Distribution-Regularized Federated Learning on Non-IID Data

Yansheng Wang[1], Yongxin Tong[1], Zimu Zhou[2], Ruisheng Zhang[1], Sinno Jialin Pan[3], Lixin Fan[4], Qiang Yang[4,5]
[1] SKLSDE Lab, School of Computer Science and Engineering, Beihang University, China
[2]City University of Hong Kong, Hong Kong, China
[3]The Chinese University of Hong Kong, Hong Kong, China
[4]AI Group, WeBank Co., Ltd., China
[5]Hong Kong University of Science and Technology, Hong Kong, China
[1]{arthur_wang, yxtong, rszhang}@buaa.edu.cn, [2]zimuzhou@cityu.edu.hk,[3]sinnopan@cuhk.edu.hk,
[4] lixinfan@webank.com, [5]qyang@cse.ust.hk

***Abstract*—Federated learning (FL) has emerged as a popular machine learning paradigm recently. Compared with traditional distributed learning, its unique challenges mainly lie in communication efficiency and non-IID (heterogeneous data) problem. While the widely adopted framework FedAvg can reduce communication overhead significantly, its effectiveness on non-IID data still lacks exploration. In this paper, we study the non-IID problem of FL from the perspective of domain adaptation. We propose a distribution regularization for FL on non-IID data such that the discrepancy of data distributions between clients is reduced. To further reduce the communication cost, we devise two novel distributed learning algorithms, namely rFedAvg and rFedAvg+, for efficiently learning with the distribution regularization. More importantly, we theoretically establish their convergence for strongly convex objectives. Extensive experiments on 4 datasets with both CNN and LSTM as learning models verify the effectiveness and efficiency of the proposed algorithms.**

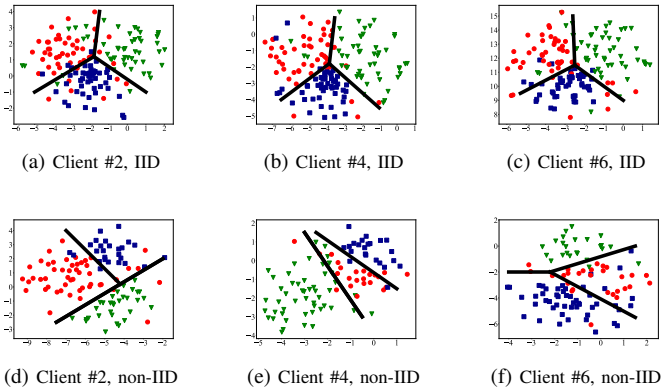
I. Introduction

Federated learning (FL) [1]–[3] is a new distributed machine learning paradigm that collaboratively trains models among multiple clients while the raw training samples possessed by each client cannot be shared. Federated learning has a wide range of real-world applications. For example, a large number of smartphone users can jointly train accurate next-word prediction models (a.k.a cross-device FL) [4], whereas enterprises or hospitals that do not have enough data for learning can cooperate to train federated models under privacy regulations (a.k.a cross-silo FL) [3].

Compared with traditional distributed learning, federated learning faces unique technical challenges. In federated learning, the data samples held by each client may not come from the same distribution (not independent and identically distributed, or *non-IID*). Also, the bandwidth and the number of interactions from the clients to the central server can be limited. These problems can severely deteriorate the effectiveness and efficiency of distributed stochastic gradient descent (SGD) algorithms, the mainstream solutions for distributed machine learning.

FedAvg [1] is a well-known framework for communication-efficient federated learning. It implements distributed SGD with client sampling and local training, where a subset of clients run multiple steps of local gradient descent and the server aggregates the local models by taking the weighted average of their model parameters. Although FedAvg shows success in improving communication efficiency of FL, it can be ineffective on non-IID data [5]–[8]. This is because the simple averaging of highly divergent models trained on non-IID data may lead to significant utility loss. Multiple studies [6], [8]–[10] have proposed remedies to improve the effectiveness of FedAvg on non-IID data. However, they mostly halt at theoretical analysis under strong assumptions [11], [12], and some [6], [7] even achieve lower accuracy than vanilla FedAvg on certain benchmarks (see Sec. VI).

