Group-Teaching: Learning Robust CNNs From Extremely Noisy Labels

YUNPING ZHENG, YUMING CHEN, AND MUDAR SAREM
1School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
2School of Software Engineering, Huazhong University of Science and Technology, Wuhan 510006, China
3General Organization of Remote Sensing, Damascus 12586, Syria

Corresponding author: Yunping Zheng (zhengyp@scut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant No. 61300134, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120172120036, the Natural Science Foundation of Guangdong Province of China under Grant No. S2011040005815, No. S201301013215, No. 2015A030313206, and No. 2017A030313349, and the Fundamental Research Funds for the Central Universities of China under Grant No. 2015ZM133.

ABSTRACT Deep convolutional neural networks have achieved tremendous success in a variety of applications across many disciplines. However, their superior performance relies on correctly annotated large-scale datasets. It is very expensive and time-consuming to get the annotated large-scale datasets, especially in the medical field. While collecting a large amount of data is relatively easy, given the amount of data available on the web, but these data are highly unreliable, and they often include a massive amount of noisy labels. The past research works have shown that these noisy labels could significantly affect the performance of the deep convolutional neural networks on image classification. However, training a robust deep convolutional neural network with extremely noisy labels is a very challenging task. Inspired by the co-teaching concept, this paper proposes a novel method for training a robust convolutional neural network with extremely noisy labels, which is called group-teaching. Specifically, we train a group of convolutional neural networks simultaneously, and let them teach each other by selecting possibly clean samples for each network in each mini-batch. Each network back propagates the samples selected by other networks except itself and then it updates itself. The empirical results on noisy versions of CIFAR-10 and CIFAR-100 datasets demonstrate that our method is superior to the state-of-the-art methods in the robustness for noisy labels. Particularly, to verify the efficacy of our group-teaching in real-world noisy labels distribution, we have also validated the effectiveness of our method on the real-world noisy WebVision1000-100 dataset. The results show that our method has achieved higher performance than the state-of-the-art methods.

INDEX TERMS Noisy labels, group-teaching, web images, deep convolutional neural networks, image classification.

I. INTRODUCTION

In recent years, the deep convolutional neural networks have achieved tremendous success in numerous computer vision tasks, achieving state-of-the-art performance on image classification [1]–[6], object detection [7]–[10], semantic segmentation [11]–[14], and so on. Yet, the deep convolutional neural networks are mostly trained in a full-supervised manner, where the large-scale manually annotated datasets are required, such as ImageNet [15], MS-COCO [16], and PASCAL VOC [17]. However, collecting massive and annotated datasets is extremely expensive and time-consuming. Because the web crawler is a low-cost and efficient tool to collect a massive amount of webly labelled images [18], an alternative solution is to use the web as a source of data and the supervision for collecting a large amount of web images automatically from the Internet by using input queries, such as text information. Such queries can be considered as labels of the images. By using this way, it will be cheap and easy to obtain a large-scale dataset. Yet, such labels are highly unreliable and they often include a massive amount of noisy labels, such as the WebVision [18], which is a large-scale benchmark on 1000-category image classification task. It has the same categories as
ImageNet [15] and it contains about 2.44 million noisy-labelled training images crawled from Flickr and Google by using 1000 class definitions from ImageNet as queries keywords. Their labels are provided by using the queries text generated from the 1000 semantic concepts of ImageNet, without any manual annotation.

The past research works have shown that these noisy labels could significantly affect the performance of the deep convolutional neural networks on image classification, as the deep convolutional neural networks have high capacity to memorize massive data and to fit noisy labels [19]. Hence, an algorithm that is robust against noisy labels for deep convolutional neural networks is necessary to resolve this potential problem.

The effects of noisy labels were studied in a lot of works [20]. Also, various approaches have been proposed about how to train a robust convolutional neural network with noisy labels in the recent years [21]–[32]. To avoid the overfitting of the noisy labels, the authors in [21]–[23] designed the robust loss function in order to label the noise with theoretical guarantees. Some other methods focused on the noisy transition matrix [24], [25] to convert each label into another class by considering the probability for each label. The method presented in [24] proposed two procedures for loss correction in order to estimate the noisy transition matrix. The authors in [26]–[28] introduced the techniques to down-weight the noisy samples by re-weighting the training samples. Both methods in [26] and [27] required a smaller clean dataset to work. Another method in [28] derived an optimal important weighting scheme for noise-robust classification.

Other methods, such as decoupling [29] and co-teaching [30], re-sampled the possibly clean samples from the noisy data and updated the networks using these selected clean data. The decoupling method [29] trained two peer networks simultaneously, and updated these networks by only using the samples which had inconsistent classifications in the two networks. Similarly, the co-teaching method [30] also trained two peer networks simultaneously, but for every mini-batch, each network of the co-teaching fed forward all the data and selected its samples with small loss which were used to update its peer network for the further training. The success of the co-teaching method depended on the diversity of the two networks. However, as the training progressed, the two networks will gradually become more consistent, just as two students learn from each other and finally their understanding of things becomes consistent. This progress of the training will lead the co-teaching method to the inability to select the correct samples for each other.

