IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Training Robust Deep Neural Networks on Noisy Labels Using Adaptive Sample Selection with Disagreement

HIROSHI TAKEDA[1],SOH YOSHIDA[2], AND MITSUJI MUNEYASU[2], (Member, IEEE)
[1]Graduate School of Science and Engineering, Kansai University, 3-3-35 Yamate-cho, Suita-shi, Osaka, Japan
[2]Faculty of Engineering Science, Kansai University, 3-3-35 Yamate-cho, Suita-shi, Osaka, Japan

Corresponding author: Soh Yoshida (e-mail: sohy@kansai-u.ac.jp).

This research was financially supported by Kansai University ORDIST Research Project. We thank Kimberly Moravec, PhD, from Edanz Group (https://jp.edanz.com/ac) for editing a draft of this manuscript.

ABSTRACT Learning with noisy labels is one of the most practical but challenging tasks in deep learning. One promising way to treat noisy labels is to use the small-loss trick based on the memorization effect, that is, clean and noisy samples are identified by observing the network's loss during training. Co-teaching+ is a state-of-the-art method that simultaneously trains two networks with small-loss selection using the "update by disagreement" strategy; however, it suffers from the problem that the selected samples tend to become noisy as the number of iterations increases. This phenomenon means that clean small-loss samples will be biased toward agreement data, which is the set of samples for which the two networks have the same prediction. This paper proposes an adaptive sample selection method to train deep neural networks robustly and prevent noise contamination in the disagreement strategy. Specifically, the proposed method calculates the threshold of the small-loss criterion by considering the loss distribution of the whole batch at each iteration. Then, the network is backpropagated by extracting samples below this threshold from the disagreement data. Combining the disagreement and agreement data of the two networks can suppress the degradation of the true-label rate of training data in a mini batch. Experiments were conducted using five commonly used benchmarks, MNIST, CIFAR-10, CIFAR-100, NEWS, and T-ImageNet to verify the robustness of the proposed method to noisy labels. The results show the proposed method improves generalization performance in an image classification task with simulated noise rates of up to 50%.

INDEX TERMS Deep neural network, learning with noisy labels, image classification, co-teaching.

I. INTRODUCTION

DEEP neural networks (DNNs) have achieved a remarkable level of performance in various applications such as image classification [1]. This result is highly dependent on the availability of a large amount of high-quality labeled data, which is difficult to obtain in practice. Instead, a common means of constructing a large labeled dataset is to use crowdsourcing systems [2], [3] such as Amazon's Mechanical Turk or search engines that query samples using a keyword, which is then used as a label [4], [5], [6]. Both approaches can facilitate the acquisition of labeled data, but contaminate these data with unreliable labels, which are called noisy labels. Real-world datasets have been reported to contain levels of noise ranging from 8.0% to 35.8% [7], [8], [9]. Furthermore, it has been found that 52% of web images retrieved using a query contain incorrect labels [10]. DNNs are highly able to fit to noisy labels [11], [12], resulting in an inevitable loss of accuracy.

Our goal is to effectively and robustly train DNNs using a training dataset with noisy labels. Various existing studies have investigated how noisy labels can be handled. A typical method is loss correction [13], [11], [14], [15], which corrects for the forward or backward loss values of the training samples by estimating the noise transition matrix. However, the accuracy of the noise transition matrix estimation decreases when there are many classes and the number of noisy data is large. Moreover, in recent years, methods based on gradient clipping [16], [17], which corrects the losses by constraining the gradient, have received much attention [18]. However, both loss correction approaches have a problem with error accumulation, where the errors in the loss correction continue to affect the network updates [19].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

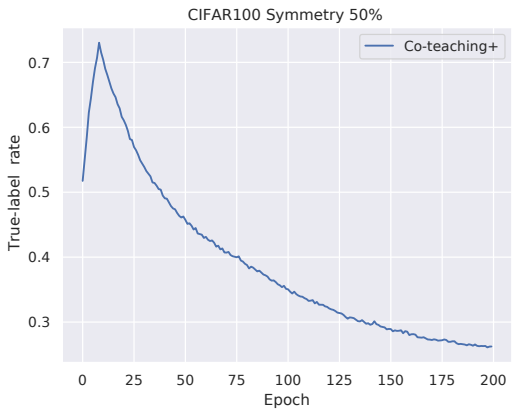

FIGURE 1. True-label rate during training on CIFAR-100 dataset using Co-teaching+ [24]. As the number of training epochs increases, training samples are selected from a subset with a high number of noisy labels. As a result, in the last stage of learning, the network is trained using noisy samples.

Recent research on DNNs has confirmed that they first learn easy (most likely clean) samples and then learn hard (most likely noisy) samples [12], which is called the memorization effect. Intuitively, suppose we could use this effect to train DNNs using only the samples with small loss. In that case, we could achieve a robust generalization performance for noisy labels without estimating the noise transition matrix. One promising approach, called sample selection, selects small-loss samples from the forward loss of the network and updates the network using backpropagation [20], [21], [22], [23], [24], [25].

Co-teaching [23] and Co-teaching+ [24] have been proposed as practical methods to deal with highly noisy data. Co-teaching trains two networks simultaneously by selecting small-loss samples at each iteration and cross-updating each network to avoid accumulating error. Co-teaching+ improves this approach by training the two networks on samples that disagree with each other's predictions to prevent their convergence and maintain their variance. This strategy is called "update by disagreement." Co-teaching+ is, to the best of our knowledge, the state of the art of sample selection-based methods. However, as the number of training epochs grows, the proportion of noisy data used for backpropagation increases, which degrades the generalization performance.

Figure 1 shows the true-label rate during training on the CIFAR-100 dataset using Co-teaching+. The true-label rate is defined as the proportion of samples with the true labels among the small-loss samples extracted from the mini batch at each iteration. In our validation study, 50% of CIFAR-100 was uniformly and randomly relabeled for each class based on symmetry flipping [26]. The results of that study indicate that the noise rate of the training samples selected by the disagreement strategy increases as the iterations progress, leading to overfitting on noisy data. It is possible to reduce the number of noisy data by lowering the rate at the end of the iterations. In fact, Yao et al. [25] used the proportion of clean data, i.e., samples without any noisy labels, as one of the parameters and tuned it using AutoML [27]. However, in practical use, it is not always possible to obtain clean data in advance.

In this paper, we propose an adaptive sample selection method to robustly train DNNs using the disagreement strategy. The key idea of the proposed method is to prevent noisy labels from becoming mixed into a training mini batch by determining a small-loss threshold at each epoch. Co-teaching+ extracts small-loss samples from the disagreement data at a defined rate throughout all iterations. However, because the small-loss samples, which are likely to be clean labels, may become biased toward disagreement or agreement as training progresses, the amount of samples to be extracted should be determined on an iteration-by-iteration basis. In the proposed method, the threshold is defined by calculating the percentile value using the data of the entire mini batch. Then, the network is backpropagated by extracting the samples below the threshold from the disagreement data. Using data combined in such a way, we can stop the true-label rate of the subset extracted from the disagreement data from decreasing. Therefore, the main contributions of this paper can be summarized as follows:

- We present a new small-loss selection method based on the memorization effect.
- We propose using a combination of agreement and disagreement data in the disagreement strategy, thus reducing the decrease in the true-label rate during the training process.
- We present the results of experiments using five commonly used benchmark datasets, MNIST, CIFAR-10, CIFAR-100, NEWS, and T-ImageNet to demonstrate that the proposed method achieves state-of-the-art results.

The remainder of this paper is structured as follows. Section II reviews the related work of deep learning with noisy data. In Section III, we propose our training method with memorization effect-based sample selection. Experimental results are discussed in Section IV, and the conclusions are given in Section V.

II. RELATED WORK

When there are noisy labels, the deep learning model eventually memorizes these incorrectly provided labels, degrading the generalization performance. This cannot be changed by choosing optimizers and network architectures or applying data augmentation [11]. The current approaches are categorized into those based on loss correction [13], [14], [15], those based on label noise cleaning [28], [29], [30], those based on dataset pruning [31], [32], [33], [34], [35], [36], [37], and those based on sample selection [21], [22], [23], [24], [38], [39]. In this section, we review these related deep learning methods for handling noisy labels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

A. LOSS CORRECTION APPROACH

The basic idea behind the loss correction approach is to correct the forward or backward loss of the DNN based on the estimated noise transition matrix. Bootstrapping [13] employs a reconstruction-based objective that uses the concept of perceptual consistency to train the network while correcting its predictions. F-correction [14] introduces a two-step method that first estimates the noise transition matrix of noisy data and corrects the output of the loss function using the forward loss correction mechanism [14]. In [14], the network is pre-trained using noisy data, and the samples with the highest output per class are assumed to be perfect samples that are likely to be clean. The noise transition matrix is then estimated using the softmax probabilities when a perfect sample is an input to the wrong class. However, F-correction is inaccurate on datasets with many classes and a small number of samples per class such as CIFAR-100. Some methods assume that clean validation data are available. Hendrycks et al. proposed gold loss correction, which estimates the matrix measuring label corruption calculated using known clean samples [15].

