Robust Federated Learning with Noisy Labels

Seunghan Yang, Hyoungseob Park, Junyoung Byun, Changick Kim
KAIST, South Korea
{seunghan, hyoungseob, bjyoung, changick}@kaist.ac.kr

Abstract

Federated learning is a paradigm that enables local devices to jointly train a server model while keeping the data decentralized and private. In federated learning, since local data are collected by clients, it is hardly guaranteed that the data are correctly annotated. Although a lot of studies have been conducted to train the networks robust to these noisy data in a centralized setting, these algorithms still suffer from noisy labels in federated learning. Compared to the centralized setting, clients' data can have different noise distributions due to variations in their labeling systems or background knowledge of users. As a result, local models form inconsistent decision boundaries and their weights severely diverge from each other, which are serious problems in federated learning. To solve these problems, we introduce a novel federated learning scheme that the server cooperates with local models to maintain consistent decision boundaries by interchanging class-wise centroids. These centroids are central features of local data on each device, which are aligned by the server every communication round. Updating local models with the aligned centroids helps to form consistent decision boundaries among local models, although the noise distributions in clients' data are different from each other. To improve local model performance, we introduce a novel approach to select confident samples that are used for updating the model with given labels. Furthermore, we propose a global-guided pseudo-labeling method to update labels of unconfident samples by exploiting the global model. Our experimental results on the noisy CIFAR-10 dataset and the Clothing1M dataset show that our approach is noticeably effective in federated learning with noisy labels.

Introduction

Modern edge devices such as smart phones have been able to access an abundant amount of data, which is suitable for training deep learning models. Since each client device should transmit its local data to the central server for conventional centralized learning, it can lead to serious data privacy issues. To address these problems, federated learning has been actively studied to shift a learning environment from the central server to each edge device. In detail, federated learning allows a server model to be trained on each client's private data without transmitting raw data to the server.

The federated learning paradigm consists of two stages: 1) In the beginning of each round, a server broadcasts the server model to selected clients, and these clients train models on their own data for multiple iterations. 2) After the clients train their models, the server aggregates the clients' model parameters. The above process iterates until the global model converges. In FedAvg (McMahan et al. 2017), model parameters of clients are aggregated in an element-wise manner with coefficients, which are proportional to the local dataset size. The global model effectively converges by FedAvg, especially when the local dataset follows an i.i.d. distribution. Many studies have been conducted to apply it to practical applications, *e.g.*, dealing with non-i.i.d. data (Li et al. 2018; Zhao et al. 2018; Shoham et al. 2019; Wang et al. 2020; Li et al. 2020b), noisy communication (Ang et al. 2020), domain adaptation (Peng et al. 2020), fair resource allocation (Li et al. 2020a), and continual learning (Yoon et al. 2020).

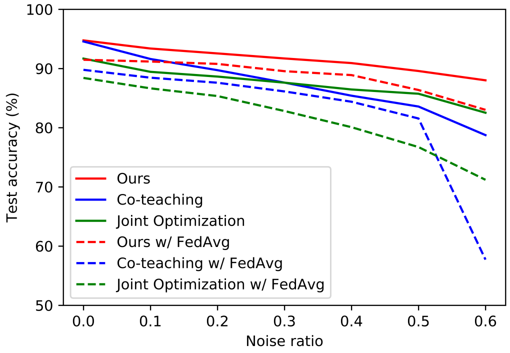


Figure 1: Test accuracy on the CIFAR-10 dataset at various noise ratios in the centralized setting (solid line) and the federated setting (dotted line). For federated learning with noisy labels, we distribute noisy data to clients in an i.i.d. fashion. Co-teaching (Han et al. 2018) and Joint Optimization (Tanaka et al. 2018) are novel methods for the centralized setting, but these algorithms combined with FedAvg (McMahan et al. 2017) suffer from performance degradation in the federated setting. Best viewed in color.

Although the above studies try to solve practical application issues related to preserving privacy, there are still remaining problems when local devices are used for training neural networks. In practice, all local data should be annotated by alternative labeling techniques such as exploiting machine-generated labels (Kuznetsova et al. 2018) due to privacy issues. These labels are inevitably corrupted unless the labeling techniques of all clients are accurate. Similarly, in the centralized setting, robust learning with noisy labels has attracted attention due to its applicability for realistic situations, and various algorithms have been proposed to train models accurately in the presence of noise. Recent algorithms have tried to minimize the effect of the noisy labels by sampling reliable data (Han et al. 2018; Wei et al. 2020; Huang et al. 2019; Guo et al. 2018), updating labels (Tanaka et al. 2018; Yi and Wu 2019), or estimating labels from matched prototypes (Han, Luo, and Wang 2019; Lee et al. 2018). These approaches have evolved into training the model with noisy labels successfully.

The aforementioned approaches for dealing with noisy labels suffer from performance degradation in the federated setting, as illustrated in Fig. 1. Unlike centralized learning, noise distributions in clients' data can be different from each other due to the discrepancy between their labeling systems or background knowledge. As a result, local models form inconsistent decision boundaries and their weights severely diverge from each other, *i.e.,* weight divergence. This causes aggregation difficulties of local models, which are a serious problem in federated learning (Li et al. 2018; Chen, Bhardwaj, and Marculescu 2020; Lim et al. 2020).

Therefore, in federated learning with noisy labels, different noise distributions in clients should be considered, and the learning directions of clients' models should be kept similar. To treat these difficulties, we introduce a new federated learning scheme that the server cooperates with local models to maintain consistent decision boundaries by interchanging class-wise centroids, as described in Fig. 2. In detail, we store local class-wise centroids on each device, which are central features of local data, and upload them on the server in every round. The server aggregates them into global centroids and broadcasts these centroids to clients. The centroids are used to update local models to maintain consistent decision boundaries with other clients, although the noise distributions in clients' data are different from each other.

In local updates, we compute local centroids based on samples with relatively small-losses to reduce the effect of noisy data, motivated by (Han et al. 2018). We adjust these centroids based on the similarity with global centroids to prevent them from being corrupted by representations of noisy data. Based on the centroids, we select confident samples to prohibit the model from fitting to noisy labels. We also utilize a global model for unconfident samples to correct the given labels, which alleviate overfitting to noisy samples in each local model.

To the best of our knowledge, this is the first federated learning algorithm dealing with noisy labels. We present a new federated learning scheme interchanging additional information called centroids and propose novel algorithms for reducing the effect of noisy data. Our approach maintains high performance on various noise ratios in the federated setting (Fig. 1).