In order to combat this drawback, we propose a novel method for training a robust convolutional neural network with extremely noisy labels, which is called group-teaching. Our method can be seen as the extension and generalization of the co-teaching method [30]. Differently from the co-teaching, our group-teaching trains a group of peer networks simultaneously and merges all of these networks outside of a selected network in order to teach that selected one. More specifically, in each mini-batch, each network is fed forward all the samples independently, and then, we merge the prediction results of all the networks except one network and select the small-loss samples according to the prediction results of all the merged networks to teach that network. Such as, for each mini-batch, we merge the prediction results \( \{p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n\} \) and generate an average prediction \( \hat{p}_i^{\text{avg}} \), given the prediction results \( \{p_1, p_2, \ldots, p_n\} \) of a group of classifications \( \{h_1, h_2, \ldots, h_n\} \). We calculate the loss of every sample according to the prediction \( \hat{p}_i^{\text{avg}} \) and select the small-loss samples as the possible clean samples to teach a classifier \( h_i \) for the further training by back propagating the loss of the selected samples. During the training process, our group-teaching can better maintain the diversity between all the peer networks, and make each network has different learning abilities. In the later stages of the training process, our method is better than co-teaching method in selecting the correct samples, and that is why our method is more robust to noisy labels than the co-teaching method.

We evaluate our proposed group-teaching on the same benchmarks used in [30] (i. e., the noisy versions of CIFAR-10 [33] and CIFAR-100 [34]). Particularly, we conduct experiments on the WebVision1000-100 dataset with real-world noisy labels, which is a 100-category subset of WebVision [18] built by random selection. Due to limited GPU resources, we do not directly perform experiments on WebVision with 2.44 million images of real-world noisy labels. However, our empirical results on noisy versions of CIFAR-10, CIFAR-100 and WebVision1000-100 datasets demonstrate that our method is superior to the state-of-the-art methods in the robustness for noisy labels.

II. RELATED WORK

Learning from noisy labels is a widely studied topic [20]. In fact, many research works have been performed about how to learn a robust convolutional neural network from noisy labels in the recent years [21]–[30]. The existing methods to learn from noisy labels can be roughly classified into several categories as follows:

A. LOSS FUNCTION

To avoid the network overfitting the noisy labels, some research works [22], [23] designed the robust loss function in order to label the noise with theoretical guarantees. Their attention was given to directly formulating a noise-robust loss function in the context of deep convolutional neural networks. Ghost et al. [22] proposed mean absolute error (MAE) as a noise-robust alternative to the commonly-used categorical cross entropy (CCE). In fact, these methods were robust to noisy labels, but they were slow in convergence and caused increased difficulty in the training process. Also, Zhang and Sabuncu [21] proposed a generalized loss function which could be seen as an extension and generalization of the mean absolute error and the categorical cross entropy.
B. NOISY TRANSITION MATRIX

Some methods focus on the noisy transition matrix [24], [25] to convert each label into another class by considering the probability for each label. Patrini et al. [24] proposed two procedures for loss correction, provided that we knew a stochastic matrix summarizing the probability of one class being flipped into another under noise. They converted each label into another class according to the noisy transition matrix. Hendrycks et al. [25] utilized trusted data by proposing a loss correction technique to mitigate the effects of the label noise on deep convolutional neural networks. Yet, it is currently challenging to estimate the noisy transition matrix for these methods.

C. RE-WEIGHTING

The authors in [26]–[28] introduced the techniques to down-weight the noisy ones by re-weighting the training samples. The idea of re-weighting each training sample has been well studied in the literature [10], [39], and [40]. A technique called MenterNet [27] used a long short-term memory (LSTM) network [38] as a mentor network to dynamically learn a curriculum. This technique weighted the training samples and used to guide the student network. Specifically, the mentor network was pre-trained with a small clean validation data otherwise, it must pre-define a curriculum. Similarly, Ren et al. [26] proposed an online reweighting method which leveraged an additional small clean validation set to adaptively assign importance weights for training samples in each mini-batch. Both of the previous methods required a small clean dataset to work. Another method presented in [28] derived an optimal importance weighting scheme for noise-robust classification.