In contrast to estimating the noise transition matrix as described above, there is an approach for correcting the loss that constrains the gradient norm to a specified value by gradient clipping [16], [17]. Menon et al. showed that noise robustness can be obtained using a partially Huberized loss, which clips only the contribution of the gradient [18].

B. LABEL NOISE CLEANING APPROACH

Label noise cleaning is an approach that identifies suspicious labels and changes them to the corresponding true ones. This approach relies on a feature extractor that maps the data into feature domains to investigate the level of noise in the noisy labels. It is an iterative framework, where the classifier and the label transformer are trained on each other and their abilities improve during training, unlike data preprocessing, where noisy labels are removed before training begins. An algorithm using this approach can be divided according to whether it requires clean data or not. If clean data are available, the obvious approach is to relabel the noise labels using the predictions of the network trained on the clean data. For relabeling, [28] uses a label blending operation, which calculates the weighted sum of the given noisy labels and the predicted labels. Alternatively, [29], [30] introduced a framework of joint optimization for both training the classifier and transforming noisy labels into clean ones. Expectation maximization is used to estimate both the parameters of the classifier and the posterior distribution of the labels to minimize the loss.

C. DATASET PRUNING APPROACH

The first approach in dataset pruning is to completely remove the noisy samples found previously and train the network on the remaining dataset. The simplest approach is to remove the samples misclassified by the network [31]. For instance, [32] used a combination of noise filters, where each noise filter assigns a level of noise to the samples. These predictions are then combined to remove samples with the highest noise levels. Luengo et al. [33] extended this method using the label correction approach. If different noise filters predict the same label for a noisy sample, the label for that sample is changed to the predicted label, otherwise it is removed from the dataset. In [34], the state of the network is varied between underfitting and overfitting by periodically adjusting the learning rate. During underfitting, noisy samples have higher losses, so this cyclic process removes the noisy samples.

The second approach is to remove only the labels of noisy samples. The traditional method employs a semi-supervised learning method [35], [36]. SELF [36] is based on a running average model called the Mean-Teacher [40], which obtains self-ensemble predictions from all samples and incrementally removes samples with labels that do not match the original labels. DivideMix [37] uses the Gaussian mixture model to divide the samples into clean and noisy samples. Using the split samples, a semi-supervised approach based on the MixMatch strategy [41] is used.

D. SAMPLE SELECTION APPROACH

This approach continuously monitors the DNNs and detects the true-labeled samples to be learned in the next training iteration. Intuitively, DNNs can achieve better generalization performance when the training data are less noisy. This approach uses the characteristic of DNNs called the memorization effect, i.e., they learn clean and simple patterns in the initial epochs, even in the presence of noisy labels. Thus, they have the ability to filter out noisy samples using their loss values. The goal is to make DNNs robust to noise by selecting only small-loss samples and eliminating mislabeled data with high losses during training iterations.

Self-paced learning [42], [43] can filter out noisy labels by assigning small weights to mislabeled samples and large weights to clean samples, thus ensuring robust model learning. Specifically, specifying a monotonically decreasing weighting function allows the classifier to focus on the easy samples first and then fit the difficult samples. For example, in the MentorNet approach [19], an additional network, called StudentNet, is trained and MentorNet is used to select clean samples to guide the training of StudentNet. If clean validation data cannot be prepared, the self-paced MentorNet uses a predefined curriculum, that is, a self-paced curriculum. The concept of the self-paced MentorNet is similar to that of the self-learning approach [44], and it inherits the problem of error accumulation.

Han et al. proposed Co-teaching [23], which trains two networks in a symmetric way. Co-teaching introduces cross-training, where a small-loss sample from one network is used as a training sample for the other network. By exchanging training samples between two networks, bias in the training samples is avoided and the accumulated error is reduced. In [38], Wang et al. proposed a method for reweighting small loss samples. Specifically, a loss function designed based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

on the ArcFace loss [45] is used to recalculate the loss of selected small-loss samples to increase the likelihood that a sample with high confidence will be selected. In [39], Chen et al. introduced the iterative noisy cross-validation (INCV) method into Co-teaching, which selects a mini-batch of samples that are estimated to be true labels using the network under training at each training iteration. However, the two networks converge to a consensus, causing a problem similar to that of the self-paced MentorNet, which uses a single network.

Co-teaching+ [24] is an improved method that introduces the concept of decoupling [22] into Co-teaching. Decoupling is similar to Co-teaching in that it simultaneously trains a pair of networks, but it updates the networks using the samples with different predictions. The weights of the two networks do not converge, allowing them to maintain divergence. Because Co-teaching+ is closely related to the proposed method, the algorithm and problem are described in the following section.

In summary, most loss correction methods have difficulty handling multi-class data, so the development of the sample selection approaches, using which uses the memorization effect, is promising. The sample selection approach continuously monitors the DNNs and selects samples to be trained learned in the following training iteration. Thus, sample selection-based methods can be incorporated into the algorithms of different approaches by simply manipulating the input stream, so a combined strategy is expected to improve accuracy. The state-of-the-art sample selection-based method is Co-teaching+, which substantially improves generalization performance using a combined selection of disagreement and small-loss data. In this paper, we point out the problems of Co-teaching+ and propose an adaptive sample selection method to improve it. Therefore, the proposed method does not assume pre-training with clean data [15], [25] is not assumed in the method proposed here, and this paper does not deal with a strategy that combines the sample selection approach with other approaches [37].

III. METHOD

The proposed method improves on existing sample selection-based methods by exploiting the memorization effect. This section first introduces the Co-teaching+ algorithm, and then describes our learning method with the proposed sample selection method (shown in Figure 2).

A. LEARNING FROM NOISY DATA

As in Co-teaching, two DNNs are trained simultaneously, but Co-teaching+ consists of two steps: disagreement update and cross update. The first step updates the mini batch data so that each network makes its own predictions and samples with predictions from the two networks that disagree are selected. Next, in the cross-update step, based on these disagreement data, each network further selects its own small-loss samples, but backpropagates those selected by the paired networks to update their parameters.

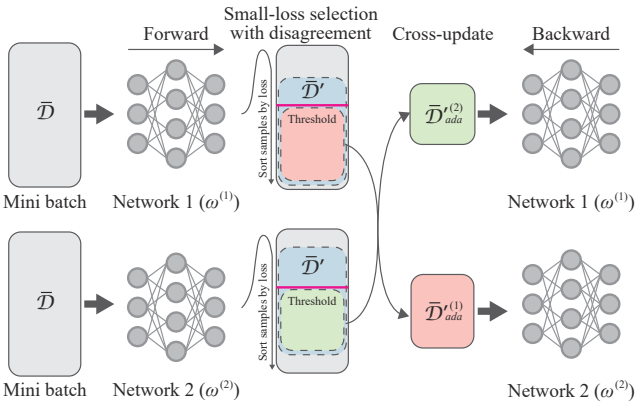

FIGURE 2. Training process of the proposed method. The forward loss values are calculated from the mini batch data $\bar{\mathcal{D}}$, and then the prediction disagreement data $\bar{\mathcal{D}}'$ of the two networks parameterized by ω_1 (resp. ω_2) are extracted. Because the loss distribution of the disagreement data is biased at each iteration, extracting small-loss samples at a fixed rate allows noisy data to be mixed in. The proposed method adaptively controls the number of small-loss samples, whose subset is denoted as $\bar{\mathcal{D}}'^{(1)}_{ada}$ (resp. $\bar{\mathcal{D}}'^{(2)}_{ada}$) from $\bar{\mathcal{D}}'$ by defining a threshold that considers the loss distribution of the whole mini batch at each iteration. Using $\bar{\mathcal{D}}'^{(1)}_{ada}$ (resp. $\bar{\mathcal{D}}'^{(2)}_{ada}$), both networks are cross-updated.

