FedCorr: Multi-Stage Federated Learning for Label Noise Correction

Jingyi Xu[1]* Zihan Chen[1,2]* Tony Q.S. Quek[1] Kai Fong Ernest Chong[1]†
[1]Singapore University of Technology and Design [2]National University of Singapore
{jinyi_xu,zihan_chen}@mymail.sutd.edu.sg {tonyquek,ernest_chong}@sutd.edu.sg

Abstract

Federated learning (FL) is a privacy-preserving distributed learning paradigm that enables clients to jointly train a global model. In real-world FL implementations, client data could have label noise, and different clients could have vastly different label noise levels. Although there exist methods in centralized learning for tackling label noise, such methods do not perform well on heterogeneous label noise in FL settings, due to the typically smaller sizes of client datasets and data privacy requirements in FL. In this paper, we propose `FedCorr`*, a general multi-stage framework to tackle heterogeneous label noise in FL, without making any assumptions on the noise models of local clients, while still maintaining client data privacy. In particular, (1)* `FedCorr` *dynamically identifies noisy clients by exploiting the dimensionalities of the model prediction subspaces independently measured on all clients, and then identifies incorrect labels on noisy clients based on per-sample losses. To deal with data heterogeneity and to increase training stability, we propose an adaptive local proximal regularization term that is based on estimated local noise levels. (2) We further finetune the global model on identified clean clients and correct the noisy labels for the remaining noisy clients after finetuning. (3) Finally, we apply the usual training on all clients to make full use of all local data. Experiments conducted on CIFAR-10/100 with federated synthetic label noise, and on a real-world noisy dataset, Clothing1M, demonstrate that* `FedCorr` *is robust to label noise and substantially outperforms the state-of-the-art methods at multiple noise levels.*

1. Introduction

Federated learning (FL) is a promising solution for large-scale collaborative learning, where clients jointly train a machine learning model, while still maintaining local data privacy [18, 24, 34]. However, in real-world FL implementations over heterogeneous networks, there may be differences in the characteristics of different clients due to diverse annotators' skill, bias, and hardware reliability [4,35]. Client data is rarely IID and frequently imbalanced. Also, some clients would have clean data, while other clients may have data with label noise at different noise levels. Hence, the deployment of practical FL systems would face challenges brought by discrepancies in two aspects i): local data statistics [5, 12, 19, 24], and ii): local label quality [4, 35]. Although recent works explored the discrepancy in local data statistics in FL, and learning with label noise in centralized learning (CL), there is at present no unified approach for tackling both challenges simultaneously in FL.

The first challenge has been explored in recent FL works, with a focus on performance with convergence guarantees [20, 25]. However, these works have the common implicit assumption that the given labels of local data are completely correct, which is rarely the case in real-world datasets.

The second challenge can be addressed by reweighting [4, 7, 28] or discarding [33] those client updates that are most dissimilar. In these methods, the corresponding clients are primarily treated as malicious agents. However, dissimilar clients are not necessarily malicious and could have label noise in local data that would otherwise still be useful after label correction. For FL systems, the requirement of data privacy poses an inherent challenge for any label correction scheme. *How can clients identify their noisy labels to be corrected without needing other clients to reveal sensitive information?* For example, [35] proposes label correction for identified noisy clients with the guidance of extra data feature information exchanged between clients and server, which may lead to privacy concerns.

Label correction and, more generally, methods to deal with label noise, are well-studied in CL. Yet, even state-of-the-art CL methods for tackling label noise [3, 8, 9, 16, 27, 30, 32, 37], when applied to local clients, are inadequate in mitigating the performance degradation in the FL setting, due to the limited sizes of local datasets. These CL methods cannot be applied on the global sever or across multiple

*Equal contributions. † Corresponding author.
Code: https://github.com/Xu-Jingyi/FedCorr

This work is supported by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG-RP-2019-015), and under its NRFF Program (NRFFAI1-2019-0005). This work is also supported in part by the SUTD Growth Plan Grant for AI.

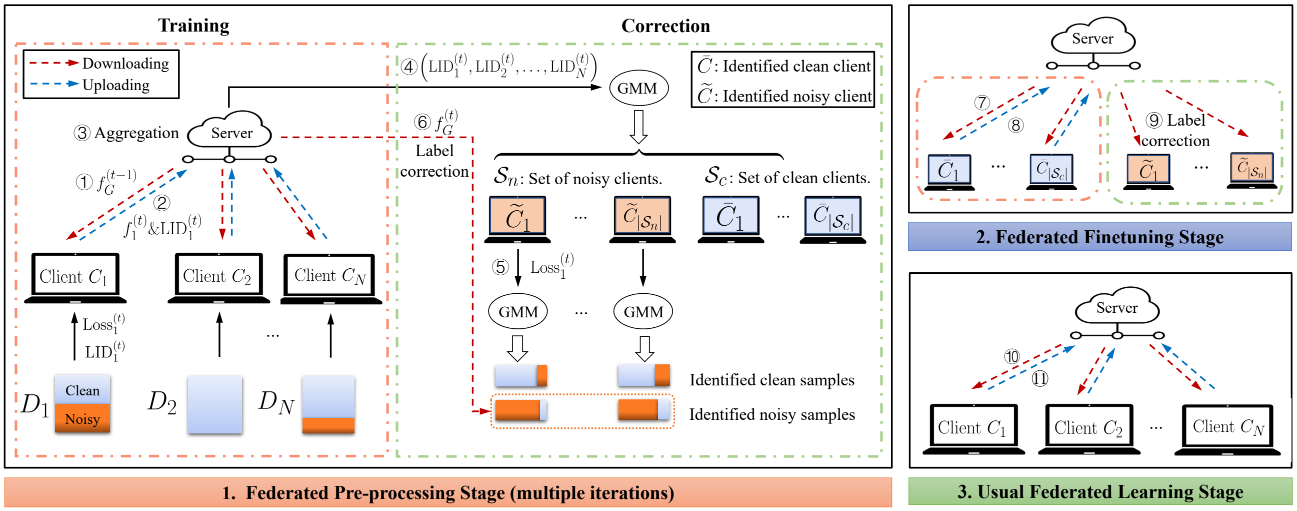
Training
Correction
Downloading
Uploading
③ Aggregation
Server
① $f_G^{(t-1)}$
② $f_1^{(t)}$&$\mathrm{LID}_1^{(t)}$
Client C_1
Client C_2
Client C_N
$\mathrm{Loss}_1^{(t)}$
$\mathrm{LID}_1^{(t)}$
...
D_1
Clean
Noisy
D_2
D_N
④ $\left(\mathrm{LID}_1^{(t)}, \mathrm{LID}_2^{(t)}, \ldots, \mathrm{LID}_N^{(t)}\right)$
GMM
$\bar{C}$: Identified clean client
$\tilde{C}$: Identified noisy client
⑥ $f_G^{(t)}$
Label correction
$\mathcal{S}_n$: Set of noisy clients.
$\mathcal{S}_c$: Set of clean clients.
$\tilde{C}_1$
$\tilde{C}_{|\mathcal{S}_n|}$
$\bar{C}_1$
$\bar{C}_{|\mathcal{S}_c|}$
⑤ $\mathrm{Loss}_1^{(t)}$
GMM
GMM
Identified clean samples
Identified noisy samples
1. Federated Pre-processing Stage (multiple iterations)
Server
⑦
⑧
⑨ Label correction
$\bar{C}_1$
$\bar{C}_{|\mathcal{S}_c|}$
$\tilde{C}_1$
$\tilde{C}_{|\mathcal{S}_n|}$
2. Federated Finetuning Stage
Server
⑩
⑪
Client C_1
Client C_2
Client C_N
3. Usual Federated Learning Stage