D. RESAMPLING

Other methods, such as decoupling [29] and co-teaching [30], re-sampled the possibly clean data from the noisy training samples and updated the network by back propagating these possibly clean data. The decoupling method [29] trained two peer networks simultaneously, and updated networks by only using the samples which had inconsistent predictions from two networks. However, the prediction of the noisy samples was irregular and the disagreement area included a number of noisy samples. Thus, the decoupling technique could not deal with the noisy labels, and that was proved by our experimental results. Similarly, to the decoupling, the co-teaching method [30] trained also two peer networks simultaneously. However, in each mini-batch, the two networks of the co-teaching were fed forward by all data and then, they selected their small-loss samples to cross-update their peer networks for further training. Compared to the decoupling method, the success of the co-teaching method depends on the diversity of the two networks. Since two networks in co-teaching have different learning abilities to filter different types of noisy labels. However, as the training progresses, the two networks will gradually become consistent, just as two students learn from each other, and finally their understanding of things becomes consistent, that will lead that the peer networks of the co-teaching method are incapable of selecting the correct samples for each other.

There are other works to relieve the impact of noisy labels, such as the works resented in [41] and [42]. In addition, differently from the above methods, Guo et al. [31] did not propose a noise-cleaning or robust loss function method. Instead, they introduced a new training strategy to improve standard network capability by leveraging curriculum learning. They explored designing a learning curriculum where all training images in each category were split into three subsets by applying a density-based clustering algorithm that measured the complexity of the training samples by using data distribution density. Reed et al. [32] proposed changing of the cross-entropy loss function by adding a regularization term that took into account the current prediction of the network, which combined the ground truth and the predictions as the final labels, and fed back the propagation. In literature [43], Li et al. presented a unified framework to distill the knowledge from the clean samples and a knowledge graph to learn a better model from the noisy labels. Tanaka et al. [45] introduced joint optimization framework for learning parameters and estimating true labels of each training sample simultaneously. Veit et al. [44] learned a noise-label cleaning network by leveraging an additional small subset of the dataset to reduce the noise in large-scare noisy labels. In addition, the solutions of regularization were also proposed to strongly constraint the model from overfitting outliers [49] and [50].
III. OUR METHOD

As mentioned before, in order to address the problem of noisy labels, we propose group-teaching method, which trains a group of peer networks simultaneously that cross-teach each other. Pipeline of group-teaching is described in Figure 1.

More specifically, in every mini-batch, each network is fed forward by all samples independently, and then, we merge the prediction results of all networks except one network forward by all samples independently, and then, we merge the prediction results of all networks except one network to teach this network. For the prediction results of all networks except one network to teach this network. For the prediction results of all networks except one network to teach this network. For the prediction results of all networks except one network to teach this network. For the prediction results of all networks except one network to teach this network.

A. DESCRIPTION OF OUR ALGORITHM

In order to better illustrate our algorithm, we define the following symbols. Let \( \{h_1, h_2, \ldots, h_n\} \) denote a group of networks. \( \{p_1, p_2, \ldots, p_n\} \) are the prediction results of \( \{h_1, h_2, \ldots, h_n\} \). \( D \) is the training set with noisy labels, and \( D_{\text{batch}} \) denotes a mini-batch from \( D \). In each mini-batch \( D_{\text{batch}} \), all the networks are fed forward by all data. For \( h_j \), we merge the prediction results \( \{p_{1}, \ldots, p_{j-1}, p_{j+1}, \ldots, p_{n}\} \) of all other networks except \( h_j \) to generate an average prediction result \( p_j^{\text{avg}} \), as it can be seen in step 9 of Algorithm 1.

In step 10 of Algorithm 1, we select small-loss samples for \( h_j \) according to the loss based on the average prediction result \( p_j^{\text{avg}} \). Firstly, we compute the loss of every sample in each mini-batch using the cross-entropy loss function, and then we rank them according to the loss from small to large and select top percentage \( R(e) \) of the small-loss samples as a useful knowledge to teach \( h_j \) updating its parameters.

The selection ratio \( R(e) \) will be discussed in late sections. And in step 11 of algorithm 1, we update \( h_j \) using the small-loss samples which are selected by \( \{h_1, \ldots, h_{j-1}, h_{j+1}, \ldots, h_n\} \).

In addition, we control the number of samples which should be selected by \( R(e) \). For each current epoch \( e \), we set the selection rate \( R(e) \) and only select top percentage \( R(e) \) of the small-loss samples in each mini-batch. For the \( R(e) \), we follow the same setting presented in [30]. The work on the memorization effect of the deep convolutional neural networks showed that the networks could firstly memorize the clean and easy samples from the training set before the noisy labels [37]. Therefore, it is not necessary to drop out the training samples at the beginning of the training process. Therefore, the initial value of \( R(e) \) equals to 1.0, denoting that all samples in the training set will be selected. Also, we set a threshold \( E_k \) which denotes maximum decay epoch of \( R(e) \), where \( R(e) \) is expressed as follows, given current epoch \( e \) and the noise rate \( \varepsilon \).

\[
R(e) = 1 - \varepsilon \cdot \min\left(\frac{e}{E_k}, 1\right).
\] (1)

In other words, at the beginning of the \( E_k \) epoch, the drop rate increases linearly from 0 to \( \varepsilon \). When \( e \) is greater than \( E_k \), the drop rate remains \( \varepsilon \). Based on the above consideration, the pseudo-code of our group-teaching algorithm is presented in Algorithm 1.

