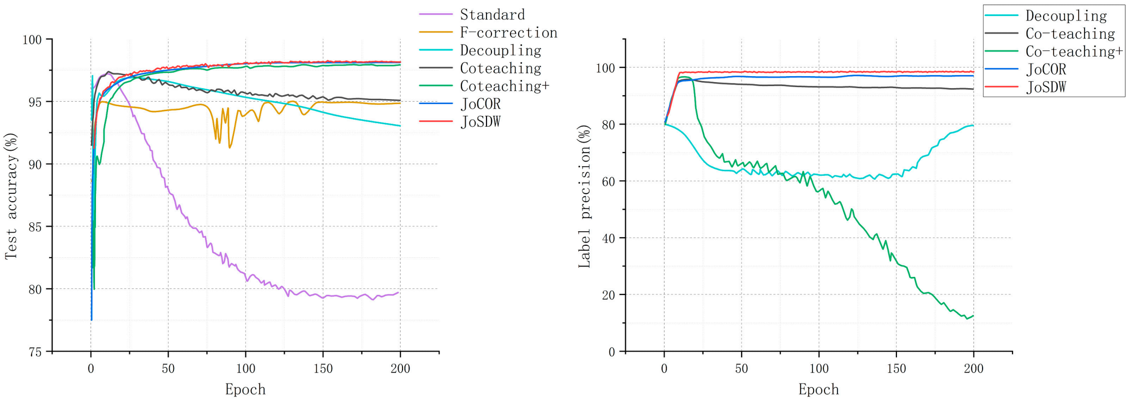

Figure 3. Results on MNIST dataset. Noise settings, Symmetry-20%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

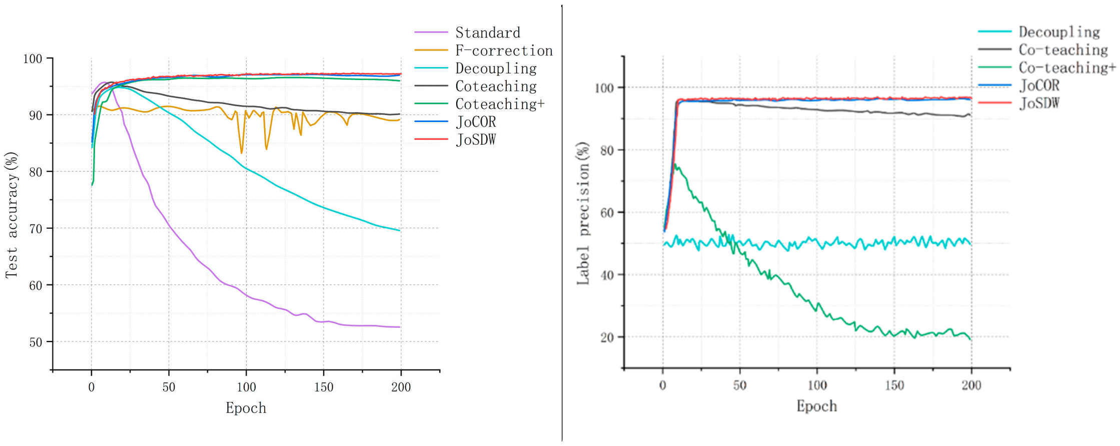

Figure 4. Results on MNIST dataset. Noise settings, Symmetry-50%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

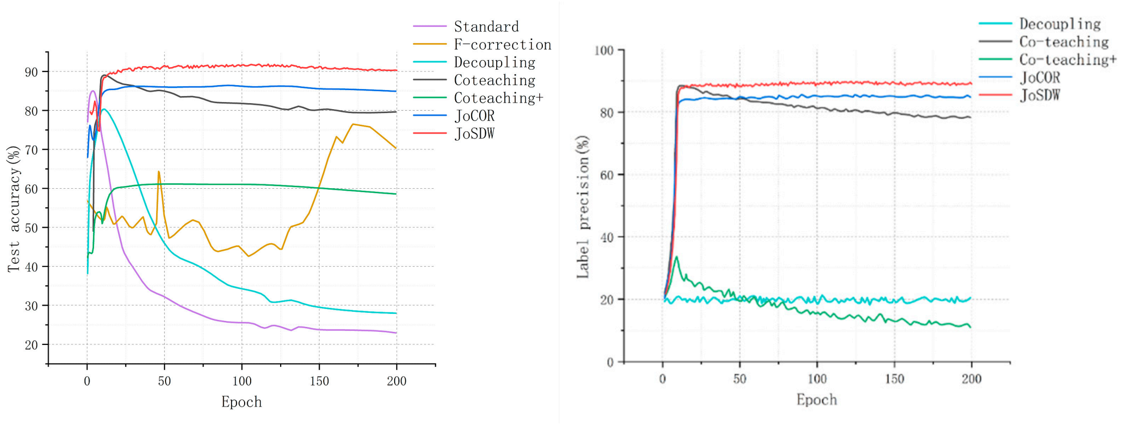

Figure 5. Results on MNIST dataset. Noise settings, Symmetry-80%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