Related work

Federated learning

Federated learning has drawn striking attention in recent years because of the increasing number of edge devices and local data collected by them. Federated learning aims to fully utilize the information of each local data without causing any serious data privacy and communication issues by transmitting the local network's parameters instead of local raw data. For preventing the server from those issues, there are several restrictions on federated learning, and they raise various problems: 1) statistical challenges (non-i.i.d. data), 2) lower network bandwidth, 3) inconsistent accuracy across devices, and 4) noisy communication. FedProx (Li et al. 2018), FedMA (Wang et al. 2020), and research about the convergence of FedAvg (Zhao et al. 2018; Li et al. 2020b) focus on the algorithms that converge the model in non-i.i.d. data. For the limitation of network bandwidth, DGC (Lin et al. 2018), signSGD (Bernstein et al. 2018), and STC (Sattler et al. 2019) only transmit important gradient changes. To maintain uniform accuracy of clients, Li et al. (Li et al. 2020a) propose fair resource allocation. For noisy communication, Ang et al. (Ang et al. 2020) aim to cope with the disturbance of noisy communication. Furthermore, various studies on specific tasks considering privacy-preserving are increased in a few years, *e.g.*, domain adaptation (Peng et al. 2020), and continual learning (Shoham et al. 2019; Yoon et al. 2020).

The aforementioned studies assume that every client has a clean dataset. However, correct annotation is not guaranteed since the local data are created by clients. Therefore, we consider that the local dataset consists of data with noisy labels and propose an algorithm to deal with it.

Learning on noisy data

There are many studies on the robustness of networks against noisy labels in the centralized setting. Since deep networks have sufficient capacity to fit on the whole noisy dataset (Zhang et al. 2016), it is essential to develop robust training methods against noisy labels. Noise cleaning-based approaches (Han et al. 2018; Guo et al. 2018; Huang et al. 2019; Lyu and Tsang 2019; Wei et al. 2020) aim to detect noisy samples and train with clean samples. In particular, Co-teaching (Han et al. 2018) introduces a pair of networks with the same structure, and each network guides its peer network by using its small-loss instances. Among the label correction approaches (Tanaka et al. 2018; Yi and Wu 2019), Joint Optimization (Tanaka et al. 2018) updates all dataset labels with pseudo-labels to prevent the network from fitting onto the noisy dataset. A new type of recent research focuses on label correction by adopting the representation power of the network to distinguish clean labels (Lee et al. 2018; Han, Luo, and Wang 2019). Deep self-learning (Han, Luo, and Wang 2019) determines the label of the sample by comparing its features with several prototypes of the categories. Furthermore, meta-learning based methods (Ren et al. 2018; Li et al. 2019; Shu et al. 2019) focus on optimizing parameters that are less prone to overfitting, and another effective approach is to design robust loss functions (Zhang

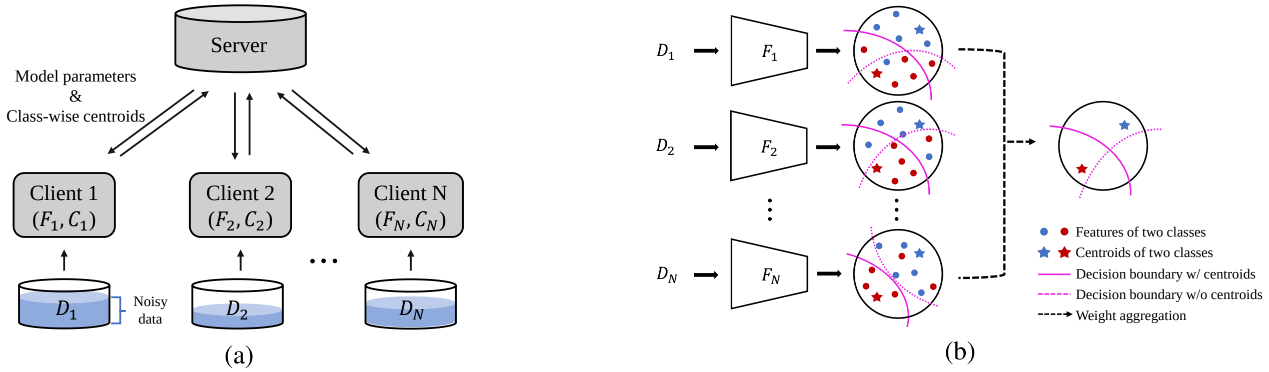


Figure 2: (a) An illustration of the federated setting with noisy labels, where each client has a dataset with a different noise ratio. Our proposed method permits the server and the clients to interchange class-wise centroids. (b) Without any restrictions on the local model, the clients' decision boundaries can be significantly different with each other since the client's model is trained on the individual noisy dataset for a large number of local epochs. Aggregating these local models can induce the server model to have incorrect boundaries. We leverage class-wise centroids to achieve local decision boundaries similar to the others.

and Sabuncu 2018; Wang et al. 2019) for a noise-tolerant model.

Previous algorithms for noisy labels aim to train networks in the centralized setting, not the federated setting. In the noisy federated setting case, clients have data with various noise distributions, and this can result in inconsistent decision boundaries and severe weight divergence in local models. To tackle these problems, we let the server cooperate with the clients to maintain consistent decision boundaries via class-wise centroids. By exploiting the centroids, we ensure that all local models have similar feature representations of classes. Moreover, we propose an algorithm for selecting confident samples and a self-training scheme suitable for the federated setting.

Robust federated learning with noisy labels

In this section, we start with the problem definition, then describe our proposed local update and global update methods.

Problem definition and notations

In the federated setting with multiple clients and a global server, local training data of the k-th client consist of images and the corresponding labels $D_k = \{(x_k^i, y_k^i)\}_{i=1}^{n_k}$, and the server cannot access any training data. In the noisy federated learning scenario, local datasets inevitably contain noise samples, where some of the given labels are not accurate, and noise distributions in clients' data are different from each other. The models can overfit to noisy data and suffer from weight divergence leading to aggregation difficulties in federated learning (Zhao et al. 2018; Lim et al. 2020).

To solve the above problem, we introduce global and local class-wise centroids, which are central features of each class in a server and clients, respectively. Local centroids are the average feature vectors from the global average pooling layer in each local dataset, and global centroids are calculated by reflecting the local centroids of selected clients, which are depicted in the next sections in detail. We denote global centroids and local centroids of the k-th client corresponding to class c by $\mathbf{f}_G^c$ and $\mathbf{f}_k^c$. In addition, $\mathbf{y}_k^i$ and $\hat{\mathbf{y}}_k^i$ indicate the one-hot vector of the ground truth label and a pseudo-label extracted by the softmax layer, respectively.

Local updates

At the beginning of each round, selected clients receive both global model parameters and global class-wise centroids from the server for local updates.

Before local updates, selected clients download the global model parameters, and their models are trained with their own local dataset by exploiting the following loss function:

$$\begin{aligned} L_c^k = & \mathbf{m}_k l_{ce}(C_k(F_k(\mathbf{x}_k)), \mathbf{y}_k) \\ & + (\mathbf{1} - \mathbf{m}_k) l_{ce}(C_k(F_k(\mathbf{x}_k)), \hat{\mathbf{y}}_k), \end{aligned} \tag{1}$$

where F_k and C_k denote the feature extractor and the classifier of the k-th client, respectively, and $l_{ce}(\cdot)$ is the cross-entropy loss function. A binary mask vector of the k-th client $\mathbf{m}_k \in \{0,1\}^{n_k}$ controls whether it learns the ground truth label or the pseudo-label. We propose a novel sampling approach to select confident samples that are used to update the mask $\mathbf{m}_k$. In addition, we introduce a global-guided pseudo-labeling method, which takes advantage of the federated setting. Instead of a naive pseudo-labeling (Tanaka et al. 2018), we obtain $\hat{\mathbf{y}}_k$ by exploiting the global model F_G and C_G. This method improves local model performance while preventing the model from overfitting to noisy data.