In co-teaching, as the training progresses, two networks will gradually converge to a consensus and finally their understanding of things becomes consistent, leading them to inability to select the correct samples for each other. Compared to the co-teaching method, our group-teaching method can better maintain the diversity between all the networks.
and make each network has different learning ability. In the later stages of the training, our method is better than the co-teaching method in selecting the correct samples, and that is why our method is more robust to noisy labels than the co-teaching method.

B. DETERMINING THE NUMBER OF NETWORKS

In order to determine the number of networks in our group-teaching method, we set up a set of experiments with the number of networks on CIFAR-10 noisy version and WebVision1000-100 datasets. The experimental results are shown in Tables 1 and Table 2 respectively. Top1 and Top5 both mean the test accuracy, which denotes the ratio of the number of samples that are correctly predicted to the total number of test dataset. And the correct prediction of top1 and top5 is defined as the label corresponding to the maximum value is consistent with trust label and five labels corresponding to the maximum first five values contain the trust label, respectively. As we can see in Table 1 and Table 2, when the number of networks is less than or equal to four, group-teaching with four networks can achieve best performance, and compared to group-teaching with four networks, when the number of networks is greater than four, the performance of group-teaching is not significantly improved. In addition, as the number of networks increases, the training time will gradually increase. Based on the trade-off between performance and training time, we determine the number of networks to be four.

### IV. EXPERIMENTAL RESULTS

In this section, we empirically evaluate our method on the closed-set [42] noisy versions of CIFAR-10 [33] and CIFAR-100 [34] datasets, and the open-set [42] real-world noisy dataset WebVision1000-100, which is a 100-category subset of WebVision [18]. WebVision is a large-scale benchmark on 1000-category image classification task. It has the same categories as ImageNet [15], and it contains about 2.44 million images with real-world noisy labels collected by using input queries from Internet. The labels of training samples are provided by using the queries text generated from the 1000 semantic concepts of ImageNet [15], without any manual annotation. Since the GPU resources are limited, we do not directly conduct our experiments to verify the efficacy of our group-teaching on WebVision with 2.44 million images of real-world noisy labels.

A. THE METHODS FOR COMPARISON

In addition, in order to verify the efficacy of our group-teaching method, we compare it with following state-of-the-art methods: (1). The standard, which only trained a single network with noisy datasets using simple training strategy. (2). The decoupling method [29], which trained two peer networks simultaneously, and updated the networks by only using the samples which had inconsistent predictions from the two trained networks. (3). The co-teaching method [30], which also trained two peer networks simultaneously, but each network of the co-teaching was fed forward by all data and, then it selected its small-loss samples to update its peer network for further training. (4). Our group-teaching method, which trains a group of peer networks simultaneously and makes each network has different learning ability. In the later stages of the training, our method is better than the co-teaching method in selecting the correct samples, and that is why our method is more robust to noisy labels than the co-teaching method.

B. BENCHMARK DATASETS

In all of our experiments, we consider three image classification benchmark datasets: CIFAR-10 [33], CIFAR-100 [34], and the real-world noisy WebVision1000-100 dataset, which is a 100-category subset of WebVision [18]. The CIFAR-10 and CIFAR-100 datasets consist of 50,000 color training images of size $32 \times 32$ having 10 and 100 categories, respectively. Also, both datasets contain 10,000 test images with data balanced. The WebVision [18] is a large noisy labels benchmark containing about 2.44 million color training images with real-world noisy labels, crawled from the Internet using the 1,000 semantic concepts of ImageNet [15], without any manual annotation. And it contains 50,000 validation images (50 images per category) manually labeled. Due to the limited number of the GPU resources, we validate our method in the WebVision1000-100, which is a 100-category subset of WebVision and contains about 240,000 color training images.
images with noisy labels and 5,000 color validation images.