Specifically, the two networks, with parameters ω_1 and ω_2 respectively, are trained using the mini-batch technique. We are given the training data $\mathcal{D}$, and split them into mini batches $\bar{\mathcal{D}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_B, y_B)\}$, where (x_i, y_i) denotes the sample and its label, and B is the batch size. Then, according to the predictions $\{\bar{y}_1^{(1)}, \bar{y}_2^{(1)}, \ldots, \bar{y}_B^{(1)}\}$ (predicted by ω_1) and $\{\bar{y}_1^{(2)}, \bar{y}_2^{(2)}, \ldots, \bar{y}_B^{(2)}\}$ (predicted by ω_2), disagreement data are extracted as follows:

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

By training the two networks using disagreement data $\bar{\mathcal{D}}'$, the two networks do not converge but maintain their divergence, similar to the decoupling algorithm [22].

To remove noisy data from disagreement data $\bar{\mathcal{D}}'$, each network selects small-loss data $\bar{\mathcal{D}}'^{(1)}$ and $\bar{\mathcal{D}}'^{(2)}$ based on their own parameters ω_1 and ω_2, respectively. Next, each network is backpropagated using their paired data. For example, parameter ω_1 is updated based on small-loss data $\bar{\mathcal{D}}'^{(2)}$. Note that, to control how many small-loss data will be selected at epoch e, the proportion of small-loss samples is defined as follows:

$$\lambda(e) = 1 - \min\left\{\frac{e}{E_k} R_{noise}, R_{noise}\right\}, \tag{2}$$

where R_{noise} is an estimate of the noise rate in training data $\mathcal{D}$. Because of the memorization effect, the DNN initially fits clean data and then gradually overfits noisy data. Therefore, a large λ is initially used, but the value of λ is quickly reduced up until epoch E_k to avoid fitting noisy data. From epoch E_k onward, it is adjusted according to the noise rate in the training data (i.e., $\lambda(e) = 1 - R_{noise}$).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

Algorithm 1 Proposed training method with adaptive small-loss sample selection.

INPUT: noisy data $\mathcal{D}$, batch size B, learning rate η, estimated noise rate R_{noise}, and epochs E_k and E_{max};

1: **for** $e = 1, 2, \cdots, E_{max}$ **do**
2: **Divide** $\mathcal{D}$ into $\frac{|\mathcal{D}|}{B}$ mini batches;
3: **Initialize** model parameters $\omega^{(1)}$ and $\omega^{(2)}$;
4: **for** $n = 1, 2, \cdots, |\mathcal{D}|$ **do**
5: **Draw** n-th mini batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
6: **Select** prediction disagreement $\bar{\mathcal{D}}'$;
7: **Calculate** the $\lambda(e)$th percentile loss $P^{(1)}_{\lambda(e)\%}$ of the mini batch $\bar{\mathcal{D}}$; ▷ threshold calculated from the losses of network 1
8: **Calculate** the $\lambda(e)$th percentile loss $P^{(2)}_{\lambda(e)\%}$ of the mini batch $\bar{\mathcal{D}}$; ▷ threshold calculated from the losses of network 2
9: **Get** $\bar{D}'^{(1)}_{ada} = \{(x_i, y_i) | \mathcal{L}(x_i, y_i; \omega^{(1)}) \leq P^{(1)}_{\lambda(e)\%}, (x_i, y_i) \in \bar{D}'\}$; ▷ extract small-loss data below the threshold $P^{(1)}_{\lambda(e)\%}$
10: **Get** $\bar{D}'^{(2)}_{ada} = \{(x_i, y_i) | \mathcal{L}(x_i, y_i; \omega^{(2)}) \leq P^{(2)}_{\lambda(e)\%}, (x_i, y_i) \in \bar{D}'\}$; ▷ extract small-loss data below the threshold $P^{(2)}_{\lambda(e)\%}$
11: **Update** $\omega^{(1)} = \omega^{(1)} - \eta \nabla \mathcal{L}\left(\bar{\mathcal{D}}'^{(2)}_{ada}; \omega^{(2)}\right)$; ▷ update $\omega^{(1)}$ by $\omega^{(2)}$
12: **Update** $\omega^{(2)} = \omega^{(2)} - \eta \nabla \mathcal{L}\left(\bar{\mathcal{D}}'^{(1)}_{ada}; \omega^{(1)}\right)$; ▷ update $\omega^{(2)}$ by $\omega^{(1)}$
13: **end for**
14: **Update** $\lambda(e) = 1 - \min\left\{\frac{e}{E_k} R_{noise}, R_{noise}\right\}$;
15: **end for**

OUTPUT: $\omega^{(1)}$ and $\omega^{(2)}$.

B. ADAPTIVE SAMPLE SELECTION

Our algorithm is described in Algorithm 1. *The key difference between the proposed method and Co-teaching+ is the introduction of an adaptive process to set the loss threshold for determining small losses at each training iteration.* Specifically, we considered the following two issues when designing Algorithm 1.

1) A decrease in the true-label rate of the training mini batch data may degrade the generalization performance.
2) The noise rate of the disagreement data $\bar{\mathcal{D}}'$ is not always the same as that of the training data $\mathcal{D}$.

As the number of training epochs increases, the small-loss samples extracted at the disagreement-update step become noisy labels, which leads to overfitting to noisy data (as described in Section I). Furthermore, Co-teaching+ controls how many small-loss data are extracted from the disagreement data by $\lambda(e)$, defined in Eq. (2). However, parameter $\lambda(e)$, which is the noise rate R_{noise}, is an estimate for the entire training data $\mathcal{D}$, and the expected noise rate for the two subsets, i.e., the agreement or disagreement data, is not always R_{noise}. In other words, it is not appropriate to fetch the same proportion of small-loss samples throughout all iterations, because the number of small-loss samples with true labels present in the disagreement subset will vary in each iteration. To avoid this problem, it may be possible to reduce the number of noisy data, for example, by making the sampling criterion more stringent, such as lowering $\lambda(e)$ at the end of training. However, such scheduling of $\lambda(e)$ requires a certain amount of clean validation data.

We first search for the $\lambda(e)$th percentile loss in network m $(= 1, 2)$, which is denoted as $P^{(m)}_{\lambda(e)\%}$, from the samples included in mini batch data $\bar{\mathcal{D}}$. Next, for each network m $(= 1, 2)$, we create subsets $\bar{D}'^{(m)}_{ada}$ using the disagreement data that satisfy the following equation:

$$\bar{D}'^{(m)}_{ada} = \{(x_i, y_i) | \mathcal{L}(x_i, y_i; \omega^{(m)}) \leq P^{(m)}_{\lambda(e)\%}, (x_i, y_i) \in \bar{D}'\}, \quad (3)$$

where $\mathcal{L}(\cdot; \omega^{(m)})$ is the loss parameterized by $\omega^{(m)}$ when the samples are given. By calculating the threshold based on mini batch data $\bar{\mathcal{D}}$, it is possible to tighten the sampling criterion when the disagreement samples are biased toward high loss data. Thus, we can address the problem of decreasing the true-label rate as training progresses. This enables adaptive small-loss sampling according to the training situation at each epoch without the need for clean validation data. Note that, when one of the two sets of data is not present in steps 9–10 (Algorithm 1), the networks are updated using disagreement data $\mathcal{D}'$ without small-loss selection, similar to the Co-teaching+ algorithm.

Finally, given a sample to be labeled, we use one of the two networks to predict the label of the sample, following the method used in Co-teaching and Co-teaching+.

TABLE 1. Details of the datasets used in our experiments.

	# of training	# of testing	# of class	image size
MNIST	60,000	10,000	10	28 × 28
CIFAR-10	50,000	10,000	10	32 × 32
CIFAR-100	50,000	10,000	100	32 × 32
NEWS	11,314	7,532	20	1000-D
T-ImageNet	100,000	10,000	200	64 × 64

IV. EXPERIMENTAL RESULTS

In this section, we confirm the effectiveness of the proposed method by simulating noise to create datasets based on the MNIST, CIFAR-10, CIFAR-100, NEWS, and T-ImageNet datasets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

TABLE 2. Architectures of the MLP used on MNIST, the CNNs used on CIFAR-10 and CIFAR-100, and the MLP used on NEWS in the experiments.