In parallel, each client loads global centroids from the server and updates its model to have similar features with the global centroids. To achieve this, we calculate local centroids on each local model depending on the similarity with global centroids, and we explicitly constrain local features to map local centroids. Note that the server and all of the clients store centroids in their own devices and transmit them in each communication round. There is an additional communication burden required for class-wise centroids, but the amount is much smaller than model parameters (0.01% to 0.03% in our experiments).

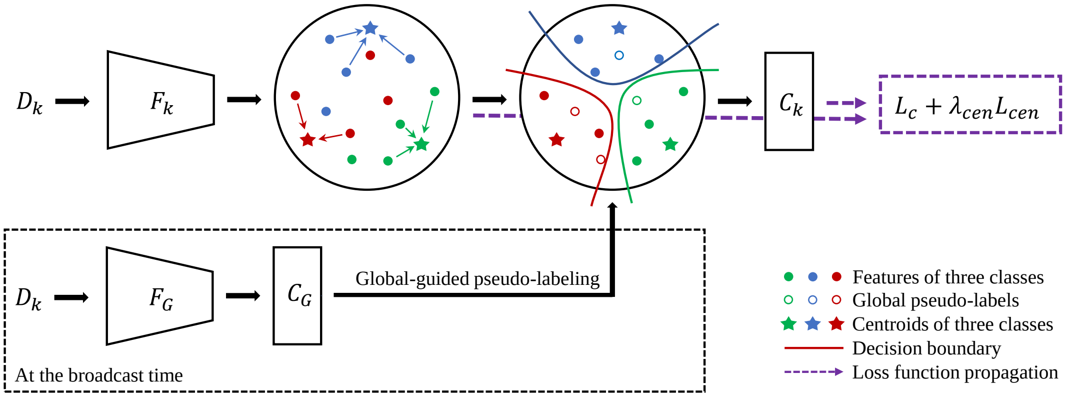

D_k
F_k
C_k
$L_c + \lambda_{cen}L_{cen}$
D_k
F_G
C_G
Global-guided pseudo-labeling
At the broadcast time
● ● ● Features of three classes
○ ○ ○ Global pseudo-labels
★ ★ ★ Centroids of three classes
Decision boundary
Loss function propagation

Figure 3: Our proposed local update algorithm. At the broadcast time, the server weights and global class-wise centroids are transmitted to each client. The client utilizes the server parameters (F_G and C_G) for global-guided pseudo-labeling and constrains local feature representations with the global centroids.

Local centroids. We use feature vectors extracted from F_k to compute local class-wise centroids $\mathbf{f}_k$. If we calculate $\mathbf{f}_k$ by using all local samples with given labels, noise labels have a negative effect on the correct formulation of centroids. Therefore, we introduce loss-based local centroids, which are motivated by (Han et al. 2018; Arpit et al. 2017). We only use features of samples with relatively small-losses to create accurate feature centroids. At first, we refine the dataset D_k by selecting $R(t)$ percentage of small-loss instances on each client as follows:

$$\hat{D}_k = \text{argmin}_{D'_k:|D'_k|\geq R(t)|D_k|} l_{ce}(D'_k), \qquad (2)$$

where D'_k is the optimization variable, $|\cdot|$ stands for the cardinality of a set the number of samples, and $R(t)$ controls how many small-loss samples should be selected in each round. Then, the k-th local model calculates naive average features of each class $\hat{\mathbf{f}}_k^c$ depending on the small-loss samples as follows:

$$\hat{\mathbf{f}}_k^c = \frac{1}{\tilde{n}_k^c} \sum_{x_k^i \in \hat{D}_k} F_k(x_k^i)\mathbb{1}(y_k^i = c), \qquad (3)$$

where $\tilde{n}_k^c$ is the number of samples corresponding to the label c in $\hat{D}_k$, and $\mathbb{1}(\cdot)$ is the indicator function returning 1 for true statements and 0 otherwise.

However, these average features may differ from the other clients'. To avoid these undesirable deviations, we derive local centroids $\mathbf{f}_k^c$ by weighted average depending on the similarity between global centroids $\mathbf{f}_G^c$ and the average features $\hat{\mathbf{f}}_k^c$ as follows:

$$\mathbf{f}_k^c = (1 - sim(\mathbf{f}_G^c, \hat{\mathbf{f}}_k^c)^2)\mathbf{f}_G^c + sim(\mathbf{f}_G^c, \hat{\mathbf{f}}_k^c)^2\hat{\mathbf{f}}_k^c, \qquad (4)$$

where $sim(\cdot,\cdot)$ can be any similarity function, but we choose cosine similarity for our experiments. $\hat{\mathbf{f}}_k^c$ is calculated by Eq. 3, and global centroids $\mathbf{f}_G^c$ are transmitted from the server reflecting entire clients' centroids, which is described in the next section.

We expect that class-wise centroids are the central features of clean samples. At the beginning of training, deep networks tend to prioritize learning simple patterns first (Arpit et al. 2017), and we exploit this property to form global centroids less susceptible to noisy samples. After that, we update local centroids to reflect similarity with these global centroids. This similarity-based update can keep centroids less corrupted by noisy data even after a large number of training epochs.

We exploit these local centroids to reduce weight divergence of clients' models. In detail, we design a loss function to map the features of the confident sample onto the centroids corresponding to the class as follows:

$$L_{cen}^k = \sum_{i=1}^{n_k} m_k^i \left\| F_k(x_k^i) - \mathbf{f}_k^{y_k^i} \right\|_2^2, \qquad (5)$$

where m_k^i denotes a binary mask of the k-th client that returns 1 for confident samples and 0 otherwise.

Confident samples. We introduce a sampling approach to select confident samples for training each client's model without a detrimental influence from noisy labels. To this end, we introduce the feature similarity-based labels as follows:

$$\tilde{y}_k^i = \text{argmax}_y sim(\mathbf{f}_k^y, F_k(x_k^i)). \qquad (6)$$

Note that we should not fully trust given labels because they may not be annotated accurately. Also they should not depend on feature similarity-based labels inducing the wrong labels for the hard samples. The complementary use of feature similarity-based and ground truth labels can help to find accurate confident samples. Therefore, we consider the similarity-based labels with local centroids and the ground truth labels at the same time. By adopting the ground truth label and the similarity-based label together, m_k^i for masking a confident sample is obtained as follows:

$$m_k^i = \mathbb{1}(\tilde{y}_k^i = y_k^i). \qquad (7)$$

We exploit this mask for L_c and L_{cen} to reduce the impact of noise samples. Since the number of confident samples is not fixed for each class, this mask can choose confident samples well regardless of different noise ratio for each class.