CIFAR-10 and CIFAR-100 are clean datasets. Yet, we need to evaluate the robustness of the proposed method against the noisy labels. Therefore, we simulate the label noises in the training set. We follow the same setting presented in [30], where the datasets were corrupted by the noisy transformation matrix and the clean label $y$ was flipped to the noisy label $y_{\text{noisy}}$. This is as same as the symmetry flipping [46] which is a simulation of fine-grained classification with noisy labels. Specifically, given $c$ categories, we randomly select for the noise rate $\varepsilon$ training images in each category, then we randomly and evenly give these images other $c-1$ category labels. The transformation matrix $Q$ of the symmetry flipping can be expressed as follows where $\varepsilon$ is the noise rate and $c$ is number of the class.

$$Q = \begin{bmatrix}
1 - \varepsilon & \varepsilon / (c - 1) & \ldots & \varepsilon & \varepsilon \\
\varepsilon / (c - 1) & 1 - \varepsilon & \ldots & \varepsilon & \varepsilon \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\varepsilon / (c - 1) & \ldots & \varepsilon / (c - 1) & 1 - \varepsilon & \varepsilon \\
\varepsilon / (c - 1) & \ldots & \varepsilon / (c - 1) & \varepsilon & 1 - \varepsilon
\end{bmatrix}.$$  \hspace{1cm} (2)

Since we focus on the robustness of our group-teaching method on extremely noisy labels, we verify the robustness of group-teaching on high-level noisy labels with a noise rate equals to 0.5. In addition, as a side product, we also verify the effectiveness of our method on low-level noisy labels with a noise rate equals to 0.2. In our experiments, we use the symmetry flipping [46] structures to build the noisy version of the CIFAR-10 and CIFAR-100 datasets with the noise rate $\varepsilon$ which is chosen from $\{0.2, 0.5\}$.

C. NETWORK ARCHITECTURE

For fairly comparing to other methods, we employ the same network architecture of the works [30], [35], and [36]. A 9-layer convolutional neural network architecture (with the detail of this architecture) is shown in Table 3. We implement this 9-layer convolutional neural network architecture which is a standard test bed for weakly-supervised. After each convolutional layer, we follow a batch normalization layer [3] and a leaky-relu activation function [47] with slope of 0.01. Also, after the first two three-layer convolutions, we use a max pooling layer and a dropout layer. Before the dense layer, we use an average pooling.

D. THE EXPERIMENTAL SETTING AND PARAMETERS

For all of the experiments, we employ the stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.1, a weight decay of $10^{-4}$, and a momentum of 0.9. For fair comparison, we implement our method and all the other compared methods with default parameters by PyTorch and perform all the experiments on a NVIDIA TITAN Xp GPU. In all of our experiments, we calculate the average test accuracy over the last ten epochs. In order to prevent the randomness of training, we did five experiments and took the average of the five experimental results as the final result.

For the CIFAR-10 and CIFAR-100 datasets, we train 350 epochs in total and the learning rate is divided by 10 at the 150th and 250th epochs and the batch size is set to 64. Before feeding the networks, the input images can be pre-processed by generating image translations, horizontally flipping, and normalizing with the mean $0.4914, 0.4822,$ and $0.4465$ and the standard deviation $0.2023, 0.1994, 0.2010$ for each RGB channel of the CIFAR-10 and CIFAR-100 datasets. For the open-set real-world noisy WebVision 1000-100 dataset, we use the ResNet-18 architecture [4] and train 200 epochs in total with batch size of 32, an initial learning rate of 0.1 and the learning rate is divided by 10 at 40th, 80th, and 120th epoch, respectively. We use single data augmentation for the input images by generating image translations and horizontally flipping, randomly cropping $224 \times 224$ patches, normalizing with the mean $0.485, 0.456,$ and $0.406$ and the standard deviation $0.229, 0.224,$ and $0.225$ for each RGB channel and performing fancy PCA [48] before feeding the networks by the input images.

For another experimental setup, we only use the noise rate $\varepsilon$ of 0.2 and 0.5, and the maximum decay epoch $E_k$ of 10 for CIFAR-10 and CIFAR-100. For WebVision1000-100, we set the noise rate $\varepsilon$ as 0.2 and the maximum decay epoch $E_k$ as 50, since this task is more difficult than that performed on the previous for CIFAR-10 and CIFAR-100 datasets. The $R(\varepsilon)$ can be computed by using Eq. (1).

The performance metrics in this paper follow the ones presented in [30] (i.e., using the test accuracy and true positive rate (TPR), where true positive rate (TPR) indicates the correct ratio in the selected images. The definition of the TPR is given as follows:

$$TPR = \frac{\text{number of clean labels}}{\text{number of selected labels}}.$$  \hspace{1cm} (3)