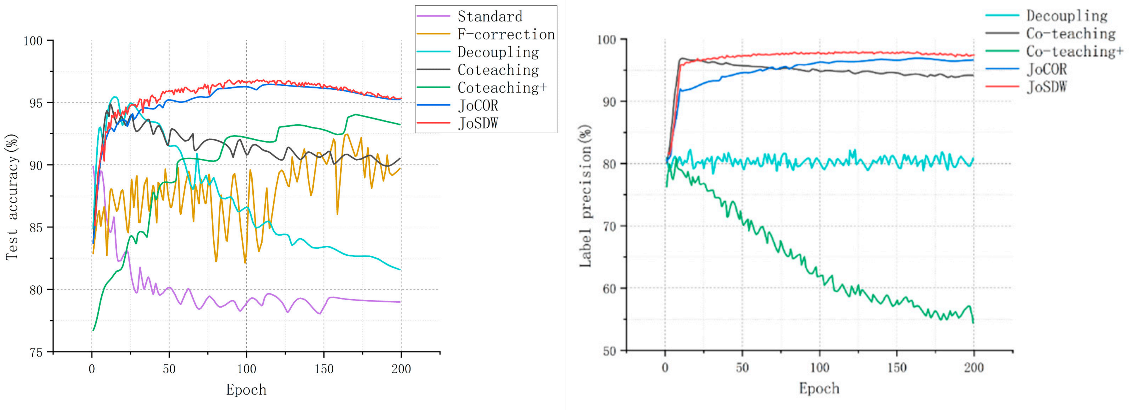

Figure 6. Results on MNIST dataset. Noise settings, Asymmetry-40%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

Table 5 compares the precision accuracy of various algorithms. In the conventional Symmetry-20% case and Symmetry-50% case, all the new methods are clearly superior to the standard method, which proves their robustness. However, when the noise label reaches the symmetry-80% case, the performance of decoupling and co-teaching+ based on disagreement declines substantially, while JoSDW is considerably better than the other methods.

Table 5. Average Precision accuracy (%) on MNIST over the last 10 epochs.

Noise Settings	**Standard**	**F-Correction**	**Decoupling**	**Co-Teaching**	**Co-Teaching+**	**JoCOR**	**JoSDW**
Symmetry-20%	79.56	95.38	93.16	95.1	97.81	98.06	98.24
Symmetry-50%	52.66	92.74	69.79	89.82	95.8	96.64	97.23
Symmetry-80%	23.43	72.96	28.51	79.73	58.92	84.89	90.31
Asymmetry-40%	79	89.77	81.84	90.28	93.28	95.24	95.38

We plot label precision vs. epochs on the right side of Figures 3–6. These experiments demonstrate the characteristics of JoSDW. Although the peak precision accuracy of JoSDW training is not as high as other algorithms, the existence of dynamic weights makes JoSDW test the accuracy that comes from behind, and there is no substantial drop in precision accuracy.

The right-hand sides of Figures 3–6 demonstrate the performance of label precision vs. epochs. From the experimental results, it can be seen that decoupling and co-teaching+ cannot effectively screen out reliable sample labels, but JoSDW, JoCOR, and co-teaching can still maintain excellent performance. The small loss sample selection strategy after secondary subdivision and dynamic weighting is higher than the traditional small loss sample selection strategy in the middle and late stages of training, and it performs well in the symmetry-80% and asymmetry-40% cases. This shows that JoSDW can better find clean examples.

4.2. Results on CIFAR-10

Table 6 shows the precision accuracy of CIFAR-10. JoSDW performs best again in all four cases. For the Symmetry-20% case, co-teaching+ performs better than co-teaching and decoupling. For the other three cases, co-teaching+ cannot even achieve the same performance as co-teaching.

Table 6. Average precision accuracy (%) on CIFAR-10 over the last 10 epochs.