MLP on MNIST	CNN on CIFAR-10	CNN on CIFAR-100	MLP on NEWS
28×28 gray image	32×32 rgb image	32×32 rgb image	1000-D text
Dense 28×28 256, ReLU	5×5 Conv, 6 ReLU 2×2 Max-pool	3×3 Conv, 64 BN, ReLU 3×3 Conv, 64 BN, ReLu 2×2 Max-pool	300 Embedding Flatten $\rightarrow 1000 \times 300$ Adaptive average-pool $\rightarrow 16 \times 300$
	5×5 Conv, 16 ReLU 2×2 Max-pool	3×3 Conv, 128 BN, ReLU 3×3 Conv, 128 BN, ReLu 2×2 Max-pool	Dense $16 \times 300 \rightarrow 4 \times 300$ 4×300 BN, Softsign
	Dense $16 \times 5 \times 5 \rightarrow 120$, ReLU Dense $120 \rightarrow 84$, ReLU	3×3 Conv, 196 BN, ReLU 3×3 Conv, 196 BN, ReLU 2×2 Max-pool	Dense $4 \times 300 \rightarrow 300$ 300 BN, Softsign
Dense $256 \rightarrow 10$	Dense $84 \rightarrow 10$	Dense $256 \rightarrow 100$	Dense $300 \rightarrow 20$

A. EXPERIMENTAL SETUP

Datasets: The details of the five datasets used in our experiments, MNIST, CIFAR-10, CIFAR-100, NEWS, and T-ImageNet, are summarized in Table 1. From those datasets, we created synthetic datasets by corrupting their labels using two noise transition matrices, symmetry flipping [26] and pair flipping [14], following [23], [24]. Note that on the NEWS dataset, [24] conducted experiments on seven classes that are groups of the original 20 classes, whereas we conducted experiments on the original 20 classes. An example of a noise transition matrix for symmetry flipping with four classes and a noise rate of R_{noise} is as follows:

$$T = \begin{bmatrix} 1 - R_{noise} & \frac{R_{noise}}{3} & \frac{R_{noise}}{3} & \frac{R_{noise}}{3} \\ \frac{R_{noise}}{3} & 1 - R_{noise} & \frac{R_{noise}}{3} & \frac{R_{noise}}{3} \\ \frac{R_{noise}}{3} & \frac{R_{noise}}{3} & 1 - R_{noise} & \frac{R_{noise}}{3} \\ \frac{R_{noise}}{3} & \frac{R_{noise}}{3} & \frac{R_{noise}}{3} & 1 - R_{noise} \end{bmatrix}. \tag{4}$$

We used symmetry flipping with $R_{noise} = \{0.2, 0.5\}$, denoted as **Symmetry 20%** and **Symmetry 50%**, respectively.

Next, we used two types of pair flipping. The first type swaps the labels between adjacent classes. An example of pair flipping, applied between adjacent classes with four classes and a noise rate of R_{noise} is as follows:

$$T = \begin{bmatrix} 1 - R_{noise} & R_{noise} & 0 & 0 \\ 0 & 1 - R_{noise} & R_{noise} & 0 \\ 0 & 0 & 1 - R_{noise} & R_{noise} \\ R_{noise} & 0 & 0 & 1 - R_{noise} \end{bmatrix}. \tag{5}$$

We used "adjacent" pair flipping for datasets with $R_{noise} = 0.45$, denoted as **Pair(adjacent) 45%**.

Unlike Pair(adjacent), the second type of pair flipping is to swap labels between two classes that are visually similar. The reason for simulating noise in this way is that we assume that real-world annotators are highly likely to mislabel classes that are similar in visual appearance. We followed [14] to define visually similar classes. For MNIST, the transitions are $2 \rightarrow 7$, $3 \rightarrow 8$, $5 \leftrightarrow 6$, $7 \rightarrow 1$. For CIFAR-10, the transitions are TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG. For CIFAR-100, because there are 20 superclasses such as aquatic mammals, fish, and flowers, the transitions are made within the same superclass. For NEWS, because there are seven news groups (comp., rec., aci., misc., talk, alt., and sci.), the transitions are made within the same group. We used "visually similar" pair flipping with $R_{noise} = 0.45$, denoted as **Pair(similar) 45%**. Note that for T-ImageNet, while it is possible to form class groups in the tree hierarchical structure defined in WordNet [46], we did not conduct experiments on Pair(similar) because the distribution of the number of classes per group is imbalanced.

In our experiments, we assume that the noise rate R_{noise} is known. However, R_{noise} is not known in practice, although an estimate can be obtained by counting the number of perfect samples [14] of each class.

Baselines: We compared the proposed method, denoted as **Proposed**, with the following state-of-the-art methods:

1) **Standard**: The networks shown in the Table 2 are trained directly using noisy data. Standard is included in the comparison to verify how much accuracy is reduced when the robust deep learning method is not used for noisy data.
2) **Co-teaching**: This method trains two networks simultaneously in a symmetric way. Reference [23] demonstrated that Co-teaching outperforms loss correction methods [13], [14], [47] and previous sample selection methods [22], [19].
3) **Co-teaching+**: An improved version of Co-teaching, which has the disagreement step in addition to the cross-update step, is a state-of-the-art method based on sample selection. Our training scheme is designed based on Co-teaching+.
4) **Huberized**: This method introduces the partially Huberized loss function [18] to Co-teaching+. A comparison of the performance of Huberized and Proposed confirms the effectiveness of the proposed adaptive sample selection.

We re-implemented all methods using public source code under the same conditions. As described above, in this study, the proposed method was compared with the methods without pre-training using a subset consisting of clean validation data.

Network structure and optimizer: The network architectures and optimization methods were changed for each dataset. For experiments using the MNIST, CIFAR-10, CIFAR-100, NEWS, and T-ImageNet datasets, we used the experimental conditions given in [24]. The architectures used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

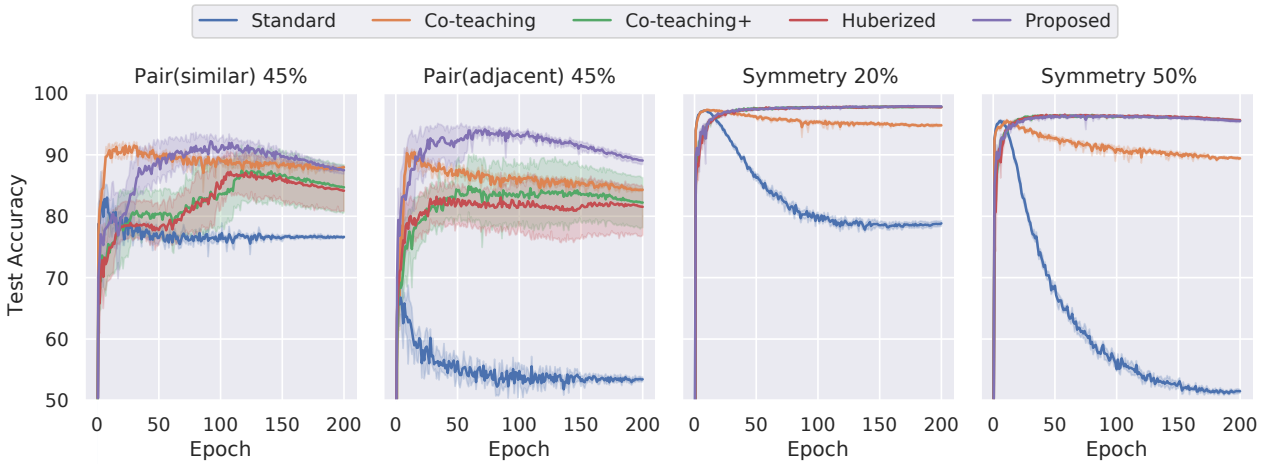

FIGURE 3. Relationship between test accuracy and number of epochs on the MNIST dataset.

TABLE 3. Average test accuracy on MNIST over the last 10 epochs. The best average result for each noise transition pattern is shown in bold, and the second-best one is underlined.

	Standard	Co-teaching	Co-teaching+	Huberized	Proposed
Pair(similar) 45%	76.61	**87.82**	84.87	84.35	87.67
Pair(adjacent) 45%	53.38	84.31	82.33	81.72	**89.22**
Symmetry 20%	78.7	94.8	97.81	97.78	**97.88**
Symmetry 50%	51.44	89.5	95.63	**95.68**	95.53

in our experiments consist of a two-layer MLP for MNIST, a five-layer CNN for CIFAR-10, a seven-layer CNN for CIFAR-100, a three-layer MLP for NEWS, and a 18-layer Preact ResNet [48] for T-ImageNet. The details of the architectures are summarized in Table 2. As an optimization method, we used Adam [49] with an initial learning rate of 0.001, linearly decreasing to zero from 80 epochs to 200 epochs, a momentum of 0.9, and a batch size of 128.