| Table 3. 9-layer CNN architecture used in our experiments on the CIFAR-10 and CIFAR-100. |
|---------------------------------------------------------------|
| CIFAR-10 | CIFAR-100 |
| 32 x 32 RGB Image | 32 x 32 RGB Image |
| 3 x 3 conv, 128 LReLU | 3 x 3 conv, 128 LReLU |
| 3 x 3 conv, 128 LReLU | 3 x 3 conv, 128 LReLU |
| 2 x 2 max pooling, stride 2 | 2 x 2 max pooling, stride 2 |
| dropout, p = 0.25 | dropout, p = 0.25 |
| 3 x 3 conv, 256 LReLU | 3 x 3 conv, 256 LReLU |
| 3 x 3 conv, 256 LReLU | 3 x 3 conv, 256 LReLU |
| 2 x 2 max pooling, stride 2 | 2 x 2 max pooling, stride 2 |
| dropout, p = 0.25 | dropout, p = 0.25 |
| 3 x 3 conv, 512 LReLU | 3 x 3 conv, 512 LReLU |
| 3 x 3 conv, 256 LReLU | 3 x 3 conv, 256 LReLU |
| 3 x 3 conv, 128 LReLU | 3 x 3 conv, 128 LReLU |
| avg-pooling | avg-pooling |
| dense 128 $\rightarrow$ 10 | dense 128 $\rightarrow$ 100 |
TABLE 4. The average test accuracy on CIFAR-10 over the last ten epochs.

| Method     | Standard | Decoupling | Co-teaching | Group-teaching |
|------------|----------|------------|-------------|---------------|
| Symmetry-20% | 78.33    | 80.09      | 88.45       | 89.50         |
| ±0.15      | ±0.09    | ±0.11      | ±0.12       |               |
| Symmetry-50% | 49.45    | 51.15      | 82.25       | 83.29         |
| ±0.18      | ±0.20    | ±0.23      | ±0.28       |               |
| Symmetry-70% | 25.34    | 26.55      | 61.24       | 71.46         |
| ±0.28      | ±0.26    | ±0.41      | ±0.39       |               |

where number of selected labels is the number of selected images and number of clean labels is the number of the really clean images in the number of selected labels.

E. EXPERIMENTAL RESULTS ANALYSIS

In this sub-section, we discuss the experimental results of this paper performed on the noisy versions of CIFAR-10 and CIFAR-100 datasets and on the real-world noisy WebVision1000-100 dataset.

1) RESULT ON CIFAR-10

In Table 4, the experimental results show that our group-teaching method consistently outperform the decoupling and the co-teaching methods in the extremely noisy CIFAR-10 dataset. Specifically, on the symmetry case with 0.2 noise rate, our proposed group-teaching method achieves about 1% higher test accuracy than the co-teaching method and about 9.5% higher test accuracy than the decoupling method. Also, on the harder symmetry flipping case with 0.5 noise rate, the decoupling method doesn’t work, but our group-teaching can still achieve 1% higher test accuracy than the co-teaching method. In addition, we conduct the experiments on the extremely noisy CIFAR-10 with 70 percent noisy labels. On the symmetry flipping case with 0.7 noise rate, the decoupling and the co-teaching methods fail to work, but our group-teaching can still achieve 1% higher test accuracy than the co-teaching method.

We show test accuracy curve in Figure 2. In all cases, we can see the effects on the performance when networks fit with the noisy labels. In symmetry-20% and symmetry-50%, the standard and the decoupling methods, as the training progressed, the networks gradually fit with the noisy data, and then the performance of the networks gradually decreases starting nearly from the 20th epoch and the 50th epoch respectively. As better robust methods for noisy labels than the standard and the decoupling methods, the co-teaching and the group-teaching can alleviate this problem. Also, our group-teaching can achieve higher test accuracy compared with the co-teaching. In symmetry-70% with extremely noisy labels, the standard and the decoupling show serious overfitting, and finally converge to a very low test accuracy, which demonstrate that the neural networks of these two methods cannot learn the correct visual representation from extremely noisy labels. Also, the test accuracy curve of group-teaching is still relatively stable and can converge to a higher test accuracy.

In order to further explain why our method can achieve better performance than the other methods, we show in Figure 3 the true positive rate (TPR) on the CIFAR-10 dataset in order to compare the ability of selecting correct clean samples on different methods. As it can be seen from Figure 2, the decoupling fails to select the correct samples. But the co-teaching and the group-teaching can successfully pick up the correct clean samples. Although at the beginning of the training, the TPRs of our group-teaching method and co-teaching method are close to each other in selecting correct clean samples, but as the training progresses, the potential problem of the co-teaching (i.e., as the training progresses, the two networks will gradually become consistent and have the inability to select the correct samples for each other) gradually appears, and our group-teaching method can alleviate this problem and achieve higher true positive rate (TPR), resulting in higher test accuracy. Thus, our approach can better maintain the diversity of the network and keep different learning skills of each network.

2) RESULT ON CIFAR-100

As we can see in Table 5, the conclusion on CIFAR-100 dataset is consistent with that on CIFAR-10 dataset.
Compared with the co-teaching method, our proposed group-teaching method can achieve about 1% higher test accuracy than the co-teaching method. However, the decoupling method does not work, regardless of the noise ratio of 0.2 or 0.5, as the classification task on CIFAR-100 is harder. Similarly, in Figure 4 and 5, we show the test accuracy and the true positive rate (TPR) on CIFAR-100 dataset, respectively. As shown in Figure 5, our notes are the same as these for CIFAR-10 dataset. The decoupling fails to pick up the correct samples, but the co-teaching and the group-teaching successfully pick up the correct clean samples. Also, the group-teaching achieves higher performance than the other methods.