Figure 1. An overview of FedCorr, organized into three stages. Algorithm steps are numbered accordingly.

clients due to FL privacy requirements. So, it is necessary and natural to adopt a more general framework that jointly considers the two discrepancies, for a better emulation of real-world data heterogeneity. Most importantly, privacy-preserving label correction should be incorporated in training to improve robustness to data heterogeneity in FL.

In this paper, we propose a multi-stage FL framework to simultaneously deal with both discrepancy challenges; see Fig. 1 for an overview. To ensure privacy, we introduce a dimensionality-based filter to identify noisy clients, by measuring the local intrinsic dimensionality (LID) [10] of local model prediction subspaces. Extensive experiments have shown that clean datasets can be distinguished from noisy datasets by the behavior of LID scores during training [22,23]. Hence, in addition to the usual local weight updates, we propose that each client also sends an LID score to the server, which is a single scalar representing the discriminability of the predictions of the local model. We then filter noisy samples based on per-sample training losses independently for each identified noisy client, and relabel the large-loss samples with the predicted labels of the global model. To improve training stability and alleviate the negative impact caused by noisy clients, we introduce a weighted proximal regularization term, where the weights are based on the estimated local noise levels. Furthermore, we finetune the global model on the identified clean clients and relabel the local data for the remaining noisy clients.

Our main contributions are as follows:

- We propose a general multi-stage FL framework FedCorr to tackle data heterogeneity, with respect to both local label quality and local data statistics.
- We propose a general framework for easy generation of federated synthetic label noise and diverse (e.g. non-IID) client data partitions.
- We identify noisy clients via LID scores, and identify noisy labels via per-sample losses. We also propose an adaptive local proximal regularization term based on estimated local noise levels.
- We demonstrate that FedCorr outperforms state-of-the-art FL methods on multiple datasets with different noise levels, for both IID and non-IID data partitions.

2. Related work

2.1. Federated methods

In this paper, we focus on three closely related aspects of FL: generation of non-IID federated datasets, methods to deal with non-IID local data, and methods for robust FL.

The generation of non-IID local data partitions for FL was first explored in [24], based on dividing a given dataset into shards. More recent non-IID data partitions are generated via Dirichlet distributions [1, 12, 28].

Recent federated optimization work mostly focus on dealing with the discrepancy in data statistics of local clients and related inconsistency issues [1, 19, 29]. For instance, FedProx deals with non-IID local data, by including a proximal term in the local loss functions [19], while FedDyn uses a dynamic proximal term based on selected clients [1]. SCAFFOLD [14] is another method suitable for non-IID local data that uses control variates to reduce client-drift. In [12] and [25], adaptive FL optimization methods for the global server are introduced, which are compatible with non-IID data distributions. Moreover, the Power-of-Choice (PoC) strategy [6], a biased client selection scheme that selects clients with higher local losses, can be used to increase the rate of convergence.

There are numerous works on improving the robustness of FL; these include robust aggregation methods [7, 17, 28], reputation mechanism-based contribution examining [33], credibility-based re-weighting [4], distillation-based semi-supervised learning [13], and personalized multi-task learning [17]. However, these methods are not designed for identifying noisy labels. Even when these methods are used to detect noisy clients, either there is no mechanism for further label correction at the noisy clients [7, 17, 28, 33], or the effect of noisy labels is mitigated with the aid of an auxiliary dataset, without any direct label correction [4, 13]. One notable exception is [35], which carries out label correction during training by exchanging feature centroids between clients and server. This exchange of centroids may lead to privacy concerns, since centroids could potentially be used as part of reverse engineering to reveal non-trivial information about raw local data.

In contrast to these methods, FedCorr incorporates the generation of diverse local data distributions with synthetic label noise, together with noisy label identification and correction, without privacy leakage.

2.2. Local intrinsic dimension (LID)

Informally, LID [10] is a measure of the intrinsic dimensionality of the data manifold. In comparison to other measures, LID has the potential for wider applications as it makes no further assumptions on the data distribution beyond continuity. The key underlying idea is that at each datapoint, the number of neighboring datapoints would grow with the radius of neighborhood, and the corresponding growth rate would then be a proxy for "local" dimension.