Evaluation metric: For the evaluation metric, we used the test accuracy, i.e., *Test Accuracy = (# of correct predictions) / (# of test dataset)*. All experiments were repeated five times, and we report the averaged results. In each figure, the 95% confidence interval is indicated by shading.

B. COMPARISON WITH DIFFERENT METHODS

1) Results on the MNIST dataset

Figure 3 compares the accuracy for each epoch up to 200 epochs on the MNIST dataset. As shown in Figure 3, for the Symmetry 20% and 50% conditions, the accuracy of Proposed is almost the same as that of Co-teaching+ and Huberized, but is better than those of the others. For Pair(similar) 45%, Proposed outperforms Co-teaching in the middle epochs. In contrast, the accuracy of Proposed is lower than that of Co-teaching at the last epoch. For Pair(adjacent) 45%, Proposed shows a significant improvement in accuracy.

Table 3 shows the average accuracy of different methods in the last 10 epochs. Proposed has the highest accuracy of 89.22% and 97.88% for Pair(adjacent) 45% and Symmetry 20%, which are 4.91 and 0.07 *pp*s[1] higher than the second-best methods, respectively. For the Pair(similar) 45% and Symmetry 50% conditions, the differences between the best method and Proposed are only 0.15 *pp*. In other words, Proposed is almost equal to the second-best method for Symmetry 20%, Pair(similar) 45%, and Symmetry 50%, but it is much more effective under the Pair(adjacent) 45% condition, with an increase of 4.91 *pp*.

When Proposed is compared with Co-teaching+, the difference is 0.07 *pp* for Symmetry 20% and 0.15 *pp* for Symmetry 50%, which are almost equal. However, under pair flipping conditions, Proposed is superior by 6.89 *pp* for Pair(adjacent) 45% and 2.8 *pp* for Pair(similar) 45%, which are substantial differences.

2) Results on the CIFAR-10 dataset

Figure 4 compares the accuracy for each epoch up to 200 epochs on the CIFAR-10 dataset. As shown in Figure 4, for Symmetry 20%, the accuracy of Proposed is almost equal to that of Co-teaching+ and Huberized, but for Symmetry 50%, that of Proposed is better than that of Co-teaching+ and Huberized. For the Pair(adjacent) 45% condition, there is an improvement in the latter epochs when compared with Co-teaching+. In contrast, for Pair(similar) 45%, Proposed has the lowest accuracy.

Table 4 shows the average accuracy of different methods in the last 10 epochs. For Pair(adjacent) 45%, Symmetry 20%, and Symmetry 50%, Proposed has the highest accuracy,

[1]We denote a percentage point as *pp*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

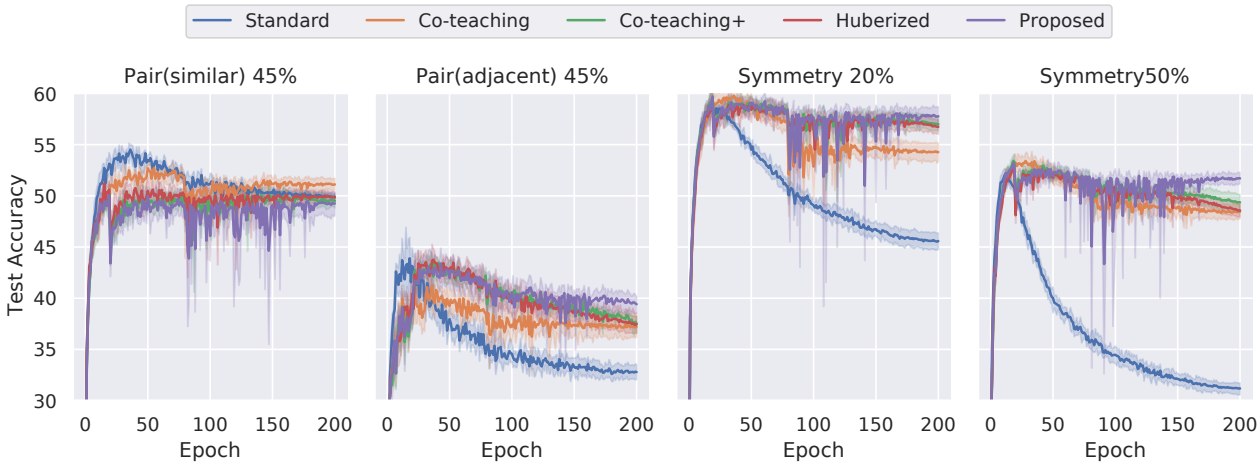

FIGURE 4. Relationship between test accuracy and the number of epochs on the CIFAR-10 dataset.

TABLE 4. Average test accuracy on CIFAR-10 over the last 10 epochs. The best average result for each noise transition pattern is shown in bold, and the second-best one is underlined.

	Standard	Co-teaching	Co-teaching+	Huberized	Proposed
Pair(similar) 45%	<u>49.93</u>	**51.14**	49.53	49.90	49.25
Pair(adjacent) 45%	32.81	37.24	<u>37.82</u>	37.56	**39.58**
Symmetry 20%	45.60	54.71	<u>57.06</u>	56.84	**57.81**
Symmetry 50%	31.22	48.45	<u>49.45</u>	48.69	**51.67**

i.e., 39.58%, 57.81%, and 51.67%, which are 1.76, 0.75, and 2.22 *pp*s higher than the second-best results. However, for Pair(similar) 45%, Proposed has the worst accuracy of 49.25%. This is 1.89 *pp* lower than the best result of Co-teaching.

Proposed outperforms Co-teaching+ under the Pair(adjacent) 45%, Symmetry 20% and Symmetry 50% conditions. Among them, the differences in accuracy for Pair(adjacent) 45% and Symmetry 50% are 1.76 and 2.22 *pp*s, respectively, indicating a substantial improvement. In contrast, under the Pair(similar) 45% condition, there is no substantial difference.

3) Results on the CIFAR-100 dataset

Figure 5 compares the accuracy for each epoch up to 200 epochs on the CIFAR-100 dataset. The accuracy of Proposed is almost the same as that of the baselines for Symmetry 20%, but for the other cases, the accuracy is substantially better than those of the baselines. In particular, in the latter epochs, the proposed method avoids overfitting on noisy data.

Table 5 shows the average accuracy in the last 10 epochs. Proposed has the highest accuracy of 32.98%, 33.07%, and 39.95% under the Pair(similar) 45%, Pair(adjacent) 45%, and Symmetry 50% conditions. When compared with Co-teaching+, the difference is 0.03 *pp* for Symmetry 20%, which is almost the same, but Proposed is better by 2.8, 4.37, and 1.93 *pp*s for Pair(similar) 45%, Pair(adjacent) 45%, and Symmetry 50%.

4) Results on the NEWS dataset

Figure 6 compares the accuracy for each epoch up to 200 epochs on the NEWS dataset. Even for this dataset, which is text-based and not visual data, the accuracy of the proposed method is better than that of the baselines, especially in the latter epochs. This result shows that the small-loss criterion based on the memorization effect is practical not only for visual data but also for other types of data.

Table 6 shows the average accuracy in the last 10 epochs. Proposed has the highest accuracy values of 18.11%, 16.25%, 19.20%, and 15.11% for each of the four noise transition patterns, outperforming Co-teaching+ with an average improvement of about 1.5 *pp*.

5) Results on the T-ImageNet dataset

To evaluate our method in a complex situation, Figure 7 shows the averaged test accuracy on T-ImageNet over the last 10 epochs. On this dataset, although the test accuracy temporarily decreases at the 80th epoch when the learning rate starts to decrease, the methods using the co-training framework with the small-loss criterion suppress the tendency of the test accuracy of the Standard method to decrease because of noisy labels. Among these methods, Proposed performs better as the number of epochs increases.