In this work, we explore federated learning on non-IID data in the lens of *domain adaptation*. Fig. 1 visualizes the features of the last fully connected (FC) layer generated by FedAvg from 3 clients on both IID and non-IID division of CIFAR10 [13]. We observe that the feature distributions from



(a) Client #2, IID (b) Client #4, IID (c) Client #6, IID

(d) Client #2, non-IID (e) Client #4, non-IID (f) Client #6, non-IID

Fig. 1: Feature visualization of FedAvg on CIFAR10 with t-SNE. The features are 512 dimension output vectors of the last FC layer from training data. The circles, triangles and squares refer to label class 0, 1 and 2 in CIFAR10. (a), (b) and (c) shows the features of 3 clients in IID partition. (d), (e) and (f) are based on non-IID partition where the majority of data in client #2, #4 and #6 comes from class 0, 1 and 2 respectively. The black lines represent possible classifiers.

different clients are consistent on IID data, which can produce consistent models (the black lines in Fig. 1a, Fig. 1b, Fig. 1c) and thus the averaging of model is effective. However, on non-IID data, the feature distributions differ from each other so the local classification models may have discrepancies (the black lines in Fig. 1d, Fig. 1e, Fig. 1f). Simple averaging of them causes confusion and drop in classification accuracy. Note that the *non-IID data problem in federated learning* resembles the *distribution shift problem in domain adaptation*, where the distributions between the source domain and target domain are different. Inspired by the solutions in domain adaptation to deal with distribution shifts between the source and the target domain, we propose a *distribution regularization* for federated learning on non-IID data such that the discrepancy of data distributions between clients is reduced. The rationale is that to minimize the discrepancy in data distributions between any two clients so that their local models tend to have consistent feature representations.

However, it is infeasible to directly apply FedAvg to federated learning with distribution regularization. This is because the regularizer measures the pairwise distances between clients, whose calculation requires communication between clients in every round of gradient descent. This will break the local training steps in FedAvg, and also bring high communication cost. Accordingly, we devise new distributed optimization algorithms based on FedAvg that can efficiently approximate the distribution regularizer and we also provide theoretical analysis to support its convergence. The main contributions of this work are summarized as follows.

- We propose the first distribution regularizer to explicitly account for the non-IID problem in federated learning.
- We devise two new communication-efficient learning algorithms (rFedAvg and rFedAvg+) for federated learning with distribution regularization. More importantly, we theoretically establish their convergence for strongly convex objectives.
- Evaluations on 4 benchmarks (MNIST [14], CIFAR10 [13], Sent140 [15] and FEMNIST [16]) with different models (CNN and LSTM) show that our proposed algorithms outperform the state-of-the-arts [6]–[8] in terms of communication rounds and test accuracy on non-IID data.

II. Related Work

Federated learning [1], [2] was first proposed for communication-efficient collaborative learning among android users and has many applications on image/text data [17], [18] and spatial data [19]–[24]. The widely recognized core issues in FL include communication efficiency, learning on non-IID data, privacy & security, etc. [3], [25], [26]. In this paper, we focus on *communication-efficient* federated learning algorithms on *non-IID* data, where our idea is inspired by techniques from *domain adaptation*.

Communication Efficiency in Federated Learning. Early research on FL optimizes the communication efficiency in a distributed learning setting. The most recognized framework is FedAvg [1], which applies client sampling and local training to reduce the communication overhead. Follow-ups such as [2] introduce more compression-based strategies like quantization, random rotations and sub-sampling. Li *et al.* [27] provide an interesting insight to improve the efficiency and reduce the communication cost. Synopses techniques like sketch are also popular in compressing gradients in FL [28], [29].

Our proposed learning algorithms are built upon FedAvg, *i.e.* applying local training steps to reduce communication rounds. Nevertheless, our algorithms are tailored for optimizing the federated learning objective with distribution regularizer, such that they outperform FedAvg in case of non-IID data.

Non-IID in Federated Learning. The non-IID problem is common in FL, since the samples held by different clients may be collected in different context *e.g.* environments and devices. The naive FedAvg easily suffers from the non-IID problem [5]. Many studies [6], [8]–[10], [12], [30], [31] have explored to deal with the non-IID data problem by modifying FedAvg in various ways. Some rectify the distribution shift in clients by using variance reduction [8] or adding proximal terms [6] in the local updating process. Others change the global objective function by considering the fairness in heterogeneous networks [7] or assuming a structured affine distribution shift in clients' data [12]. Additional methods include adaptive sampling of clients [9], [10] and new aggregation strategies at the server [9], [30], [31]. An experimental study of these methods in cross-silo setting is made in [32].

In our work, we follow the methods on changing the global objective function [7], [12] and explicitly account for the non-IID problem by adding a distribution regularizer. Unlike [7] which aims to improve the performance of the worst clients while preserving similar overall performance of FedAvg, and [12] which introduces worst disturbance to make the learned overall model more robust, this approach can improve the overall performance and performance of worst nodes together with the distribution regularization and holds the potential to generalize to more than supervised learning tasks. We further devise two communication-efficient algorithms to cope with this new regularizer and empirically show that our method outperforms the state-of-the-arts [6]–[8].