Global-guided pseudo-labeling. To fully utilize the local data information, we exploit the well-known label correction method (Tanaka et al. 2018). Although this self-learning strategy with pseudo-labeling is powerful for label correction in the centralized setting, it leads local models to be self-biased (Arazo et al. 2019). Therefore, we propose global-guided pseudo-labeling, which corrects labels of local data by employing the server model. Our technique for the label estimation prevents local models from being self-biased. Each client receives the global model at the broadcast time and uses the model to generate global-guided pseudo-labels $\hat{\mathbf{y}}_k$ as follows:

$$\hat{\mathbf{y}}_k = C_G(F_G(\mathbf{x_k})), \tag{8}$$

where C_G and F_G are the client's networks with global parameters. After that, each client trains its network with these global-guided pseudo-labels by Eq. 1.

Finally, the *k*-th local model is trained to minimize the sum of three losses:

$$L^k_{total} = L^k_c + \lambda_{cen} L^k_{cen} + \lambda_e L^k_e, \tag{9}$$

where L^k_e is the entropy regularization of prediction results. Note that this term forces probability distribution of each softmax output to a single class. $\mathbf{p}^i$ indicates softmax outputs $C_k(F_k(x^i_k))$, and the loss L^k_e is calculated by $-\sum_i \mathbf{p}^i \text{log} \mathbf{p}^i$. λ_{cen} and λ_e indicate trade-off parameters. Our complete algorithm is illustrated in Fig. 3.

Global updates

After the local update in each round, the clients upload model parameters and local centroids to the server. We exploit FedAvg (McMahan et al. 2017) for weight aggregation, which is well known as an effective algorithm for i.i.d. data. For centroid aggregation, the server updates global centroids by a similarity-based aggregation of uploaded local centroids. This leads the server model to explicitly deal with the different noise ratios in clients. Moreover, since it performs a class-wise summation of local centroids, it is less affected by different noise ratios in classes.

Weight aggregation. We execute FedAvg (McMahan et al. 2017) for weight aggregation, which is suitable for an i.i.d. dataset. Since only noisy labels are added in i.i.d. data, we expect that the FedAvg algorithm works well enough in our experimental settings. FedAvg takes a weighted average of local parameters θ_L as follows:

$$\theta_G = \sum_{k \in K} \frac{n_k}{n} \theta_{L,k}, \tag{10}$$

where θ_G is global parameters, and n and n_k indicate the total number of data and the number of the *k*-th client's data, respectively.

Global centroid aggregation. To tackle the different noise distributions in clients explicitly, we adjust global centroids to employ the similarity-based summation of local centroids. In every round, local centroids of selected clients update global centroids by using the similarity to previous global centroids in the server. Let K be the set of indices of clients selected in the current round, then global centroids are updated as follows:

$$\mathbf{f}^c_G = \frac{1}{\sum_{k \in K} w^c_k} \sum_{k \in K} w^c_k \mathbf{f}^c_k, \tag{11}$$

where w^c_k indicates similarity between stored global centroids $\hat{\mathbf{f}}^c_G$ and the uploaded the *k*-th client centroids of class c, and it is obtained by:

$$w^c_k = sim(\hat{\mathbf{f}}^c_G, \mathbf{f}^c_k). \tag{12}$$

Therefore, this weight update rule allows global centroids to reflect the similarity with local centroids, which depends on the client and class. The complete pseudo-code is shown in supplementary material.

Experiments

We adopt the following federated setting from FedAvg (McMahan et al. 2017). We set the number of clients to 100, and distribute each dataset to clients in an i.i.d. fashion. We select local epoch and local mini-batch to 5 and 50, respectively, considering communication efficiency and memory limitations of local devices.

Datasets

CIFAR-10. CIFAR-10 (Krizhevsky and Hinton 2009) is a benchmark dataset of 10 categories, which contains 50,000 images for training and 10,000 images for testing. We replace the ground truth labels of CIFAR-10 with two types of noisy labels: symmetric flipping (Van Rooyen, Menon, and Williamson 2015) and pair flipping (Han et al. 2018), which are described in supplementary material. Since our paper mainly focuses on the robustness in federated learning with various noisy ratios, the noise ratio ϵ is chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ for symmetric flipping and $\{0.1, 0.2, 0.3, 0.4, 0.45\}$ for pair flipping. In detail, we give noisy labels to the entire dataset with the designated noise ratio, and then randomly distribute it to 100 clients. This process induces different noise distributions in clients, and we fix the seed for a fair comparison. In ablation studies, we experiment with extremely different noise ratios.

Clothing1M. Clothing1M (Xiao et al. 2015) is a large real-world dataset of 14 categories, which contains 1 million images of clothing with noisy labels since it is obtained from several online shopping websites. In (Xiao et al. 2015), it is reported that the overall noise ratio is approximately 38.46%. The Clothing1M dataset also contains 50k, 14k, and 10k of clean data for training, validation, and testing, respectively, but we do not use the clean training data. For federated learning, we randomly divide the clothing1M dataset into 100 groups, which indicates the number of clients, and we set them as local datasets.

Table 1: Test accuracy on the CIFAR-10 dataset with symmetric and pair flipping noise. We report the average accuracy over the last 10 rounds.

Method	Test Accuracy (%)											
Noise type	Symmetric flipping							Pair flipping				
Noise ratio	0	0.1	0.2	0.3	0.4	0.5	0.6	0.1	0.2	0.3	0.4	0.45
Cross Entropy Loss (McMahan et al. 2017)	89.5	81.8	73.1	63.1	54.8	41.9	35.4	83.2	73.7	65.4	56.5	49.5
Co-teaching (Han et al. 2018)	89.8	88.5	87.6	86.1	84.4	81.6	57.7	88.4	86.3	73.1	57.5	55.2
Joint Optimization (Tanaka et al. 2018)	88.4	86.7	85.4	82.8	80.1	76.7	71.2	86.8	86.2	86.0	85.4	83.1
Ours	91.5	91.2	90.8	89.6	88.7	86.4	83.0	90.9	90.5	89.8	89.2	88.8

Table 2: Test accuracy on the Clothing1M dataset in the centralized (C. L.) and federated settings (F. L.).

Method	Test Acc. (%)	
Setting	C. L.	F. L.
Cross Entropy Loss (McMahan et al. 2017)	69.5	70.8
Co-teaching (Han et al. 2018)	71.0	71.8
Joint Optimization (Tanaka et al. 2018)	72.2	73.5
Deep self-learning (Han, Luo, and Wang 2019)	74.5	73.6
Ours	72.5	76.4

Table 3: Test accuracy with different noise ratios in clients (left) and classes (right) on the CIFAR-10 dataset.

η	Test Acc. (%)
ϵ	0.4
0	88.7
0.1	88.5
0.2	88.7
0.3	88.7
0.4	89.0

ϵ	Test Acc. (%)
0.1	91.1
0.2	90.5
0.3	89.6
0.4	88.6
0.5	86.8
0.6	83.1

Implementation details

We experiment with our proposed method, Cross Entropy Loss (McMahan et al. 2017), Co-teaching (Han et al. 2018), Joint Optimization (Tanaka et al. 2018), and Deep self-learning (Han, Luo, and Wang 2019) in our federated setting[1]. Note that we experiment these algorithms with FedAvg (McMahan et al. 2017) for weight aggregation. For CIFAR-10, we implemented the 9-Layer CNN applied in Co-teaching (Han et al. 2018). We do not report Deep self-learning because the 9-Layer CNN is not a pre-trained model, which does not have enough representation power to extract valid prototypes. Training on the Clothing1M dataset, we exploited ResNet-50 (He et al. 2016) pre-trained on ImageNet (Deng et al. 2009) by following (Tanaka et al. 2018; Han, Luo, and Wang 2019). To prevent overfitting to a small number of training data in each local dataset, we augment training data by resizing, normalizing, and cropping images. The detailed experimental setup is described in supplementary material.