Table 7 shows the test accuracy in the last 10 epochs. Proposed consistently achieves higher accuracy regardless of noise transition pattern. The differences between Proposed and Co-teaching+ are 4.23 *pp* for Pair(adjacent) 45%, 2.5 *pp* for Symmetry 20%, and 2.66 *pp* for Symmetry 50%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

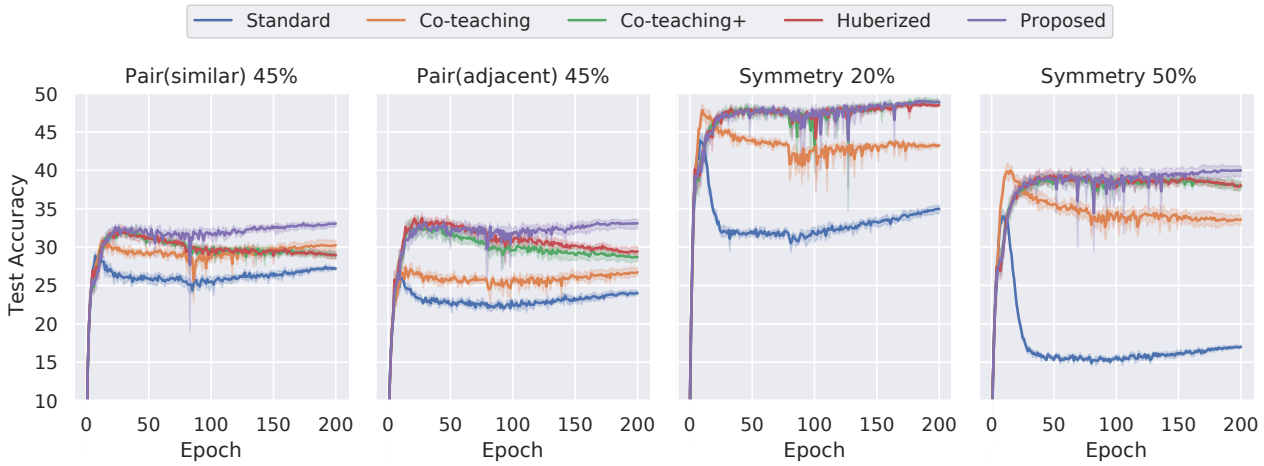

FIGURE 5. Relationship between test accuracy and number of epochs on the CIFAR-100 dataset.

TABLE 5. Average test accuracy on CIFAR-100 over the last 10 epochs. The best average result for each noise transition pattern is shown in bold, and the second-best one is underlined.

	Standard	Co-teaching	Co-teaching+	Huberized	Proposed
Pair(similar) 45%	27.24	<u>30.18</u>	29.02	28.98	**32.98**
Pair(adjacent) 45%	23.94	26.62	28.70	<u>29.38</u>	**33.07**
Symmetry 20%	34.84	43.21	<u>48.88</u>	48.47	**48.91**
Symmetry 50%	16.94	33.53	38.02	<u>38.04</u>	**39.95**

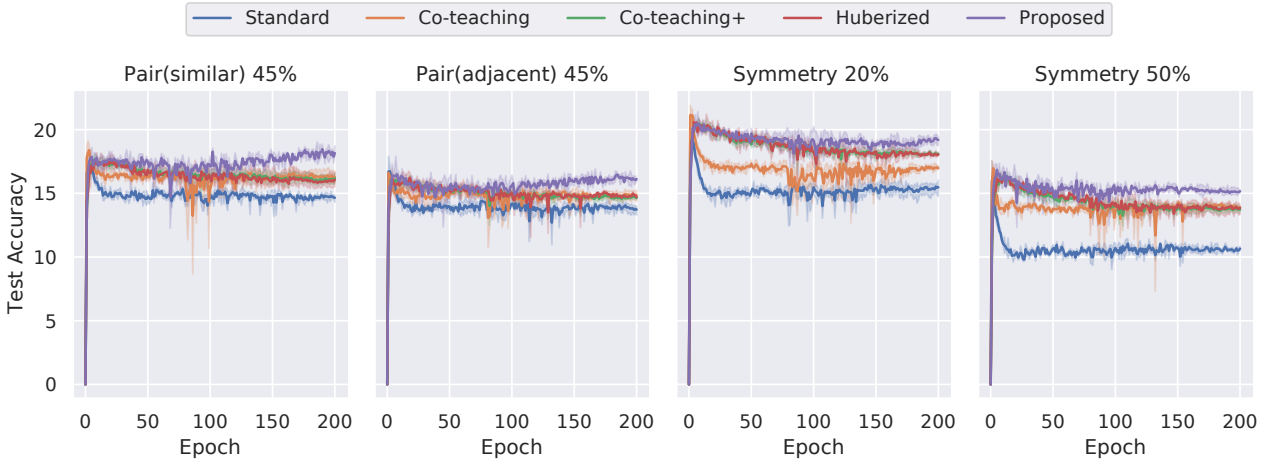

FIGURE 6. Relationship between test accuracy and the number of epochs on the NEWS dataset.

The results of the experiments on the five datasets show that Huberized has almost the same accuracy as Co-teaching+ with a difference of no more than 1 *pp*. In contrast, Proposed improves the accuracy by up to 6.89 *pp* when compared with Co-teaching+, i.e., on the MNIST dataset with Pair(adjacent) 45%, and the worst case is 0.28 *pp*, i.e., on the CIFAR-10 dataset with Pair(similar) 45%. Note that improvements in accuracy are observed for all noise transition patterns on the NEWS dataset, which is not a visual dataset, and the CIFAR-100 and T-ImageNet datasets, which are close to a real environment and have a large number of classes.

C. TRUE-LABEL RATE DISCUSSION

In this section, we compare the true-label rate of Co-teaching+ and that of Proposed. Our objective is to reduce the decrease in the true-label rate of deep learning by combining the cross-update and disagreement strategy. Therefore, we verify the effectiveness of Proposed by confirming whether the introduction of our sample selection method improves the true-label rate. Note that to calculate the true-label rate, we used the ground-truth labels before the label transitions, but they were used only for reference.

First, Figure 8 (a) compares the true-label rate on the MNIST dataset, and it can be seen that the true-label rate of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

TABLE 6. Average test accuracy on NEWS over the last 10 epochs. The best average result for each noise transition pattern is shown in bold, and the second-best one is underlined.

	Standard	Co-teaching	Co-teaching+	Huberized	Proposed
Pair(similar) 45%	14.72	<u>16.31</u>	16.11	15.97	**18.11**
Pair(adjacent) 45%	13.83	<u>14.88</u>	14.68	14.79	**16.25**
Symmetry 20%	15.39	17.03	<u>18.10</u>	18.05	**19.20**
Symmetry 50%	10.56	<u>13.92</u>	13.77	13.87	**15.11**

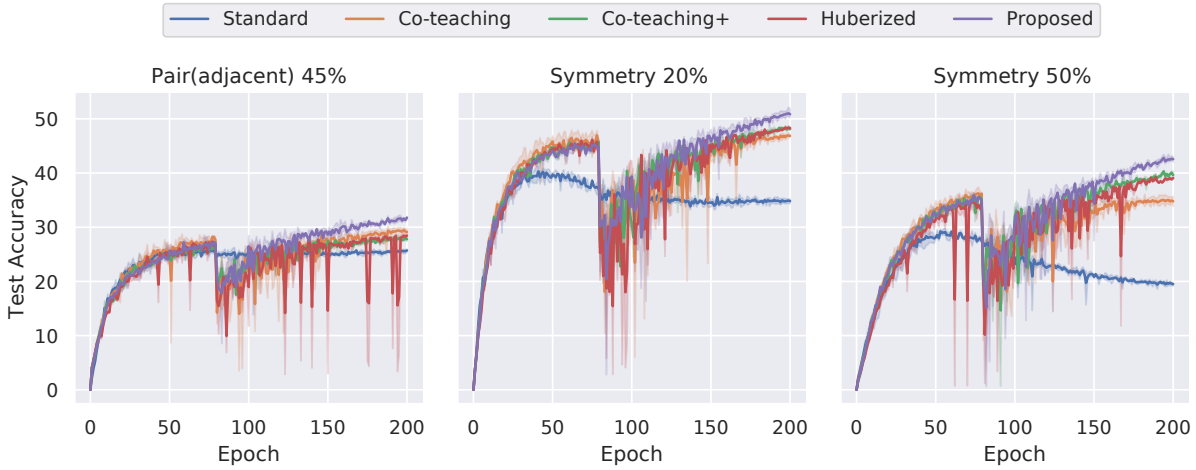

FIGURE 7. Relationship between test accuracy and the number of epochs on the T-ImageNet dataset.

TABLE 7. Average test accuracy on T-ImageNet over the last 10 epochs. The best average result for each noise transition pattern is shown in bold, and the second-best one is underlined.