LID builds upon this idea [11] via the geometric intuition that the volume of an m-dimensional Euclidean ball grows proportionally to r^m when its radius is scaled by a factor of r. Specifically, when we have two m-dimensional Euclidean balls with volumes V_1, V_2, and with radii r_1, r_2, we can compute m as follows:

$$\frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^m \Rightarrow m = \frac{\log(V_2/V_1)}{\log(r_2/r_1)}. \tag{1}$$

We shall now formally define LID. Suppose we have a dataset consisting of vectors in $\mathbb{R}^n$. We shall treat this dataset as samples drawn from an n-variate distribution $\overline{\mathcal{D}}$. For any $x \in \mathbb{R}^n$, let Y_x be the random variable representing the (non-negative) distance from x to a randomly selected point y drawn from $\overline{\mathcal{D}}$, and let $F_{Y_x}(t)$ be the cumulative distribution function of Y_x. Given $r > 0$ and a sample point x drawn from $\overline{\mathcal{D}}$, define the *LID of x at distance r* to be

$$\mathrm{LID}_x(r) := \lim_{\varepsilon \to 0} \frac{\log F_{Y_x}((1+\varepsilon)r) - \log F_{Y_x}(r)}{\log(1+\varepsilon)},$$

provided that it exists, i.e. provided that $F_{Y_x}(t)$ is positive and continuously differentiable at $t = r$. The *LID at x* is defined to be the limit $\mathrm{LID}_x = \lim_{r\to 0} \mathrm{LID}_x(r)$. Intuitively, the LID at x is an approximation of the dimension of a smooth manifold containing x that would "best" fit the distribution $\overline{\mathcal{D}}$ in the vincinity of x.

Estimation of LID: By treating the smallest neighbor distances as "extreme events" associated to the lower tail of the underlying distance distribution, [2] proposes several estimators of LID based on extreme value theory. In particular, given a set of points $\mathcal{X}$, a reference point $x \in \mathcal{X}$, and its k nearest neighbors in $\mathcal{X}$, the maximum-likelihood estimate (MLE) of x is:

$$\widehat{\mathrm{LID}}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(x)}{r_{max}(x)}\right)^{-1}, \tag{2}$$

where $r_i(x)$ denotes the distance between x and its i-th nearest neighbor, and $r_{max}(x)$ is the maximum distance from x among the k nearest neighbors.

3. Proposed Method

In this section, we introduce FedCorr, our proposed multi-stage training method to tackle heterogeneous label noise in FL systems (see Algorithm 1). Our method comprises three stages: pre-processing, finetuning and usual training. In the first stage, we sample the clients without replacement using a small fraction to identify noisy clients via LID scores and noisy samples via per-sample losses, after which we relabel the identified noisy samples with the predicted labels of the global model. The noise level of each client is also estimated in this stage. In the second stage, we finetune the model with a typical fraction on relatively clean clients, and use the finetuned model to further correct the samples for the remaining clients. Finally, in the last stage, we train the model via the usual FL method (FedAvg [24]) using the corrected labels at the end of the second stage.

3.1. Preliminaries

Consider an FL system with N clients and an M-class dataset $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^N$, where each $\mathcal{D}_k = \{(x_k^i, y_k^i)\}_{i=1}^{n_k}$ denotes the local dataset for client k. Let $\mathcal{S}$ denote the set of all N clients, and let $w_k^{(t)}$ (resp. $w^{(t)}$) denote the local model weights of client k (resp. global model weights obtained by aggregation) at the end of communication round t. At the end of round t, the global model $f_G^{(t)}$ would have its weights $w^{(t)}$ updated as follows:

$$w^{(t)} \leftarrow \sum_{k\in\mathcal{S}_t} \frac{|\mathcal{D}_k|}{\sum_{i\in\mathcal{S}_t}|\mathcal{D}_i|} w_k^{(t)}, \tag{3}$$

where $\mathcal{S}_t \subseteq \mathcal{S}$ is the subset of selected clients in round t.

For the rest of this subsection, we shall give details on client data partition, noise model simulation, and LID score computation. These are three major aspects of our proposed approach to emulate data heterogeneity, and to deal with the discrepancies in both local data statistics and label quality.

Data partition. We consider both IID and non-IID heterogeneous data partitions in this work. For IID partitions,

Algorithm 1 FedCorr (Red and Black line numbers in the pre-processing stage refer to operations for clients and server, respectively.)

Inputs: N (number of clients), T_1, T_2, T_3, $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{N}$ (dataset), $w^{(0)}$ (initialized global model weights).
Output: Global model f_G^{final}

```
    // Federated Pre-processing Stage
 1: $(\hat{\mu}_1^{(0)}, \dots, \hat{\mu}_N^{(0)}) \leftarrow (0, \dots, 0)$   // estimated noise levels
 2: for $t = 1$ to $T_1$ do
 3:     $\mathcal{S}$ =Shuffle($\{1, \dots, N\}$)
 4:     $w_{inter} \leftarrow w^{(t-1)}$   // intermediary weights
 5:     for $k \in \mathcal{S}$ do
 6:         $w_k^{(t)} \leftarrow$ weights that minimize loss function (5)
 7:         Upload weights $w_k^{(t)}$ and LID score to server
 8:     Update global model $w^{(t)} \leftarrow w_{inter}$
 9:     Divide all clients into clean set $\mathcal{S}_c$ and noisy set $\mathcal{S}_n$
        based on cumulative LID scores via GMM
10:     for noisy client $k \in \mathcal{S}_n$ do
11:         Divide $\mathcal{D}_k$ into clean subset $\mathcal{D}_k^c$ and noisy subset
            $\mathcal{D}_k^n$ based on per-sample losses via GMM
12:         $\hat{\mu}_k^{(t)} \leftarrow \frac{|\mathcal{D}_k^n|}{|\mathcal{D}_k|}$   // update estimated noise level
13:         $y_k^{(i)} \leftarrow \arg\max f(x_k^{(i)}; w^{(i)}), \forall (x_k^{(i)}, y_k^{(i)}) \in \mathcal{D}_k^n$
    // Federated Finetuning Stage
14: $\mathcal{S}_c \leftarrow \{k | k \in \mathcal{S}, \mu_k < 0.1\}$, $\mathcal{S}_n \leftarrow \mathcal{S} \setminus \mathcal{S}_c$.
15: for $t = T_1 + 1$ to $T_1 + T_2$ do
16:     Update $w_k^{(t)}$ by usual FedAvg among clients in $\mathcal{S}_c$
17: for Noisy client $k \in \mathcal{S}_n$ do
18:     $y_k^{(i)} \leftarrow \arg\max f(x_k^{(i)}; w^{(i)}), \forall (x_k^{(i)}, y_k^{(i)}) \in \mathcal{D}_k$
    // Usual Federated Learning Stage
19: for $t = T_1 + T_2 + 1$ to $T_1 + T_2 + T_3$ do
20:     Update $w_k^{(t)}$ by usual FedAvg among all clients
21: return $f_G^{\text{final}} := f(\cdot; w^{(T_1+T_2+T_3)})$
```