Domain Adaptation. Domain Adaptation (DA) [33] aims to transfer the knowledge learned from a source domain to a target domain with different data distributions. The key idea is to reduce distribution discrepancy by finding domain-invariant structures [34]. Prior solutions [35]–[37] mainly focus on minimizing a distance metric of domain discrepancy like the maximum mean discrepancy distance. More recent proposals [38], [39] use deep neural networks to learn transferable features, taking the minimization of domain discrepancy as an adaptation regularizer in the empirical risk minimization problem. Adversarial learning has also emerged as an effective solution to DA [40], [41].

Domain adaptation in the federated learning setting was first considered in [42]. It aims to generalize the model to

new target devices in different domains while the distributed data sources are still assumed to be IID. In this work, we extend the idea of domain adaptation by taking all the clients in FL as both the sources and targets to minimize the overall distribution discrepancy. We further devise distributed learning algorithms to efficiently calculate the regularizer and provide convergence analysis.

III. Federated Learning with Distribution Regularization

In this section, we first introduce the standard federated learning setting (Sec. III-A) and then present our distribution regularization dedicated for federated learning on non-IID data (Sec. III-B).

A. Standard Federated Learning

We consider the following general federated learning model with N clients [1], [2], [6]–[8], [11]:

$$min_w \left\{ F(w) \ , \ \sum_{k=1}^{N} p_k F_k(w) \right\}. \tag{1}$$

where p_k is the weight of client k with $\sum_{k=1}^{N} p_k = 1$ and $F_k(w)$ is the local objective for client k with model parameter w. For general supervised learning, the local objective is to minimize the empirical risk, *i.e.* $F_k(w) = \sum_{j=1}^{n_k} l(w, x_{k,j})$, where n_k is the number of samples held by client k. Without loss of generality, we assume the loss function of each client is the same. Let the samples owned by client k be $(\boldsymbol{x_k}, \boldsymbol{y_k}) = ((x_{k,1}, y_{k,1}), (x_{k,2}, y_{k,2}), \cdots, (x_{k,n_k}, y_{k,n_k}))$.

In federated learning, the data partitions $\{(\boldsymbol{x_k}, \boldsymbol{y_k})\}$ cannot be shared among parties. Instead, only intermediate results are communicated to a central server for optimization. In addition, the samples from different parties can be non-independent and identical distributed (non-IID). Therefore, a good federated learning algorithm should optimize the objective in Eq. (1) with *minimal communication cost* and work in case of *non-IID data*.

Our solution is built upon FedAvg [1], a popular communication-efficient framework for federated learning. However, the FedAvg framework does not address the non-IID data problem. We explicitly account for learning on non-IID data by introducing a *distribution regularization*, as explained below.

B. Distribution Regularization

We assume that the samples are from the same distribution for each client but the distributions vary across clients [8], [11], [12]. This is reasonable because data generated from the same client usually undergo the same physical- and device-dependent context, whereas such context varies across clients. However, we do not restrict the difference in distributions as affine shifts as in [12].

For effective learning on differently distorted data distributions, we propose to project these distributions to a common space where the distances among the projected distributions are minimized. The idea has been widely adopted in domain adaptation [35], [36], [38], [39] but we are the first work to apply it for the non-IID problem in federated learning. Specifically, for two clients i and j with different marginal distributions $P(\boldsymbol{x_i}) \neq P(\boldsymbol{x_j})$, we aim to find a mapping $\phi(\cdot)$ which projects the two marginal distributions to a reproducing Kernel Hilbert Space (RKHS) such that $P(\phi(\boldsymbol{x_i})) \approx P(\phi(\boldsymbol{x_j}))$. As a proof-of-concept, we adopt the widely used empirical estimate of maximum mean discrepancy (MMD) [35] as the distance between the data distributions of clients i and j:

$$MMD(\boldsymbol{x_i}, \boldsymbol{x_j}) = \left\| \frac{1}{n_i} \sum_{k=1}^{n_i} \phi(x_{i,k}) - \frac{1}{n_j} \sum_{k=1}^{n_j} \phi(x_{j,k}) \right\|. \tag{2}$$

In the empirical study of this work, we use a deep neural network to approximate ϕ. Accordingly, we can reformulate the standard federated learning model Eq. (1) by adding a new local objective that explicitly captures the pairwise data distribution discrepancy between clients.

$$min_w \left\{ F(w) = \sum_{k=1}^{N} p_k (f_k(w) + \lambda r_k(w)) \right\}. \tag{3}$$

where

$$f_k(w) = \sum_{j=1}^{n_k} l(w, x_{k,j}). \tag{4}$$

and

$$r_k(w) = \frac{1}{N-1} \sum_{j \neq k} d^2 \left(\phi(x_k; \varpi,), \phi(x_j; \varpi)\right). \tag{5}$$

Here $d(\cdot, \cdot)$ is the MMD distance between two clients, ϖ is parameter of $\phi(\cdot)$ which denotes the parameters of $f_k(w)$ except for the output layer and λ is the weight coefficient which also works as the normalization factor of r_k.