	Standard	Co-teaching	Co-teaching+	Huberized	Proposed
Pair(adjacent) 45%	25.54	<u>29.27</u>	27.78	26.82	**31.51**
Symmetry 20%	34.83	46.72	<u>48.14</u>	48.09	**50.64**
Symmetry 50%	19.67	34.89	<u>39.71</u>	38.70	**42.37**

Proposed gradually becomes lower than that of Co-teaching+ for Pair(Similar) and Pair(adjacent). However, Figure 3 and Table 3 show that Proposed outperforms all baselines shown. This result can be explained as follows: (i) on MNIST, the predictions of the two networks agreed on many of the samples, (ii) Proposed did not use the small-loss trick after the middle epochs, and (iii) Proposed suppressed the decrease in the number of training samples in the initial epochs. Whereas the true-label rate of Co-teaching+ outperforms that of Proposed, Co-teaching+ suffers from insufficient learning due to the small number of small-loss samples in the backpropagation. The degradation of generalization performance due to insufficient learning can be confirmed by the performance difference between Co-teaching+ and Co-teaching shown in Figure 3. In the initial epochs, there is no significant difference in the true-label rate between Proposed and Co-teaching+. However, Proposed, which determines the small-loss criterion by considering the loss distribution of the whole mini batch, used a larger number of samples for backpropagation than Co-teaching+. This effect leads to the performance difference in the initial epochs. Moreover, as shown for Pair(adjacent), the true-label rate of Proposed starts to decrease from the middle epochs. Both Proposed and Co-teaching+ use all the disagreement data including noisy labels when the number of small-loss samples becomes zero. The reason for the decrease in the true-label rate of Proposed is that this process switching occurs. Hence, whereas the true-label rate of Co-teaching+ is higher than that of Proposed, its generalization performance is not improved due to the small number of samples. Therefore, the proposed sample selection method is able to suppress the decrease in the number of training samples in the initial epochs, which occurs when the predictions of the two networks agree frequently.

Second, Figure 8 (b) compares the true-label rate on the CIFAR-10 dataset, where a substantial improvement is observed for Symmetry 20% and 50%. In contrast, the two true-label rates for Pair(similar) are almost the same, whereas for Pair(adjacent), there is a slight improvement in the true-label rate but an increase in the accuracy. Under the Pair(adjacent) condition, even a small increase in the true-label rate contributes to an improvement in accuracy.

Finally, Figure 8 (c) shows the true-label rate on the CIFAR-100 dataset, where improvements in the true-label rate is confirmed in all cases. As shown in Table 5, Proposed outperforms the baselines in all noise transitions, indicating that the improvement in the true-label rate contributes to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119582, IEEE Access

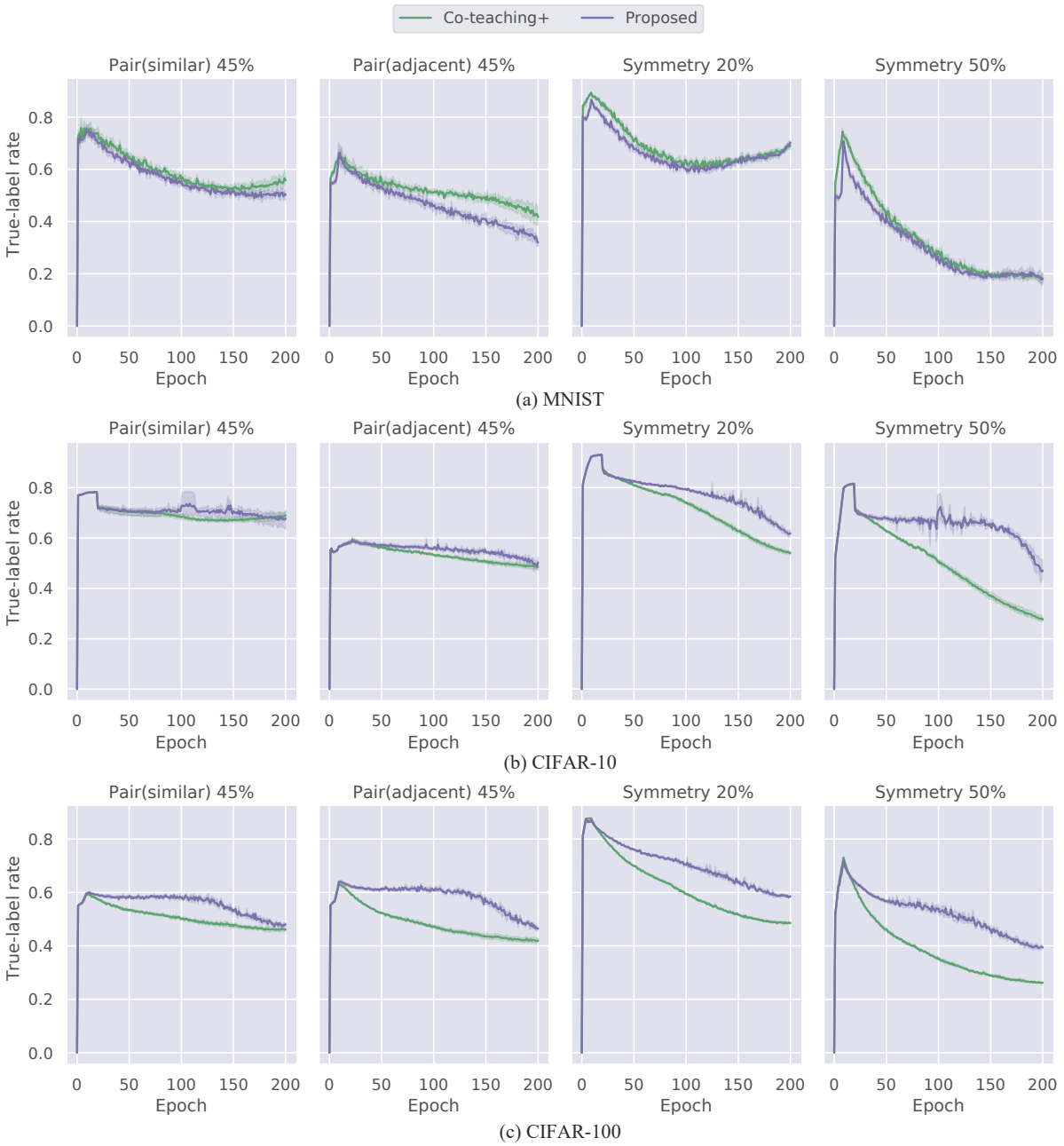

FIGURE 8. Comparisons of the average true-label rate when using the proposed method and Co-teaching+ on (a) MNIST, (b) CIFAR-10, and (c) CIFAR-100.

improvement in the accuracy. This result can be explained as follows. (i) Proposed suppressed the decrease in the true-label rate and trained the network with fewer noisy samples. (ii) Proposed accelerated the fit to the hard samples. The comparison of the true-label rate between Proposed and Co-teaching+ in Figure 8 (c) shows that the true-label rate of Proposed exceeds that of Co-teaching+ throughout almost all epochs. In this paper, the true-label rate is defined as the proportion of samples with the true label among the small-loss samples extracted from the mini-batch at each iteration. Thus, by maintaining a high true-label rate, Proposed can train the network with more true-labeled samples than Co-teaching+. Second, DNNs tend to learn simple patterns first and then gradually memorize all the samples [12]. Therefore, Proposed can be considered to be fit to hard samples, which improves the generalization performance of the classifier, especially by suppressing the decrease in the true-label rate in the latter half of training.

In summary, we can confirm the improvement of the true-label rate on the CIFAR-10 and CIFAR-100 datasets, and this effect improves accuracy. The results of four different noise simulations, especially on CIFAR-100, show that the

TABLE 8. Average computation time per iteration on CIFAR-100 with Symmetry 50% over the 200 epochs.

Methods	time(s)
Standard	9.66
Co-teaching	16.14
Co-teaching+	16.41
Huberized	16.32
Proposed	16.25

proposed method reduces the decrease in the true label rate from the middle to the latter half of training and improves the testing accuracy. For the MNIST dataset, the proposed method successfully avoids the problem of decreasing the number of training samples, which occurs when the number of agreement samples is large. However, in such a case, the proposed sample selection method has a weak impact on the latter half of training. Therefore, the effectiveness of the proposed method, which sets an adaptive loss threshold for small-loss samples for each epoch, is confirmed.