the whole dataset $\mathcal{D}$ is uniformly distributed at random among N clients. For non-IID partitions, we first generate an $N \times M$ indicator matrix Φ, where each entry Φ_{ij} indicates whether the local dataset of client i contains class j. Each Φ_{ij} shall be sampled from the Bernoulli distribution with a fixed probability p. For each $1 \leq j \leq M$, let v_j be the sum of entries in the j-th column of Φ; this equals the number of clients whose local datasets contain class j. Let $\boldsymbol{q}_j$ be a vector of length v_j, sampled from the symmetric Dirichlet distribution with the common parameter $\alpha_{Dir} > 0$. Using $\boldsymbol{q}_j$ as a probability vector, we then randomly allocate the samples within class j to these v_j clients. Note that our non-IID data partition method provides a general framework to control the variability in both class distribution and the sizes of local datasets (see Fig. 2).

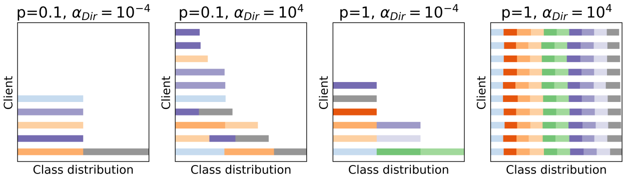

Figure 2. Depiction of non-IID partitions for different parameters.

Noise model. To emulate label noise in real-world data, we shall introduce a general federated noise model framework. For simplicity, this work only considers instance-independent label noise. This framework has two parameters ρ and τ, where ρ denotes the system noise level (ratio of noisy clients) and τ denotes the lower bound for the noise level of a noisy client. Every client has a probability ρ of being a noisy client, in which case the local noise level for this noisy client is determined randomly, by sampling from the uniform distribution $U(\tau, 1)$. Succinctly, the noise level of client k (for $k = 1, \dots, N$) is

$$\mu_k = \begin{cases} u \sim U(\tau, 1), & \text{with probability } \rho; \\ 0, & \text{with probability } 1 - \rho. \end{cases} \tag{4}$$

When $\mu_k \neq 0$, the $100 \cdot \mu_k\%$ noisy samples are chosen uniformly at random, and are assigned random labels, selected uniformly from the M classes.

LID scores for local models. In this paper, we associate LID scores to local models. Consider an arbitrary client with local dataset $\mathcal{D}$ and current local model $f(\cdot)$. Let $\mathcal{X} := \{f(x)\}_{x \in \mathcal{D}}$ be the set of prediction vectors, and for each $x \in \mathcal{D}$, compute $\widehat{LID}(f(x))$ w.r.t. the k nearest neighbors in $\mathcal{X}$, as given in (2). We define the *LID score* of $(\mathcal{D}, f)$ to be the average value of $\widehat{LID}(f(x))$ over all $x \in \mathcal{D}$. Note that as the local model $f(\cdot)$ gets updated with each round, the corresponding LID score will change accordingly.

Experiments have shown that given the same training process, models trained on a dataset with label noise tend to have larger LID scores as compared to models trained on the same dataset with clean labels [22, 23]. Intuitively, the prediction vectors of a well-trained model, trained on a clean dataset, would cluster around M possible one-hot vectors, corresponding to the M classes. However, as more label noise is added to the clean dataset, the prediction vector of a noisy sample would tend to be shifted towards the other clusters, with different noisy samples shifted in different directions. Hence, the prediction vectors near each one-hot vector would become "more diffuse" and would on average span a higher dimensional space.

3.2. Federated pre-processing stage

FedCorr begins with the pre-processing stage, which iteratively evaluates the quality of the dataset of each client, and relabels identified noisy samples. This pre-processing stage differs from traditional FL in the following aspects:

- All clients will participate in each iteration. Clients are selected without replacement, using a small fraction.
- An adaptive local proximal term is added to the loss function, and mixup data augmentation is used.
- Each client computes its LID score and per-sample cross-entropy loss after local training and sends its LID score together with local model updates to the server.

Client iteration and fraction scheduling. The pre-processing stage is divided into T_1 iterations. In each iteration, every client participates exactly once. Every iteration is organized by communication rounds, similar to the usual FL, but with two key differences: a small fraction is used, and clients are selected without replacement. Each iteration ends when all clients have participated.

It is known that large fractions could help improve the convergence rate [24], and a linear speedup could even be achieved in the case of convex loss functions [26]. However, large fractions have a weak effect in non-IID settings, while intuitively, small fractions would yield aggregated models that deviate less from local models; cf. [21]. These observations inspire us to propose a fraction scheduling scheme that combines the advantages of both small and large fractions. Specifically, we sample clients using a small fraction without replacement in the pre-processing stage, and use a typical larger fraction with replacement in the latter two stages. By sampling without replacement during pre-processing, we ensure all clients participate equally for the evaluation of the overall quality of labels in local datasets.