Noise Settings	Standard	F-Correction	Decoupling	Co-Teaching	Co-Teaching+	JoCOR	JoSDW
Symmetry-20%	69.18	68.74	69.32	78.23	78.71	85.73	86.28
Symmetry-50%	42.71	42.19	40.22	71.3	57.05	79.41	79.75
Symmetry-80%	16.24	15.88	15.31	25.58	24.19	27.78	32.31
Asymmetry-40%	69.43	70.6	68.72	73.78	68.84	76.36	77.04

Figures 7–10 show the precision accuracy and label precision vs. epochs. JoSDW is superior to all other comparison methods in terms of precision accuracy and label accuracy. In terms of label accuracy, decoupling and co-teaching+ relying on disagreement does not find a clean instance. When the noisy label reaches Symmetry-80%, JoSDW is considerably better than other methods in terms of precision accuracy and label accuracy, but in Asymmetry-40%, although in terms of label accuracy JoSDW is substantially better than other methods, there is no obvious advantage in precision accuracy. Additionally, although some methods outperform JoSDW in the initial stage, in all subsequent epochs, JoSDW consistently outperforms other approaches.

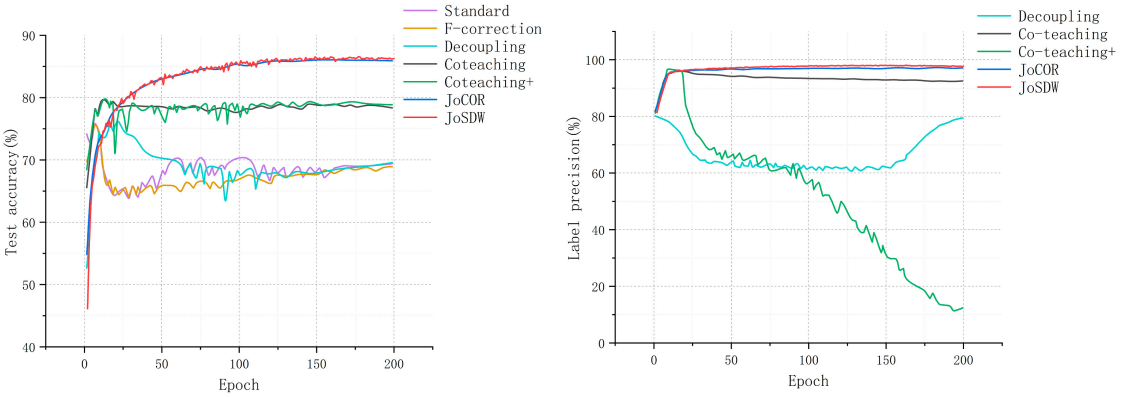

Figure 7. Results on the CIFAR-10 dataset. Noise settings, Symmetry-20%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

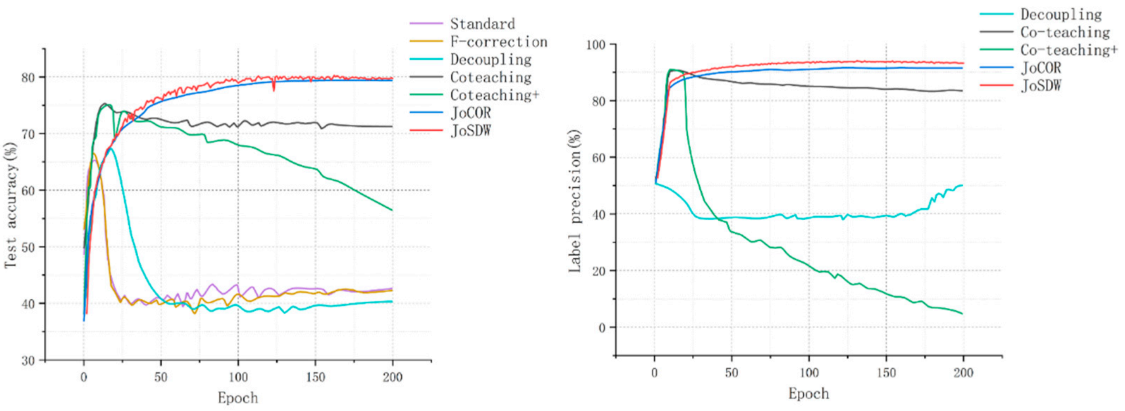

Figure 8. Results on the CIFAR-10 dataset. Noise settings, Symmetry-50%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