We note that optimizing Eq. (3) demands new federated learning algorithms due to the following two reasons. First, naive adoption of the generic FedAvg framework [1] can lead to high communication cost due to the distribution regularization term. Also, there is no guarantee on convergence when optimizing Eq. (3). As next, we propose new learning algorithms for federated learning with distribution regularization that *enjoy low communication cost* (see Sec. IV) and *guarantees to converge* (see Sec. V).

IV. Learning Algorithms

The optimization objective in Eq. (3) consists of two parts: the standard federated learning objective and the distribution regularization term. The standard federated learning objective can be optimized by communication-efficient algorithms such as FedAvg [1]. However, exact calculation of the regularization term requires extra communication between every pair of clients to compute the pairwise MMD distances, which incurs at least $O(N^2)$ of communication overhead in a single round.

We propose to reduce the communication cost by calculating the regularization term *approximately*. The idea is to use *delayed* distribution mapping $\phi(\cdot)$ rather than the *up-to-date* one in computing the regularization. We first review the

FedAvg framework (Sec. IV-A) for the standard federated learning objective and then present two new communication-efficient algorithms (Sec. IV-B and Sec. IV-C) for federated learning with distribution regularization.

A. Preliminaries: FedAvg

FedAvg [1] is a recognized framework for communication-efficient optimization on the standard federated learning objective in Eq. (1). It is based on synchronous distributed large-batch SGD, which has two main steps: (1) *local updating* by clients and (2) *global aggregating* by the server. In each round t, a subset of clients is sampled with sample ratio SR among which client k will perform local mini-batch SGD with learning rate η_t: $w_{t+1}^k \leftarrow w_t^k - \eta_t \cdot \nabla F_k$ with batch size B for E steps. Afterwards, the central server aggregates the local models by taking a weighted average of them, *i.e.*, $w_{t+1} = \sum_{k=1}^{K} p_k w_{t+1}^k$. In FedAvg, the sample ratio SR, the number of local steps E and the mini-batch size B together control the computation and communication overhead. With $SR = 1$ and $E = 1$, FedAvg is reduced to the standard synchronous distributed SGD.

Our proposed algorithms are built upon the FedAvg framework, *i.e.* local updating with SGD and global model averaging. However, our algorithms are tailored for federated learning with distribution regularization by using a delayed mapping to update the regularization term, as explained below.

B. rFedAvg Algorithm

Basic Idea. As mentioned before, directly applying FedAvg results in calculating the distances $||\frac{1}{n_k}\sum_{j=1}^{n_k}\phi_t(x_{k,j}) - \frac{1}{n_{k^0}}\sum_{j=1}^{n_{k^0}}\phi_t(x_{k^0,j})||^2$ between clients k and k^0 in each round t. The basic idea of rFedAvg is to use a delayed mapping to avoid such all-round communications. Specifically, we define a local mapping operator as $\delta_t^k = \frac{1}{n_k}\sum_{j=1}^{n_k}\phi_t(x_{k,j})$ so that the distance becomes $||\delta_t^k - \delta_t^{k^0}||^2$. A delayed mapping refers to that, for client k at round t, we use the local mapping of k^0 at some previous round $t^0 < t$, *i.e.*, $\delta_{t^0}^{k^0}$ to calculate their distance. The synchronization of local mappings δ follows that in FedAvg, *i.e.*, synchronizing in every E local steps.

Algorithm Details. Algorithm 1 illustrates the rFedAvg algorithm. Note that the number of iteration t (*i.e.*, steps of gradient descent) in FedAvg does not differ local and global steps. We use an extra notion c to represent the global communication (synchronization) steps to avoid ambiguity. At each communication step, the number of iteration t is $t = c \cdot E$. The algorithm runs for C communication rounds in total, which equals $C \cdot E$ iterations.