Analysis

CIFAR-10. We report the results of CIFAR-10 with symmetric flipping and pair flipping in Table 1. As shown in Table 1, our method achieves better overall test accuracy at various noise ratios.

Co-teaching selects a fixed number of loss-based samples, which is vulnerable to different noise distributions in clients. It causes serious performance degradation of the server model since each client model is affected by noisy data. Joint Optimization follows a self-training scheme, which is a naive pseudo-labeling method. In federated learning, this self-training method may lead the network to be self-biased due to a large number of local epochs, and the weights of self-biased local models can be diverged severely. Notably in extremely noisy cases, clients cannot be trained properly due to high noise ratios, and when these local parameters are aggregated in the server, performance is further deteriorated. Our proposed method is also dependent on a loss-based algorithm but robust to different noise distributions in clients because of the similarity-based centroids update. Moreover, we exploit a global-guided pseudo-labeling method, which mitigates self-bias of each client, and we validate the effectiveness of this algorithm in ablation studies.

Furthermore, neither of the two previous methods can guarantee that all local models are trained to form similar decision boundaries, which makes aggregation of local models unsuccessful. Our proposed method constrains class representations of all client models not to diverge from global class representations. This induces all local models to be trained to have similar boundaries, and we demonstrate the efficacy of the algorithm in noisy federated learning.

Clothing1M. Different from the artificial noise in CIFAR-10, Clothing1M is a real-world noisy label dataset, including lots of unknown structure noise. By comparing the results in Table 2, we can see that our proposed method outperforms the others by a large margin in the federated setting. In the case of Deep self-learning (Han, Luo, and Wang 2019), which corrects the labels of the data comparing the similarity with the several prototypes of the features, it achieves great performance improvement in the centralized setting. However, in the federated setting, this algorithm suffers from the significant performance degradation since it does not constrain local models from having similar decision boundaries. We get the centroids with relatively small-losses and update them to be similar with global centroids. It can keep centroids less corrupted by noise data as well as achieve local decision boundaries similar to others. Our algorithm is ef-

[1] We use official codes for Co-teaching and Joint Optimization, and reproduce the code for Deep self-learning according to the paper.

Table 4: Effect of our sampling and pseudo-labeling methods.

mask	pseudo-label	Acc. (%)
x	x	74.5
o	x	77.8
x	naive	74.7
o	naive	86.7
o	global-guided	88.7

Table 5: Noisy label detection.

Noise type	Symmetric flipping					
Noise ratio	0.1	0.2	0.3	0.4	0.5	0.6
Precision	0.997	0.994	0.989	0.980	0.968	0.944
Recall	0.933	0.937	0.909	0.903	0.892	0.840

fective for the real-world noisy label dataset corrupted by unknown structure noise.

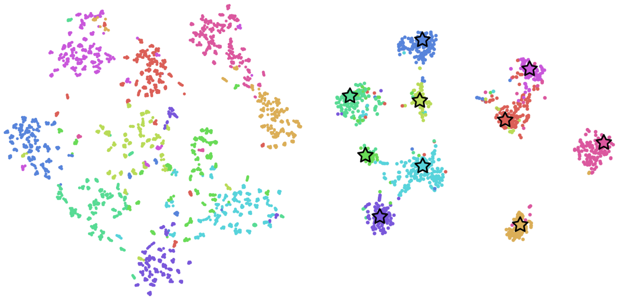
Ablation studies

We conduct ablation studies to show that each proposed algorithm is effective in federated learning with noisy labels.

Different noise ratios in clients. In the federated setting, clients may have different amounts of noise because of the discrepancy between clients' labeling systems. In detail, we split clients into five groups and assign different noise ratios. We set noise variance η, then divide noise range $[\epsilon - \eta, \epsilon + \eta]$ equally into five noise ratios and assign the noise ratio to each group. For example, if we set the noise ratio ϵ and noise variance η to 0.4 and 0.2, respectively, the noise ratio in each group is assigned one of $\{0.2, 0.3, 0.4, 0.5, 0.6\}$. In Table 3, we experiment by fixing the noise ratio to 0.4 and changing noise variance. Our approach achieves consistent performance regardless of the different noise ratios in clients.

Different noise ratios in classes. Due to background knowledge of the client user, the local data can have different noise ratios in classes. We assume an extreme situation where each client has totally erroneous samples for a single class. In detail, we force clients to have wrong labels for a single random class entirely and replace the labels of other classes with noisy labels of the designated noise ratio ϵ. We show that the proposed algorithm is robust to different noise ratios in classes in Table 3.

Confident samples. We evaluate the effectiveness of our sampling approach in Table 4. Note that we set the noise ratio to 0.4 by using symmetric flipping. As shown in Table 4, the mask for confident samples and the pseudo-labeling method complement each other. The network trained only with the selected samples by removing unconfident samples has better performance than the one trained with all samples, and the performance increases considerably when the unconfident samples' labels are replaced by pseudo-labels.



(a) Ours w/o global centroids (b) Ours w/ global centroids

Figure 4: Feature visualization of selected clients with t-SNE. We plot features of clients and global centroids. (a) Data are well clustered by their categories, however varying from clients, which hinders weight aggregation in the server. (b) Most clients have similar features around global class-wise centroids marked by stars. Best viewed in color.

Moreover, we show the precision and recall of noisy label detection of our sampling approach in Table 5. The precision means the number of correctly detected noisy labels over the entire number of detected noisy labels and the recall means the number of correctly detected noisy labels over the entire number of noisy labels in data. Our sampling approach selects confident samples by the complementary use of feature similarity-based and ground truth labels. It leads to high accuracy for the precision of noise labels.

Global-guided pseudo-labeling. We have conducted the experiments by removing global-guided pseudo-labeling or replacing the proposed method with naive pseudo-labeling in Table 4. Although the self-learning strategy with pseudo-labeling is powerful for label correction in centralized learning, it leads local models to be self-biased to their own datasets. Our global-guided pseudo-labeling outperforms a naive approach to prevent local models from being self-biased.

Interchanging class-wise centroids. To show the effectiveness of interchanging centroids, we experiment our algorithm while local models are updated without using global centroids. In detail, we use the loss function Eq. 5 by calculating local centroids without using Eq. 4, which cannot explicitly constrain local models to have similar boundaries. Figure 4 shows that global centroids help all clients to have similar feature representations, which leads to reducing weight divergence.

Conclusion

In this paper, we have considered that each local dataset may consist of noisy labels in the practical federated learning scenario. Our proposed approach is to interchange additional information, which is global and local feature centroids of classes. We demonstrate that our approach is clearly effective in the noisy federated setting by reducing weight divergence. Moreover, we propose a novel algorithm for local updates including similarity-based confident example sampling and global-guided pseudo-labeling. In extensive experiments, we have shown that our approach outperforms existing state-of-the-art methods on CIFAR-10 and Clothing1M.