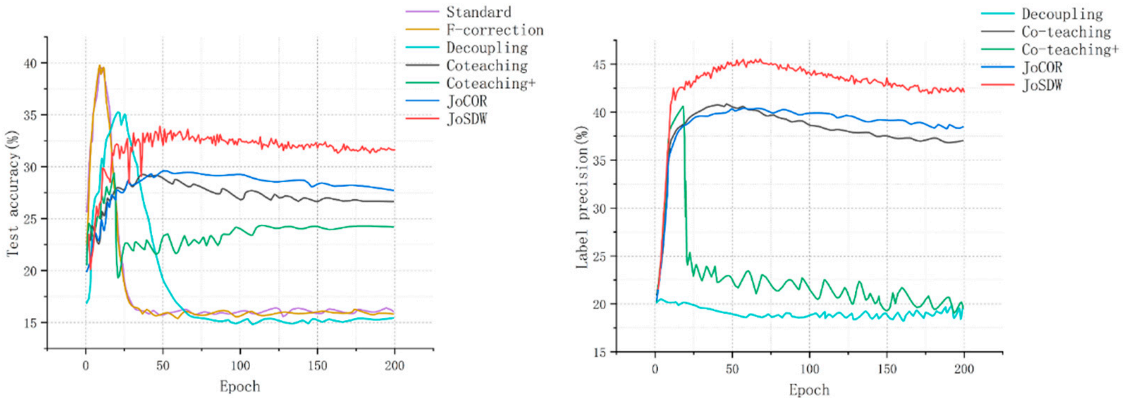

Figure 9. Results on the CIFAR-10 dataset. Noise settings, Symmetry-80%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

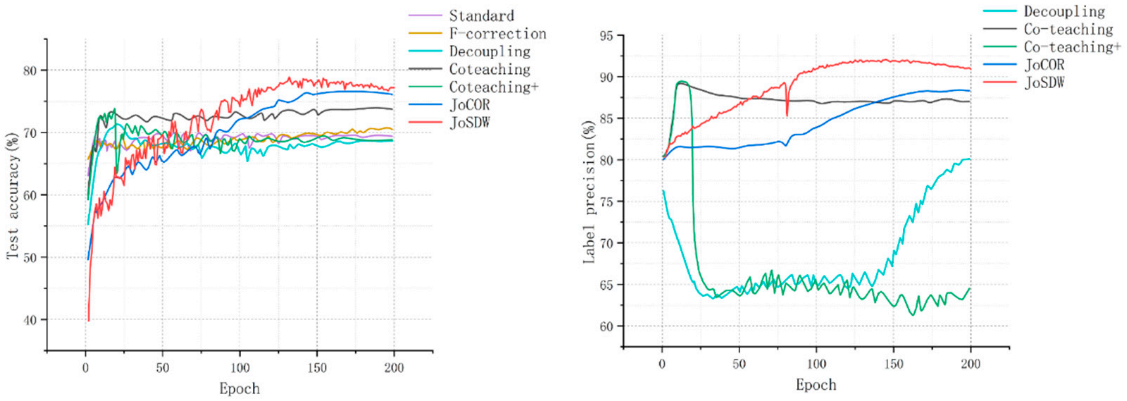

Figure 10. Results on the CIFAR-10 dataset. Noise settings, Asymmetry-40%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

4.3. Results on CIFAR-100

Table 7 displays the precision accuracy. The precision accuracy and label precision vs. epochs are shown in Figures 11–14. In the MNIST and CIFAR-10 datasets, there are just 10 classes.

Table 7. Average precision accuracy (%) on CIFAR-100 over the last 10 epochs.

Noise Settings	Standard	F-Correction	Decoupling	Co-Teaching	Co-Teaching+	JoCOR	JoSDW
Symmetry-20%	35.14	37.95	33.1	43.73	49.27	53.01	53.28
Symmetry-50%	16.97	24.98	15.25	34.96	40.04	43.49	45.25
Symmetry-80%	4.41	2.1	3.89	15.15	13.44	15.49	14.01
Asymmetry-40%	27.29	25.94	26.11	28.35	33.62	32.7	34.63