In the i^{th} local training steps after a global step c, client k calculates the gradient F_k^0, where $F_k^0(w, w_{t_0}) = f_k(w) + \lambda r_k^0(w, w_{t_0})$, $r_k^0(w, w_{t_0}) = \sum_{j \in k} d^2(\phi(w, x_k), \phi(w_{t_0}, x_j))$. The delayed maps δ_{cE} is broadcast by the server at global step c and thus is delayed for i steps. After the local training, each client sends their $\delta_{(c+1)E}^k$ as well as the local model parameters to the server for aggregation and following communications.

Remarks. The rFedAvg algorithm has two shortcomings.

Algorithm 1: rFedAvg Algorithm

input : C: communication rounds; E: local steps; η: learning rate; (p_k): weights of clients; λ: objective weight parameter.

output: w_{CE}: the final global model.

Server initializes w_0, δ_0;
for $c = 0, 1, \cdots, C-1$ **do**
  Server sends w_{cE}, δ_{cE} to each client;
  for *Client* $k = 1, 2, \cdots, N$ **do**
    $w_{cE}^k \leftarrow w_{cE}$;
    for *each local epoch* $i = 1, 2, \cdots, E$ **do**
      $t \leftarrow c \cdot E + i - 1$;
      Randomly samples ξ_t^k from local data of client k;
      $w_{t+1}^k \leftarrow w_t^k - \eta_t \cdot \nabla F_k^0(w_t^k, \xi_t^k, \delta_{cE})$;
    $\delta_{(c+1)E}^k \leftarrow \frac{1}{n_k}\sum_{j=1}^{n_k}\phi(w_{cE}^k, x_{k,j})$;
    Client sends $w_{(c+1)E}^k$, $\delta_{(c+1)E}^k$ to the server;
  Server updates $w_{(c+1)E} \leftarrow \sum_{k=1}^{N} p_k w_{(c+1)E}^k$;
  Server updates $\delta_{(c+1)E} \leftarrow (\delta_{(c+1)E}^1, \delta_{(c+1)E}^2, \cdots, \delta_{(c+1)E}^N)$
return w_{CE}

Assume δ is a d-dimension vector. Then the communication overhead in a single round is at least $O(dN^2)$ since the server has to broadcast a copy of $N \cdot d$-dimension vector to N clients.

Each delayed $\delta_{cE}^{k^0}$ is calculated with each client's local model parameter $w_{cE}^{k^0}$, which may aggravate the discrepancy between clients.

These drawbacks lead us to devise an improved algorithm: rFedAvg+, which is explained below.

C. rFedAvg+ Algorithm

Basic Idea. To further reduce the communication cost and to avoid the inconsistent calculation of mappings, we propose the rFedAvg+ algorithm, which modifies rFedAvg in the following aspects:

We add a synchronization step in each round to obtain consistent global models before calculating the mappings.

We reduce the communication overhead by taking the average of all δ^{k^0} with $k^0 \neq k$ rather than calculate their distances.

Algorithm Details. Algorithm 2 illustrates the rFedAvg+ algorithm. We mainly describe the two modifications in details. Firstly, we allow the server and clients to communicate twice in each communication round. At the first time, the server and the clients only synchronize the global model. At the second time, each client calculates $\delta_{(c+1)E}^k$ with the global model and then sends it back to the server. So the clients can reach consensus on the models to calculate their distances. As we will see in Theorem 1 and Theorem 2, this modification can decrease the constant term in convergence rate. Secondly,

Algorithm 2: rFedAvg+ Algorithm

input : C: communication rounds; E: local steps; η: learning rate; (p_k): weights of clients;λ: objective weight parameter.

output: $w_{C\,E}$: the final global model.

Server initializes w_0, $\quad_0 = (\delta_0{}^1, \cdots \delta_0{}^N)$;
Server sends w_0 to each client ;
for $c = 0, 1, \cdots, C-1$ **do**
  Server sends $\delta_{c\,E}^{\ k}$ to client k ;
  for *Client* $k = 1, 2, \cdots, N$ **do**
    $w_{cE}^k \leftarrow w_{cE}$;
    for *each local epoch* $i = 1, 2, \cdots, E$ **do**
      $t \leftarrow c \cdot E + i - 1$;
      Randomly samples ξ_t^k from local data of client k;
      $w_{t+1}^k \leftarrow w_t^k - \eta_t \cdot \nabla F_k^0(\varpi_t^k, \xi_t^k, \quad_{cE}^{\ k})$;
    Client sends $w_{(c+1)E}^k$ to the server;
  Server updates $w_{(c+1)E} \leftarrow \sum_{k=1}^N p_k w_{(c+1)E}^k$;
  Server sends $w_{(c+1)E}$ to each client ;
  for *Client* $k = 1, 2, \cdots, N$ **do**
    $\delta_{(c+1)E}^k \leftarrow \frac{1}{n_k} \sum_{j=1}^{n_k} \phi(\varpi_{(c+1)E}, x_{k,j})$;
    Client sends $\delta_{(c+1)E}^k$ to the server ;
  for $k = 1, 2, \cdots, N$ **do**
    Server updates $\delta_{(c+1)E}^{\ k} \leftarrow \frac{1}{N-1} \sum_{j \neq k} \delta_{(c+1)E}^j$;
return $w_{C\,E}$