Mixup and local proximal regularization. Throughout the pre-processing stage, for client k with batch $(X_b, Y_b) = \{(x_i, x_j)\}_{i=1}^{n_b}$ (where n_b denotes batch size), we use the following loss function:

$$L(X_b) = L_{CE}\left(f_k^{(t)}(\tilde{X}_b), \tilde{Y}_b\right) + \beta \hat{\mu}_k^{(t-1)} \left\| w_k^{(t)} - w^{(t-1)} \right\|^2. \quad (5)$$

Here, $f_k^{(t)} = f(\cdot; w_k^{(t)})$ denotes the local model of client k in round t, and $w^{(t-1)}$ denotes the weights of the global model obtained in the previous round $t-1$. The first term in (5) represents the cross-entropy loss on the mixup augmentation of (X_b, Y_b), while the second term in (5) is an adaptive local proximal regularization term, where $\hat{\mu}_k^{(t-1)}$ is the estimated noise level of client k to be defined later. It should be noted that our local proximal regularization term is only applied in the pre-processing stage.

Recall that mixup [38] is a data augmentation technique that favors linear relations between samples, and that has been shown to exhibit strong robustness to label noise [3,16]. Mixup generates new samples $(\tilde{x}, \tilde{y})$ as convex combinations of randomly selected pairs of samples (x_i, y_i) and (x_j, y_j), given by $\tilde{x} = \lambda x_i + (1-\lambda) x_j$, $\tilde{y} = \lambda y_i + (1-\lambda) y_j$, where $\lambda \sim \text{Beta}(\alpha, \alpha)$, and $\alpha \in (0, \infty)$. (We use $\alpha = 1$ in our experiments.) Intuitively, mixup achieves robustness to label noise due to random interpolation. For example, if $(x_i, \hat{y}_i)$ is a noisy sample and if y_i is the true label, then the negative impact caused by an incorrect label $\hat{y}_i$ is alleviated when paired with a sample whose label is y_i.

Our adaptive local proximal regularization term is scaled by $\hat{\mu}_k^{(t-1)}$, which is the estimated noise level of client k computed at the end of round $t-1$. (In particular, this term would vanish for clean clients.) The hyperparameter β is also incorporated to control the overall effect of this term. Intuitively, if a client's dataset has a larger discrepancy from other local datasets, then the corresponding local model would deviate more from the global model, thereby contributing a larger loss value for the local proximal term.

Identification of noisy clients and noisy samples. To address the challenge of heterogeneous label noise, we shall iteratively identify and relabel the noisy samples. In each iteration of this pre-processing stage, where all clients will participate, every client will compute the LID score and per-sample loss for its current local model (see Algorithm 1, lines 3-9). Specifically, when client k is selected in round t, we train the model $f_k^{(t)}$ on the local dataset $\mathcal{D}_k$ and then compute the LID score of $(\mathcal{D}_k, f_k^{(t)})$ via (2). Note that our proposed framework preserves the privacy of client data, since in comparison to the usual FL, there is only an additional LID score sent to the server, which is a single scalar that reflects only the predictive discriminability of the local model. Since the LID score is computed from the predictions of the output layer (of the local model), knowing this LID score does not reveal information about the raw input data. This additional LID score is a single scalar, hence it has a negligible effect on communication cost.

At the end of iteration t, we shall perform the following three steps:

1. The server first computes a Gaussian Mixture Model (GMM) on the cumulative LID scores of all N clients. Using this GMM, the set of clients $\mathcal{S}$ is partitioned into two subsets: $\mathcal{S}_n$ (noisy clients) and $\mathcal{S}_c$ (clean clients).
2. Each noisy client $k \in \mathcal{S}_n$ locally computes a new GMM on the per-sample loss values for all samples in the local dataset $\mathcal{D}_k$. Using this GMM, $\mathcal{D}_k$ is partitioned into two subsets: a clean subset $\mathcal{D}_k^c$, and a noisy subset $\mathcal{D}_k^n$. We observe that the large-loss samples are more likely to have noisy labels. The local noise level of client k can then be estimated by $\hat{\mu}_k^{(t)} = |\mathcal{D}_k^n| / |\mathcal{D}_k|$ if $k \in \mathcal{S}_n$, and $\hat{\mu}_k^{(t)} = 0$ otherwise.
3. Each noisy client $k \in \mathcal{S}_n$ performs relabeling of the noisy samples by using the predicted labels of the global model as the new labels. In order to avoid overcorrection, we only relabel those samples that are identified to be noisy with high confidence. This partial relabeling is controlled by a relabel ratio π and a confidence threshold θ. Take noisy client k for example: We first choose samples from $\mathcal{D}_k^n$ that corresponds to the top-$\pi \cdot |\mathcal{D}_k^n|$ largest per-sample cross-entropy losses.

Next, we obtain the prediction vectors of the global model, and relabel a sample only when the maximum entry of its prediction vector exceeds θ. Thus, the subset $\widetilde{\mathcal{D}_k^n}'$ of samples to be relabeled is given by

$$\widetilde{\mathcal{D}_k^n} = \operatorname*{arg\,max}_{\substack{\tilde{\mathcal{D}}\subseteq\mathcal{D}_k^n\\ |\tilde{\mathcal{D}}|=\pi\cdot|\mathcal{D}_k^n|}} L_{CE}(\tilde{\mathcal{D}}; f_G^{(t)}); \tag{6}$$

$$\widetilde{\mathcal{D}_k^n}' = \{(x,y)\in\widetilde{\mathcal{D}_k^n} \,|\, \max(f_G^{(t)}(x)) \geq \theta\}; \tag{7}$$

where $f_G^{(t)}$ is the global model at the end of iteration t.

Why do we use cumulative LID scores in step 1?
In deep learning, it has been empirically shown that when training on a dataset with label noise, the evolution of the representation space of the model exhibits two distinct phases: (1) an early phase of dimensionality compression, where the model tends to learn the underlying true data distribution, and (2) a later phase of dimensionality expansion, where the model overfits to noisy labels [23].