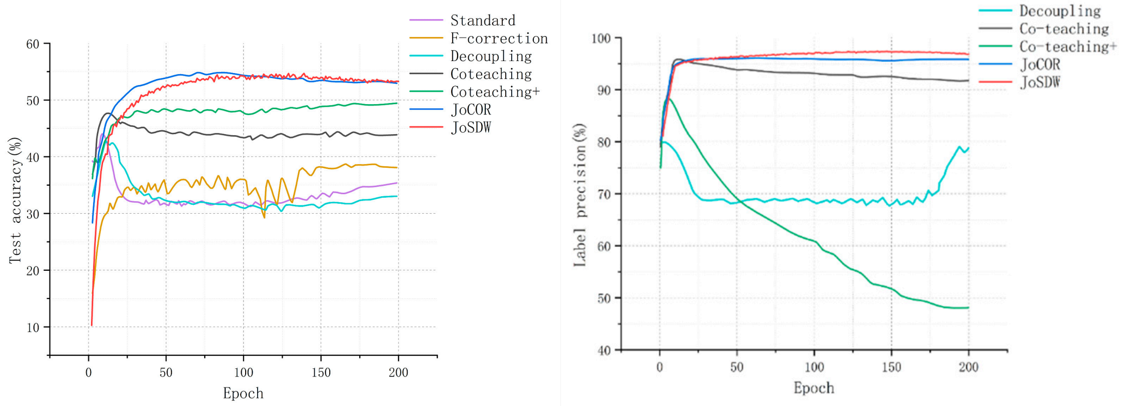

Figure 11. Results on CIFAR-100 dataset. Noise settings, Symmetry-20%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

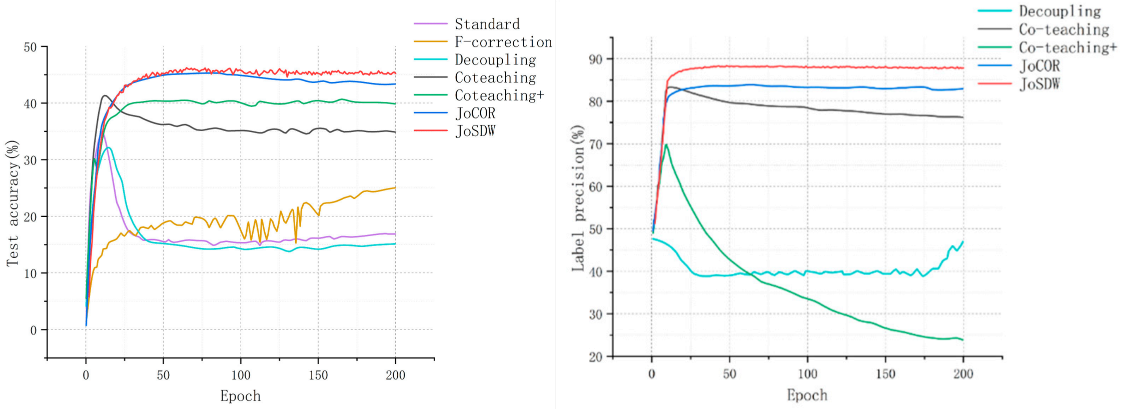

Figure 12. Results on CIFAR-100 dataset. Noise settings, Symmetry-50%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

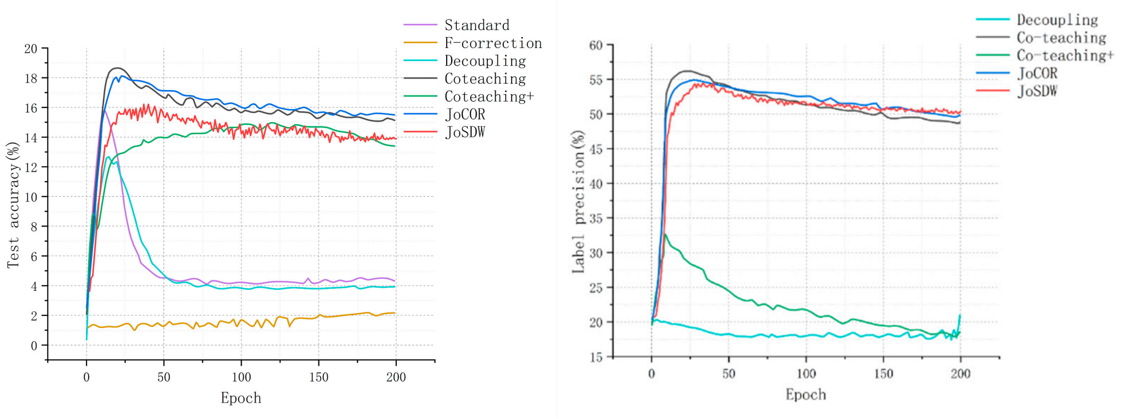

Figure 13. Results on CIFAR-100 dataset. Noise settings, Symmetry-80%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