the server will use the average of of clients rather than the N-dimension vector of . Therefore the communication overhead is reduced from $O(dN^2)$ to $O(dN)$. In this case, the objective of r_k will change from $r_k = \frac{1}{N-1} \sum_{j \neq k} ||\delta^k - v^j||^2$ to $\tilde{r}_k = ||\delta^k - \frac{1}{N-1} \sum_{j \neq k} \delta^j||^2$. Note that r_k and $\tilde{r}_k$ have the same gradients with respect to v^k so the convergence can still hold, while $\tilde{r}_k$ can also be considered as a tight lower bound of r_k.

Remarks. rFedAvg+ reduces the total communication overhead from $O(dN^2)$ to $O(dN)$, although the clients need to communicate with the server twice in each training round. As we will show in the evaluations, rFedAvg+ generally outperforms rFedAvg in terms of test accuracy, and it is also more efficient in training time per round. It is also worth mentioning that although we describe rFedAvg and rFedAvg+ by assuming *full participation* of clients, our empirical studies show that they are also effective in case of *partial participation*. However, the proposed methods still have some limitations. For example, they can only alleviate the data heterogeneity problem but cannot fully address it especially in case of extreme non-IID (*i.e.* with outliers). In this case, a potential remedy is to eliminate the outliers first, and then our approach will be feasible.

V. Theoretical Analysis

This section theoretically analyzes the convergence of rFedAvg and rFedAvg+ on non-IID data.

A. Notations and Assumptions

Recall that our objective function, *i.e.*, Eq. (3) is

$$min_w\{F(w) \triangleq \sum_{k=1}^N p_k F_k(w)\}.$$

where $F_k(w) \triangleq f_k(w) + \lambda r_k(w)$, $f_k(w) = \sum_{j=1}^{n_k} l_j(w, x_{kj})$, $r_k(w) = \sum_{j \neq k} d^2(\phi(\varpi, x_k), \phi(\varpi, x_j))$, $w = (\varpi, \tilde{\varpi})$. ϖ is parameter of ϕ and $\tilde{\varpi}$ is parameter of the classification model.

From [11], the process of gradient descent in FedAvg can be represented by the following sequence.

$$v_{t+1}^k \triangleq w_t^k - \eta_t \nabla F_k(w_t^k, \xi_t^k).$$

$$w_{t+1}^k \triangleq \begin{cases} v_{t+1}^k & \text{if } E \nmid (t+1) \\ \sum_{k=1}^N p_k v_{t+1}^k & \text{if } E | (t+1) \end{cases}$$

where w_t^k is the model parameter of client k at t^{th} round. It also defines two virtual sequences: $v_t \triangleq \sum_k p_k v_t^k$, $w_t \triangleq \sum_k p_k w_t^k$.

In our methods, the gradient ∇F_k is inaccurate due to the approximated calculation of r_k. We use the apostrophe on the initial symbols to represent the approximation. Then the gradient descent process in rFedAvg and rFedAvg+ is

$$v'^k_{t+1} \triangleq w'^k_t - \eta_t \nabla F_k'(w'^k_t, \xi_t^k, \varpi'_0).$$

$$w'^k_{t+1} \triangleq \begin{cases} v'^k_{t+1} & \text{if } E \nmid (t+1) \\ \sum_{k=1}^N p_k v'^k_{t+1} & \text{if } E | (t+1) \end{cases}$$

where $F_k'(w, \varpi'_0) \triangleq f_k(w) + \lambda r_k'(w, \varpi'_{t_0})$, $r_k'(w, \varpi'_{t_0}) \triangleq \sum_{j \neq k} d^2(\phi(\varpi, x_k), \phi(\varpi'_{t_0}, x_j))$.
We also define two virtual sequences: $v'_t \triangleq \sum_k p_k v'^k_t$, $w'_t \triangleq \sum_k p_k w'^k_t$.

We make the following assumptions.

A1. $F_1, F_2, ..., F_N$ are all $L-smooth$ and $\mu-strongly\ convex$.

A2. $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k) - \nabla F_k(w_t^k)\|^2 \leq \sigma_k^2$.

A3. $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k)\|^2 \leq G^2$.