We observed that clients with larger noise levels tend to have larger LID scores. Also, the overlap of LID scores between clean and noisy clients would increase during training. This increase could be due to two reasons: (1) the model may gradually overfit to noisy labels, and (2) we correct the identified noisy samples after each iteration, thereby making the clients with low noise levels less distinguishable from clean clients. Hence, the cumulative LID score (i.e., the sum of LID scores in all past iterations) is a better metric for distinguishing noisy clients from clean clients; see the top two plots in Fig. 3 for a comparison of using LID score versus cumulative LID score. Furthermore, the bottom two plots in Fig. 3 show that cumulative LID score has a stronger linear relation with local noise level.

3.3. Federated finetuning stage

We aim to finetune the global model f_G on relatively clean clients over T_2 rounds and further relabel the remaining noisy clients. The aggregation at the end of round t is given by the same equation (3), with one key difference: $\mathcal{S}_t$ is now a subset of $\mathcal{S}_c = \{k|1 \leq k \leq N, \hat{\mu}_k^{(T_1)} \leq \kappa\}$, where κ is the threshold used to select relatively clean clients based on the estimated local noise levels $\hat{\mu}_1^{(T_1)}, ..., \hat{\mu}_N^{(T_1)}$.

At the end of the finetuning stage, we relabel the remaining noisy clients $\mathcal{S}_n = \mathcal{S} \setminus \mathcal{S}_c$ with the predicted labels of f_G. Similar to the correction process in the pre-processing stage, we use the same confidence threshold θ to control the subset of samples to be relabeled; see (7).

3.4. Federated usual training stage

In this final stage, we train the global model over T_3 rounds via the usual FL (FedAvg) on all the clients, using the labels corrected in the previous two training stages. We also incorporate this usual training stage with three FL methods to show that methods based on different techniques can be well-incorporated with FedCorr, even if they are not designed specifically for robust FL; see Sec. 4.2.

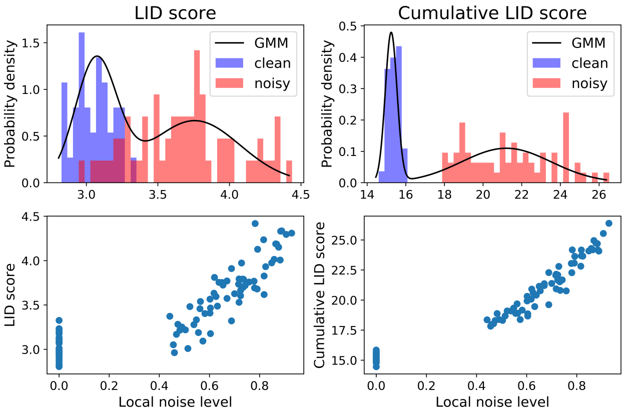

Figure 3. Empirical evaluation of LID score (left) and cumulative LID score (right) after 5 iterations on CIFAR-10 with noise model $(\rho, \tau) = (0.6, 0.5)$, and with IID data partition, over 100 clients. Top: probability density function and estimated GMM; bottom: LID/cumulative LID score vs. local noise level for each client.

Dataset	CIFAR-10	CIFAR-100	Clothing1M
Size of $\mathcal{D}_{train}$	50,000	50,000	1,000,000
# of classes	10	100	14
# of clients	100	50	500
Fraction γ	0.1	0.1	0.02
Architecture	ResNet-18	ResNet-34	pre-trained ResNet-50

Table 1. List of datasets used in our experiments.

4. Experiments

In this section, we conduct experiments in both IID (CIFAR-10/100 [15]) and non-IID (CIFAR-10, Clothing1M [31]) data settings, at multiple noise levels, to show that FedCorr is simultaneously robust to both local label quality discrepancy and data statistics discrepancy. To demonstrate the versatility of FedCorr, we also show that various FL methods can have their performances further improved by incorporating the first two stages of FedCorr. We also conduct an ablation study to show the effects of different components of FedCorr. Details on data partition and the noise model used have already been given in Sec. 3.1.

4.1. Experimental Setup

Baselines. There are two groups of experiments.

In the first group, we demonstrate that FedCorr is robust to discrepancies in both data statistics and label quality. We compare FedCorr with the following state-of-the-art methods from three categories: (1) methods to tackle label noise in CL (JointOpt [27] and DivideMix [16]) applied to local clients; (2) classic FL methods (FedAvg [24]

Setting	Method	Best Test Accuracy (%) ± Standard Deviation (%)						
		$\rho = 0.0$	$\rho = 0.4$		$\rho = 0.6$		$\rho = 0.8$	
		$\tau = 0.0$	$\tau = 0.0$	$\tau = 0.5$	$\tau = 0.0$	$\tau = 0.5$	$\tau = 0.0$	$\tau = 0.5$
Centralized (for reference)	JointOpt	93.73±0.21	92.29±0.37	92.11±0.21	91.26±0.46	88.42±0.33	89.18±0.29	85.62±1.17
	DivideMix	95.64±0.05	96.39±0.09	96.17±0.05	96.07±0.06	94.59±0.09	94.21±0.27	94.36±0.16
Federated	FedAvg	93.11±0.12	89.46±0.39	88.31±0.80	86.09±0.50	81.22±1.72	82.91±1.35	72.00±2.76
	FedProx	92.28±0.14	88.54±0.33	88.20±0.63	85.80±0.41	85.25±1.02	84.17±0.77	80.59±1.49
	RoFL	88.33±0.07	88.25±0.33	87.20±0.26	87.77±0.83	83.40±1.20	87.08±0.65	74.13±3.90
	ARFL	92.76±0.08	85.87±1.85	83.14±3.45	76.77±1.90	64.31±3.73	73.22±1.48	53.23±1.67
	JointOpt	88.16±0.18	84.42±0.70	83.01±0.88	80.82±1.19	74.09±1.43	76.13±1.15	66.16±1.71
	DivideMix	77.96±0.15	77.35±0.20	74.40±2.69	72.67±3.39	72.83±0.30	68.66±0.51	68.04±1.38
	Ours	**93.82±0.41**	**94.01±0.22**	**94.15±0.18**	**92.93±0.25**	**92.50±0.28**	**91.52±0.50**	**90.59±0.70**

Table 2. Average (5 trials) and standard deviation of the best test accuracies of various methods on CIFAR-10 with IID setting at different noise levels (ρ: ratio of noisy clients, τ: lower bound of client noise level). The highest accuracy for each noise level is boldfaced.