Ethics statement

Our study suggests a practical learning scenario, especially learning with noisy labels. Based on our proposed federated learning with noisy labels, contributors in various fields such as healthcare, fairness, and recommendation system can indirectly benefit from our global guided update scheme. In the case of healthcare, medical data with wrong annotations can be a potential threat to the smart healthcare system. Our scheme can help prevent a medical accident due to erroneous data from occurring in federated learning, and it would play a crucial role in the development of smart healthcare. Our approach promotes social trends shifting a learning environment from the central server to various edge devices by allowing the models to learn without precise data. It enables client-side learning without precise data, which does not require an expert for annotating specialized data for each device. In our setting, we only focus on dealing with noisy labels on i.i.d. data. More work is needed for noisy labels in federated learning with non-i.i.d. data.

References

Ang, F.; Chen, L.; Zhao, N.; Chen, Y.; Wang, W.; and Yu, F. R. 2020. Robust Federated Learning With Noisy Communication. *IEEE Transactions on Communications* .

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2019. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983* .

Arpit, D.; Jastrzębski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Bernstein, J.; Zhao, J.; Azizzadenesheli, K.; and Anandkumar, A. 2018. signSGD with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291* .

Chen, W.; Bhardwaj, K.; and Marculescu, R. 2020. FedMAX: Mitigating Activation Divergence for Accurate and Communication-Efficient Federated Learning. *arXiv preprint arXiv:2004.03657* .

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE conference on computer vision and pattern recognition*. Ieee.

Guo, S.; Huang, W.; Zhang, H.; Zhuang, C.; Dong, D.; Scott, M. R.; and Huang, D. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision*.

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

Han, J.; Luo, P.; and Wang, X. 2019. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Huang, J.; Qu, L.; Jia, R.; and Zhao, B. 2019. O2U-Net: A Simple Noisy Label Detection Approach for Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical report* .

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Duerig, T.; et al. 2018. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982* .

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* .

Lee, K.-H.; He, X.; Zhang, L.; and Yang, L. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Li, J.; Wong, Y.; Zhao, Q.; and Kankanhalli, M. S. 2019. Learning to Learn From Noisy Labeled Data. In *Proceedings of IEEE conference on computer vision and pattern recognition*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. In *Proceedings of the 3rd MLSys Conference*.

Li, T.; Sanjabi, M.; Beirami, A.; and Smith, V. 2020a. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=ByexElSYDr.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020b. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=HJxNAnVtDS.

Lim, W. Y. B.; Luong, N. C.; Hoang, D. T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; and Miao, C. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials* .

Lin, Y.; Han, S.; Mao, H.; Wang, Y.; and Dally, B. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=SkhQHMW0W.

Lyu, Y.; and Tsang, I. W. 2019. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045* .

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas Aguera y, B. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch .

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1944–1952.

Peng, X.; Huang, Z.; Zhu, Y.; and Saenko, K. 2020. Federated Adversarial Domain Adaptation. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=HJezF3VYPB.

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* .

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050* .

Sattler, F.; Wiedemann, S.; Müller, K.-R.; and Samek, W. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* .

Shoham, N.; Avidor, T.; Keren, A.; Israel, N.; Benditkis, D.; Mor-Yosef, L.; and Zeitak, I. 2019. Overcoming Forgetting in Federated Learning on Non-IID Data. *arXiv preprint arXiv:1910.07796* .

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*.

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Van Rooyen, B.; Menon, A.; and Williamson, R. C. 2015. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated Learning with Matched Averaging. In *International Conference on Learning Representations*.

Wang, L.; and Wong, A. 2020. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. *arXiv preprint arXiv:2003.09871* .

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*.

Wei, H.; Feng, L.; Chen, X.; and An, B. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. *arXiv preprint arXiv:2003.02752* .

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yi, K.; and Wu, J. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yoon, J.; Jeong, W.; Lee, G.; Yang, E.; and Hwang, S. J. 2020. Federated Continual Learning with Adaptive Parameter Communication. *arXiv preprint arXiv:2003.03196* .

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* .

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* .

Supplementary Material: Robust Federated Learning with Noisy Labels

Algorithm details

Initialization of global centroids. Since we update local centroids using similarities with global centroids, randomly initialized global centroids hinder local models from deriving accurate local centroids. For this reason, average features $\hat{\mathbf{f}}_k$ are used instead of global centroids in the first round. After that, global centroids are computed by aggregating clients' local centroids.

Loss-based centroids. At the beginning of training, deep networks tend to prioritize learning simple patterns first (Arpit et al. 2017). We utilize this property to keep more instances at the start, *i.e.*, $R(t)$ is large in Eq. 2. As the training proceeds, we gradually reduce $R(t)$ to prevent local models from fitting to noise samples following (Han et al. 2018). We set T and τ to 10 and ϵ in our experiments, and we show that our approach is robust to these parameters in the experimental section.

Pseudo-labels. In Eq. 1, we train the network with unconfident samples by using pseudo-labels. At the early stage, the network cannot generate accurate pseudo-labels $\hat{\mathbf{y}}$ due to insufficient training time. Therefore, at first, we replace pseudo-labels $\hat{\mathbf{y}}$ with ground truth labels $\mathbf{y}$. After the number of rounds reaches predefined number (T_{pl}), we exploit pseudo-labels, then we train the network jointly by $\mathbf{y}$ and $\hat{\mathbf{y}}$.

Scheduling for λ_{cen}. To avoid the noisy mask at the early stage of the training procedure, we initialize λ_{cen} to 0 and gradually increase it to a predefined number.

Mini-batch algorithm. We modify Eq. 4 for mini-batch SGD as follows:

$$\mathbf{f}_{k,j} = (1 - sim(\mathbf{f}_{k,j-1}, \hat{\mathbf{f}}_{k,j})^2)\mathbf{f}_{k,j-1} + sim(\mathbf{f}_{k,j-1}, \hat{\mathbf{f}}_{k,j})^2\hat{\mathbf{f}}_{k,j}, \tag{13}$$

where $\mathbf{f}_{k,j}$ indicates local centroids at j-th iteration. The full algorithm for our local and global updates is given in Algorithm 1. Note that $\mathbf{f}_{k,0}$ indicates global centroids $\mathbf{f}_G$ in the first local epoch.