In addition, from Figure 2 and 4, we can see that the test accuracy is improved abruptly in the 150th epoch, since the learning rate is divided by 10. In general, in manual closed-set noisy datasets, our group-teaching method achieves higher performance compared with the other state-of-the-art methods.

3) RESULT ON WebVision1000-100
In literature [30], Han et al. only conducted the experiments on the manual closed-set noisy datasets, such as the noisy version of CIFAR-10 and CIFAR-100 datasets, but they did...
Y. Zheng et al.: Group-Teaching: Learning Robust CNNs From Extremely Noisy Labels

FIGURE 6. Test accuracy of standard, decoupling, co-teaching, and group-teaching on WebVision1000-100.

TABLE 6. The performances on WebVision1000-100 with different noise rate $\varepsilon$.

| noise rate $\varepsilon$ | 0.1     | 0.2     | 0.3     | 0.4     |
|-------------------------|---------|---------|---------|---------|
| Top1                    | 78.72   | 79.32   | 78.40   | 78.18   |
| Top5                    | 92.34   | 92.96   | 90.48   | 90.80   |

TABLE 7. The average test accuracy on WebVision1000-100 over the last ten epochs.

| Method       | Standard | Decoupling | Co-teaching | Group-teaching |
|--------------|----------|------------|-------------|----------------|
| Noise rate 20%| 78.26    | 72.40      | 79.32       | 80.36          |
| Std          | ±0.15    | ±0.21      | ±0.20       | ±0.18          |

not verify the efficacy of the co-teaching method in real-world noisy scenario. In order to verify the robustness of our group-teaching in real-world noisy labels distribution, we perform experiments on the open-set [42] real-world noisy WebVision1000-100 dataset, which is a 100-category subset of the WebVision [18]. Since we do not know the noise rate of the WebVision1000-100, we set up a set of experiments on the noisy ratio based on co-teaching method. As we can see in Table 6, when the noise rate is 0.2, the co-teaching method can achieve the best performance compared with different noise rates. Based on this set of experiments, all our experiments use a noise ratio of 0.2 on WebVision1000-100.

The experimental results are shown in Table 7. As it can be seen from Table 7, the group-teaching method achieves higher performance than the other methods, i.e. an average test accuracy of 80.36% for group-teaching vs. 79.32% for co-teaching). Also, as we can see in Figure 6, the decoupling method did not converge on WebVision1000-100 dataset. It failed to deal with the open-set real-world noisy labels distribution. Surprisingly, the standard method achieves a relatively good test accuracy and does not cause a significant overfitting, since this task on the WebVision1000-100 dataset is too difficult and the capacity of the ResNet-18 [4] is too small to completely fit the noisy data. However, the group-teaching and the co-teaching methods show better performance than the Standard method. Among them, the group-teaching performance is the best.

TABLE 8. On the symmetry flipping case with 0.5 noise rate, the performance on CIFAR-10 noisy version with the different network architectures.

| architecture      | Top1   | Co-teaching | Group-teaching |
|--------------------|--------|-------------|----------------|
| CNN9               | 49.45  | --          | --             |
| Res50              | 49.08  | 79.98       | 83.29          |
| Res50+Res50        | 76.01  | --          | --             |
| CNN9+CNN9          | 83.29  | --          | --             |
| Res50+CNN9         | 78.96  | 89.24       | 83.11          |

TABLE 9. The test accuracy of each network on decoupling, co-teaching, and group-teaching on noisy version of CIFAR-10 dataset.

| Dataset          | Network id | Decoupling | Co-teaching | Group-teaching |
|------------------|------------|------------|-------------|----------------|
| CIFAR-10 with    | Network-0  | 80.09      | 88.25       | 89.06          |
| Symmetric-0.2 noise rate | Network-1 | 79.98      | 88.45       | 89.45          |
| Symmetric-0.5 noise rate | Network-2 | --         | --          | 89.50          |
| Mean             | 80.04      | 88.35      | 89.24       | 83.12          |
| Std              | 0.05       | 0.10       | 0.24        | 0.12           |

In summary, in all of our experiments, our proposed group-teaching method achieves better performance than the other competitors.

F. DISCUSSING NETWORK ARCHITECTURE

We perform the experiments with the different network architectures on CIFAR-10 noisy version on the symmetry flipping case with 0.5 noise rate. We use two networks, Resnet-50 [4] and CNN9 [30], to randomly match and conduct 5 experiments. As we can see in Table 8, the experiment with two CNN9 achieves the best performance and although the network of Resnet50 is more powerful than CNN9, in the noisy version of CIFAR-10 dataset, the performance of Resnet50 is worse than CNN9. This may be because Resnet50 is so powerful that it can remember too many noisy labels.