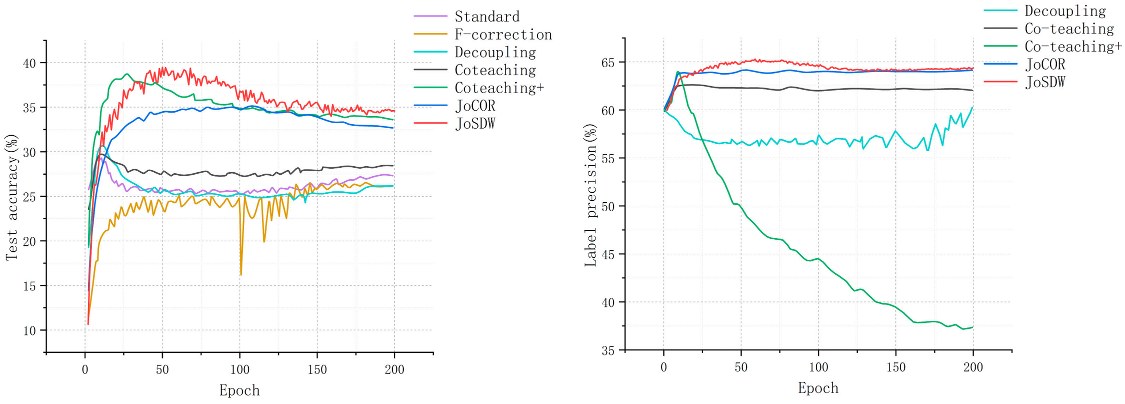

Figure 14. Results on CIFAR-100 dataset. Noise settings, Asymmetry-40%. **Left**: precision accuracy (%) vs. epochs; **Right**: label precision (%) vs. epochs.

However, JoSDW still achieves high precision accuracy on these datasets. In the simplest case of symmetric-20% and symmetric-50%, the effect of JoSDW is considerably better than other methods. In the most difficult case of symmetry-80%, JoSDW can still obtain higher precision accuracy, and JoCOR and co-teaching are combined.

4.4. Ablation Study

Ablation studies were carried out to determine the impacts of secondary subdivision and dynamic weighting, we conduct experiments on the MNIST dataset with Symmetry-50% noise and the CIFAR-10 dataset with Symmetry-20% noise. To eliminate the influence of secondary subdivision and dynamic weighting, we train all samples obtained by the small loss sample strategy. To verify the effect of dynamic weighting, we use fixed weighting coefficients for the pure sample and the complex sample. According to the previous analysis, these two methods should play their respective roles in the training process.

Sample selection: The state-of-the-art performance of JoSDW is largely due to precise and dependable sample selection. We use graphs and tables to show the accuracy of sample selection. The graph shows the accuracy of the clean sample selection to study and verify the advantage of the sample selection method. It can be seen from the figure that JoSDW is efficient in accurately and reliably selecting clean samples. In every case, JoSDW outperforms the most advanced sample selection algorithms in terms of picking clean sample data. In addition, in demanding scenarios (i.e., Asymmetry-40%), although all other methods have an impact on finding clean samples, the accuracy of JoSDW's clean sample selection steadily improves as training progresses. These findings support the validity of our clean sample selection method. The last period in this table shows the best periods and last period's selection accuracy, respectively. The results confirm the effectiveness of JoSDW in selecting pure samples and complex samples.

Tables 8 and 9 show the impact of various phases in our strategy. In training, the JoSDW-S denotes the case where pure samples and complex samples are used. In training, JoSDW-SD represents the utilization of clean samples and complex samples with dynamic weights. Finally, the JoSDW denotes the suggested approach in its ultimate form. The experimental process is shown in Figure 15.

Table 8. Average precision accuracy (%) on CIFAR-10 Symmetry-20% over the last 10 epochs.

Index	Methods	Label Precision
1	JoSDW-S	81.12
2	JoSDW-SD	84.96
3	JoSDW	86.28

Table 9. Average precision accuracy (%) on Mnist Symmetry-50% over the last 10 epochs.