Two types of noisy labels

Since CIFAR-10 (Krizhevsky and Hinton 2009) and MNIST (LeCun et al. 1998) are clean, following (Reed et al. 2014; Patrini et al. 2017), we manually corrupt these datasets by the label transition matrix Q, where $Q_{ij} = \Pr(\tilde{y} = j | y = i)$ given that noisy $\tilde{y}$ is flipped from clean y. For symmetric flipping, we inject the symmetric label noise as follows:

$$Q = \begin{bmatrix} 1-\epsilon & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} & \frac{\epsilon}{n-1} \\ \frac{\epsilon}{n-1} & 1-\epsilon & & & \frac{\epsilon}{n-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\epsilon}{n-1} & & & 1-\epsilon & \frac{\epsilon}{n-1} \\ \frac{\epsilon}{n-1} & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} & 1-\epsilon \end{bmatrix}, \tag{14}$$

where n is the number of classes and ϵ indicates the noise ratio. Pair flipping is a well-known noise generation method that focuses on fine-grained classification with noisy labels, and its noise transition matrix Q is obtained as follows:

$$Q = \begin{bmatrix} 1-\epsilon & \epsilon & 0 & \cdots & 0 \\ 0 & 1-\epsilon & \epsilon & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1-\epsilon & \epsilon \\ \epsilon & 0 & \cdots & 0 & 1-\epsilon \end{bmatrix}. \tag{15}$$

Implementation details

We used the Pytorch framework (Paszke et al. 2017) to implement the model, and training was done on a GTX 1080Ti and Intel i7-6700k@4.00GHz. We utilized the official code provided by authors for Co-teaching (Han et al. 2018). We modified the official code to Pytorch version of Joint Optimization (Tanaka et al. 2018) and reproduced the model in Deep self-learning (Han, Luo, and Wang 2019) according to the paper. In our federated setting, we set the number of clients to 100, and 10 clients are selected at every round. Batch size and local epoch are 50 and 5, and we trained the network during 100, 1000, and 40 rounds for MNIST, CIFAR-10, and Clothing1M, respectively. We used SGD optimizer with a momentum of 0.5, a weight decay of 10^{-4} for all algorithms. Next, we describe the parameters for each algorithm.

Ours. We determined balance parameters (λ_{cen} and λ_e) and T_{pl} based on ablation studies and the previous work (Tanaka et al. 2018). We set λ_{cen} and λ_e to 1.0 and 0.8 for all datasets, and T_{pl} to 30 for MNIST, 100 for CIFAR-10, and 5 for Clothing1M. The initial learning rate is 0.25 for MNIST and CIFAR-10, and 0.01 for Clothing1M. For the Clothing1M dataset, the learning rate was decreased by 10 every 10 rounds. In addition, to obtain loss-based centroids, we set $T = 10$ and $\tau = \epsilon$ by following (Han et al. 2018). We show that our approach is robust to all these parameters in ablation studies. For the centralized setting, we modified our algorithm by replacing global-guided pseudo-labeling with naive pseudo-labeling and calculating global centroids to use all data.

Co-teaching. We took two networks with the same architecture but different initializations as two classifiers. Different from (Han et al. 2018), we used SGD optimizer with an initial learning rate of 0.15 for federated learning (McMahan et al. 2017). We found that this setting has similar test accuracy to the original setting in the centralized setting through experiments. We set $R(t) = 1 - \tau \min(t/T, 1)$ with $T = 10$ for MNIST and CIFAR-10 and $T = 5$ for Clothing1M, and $\tau = \epsilon$ by following (Han et al. 2018; Wei et al. 2020). We initialized the learning rate to 0.15 for MNIST and CIFAR-10, and 0.001 for Clothing1M. For Clothing1M, the learning rate was decreased by 10 every 10 rounds.

Joint Optimization. We determined balance parameters (λ_α and λ_β), start epoch, and learning rate based on (Tanaka et al. 2018). For the MNIST and CIFAR-10 datasets, we used the different learning rates suitable for each noise ratio following (Tanaka et al. 2018). We set α, β, and start epoch to 1.2, 0.8, and 100 respectively in MNIST and CIFAR-10. For Clothing1M, we used a learning rate of 8×10^{-4}, and used 2.4 for α and 0.8 for β.

Deep self-learning. The initial learning rate was 0.002 and decreased by 10 every 5 epochs. We followed the paper (Han, Luo, and Wang 2019) to choose hyper-parameters except the number of selected samples and prototypes. In the federated setting, each client only has a small portion of the original dataset. For this reason, in the label correction phase, we randomly sample 128 images for each class and 3 class prototypes are picked out for each class.

More experimental results

Experimental results on MNIST. MNIST (LeCun et al. 1998) is a benchmark dataset of 10 categories, which contains

```
Algorithm 1: Robust Federated Learning with Noisy Labels
Input: global weights $\theta_G$, global centroids $\mathbf{f}_G$, learning rate $\eta$, start round that uses pseudo-labels $T_{pl}$, fixed $\tau$,
round $T$ and $T_{max}$;
Server executes:
    initialize $\theta_G$;
    for each round $t = 1, ..., T_{max}$ do
        $S_t \leftarrow$ (random set of m clients);
        for each client $k \in S_t$ in parallel do
            Load $\theta_k$, $\mathbf{f}_k \leftarrow$ LocalUpdate($k$, $t$, $\theta_G$, $\mathbf{f}_G$, $R(t)$);
        Update global weights $\theta_G$ by Eq. 10;
        Update global centroids $\mathbf{f}_G$ by Eq. 11;
        Update $R(t) = 1 - \min\{\frac{t}{T}\tau, \tau\}$;

function LocalUpdate($k$, $t$, $\theta_G$, $\mathbf{f}_G$, $R(t)$): // Run on client k
    Load $\theta_k \leftarrow \theta_G$;
    if $t = 1$ then // Initialization of global centroids
        Obtain naive average features $\hat{\mathbf{f}}_k$ by Eq. 3 from $D_k$;
        Load $\mathbf{f}_{k,0} \leftarrow \hat{\mathbf{f}}_k$;
    else
        Load $\mathbf{f}_{k,0} \leftarrow \mathbf{f}_G$
    Obtain global-guided pseudo labels $\hat{\mathbf{y}}$ by Eq. 8 from $D_k$;
    for each local epoch $i = 1, ..., E$ do
        Shuffle training set $D_k$;
        for $j = 1, ..., N_{max}$ do
            Fetch mini batch $D_{k,j}$ from $D_k$;
            Obtain small-loss sets $\hat{D}_{k,j}$ by Eq. 2 from $D_{k,j}$;
            Obtain confident mask vector $\mathbf{m}_{k,j}$ by Eq. 7;
            if $t < T_{pl}$ then
                Load $\hat{\mathbf{y}} \leftarrow \mathbf{y}$; // Replacing pseudo-labels with ground truth labels
            Update local weights $\theta_k$ by minimizing Eq. 9;
            Obtain loss-based average features $\hat{\mathbf{f}}_{k,j}$ by Eq. 3 from $\hat{D}_{k,j}$;
            Update local centroids $\mathbf{f}_{k,j}$ by Eq. 13;
        Load $\mathbf{f}_{k,0} \leftarrow \mathbf{f}_{k,N_{max}}$;
    Output: $\theta_k$ and $\mathbf{f}_{k,0}$.
```

Table 6: Test accuracy on the MNIST dataset with symmetric and pair flipping noise. We report the average accuracy over the last 10 rounds.