D. COMPUTATIONAL COST

In this section, we compare the computation time of the proposed method with that of other methods. This experiment was conducted by using the CIFAR-100 dataset with Symmetry 50%. We used the PyTorch framework [50] to implement each model, and training was performed on two RTX A6000 GPUs with NVLINK and AMD EPYC 7402P @ 2.8 GHz. Table 8 shows the results of the average computation time per iteration for all 200 epochs. Standard, which learns a single network, has the shortest computation time of the five methods. The Proposed method and the comparison methods, which train two networks simultaneously, have longer computation times. From Table 8, we can confirm that the computation times of the Co-teaching and Proposed methods are very similar.

V. CONCLUSIONS

In this paper, we presented a method to robustly train DNNs under real-world conditions where noisy labels are expected to be heavily present in the training data. DNN training methods that use the sample-selection approach, which uses the small-loss trick based on the memorization effect, has recently become a promising method for scaling to a large number of classes. Among them, Co-teaching+ is a state-of-the-art method that improves robustness by training two networks simultaneously using disagreement data. However, in Co-teaching+, the data selected by the small-loss criterion become noisy as the number of epochs increases. In this paper, we proposed a practical solution to this problem. The key idea of the proposed method is to prevent noisy labels from becoming mixed in the mini batch data by determining the small-loss threshold at each epoch. Extensive experiments on five benchmarks demonstrate that the proposed method achieves a state-of-the-art performance. Further, the improvement in the true-label rate was confirmed on a dataset that closely simulates a practical environment.

One of the limitations of the proposed method is that it relies on the disagreement strategy. Therefore, when the predictive agreement between the two networks is high, the proposed sample selection method is unlikely to be effective. The other limitation is that very difficult but clean samples are indistinguishable from noisy samples. Such samples are helpful for improving the robustness of classifiers. Our future work is to develop a method to incorporate them into the training samples.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2012, pp. 1097–1105.

[2] P. Welinder, S. Branson, S. Belongie, and P. Perona, "The multidimensional wisdom of crowds," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2010, pp. 2424–2432.

[3] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," Mach. Learn., vol. 95, no. 3, pp. 291–327, 2014.

[4] X. Chen, A. Shrivastava, and A. Gupta, "Neil: Extracting visual knowledge from web data," in Proc. Int. Conf. Comput. Vis. (ICCV), 2013, pp. 1409–1416.

[5] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 4, pp. 754–766, 2011.

[6] L. Niu, W. Li, and D. Xu, "Visual recognition by learning from web data: A weakly supervised domain generalization approach," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2015, pp. 2774–2783.

[7] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2015, pp. 2691–2699.

[8] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, "WebVision database: Visual learning and understanding from web data," arXiv:1708.02862, 2017.

[9] H. Song, M. Kim, and J.-G. Lee, "SELFIE: Refurbishing unclean samples for robust deep learning," in Proc. Int. Conf. Mach. Learn. (ICML), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 5907–5915.

[10] 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. Comp. Vis. (ECCV), 2018, pp. 139–154.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.

[12] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," in Proc. Int. Conf. Mach. Learn. (ICML), 2017, pp. 233–242.

[13] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.

[14] 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. Conf. Comput. Vis. Pattern Recog. (CVPR), 2017, pp. 1944–1952.

[15] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 10 477–10 486.

[16] T. Mikolov, "Statistical language models based on neural networks," Ph.D. dissertation, Brno University of Technology, 2012.

[17] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), 2017, pp. 1310–1318.

[18] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, "Can gradient clipping mitigate label noise?" in Proc. Int. Conf. Learn. Represent. (ICLR), 2020, pp. 1–14.

[19] L. Jiang, Z. Zhou, T. Leung, 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. (ICML), 2018, pp. 92–100.

[20] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2010, pp. 1189–1197.

[21] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in Proc. AAAI Conf. Artif. Intell. (AAAI), 2015, pp. 2694–2700.

[22] E. Malach and S. Shalev-Shwartz, "Decoupling when to update from how to update," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 960–970.

[23] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networkswith extremely noisy labels," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 8535–8545.

[24] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 7164–7173.

[25] Q. Yao, H. Yang, B. Han, G. Niu, and J. T.-Y. Kwok, "Searching to exploit memorization effect in learning with noisy labels," in Proc. Int. Conf. Mach. Learn. (ICML), 2020, pp. 10 789–10 798.

[26] B. V. Rooyen, A. Menon, and R. C. Williamson, "Can gradient clipping mitigate label noise?" in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 1–14.

[27] F. Hutter, L. Kotthoff, and J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges. Springer, 2018.

[28] L. Jaehwan, Y. Donggeun, and K. Hyo-Eun, "Photometric transformer networks and label adjustment for breast density prediction," in Proc. Int. Conf. Comput. Vis. Workshops (ICCVW), 2019.

[29] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2018, pp. 5552–5560.

[30] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2019, pp. 7017–7025.

[31] S. J. Delany, N. Segata, and B. Mac Namee, "Profiling instances in noise reduction," Know.-Based Syst., vol. 31, pp. 28–40, 2012.

[32] L. P. Garcia, J. A. Sáez, J. Luengo, A. C. Lorena, A. C. de Carvalho, and F. Herrera, "Using the one-vs-one decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems," Know.-Based Syst., vol. 90, pp. 153–164, 2015.

[33] J. Luengo, S.-O. Shim, S. Alshomrani, A. Altalhi, and F. Herrera, "CNC-NOS: Class noise cleaning by ensemble filtering and noise scoring," Know.-Based Syst., vol. 140, pp. 27–49, 2018.

[34] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2U-Net: A simple noisy label detection approach for deep neural networks," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2019.

[35] Y. Ding, L. Wang, D. Fan, and B. Gong, "A aemi-aupervised two-stage approach to learning from noisy labels," in Proc. Winter Conf. Appli. Comput. Vis. (WACV), 2018, pp. 1215–1224.

[36] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, "SELF: Learning to filter noisy labels with self-ensembling," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.

[37] J. Li, R. Socher, and S. C. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.

[38] X. Wang, S. Wang, H. Shi, J. Wang, and T. Mei, "Co-Mining: Deep face recognition with noisy labels," in Proc. Int. Conf. Comput. Vis. (ICCV), 2019, pp. 9357–9366.

[39] P. Chen, B. B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 1062–1070.

[40] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 1195–1204.

[41] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2019.

[42] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 5, pp. 865–878, 2017.

[43] C. Shi, Z. Gu, C. Duan, and Q. Tian, "Multi-view adaptive semi-supervised feature selection with the self-paced learning," Signal Processing, vol. 168, p. 107332, 2020.

[44] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in Proc. Assoc. Comput. Linguistics (ACL), 1995, pp. 189–196.

[45] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in Proc. Conf. Comput. Vis. Pattern Recog. (CVPR), 2019, pp. 4685–4694.

[46] C. Fellbaum, WordNet: An Electronic Lexical Database. Bradford Books, 1998.

[47] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in Proc. Eur. Conf. Comp. Vis. (ECCV), 2016, pp. 630–645.

[49] D. P. Kingma and J. Ba, "Can gradient clipping mitigate label noise?" in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), 2019, pp. 8024–8035.

HIROSHI TAKEDA received the B.S. and M.E. degrees in Engineering Science from Kansai University, Japan, in 2019 and 2021, respectively. In 2021, he joined Accenture PLC. His research interests is learning with noisy labels and social media retrieval.

SOH YOSHIDA received his B.S., M.S., and Ph.D. degrees in Engineering and Information Science from Hokkaido University, Japan, in 2012, 2014, and 2016, respectively. He joined the Faculty of Engineering, Kansai University, Japan, in 2016. He is currently an Assistant Professor. His research interests are AI technologies related to Multimedia Processing and Information Retrieval. He is a member of the ACM, IEEE, IEICE, and ITE.

MITSUJI MUNEYASU received his B.S. and M.S. degrees in System Engineering from Kobe University, Japan, in 1982 and 1984, respectively, and his Ph.D. degree in Engineering from Hiroshima University, Japan, in 1993. He joined Oki Electric Industry Co., Ltd., Tokyo, Japan, in 1984. From 1990 to 1991, he was a Research Assistant at the Faculty of Engineering, Tottori University, Tottori, Japan. From 1991 to 2001, he was a Research Assistant and Associate Professor at the Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, Japan. He joined the Faculty of Engineering, Kansai University, Osaka, Japan, in 2001. He is currently a Professor. His research interests are image processing theory and nonlinear digital signal processing. He is a fellow of the IEICE and a member of the IEEE and IPSJ.