Index	Methods	Label Precision
1	JoSDW-S	95.03
2	JoSDW-SD	96.66
3	JoSDW	97.23

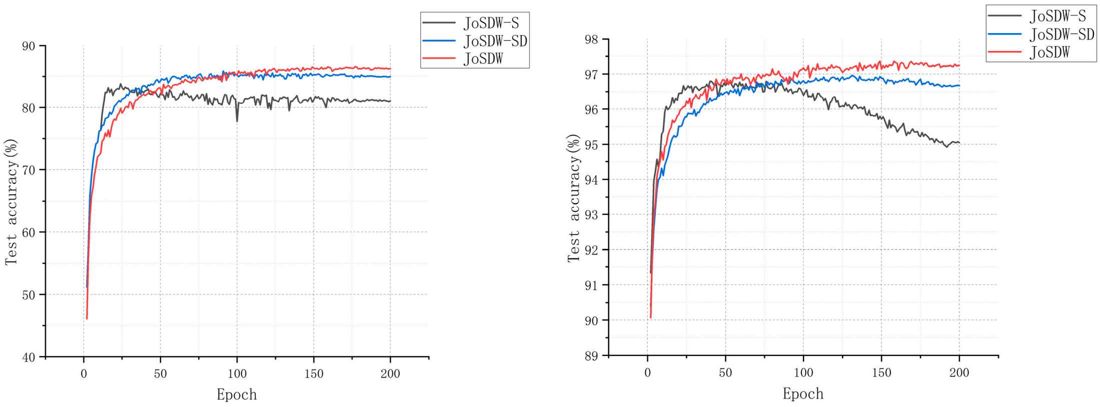

Figure 15. Left: Results on the CIFAR-10 dataset. Noise settings, Symmetry-20%. precision accuracy (%) vs. epochs; **Right**: Results on Mnist dataset Noise settings, Symmetry-50%. precision accuracy (%) vs. epochs.

5. Conclusions

In this section, we will discuss the problem of sample classification.

As shown in Figure 16, the pure samples always maintain a very high purity rate, while the performance of complex samples is complex, with a rapid decline from the average purity rate of the data. Dirty samples always have low purity. This result justifies our idea of distinguishing the two samples.

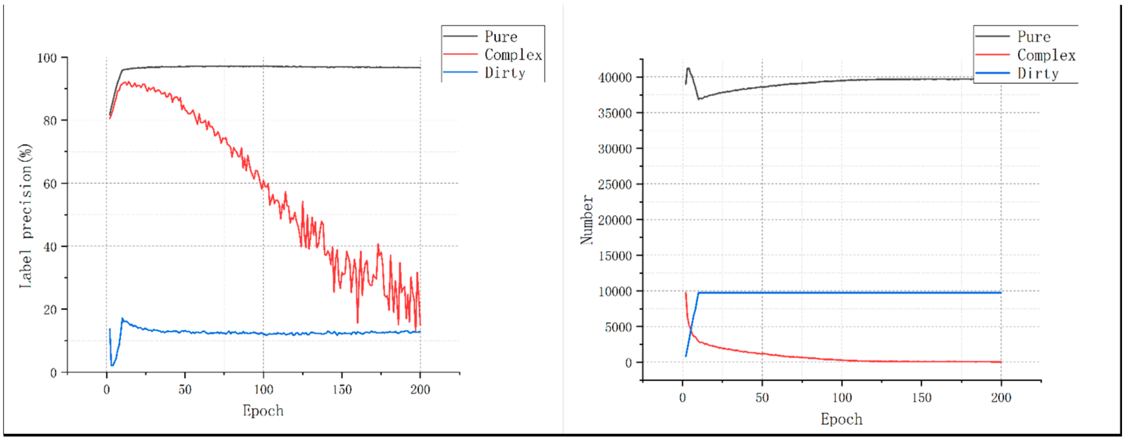

Figure 16. Results on the CIFAR-10 dataset. Noise settings, Symmetry-20%. **Left**: Label precision (%) vs. epochs; **Right**: Number vs. epochs.

The complex sample can guide model training at the beginning and solves the problem of "when to update". In the middle and late stages of the experiment, a part of the complex sample with the ability to guide the training of the model became the pure sample. The remaining complex samples have an increasing proportion of noisy labels. The complex samples gradually lose the ability to guide model updates.

This paper studies the problem of noisy label learning in deep learning, which is an important issue in the cheap and fast implementation of deep learning. This paper proposes a robust JoSDW to improve the performance of deep neural networks with noisy labels. The key idea of JoSDW is to train two neural networks with the same structure at the same time and select high clean sample labels through a small-loss sample selection strategy. Then, the samples obtained in the first screening continue to be classified by the prediction results of the two neural networks. Those with the same prediction result are pure samples, and those with different prediction results are called complex samples. Simultaneously update according to the respective cross-entropy loss and a joint loss. We conduct experiments on three datasets (MNIST, CIFAR-10, CIFAR-100) to prove that JoSDW can train the depth model robustly under the noisy label.