Note that A1, A2 and A3 are used in the convergence analysis of FedAvg [11]. We further make assumptions on the mapping ϕ: its gradient and diameter are bounded and the mapping is convex.

A4. $\|\nabla \phi(\varpi, x)\|^2 \leq H^2$, $\mathbb{E}\|\nabla F_k'(w_t^k, \xi_t^k)\|^2 \leq G'^2$.

A5. $max_{i \neq j} \|\phi(\varpi, x_i) - \phi(\varpi, x_j)\|^2 \leq \tau^2$.

A6. ϕ is a convex mapping.

B. Convergence Results

We first review the convergence of FedAvg on non-IID data in Lemma 1, then derive the convergence of rFedAvg+ in Theorem 1, and finally extend the conclusion to rFedAvg in Theorem 2.

According to [11], the gradient descent in FedAvg has a convergence rate of $O(1/T)$.

Lemma 1. *(Theorem 1. in [11]) Assume A1, A2 and A3. Choose* $\kappa = L/\mu$, $\gamma = max(8\kappa, E)$, $\beta = \frac{2}{\mu}$, $v = max(\frac{\beta^2 B}{\beta\mu_1}, (\gamma+1)\mathsf{E}\|w_1 - w\|^2)$ *and the learning rate* $\eta_t = \frac{2}{\mu(\gamma+t)}$. *Then*

$$\mathsf{E}\|w_t^2 - w\|^2 \leq \frac{v}{t+\gamma}.$$

Then by Assumption 3 and 4, the deviation between the local model $w_t^k(w^{0k}_t)$ and the global model w_{t_0} can be bounded by local steps E and learning rate η_{t_0}.

Lemma 2. *Assume A3 and A4. Then*

$$\mathsf{E}\|w_t^k - w_{t_0}\|^2 \leq E^2\eta_{t_0}^2 G^2, \mathsf{E}\|w^{0k}_t - w_{t_0}\|^2 \leq E^2\eta_{t_0}^2 G^{02}.$$

Proof. By the convexity of $\|\cdot\|^2$:
$\mathsf{E}\|w_t^k - w_{t_0}\|^2 = \mathsf{E}\|\sum_{t=t_0}^{t-1}\eta_t\nabla F_k(w_t^k,\xi_t^k)\|^2 \leq E\sum_{t=t_0}^{t-1}\eta_t^2\mathsf{E}\|\nabla F_k(w_t^k,\xi_t^k)\|^2 \leq E^2\eta_{t_0}^2G^2$, where $t-t_0 < E$ and $E|t_0$.
$\mathsf{E}\|w^{0k}_t - w_{t_0}\|^2 = \mathsf{E}\|\sum_{t=t_0}^{t-1}\eta_t\nabla F_k^0(w^{0k}_t,\xi_t^k)\|^2 \leq E\sum_{t=t_0}^{t-1}\eta_t^2\mathsf{E}\|\nabla F_k^0(w_t^k,\xi_t^k)\|^2 \leq E^2\eta_{t_0}^2G^{02}$.
□

Lemma 1 shows that w_t can converge to w . However, w_t represents the model in FedAvg. In our methods, the local model cannot update in every local epoch. Next we prove that the difference between v_t^0 and v_t can be bounded by $O(\eta_{t_0}^2)$.

Lemma 3. *Assume A4, A5, A6 and* $\eta_{t+1} \leq \eta_t$. *Then*

$$\mathsf{E}\|v_t^0 - v_t\|^2 \leq \eta_{t_0}^2 C_1 + \eta_{t_0}^4 C_2.$$

where $C_1 = \sum_k p_k(2E^2(G^2 + G^{02} + 2GG^0) + 16G^2 + 32m^2H^2\tau^2)$ *and* $C_2 = \sum_k 16p_k m^2E^2H^4(3G^2 + G^{02})$

Proof.

$$\begin{aligned}
&\mathsf{E}\|v^0_{t+1} - v_{t+1}\|^2 = \mathsf{E}\|\sum_k p_k(v^{0k}_{t+1} - v^k_{t+1})\|^2\\
\leq&\mathsf{E}\sum_k p_k\|v^{0k}_{t+1} - v^k_{t+1}\|^2\\
=&\mathsf{E}\sum_k p_k\|w^{0k}_t - w_t^k + \eta_t\nabla F_k^0(w^{0k}_t,\xi^{0k}_t,w_{t_0})\\
&-\eta_t\nabla F_k(w_t^k,\xi_t^k)\|^2\\
\leq&2(\sum_k p_k\mathsf{E}\|w^{0k}_t - w_t^k\|^2 +\\
&\eta_t^2\sum_k p_k\mathsf{E}\|\nabla F_k^0(w^{0k}_t,\xi^{0k}_t,w_{t_0}) - \nabla F_k(w_t^k,\xi_t^k)\|^2). \quad (6)
\end{aligned}$$