G. DISCUSSING THE DIVERSITY OF EACH NETWORK

In this section, we intuitively discuss the diversity of each network in different methods. In Table 9 and 10, we report the individual performance of each network and the mean and standard deviation in decoupling method, co-teaching method and our proposed group-teaching method. As shown in Table 9 and 10, group-teaching achieve best performance and maximum standard deviation, which intuitively
In this section, we analyze the impact of noisy labels on image classification in more detail. We conduct the experiments of a single model on CIFAR datasets with all clean labels and noisy labels, separately. We report the performances on CIFAR-10 and CIFAR-100 with different noisy rate in Figure 7 (a) and (b), where standard-clean and standard-symmetric-∗ separately represent the test accuracy curve for training a single network on the origin CIFAR-10 and CIFAR-100 with all clean labels and on the noisy version of CIFAR-10 and CIFAR-100 with symmetric-∗ noisy labels. As shown in Figure 7 (a) and (b), compared to the performance on datasets with all clean label, the performance of the neural network is greatly affected, even with only 20% noisy labels. When the noisy labels reach 70 percent, the neural networks can no longer work. Therefore, we do not consider it necessary to perfume experiments with noise rate greater than 70 percent.
S. Hong, H. Noh, and B. Han, “You only look once: Unified, real-time object detection,” in Proc IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 371–385.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2999–3007.

E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 3431–3440.

S. Hong, H. Noh, and B. Han, “Decoupled deep neural network for semi-supervised semantic segmentation,” in Proc. Neuro Inf. Process. Syst., vol. 1, 2015, pp. 1495–1503.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” 2014, arXiv:1412.7062. [Online]. Available: https://arxiv.org/abs/1412.7062

K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask CNN,” in Proc IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980–2988.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Jun. 2009, pp. 248–255.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. ECCV, 2014, pp. 740–755.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010.

W. Li, L. Wang, W. Li, E. Agustsson, J. Berent, A. Gupta, R. Sukthankar, and L. Van Gool, “WebVision challenge: Visual learning and understanding with Web data,” 2017, arXiv:1705.05640. [Online]. Available: http://arxiv.org/abs/1705.05640

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” 2016, arXiv:1611.03530. [Online]. Available: http://arxiv.org/abs/1611.03530

B. Frenay and M. Verleysen, “Classification in the presence of label noise: A survey,” IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 5, pp. 845–869, May 2014.

Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in Proc. Int. Conf. Intell. Inform. Process. Syst. (NeurIPS), 2018, pp. 8792–8802.

A. Ghosh, H. Kumar, and P. S. Sastry, “Robust loss functions under label noise for deep neural networks,” in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 1919–1925.

A. Ghosh, N. Manwani, and P. S. Sastry, “Making risk minimization tolerant to label noise,” Neurocomputing, vol. 160, pp. 93–107, Jul. 2015.

G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2233–2241.

D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, “Using trusted data to train deep networks on labels corrupted by severe noise,” in Proc. Int. Conf. Intell. Inform. Process. Syst., 2018, pp. 10477–10486.

M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in Proc. Int. Conf. Mach. Learn., vol. 80, 2018, pp. 4334–4343.

L. Jiang, Z. Zhou, T. Leung, J.-L. Li, and L. Fei-Fei, “MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in Proc. Int. Conf. Mach. Learn., vol. 80, 2018, pp. 2304–2313.

T. Liu and D. Tao, “Classification with noisy labels by importance reweighting,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 3, pp. 447–461, Mar. 2016.

E. Malach and S. Shalev-Shwartz, “Decoupling when to update from ‘how to update’,” in Proc. Int. Conf. Intell. Inform. Process. Syst., 2017, pp. 961–971.

B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I.-W. H. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in Proc. Int. Conf. Intell. Inform. Process. Syst., 2018, pp. 8536–8546.

S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, “CurriculumNet: Weakly supervised learning from large-scale Web images,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 139–154.
YUMING CHEN was born in Guangdong, China, in 1993. He received the B.E. degree in network engineering from South China Normal University, Guangzhou, in 2017. He is currently pursuing the master’s degree with the Department of Computer Technology, South China University of Technology.

His research interests include image processing and computer vision.

MUDAR SAREM received the B.S. degree in electronic engineering from Tishreen University, Lattakia, Syria, in 1989, and the M.S. and Ph.D. degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2002, respectively.

He is currently an Associate Professor with the School of Software Engineering, Huazhong University of Science and Technology. He has published more than 80 articles in refereed conferences and journals. His research interests include image processing, computer networks, and distributed systems.