Method	Best Test Accuracy (%) ± Standard Deviation(%)			
	$\rho = 0.0$ $\tau = 0.0$	$\rho = 0.4$ $\tau = 0.5$	$\rho = 0.6$ $\tau = 0.5$	$\rho = 0.8$ $\tau = 0.5$
JointOpt (CL)	72.94±0.43	65.87±1.50	60.55±0.64	59.79±2.45
DivideMix (CL)	75.58±0.14	75.43±0.34	72.26±0.58	71.02±0.65
FedAvg	72.41±0.18	64.41±1.79	53.51±2.85	44.45±2.86
FedProx	71.93±0.13	65.09±1.46	57.51±2.01	51.24±1.60
RoFL	67.89±0.65	59.42±2.69	46.24±3.59	36.65±3.36
ARFL	72.05±0.28	51.53±4.38	33.03±1.81	27.47±1.08
JointOpt	67.49±0.36	58.43±1.88	44.54±2.87	35.25±3.02
DivideMix	45.91±0.27	43.25±1.01	40.72±1.41	38.91±1.25
Ours	**72.56±2.07**	**74.43±0.72**	**66.78±4.65**	**59.10±5.12**

Table 3. Average (5 trials) and standard deviation of the best test accuracies on CIFAR-100 with IID setting.

Method\(p, α_{Dir})	$(0.7, 10)$	$(0.7, 1)$	$(0.3, 10)$
FedAvg	78.88±2.34	75.98±2.92	67.75±4.38
FedProx	83.32±0.98	80.40±0.94	73.86±2.41
RoFL	79.56±1.39	72.75±2.21	60.72±3.23
ARFL	60.19±3.33	55.86±3.30	45.78±2.84
JointOpt	72.19±1.59	66.92±1.89	58.08±2.18
DivideMix	65.70±0.35	61.68±0.56	56.67±1.73
Ours	**90.52±0.89**	**88.03±1.08**	**81.57±3.68**

Table 4. Average (5 trials) and standard deviation of the best test accuracies of different methods on CIFAR-10 with different non-IID setting. The noise level is $(\rho, \tau) = (0.6, 0.5)$.

Settings	FedAvg	FedProx	RoFL	ARFL	JointOpt	Dividemix	Ours
FL	70.49	71.35	70.39	70.91	71.78	68.83	**72.55**
CL	-	-	-	-	72.23	74.76	-

Table 5. Best test accuracies on Clothing1M with non-IID setting. CL results are the accuracies reported in corresponding papers.

and FedProx [19]); and (3) FL methods designed to be robust to label noise (RoFL [35] and ARFL [7]). For reference, we also report experimental results on JointOpt and DivideMix in CL, so as to show the performance reduction of these two methods when used in FL.

In the second group, we demonstrate the versatility of FedCorr. We examine the performance improvements of three state-of-the-art methods when the first two stages of FedCorr are incorporated. These methods are chosen from three different aspects to improve FL: local optimization (FedDyn [1]), aggregation (Median [36]) and client selection (PoC [6]).

Implementation details. We choose different models and number of clients N for each dataset; see Tab. 1. For data pre-processing, we perform normalization and image augmentation using random horizontal flipping and random cropping with padding=4. We use an SGD local optimizer with a momentum of 0.5, with a batch size of 10 for CIFAR-10/100 and 16 for Clothing1M. With the exception of JointOpt and DivideMix used in FL settings, we shall always use 5 local epochs across all experiments. For FedCorr, we always use the same hyperparameters on the same dataset. In particular, we use $T_1 = 5, 10, 2$ for CIFAR-10, CIFAR-100, Clothing1M, respectively. For fraction scheduling, we use the fraction $\gamma = \frac{1}{N}$ in the pre-processing stage, and we use the fractions specified in Tab. 1 for the latter two stages. Further implementation details can be found in the supplementary material.

4.2. Comparison with state-of-the-art methods

IID settings. We compare FedCorr with multiple baselines at different noise levels, using the same configuration. Tab. 2 and Tab. 3 show the results on CIFAR-10 and CIFAR-100, respectively. In summary, FedCorr achieves best test accuracies across all noise settings tested on both datasets, with particularly significant outperformance in the case of high noise levels. Note that we have implemented

Method	Best Test Accuracy (%) ± Standard Deviation (%)						
	$\rho = 0.0$	$\rho = 0.4$		$\rho = 0.6$		$\rho = 0.8$	
	$\tau = 0.0$	$\tau = 0.0$	$\tau = 0.5$	$\tau = 0.0$	$\tau = 0.5$	$\tau = 0.0$	$\tau = 0.5$
Ours	**93.82±0.41**	**94.01±0.22**	**94.15±0.18**	**92.93±0.25**	**92.50±0.28**	**91.52±0.50**	**90.59±0.70**
Ours w/o correction	92.85±0.66	93.71±0.20	93.60±0.21	92.15±0.29	91.77±0.65	90.48±0.56	88.77±1.10
Ours w/o frac. scheduling	86.05±1.47	85.59±1.10	78.44±7.90	80.29±2.62	77.96±3.65	76.67±3.48	72.71±5.03
Ours w/o local proximal	93.37±0.05	93.64±0.15	93.46±0.17	92.34±0.14	91.74±0.47	90.45±0.94	88.74±1.72
Ours w/o finetuning	92.71±0.18	93.06±0.15	92.62±0.28	91.41±0.14	89.31±0.90	89.62±0.40	83.81±2.59
Ours w/o usual training	93.11±0.10	93.53±0.17	93.46±0.14	92.16±0.24	91.50±0.51	90.62±0.59	88.97±1.37
Ours w/o mixup	90.63±0.70	88.83±1.88	91.34±0.39	87.79±0.89	87.50±1.33	87.86±0.53	83.29±1.78

Table 6. Ablation study results (average and standard deviation of 5 trials) on CIFAR-10.