Method	Test Accuracy (%)											
Noise type	Symmetric flipping							Pair flipping				
Noise ratio	0	0.1	0.2	0.3	0.4	0.5	0.6	0.1	0.2	0.3	0.4	0.45
Cross Entropy Loss (McMahan et al. 2017)	99.6	98.7	97.4	94.2	88.1	79.2	62.5	98.7	95.6	86.8	69.4	61.6
Co-teaching (Han et al. 2018)	99.4	99.4	99.2	99.1	98.9	98.1	96.7	99.3	99.2	99.1	96.6	92.3
Joint Optimization (Tanaka et al. 2018)	98.1	97.9	97.9	97.3	97.0	97.1	94.1	98.0	97.3	97.3	97.3	97.2
Ours	99.5	99.4	99.3	99.3	99.2	99.1	98.8	99.4	99.4	99.3	99.2	99.1

Figure 5: Image variances in intensity.

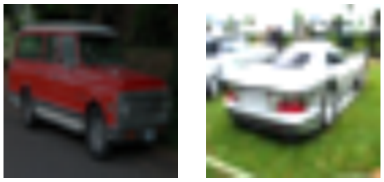
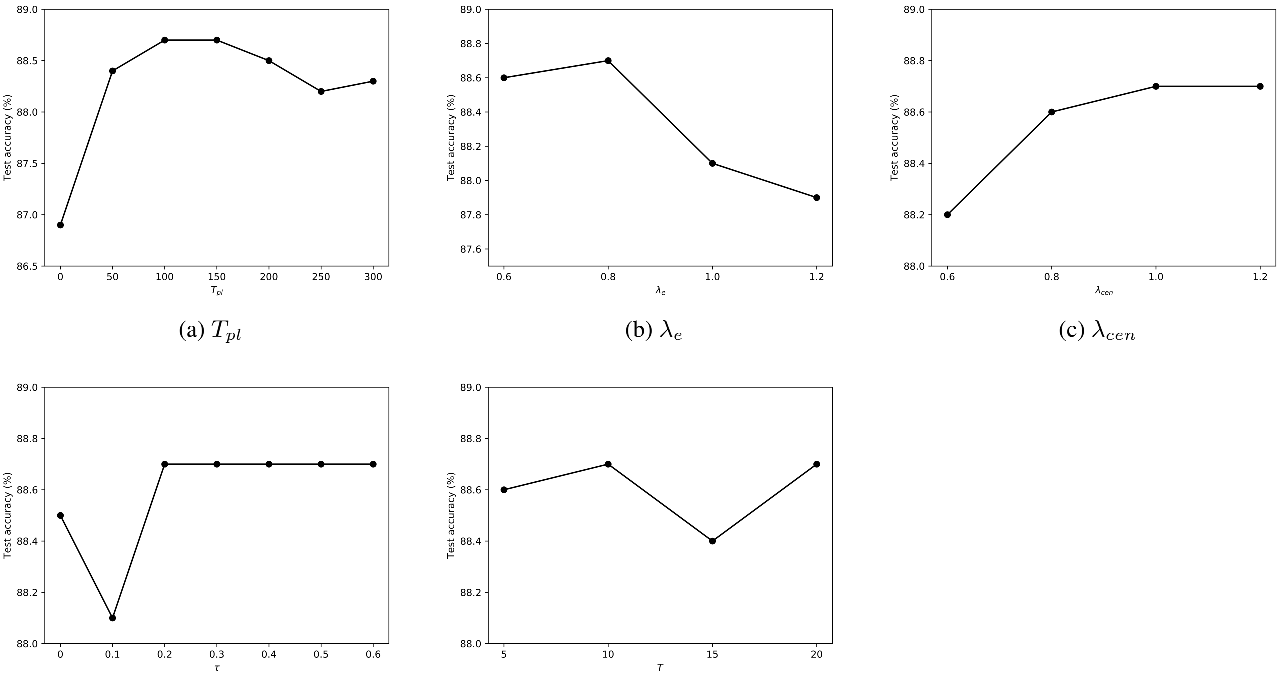


Table 7: Test accuracy on the various number of participating clients in each round.

# of clients	1	2	5
Acc. (%)	78.6	85.9	88.2
# of clients	20	50	100
Acc. (%)	89.6	90.1	89.9

Table 8: Computational time. GTX 1080Ti and Intel i7-6700k@4.00GHz.

Method	time (s)
Co-teaching	45.8
Joint Optimization	24.9
Ours	27.8



(a) T_{pl} (b) λ_e (c) λ_{cen}

(d) τ (e) T

Figure 6: Performance dependency of hyper-parameters.

60,000 images for training and 10,000 images for testing. We replace the ground truth labels of MNIST with two types of noisy label: symmetric and pair flipping. We implemented the 9-Layer CNN applied in Co-teaching (Han et al. 2018). In Table 6, our proposed approach maintain high performance at various noise ratios.

Image variances in different domains. In the real world, the data would differ from client to client not only in terms of noise in the labels but also the features themselves. We investigate this situation in federated learning. We assume that clients are in different environments, especially light intensity, *e.g.*, a photo of a car in one client would be different from that in others in terms of light intensity, as illustrated in Fig. 5. In detail, we use the noisy CIFAR-10 dataset with noise ratio 0.4, and change the light intensity of each client dataset at a rate of specific value within 0.5 to 1.5 times by using the ImageEnhance module in the PIL package. Our algorithm achieves 88.4%, although the noise exists in both labels and features (88.7% in the original setting).

Real federated learning scenario. For centralized learning, transmitting medical data from each medical center to the central server causes privacy issues. In consideration of these issues, we experiment our algorithm under the assumption that the medical dataset is in each medical center. We choose the COVIDx Dataset (Wang and Wong 2020), which consists of 15,282 chest x-ray images, and distribute the dataset to 100 medical centers (clients). Our approach achieves 93.5%, which is comparable to the centralized setting (93.6%).

Number of participating clients. In real-world federated learning scenarios, the population base can be significantly larger and a considerably smaller portion can be selected every round. We provide experimental results on the noisy CIFAR-10 dataset with noise ratio 0.4 by changing the number of participating clients per round in Table 7. Note that the client population is 100. Even if only two clients participate in communication, it shows comparable performance by reducing weight divergence of clients' models.

Computational cost. Since our proposed algorithm increases computation-cost not only for local updates but also for global updates, we measure the time from the start of the round to the next round. As shown in Table 8, the speed of our algorithm is similar to that of Joint Optimization (Tanaka et al. 2018). Due to the use of similarity-based updates and confident samples, our algorithm has a marginal increase in computational cost. Since Co-teaching (Han et al. 2018) exploits two networks, its computational time is longer than others that use only one network.

Performance dependency of hyper-parameters. We use the noisy CIFAR-10 dataset and set the noise ratio ϵ to 0.4 by using symmetric flipping. We set T_{pl}, λ_e, λ_{cen}, τ, T to 100, 0.8, 1.0, 0.4, 10, respectively. Then, we have conducted various experiments changing hyper-parameters, *i. e.,* T_{pl}, λ_e, λ_{cen}, τ, and T. As shown in Fig. 6, the prediction accuracy is robust to the hyper-parameters except T_{pl} that is related to replacing ground truth labels with pseudo labels. Since the network cannot generate accurate pseudo-labels $\hat{\mathbf{y}}$ at the early stage, it achieves lower performance compared to networks trained with high T_{pl}.