In the future, our work can be divided into three points: first, we will explore label correction methods to recover discarded low-confidence sample labels to improve the utilization of samples. Second, we will try an ensemble of noisy label learning methods, which will be combined with methods such as boosting ensembles [8] and Nested Dropout [18]; Third, the current methods of noisy learning are mainly applied to artificially set noisy datasets, and with the rise of unsupervised learning, we will focus on the combination of pseudo-labels generated by clustering and noisy learning.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; software, Y.Z.; validation, M.B.; formal analysis, J.X.; investigation, M.B.; resources, J.X.; data curation, M.B.; writing—original draft preparation, Y.Z.; writing—review and editing, J.X.; visualization, Y.Z.; supervision, H.X.; project administration, H.X.; funding acquisition, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://yann.lecun.com/exdb/mnist/, https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 30 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, W.; Wang, L.; Li, W.; Agustsson, E.; Van Gool, L. Webvision database: Visual learning and understanding from web data. *arXiv* **2017**, arXiv:1708.02862.
2. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 181–196.
3. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A. The open images dataset v4. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
4. Tanno, R.; Saeedi, A.; Sankaranarayanan, S.; Alexander, D.C.; Silberman, N. Learning from noisy labels by regularized estimation of annotator confusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11244–11253.
5. Arnaiz-González, Á.; Fernández-Valdivielso, A.; Bustillo, A.; de Lacalle, L.N.L. Using artificial neural networks for the prediction of dimensional error on inclined surfaces manufactured by ball-end milling. *Int. J. Adv. Manuf. Technol.* **2016**, *83*, 847–859. [CrossRef]
6. Jacob, E.; Astorga, J.; Jose Unzilla, J.; Huarte, M.; Garcia, D.; Lopez de Lacalle, L.N. Towards a 5G compliant and flexible connected manufacturing facility. *Dyna* **2018**, *93*, 656–662. [CrossRef]
7. Liu, S.; Niles-Weed, J.; Razavian, N.; Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *arXiv* **2020**, arXiv:2007.00151.
8. Andres, B.; Gorka, U.; Perez, J. Smart optimization of a friction-drilling process based on boosting ensembles. *J. Manuf. Syst.* **2018**, *48*, 108–121.
9. Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv* **2014**, arXiv:1412.6596.
10. Tanaka, D.; Ikami, D.; Yamasaki, T.; Aizawa, K. Joint optimization framework for learning with noisy labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5552–5560.

11. Arpit, D.; Jastrzębski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y. A closer look at memorization in deep networks. In Proceedings of the International Conference on Machine Learning, Singapore, 24–26 February 2017; pp. 233–242.
12. Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2304–2313.
13. Shen, Y.; Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In Proceedings of the International Conference on Machine Learning, Zhuhai, China, 22–24 February 2019; pp. 5739–5748.
14. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv* **2018**, arXiv:1804.06872.
15. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning, Zhuhai, China, 22–24 February 2019; pp. 7164–7173.
16. Malach, E.; Shalev-Shwartz, S. Decoupling "when to update" from "how to update". *arXiv* **2017**, arXiv:1706.02613.
17. Wei, H.; Feng, L.; Chen, X.; An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13726–13735.
18. Chen, Y.; Shen, X.; Hu, S.X.; Suykens, J.A.K. Boosting Co-teaching with Compression Regularization for Label Noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nashville, TN, USA, 19–25 June 2021; pp. 2688–2692.
19. Kumar, M.P.; Packer, B.; Koller, D. Self-Paced Learning for Latent Variable Models. In Proceedings of the NIPS, Vancouver, BC, Canada, 6–11 December 2010; p. 2.
20. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv* **2020**, arXiv:2002.05709.
21. Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
22. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; pp. 1735–1742.
23. Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J.T.; Peng, X. Contrastive clustering. In Proceedings of the 2021 AAAI Conference on Artificial Intelligence (AAAI), Virtual, 2–9 February 2021.
24. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
25. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
26. Gutmann, M.; Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Setti Ballas, Italy, 13–15 May 2010; pp. 297–304.
27. Blum, A.; Mitchell, T.M. Combining Labeled and Unlabeled Sata with Co-Training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, WI, USA, 24–26 July 1998.
28. Sindhwani, V.; Niyogi, P.; Belkin, M. A co-regularization approach to semi-supervised learning with multiple views. In Proceedings of the ICML Workshop on Learning with Multiple Views, Bonn, Germany, 11 August 2005; pp. 74–79.
29. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
30. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~{}kriz/learning-features-2009-TR.pdf (accessed on 30 December 2021).
31. Goldberger, J.; Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In Proceedings of the ICLR 2017 Conference, Toulon, France, 24–26 April 2017.
32. Kiryo, R.; Niu, G.; du Plessis, M.C.; Sugiyama, M. Positive-unlabeled learning with non-negative risk estimator. *arXiv* **2017**, arXiv:1703.00593.