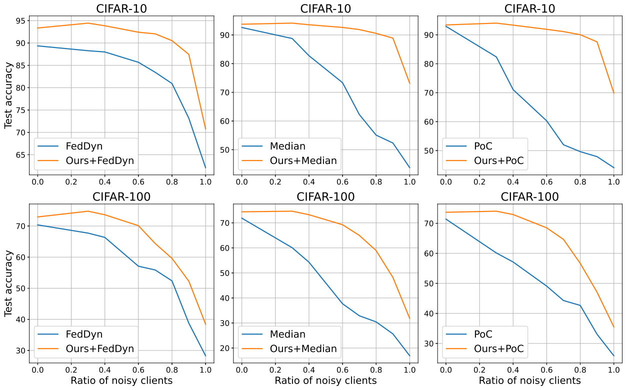

Figure 4. Best test accuracies of three FL methods combined with FedCorr on CIFAR-10/100 with multiple ρ and fixed $\tau = 0.5$.

JointOpt and DivideMix in both centralized and federated settings to show the performance reduction ($10\% \sim 30\%$ lower for best accuracy) when these CL methods are applied to local clients in FL. Furthermore, the accuracies in CL can also be regarded as upper bounds for the accuracies in FL. Remarkably, the accuracy gap between DivideMix in CL and FedCorr in FL is $< 4\%$ even in the extreme noise setting $(\rho, \tau) = (0.8, 0.5)$. In the centralized setting, we use the dataset corrupted with exactly the same scheme as in the federated setting. For the federated setting, we warm up the global model for 20 rounds with FedAvg to avoid introducing additional label noise during the correction process in the early training stage, and we then apply JointOpt or DivideMix locally on each selected client, using 20 local training epochs.

Non-IID settings. To evaluate FedCorr in more realistic heterogeneous data settings, we conduct experiments using the non-IID settings as described in Sec. 3.1, over different values for (p, α_{Dir}). Tab. 4 and Tab. 5 show the results on CIFAR-10 and Clothing1M, respectively. Note that we do not add synthetic label noise to Clothing1M, since it already contains real-world label noise. For CIFAR-10, FedCorr consistently outperforms all baselines by at least 7%. For Clothing1M, FedCorr also achieves the highest accuracy in FL, and this accuracy is even higher than the reported accuracy of JointOpt in CL.

Combination with other FL methods. We also investigate the performance of three state-of-the-art methods, when the first two stages of FedCorr are incorporated. As shown in Fig. 4, we consistently obtain significant accuracy improvements on CIFAR-10/100 for various ratios of noisy clients.

4.3. Ablation study

Tab. 6 gives an overview of the effects of the components in FedCorr. Below, we consolidate some insights into what makes FedCorr successful:

- All components help to improve accuracy.
- Fraction scheduling has the largest effect. The small fraction used in the pre-processing stage helps to capture local data characteristics, as it avoids information loss brought by aggregation over multiple models.
- The highest accuracy among different noise levels is primarily achieved at a low noise level (e.g. $\rho = 0.4$) and not at the zero noise level, since additional label noise could be introduced during label correction.

5. Conclusion

We present FedCorr, a general FL framework that jointly tackles the discrepancies in both local label quality and data statistics, and that performs privacy-preserving label correction for identified noisy clients. Our experiments demonstrate the robustness and outperformance of FedCorr at multiple noise levels and diverse data settings.

In its current formulation, FedCorr does not consider dynamic participation in FL, whereby clients can join or leave training at any time. New clients joining much later would always have relatively lower cumulative LID scores, which means new noisy clients could be categorized incorrectly as clean clients. Thus, further work is required to handle dynamic participation.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020. 2, 7

[2] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Ken-ichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2015. 3

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019. 1, 5

[4] Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Biao Chen, and Zhiqi Shen. Focus: Dealing with label quality disparity in federated learning. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with IJCAI(FL-IJCAI'20)*, 2020. 1, 3

[5] Zihan Chen, Kai Fong Ernest Chong, and Tony QS Quek. Dynamic attention-based communication-efficient federated learning. In *International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI (FTL-IJCAI'2021)*, 2021. 1

[6] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020. 2, 7

[7] Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. In *AAAI Workshop Towards Robust, Secure and Efficient Machine Learning*, 2021. 1, 2, 3, 7

[8] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020. 1

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 1

[10] Michael E Houle. Dimensionality, discriminability, density and distance distributions. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 468–473. IEEE, 2013. 2, 3

[11] Michael E Houle. Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications. In *International Conference on Similarity Search and Applications*, pages 64–79. Springer, 2017. 3

[12] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 1, 2

[13] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 2021. 3

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[16] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 1, 5, 6

[17] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 2, 3

[18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. 1, 2, 7

[20] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019. 1

[21] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019. 5

[22] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. 2, 4

[23] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018. 2, 4, 6

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 5, 6

[25] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. 1, 2

[26] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018. 5

[27] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning

with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018. 1, 6

[28] Ching Pui Wan and Qifeng Chen. Robust federated learning with attack-adaptive aggregation. In *International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI (FTL-IJCAI'2021)*, 2021. 1, 2, 3

[29] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 2020. 2

[30] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *NeurIPS*, 33, 2020. 1

[31] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 6

[32] Jingyi Xu, Tony QS Quek, and Kai Fong Ernest Chong. Training classifiers that are universally robust to all label noise levels. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 1

[33] Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML(FL-ICML'21)*, 2021. 1, 3

[34] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019. 1

[35] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. *IEEE Intelligent Systems*, 2022. 1, 3, 7

[36] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659. PMLR, 10–15 Jul 2018. 7

[37] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pages 7164–7173. PMLR, 2019. 1

[38] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 5