By Lemma 2, we can get:

$$\begin{aligned}
&\mathsf{E}\|w^{0k}_t - w_t^k\|^2 = \mathsf{E}\|w^{0k}_t - w_{t_0} + w_{t_0} - w_t^k\|^2\\
\leq&\mathsf{E}\|w^{0k}_t - w_{t_0}\|^2 + \mathsf{E}\|w_{t_0} - w_t^k\|^2 +\\
&2\mathsf{E}\|w^{0k}_t - w_{t_0}\|\cdot\mathsf{E}\|w_{t_0} - w_t^k\|\\
\leq&E^2\eta_{t_0}^2(G^2 + G^{02} + 2GG^0). \quad (7)
\end{aligned}$$

For the second term in (6):

$$\begin{aligned}
&\mathsf{E}\|\nabla F_k^0(w^{0k}_t,\xi^{0k}_t,w_{t_0}) - \nabla F_k(w_t^k,\xi_t^k)\|^2\\
=&\mathsf{E}\|\nabla F_k^0(w^{0k}_t,\xi^{0k}_t,w_{t_0}) - \nabla F_k^0(w_t^k,\xi_t^k,w_{t_0}) +\\
&\nabla F_k^0(w_t^k,\xi_t^k,w_{t_0}) - \nabla F_k(w_t^k,\xi_t^k)\|^2\\
\leq&2(\underbrace{\mathsf{E}\|\nabla F_k^0(w^{0k}_t,\xi^{0k}_t,w_{t_0}) - \nabla F_k^0(w_t^k,\xi_t^k,w_{t_0})\|^2}_{B_1} +\\
&\underbrace{\mathsf{E}\|\nabla F_k^0(w_t^k,\xi_t^k,w_{t_0}) - \nabla F_k(w_t^k,\xi_t^k)\|^2}_{B_2}). \quad (8)
\end{aligned}$$

To bound B_1, we rewrite B_1 as:

$$\begin{aligned}
&\mathsf{E}\|\nabla F_k^0(w^{0k}_t,\xi_t^k,w_{t_0}) - \nabla F_k^0(w_t^k,\xi_t^k,w_{t_0})\|^2\\
=&\mathsf{E}\|\nabla f_k(w^{0k}_t,\xi^{0k}_t) - \nabla f_k(w_t^k,\xi_t^k) + \lambda\nabla r_k^0(w^{0k}_t,w_{t_0}) -\\
&\lambda\nabla r_k^0(w_t^k,w_{t_0})\|^2\\
\leq&2(\mathsf{E}\|\nabla f_k(w^{0k}_t,\xi^{0k}_t) - \nabla f_k(w_t^k,\xi_t^k)\|^2 +\\
&\lambda^2\mathsf{E}\|\nabla r_k^0(w^{0k}_t,w_{t_0}) - \nabla r_k^0(w_t^k,w_{t_0})\|^2)\\
\leq&4G^2 + 4\lambda^2(\mathsf{E}\|\nabla r_k^0(w^{0k}_t,w_{t_0})\|^2 + \mathsf{E}\|\nabla r_k^0(w_t^k,w_{t_0})\|^2). \quad (9)
\end{aligned}$$

To bound $\|\nabla r_k^0(w^{0k}_t,w_{t_0})\|^2$, we rewrite it as:

$$\begin{aligned}
&\mathsf{E}\|\nabla\sum_{j\in\bar{k}} d^2(\phi(\tilde{w}^{0k}_t,x_k),\phi(w_{t_0},x_j))\|^2\\
\leq&m\sum_{j\in\bar{k}}\mathsf{E}\|\nabla d^2(\phi(\tilde{w}^{0k}_t,x_k),\phi(w_{t_0},x_j))\|^2\\
=&m\sum_{j\in\bar{k}}\mathsf{E}\|(\phi(\tilde{w}^{0k}_t,x_k) - \phi(w_{t_0},x_j))\nabla\phi(\tilde{w}^{0k}_t,x_k)\|^2\\
\leq&m^2H^4\mathsf{E}\|w^{0k}_t - w_{t_0}\|^2 \leq m^2E^2G^{02}H^4\eta_{t_0}^2.
\end{aligned}$$

where the third inequality results from the convexity of ϕ and the last inequality from Lemma 2.
In a similar way, we can get a bound for $\mathsf{E}\|\nabla r_k^0(w_t^k,w_{t_0})\|^2$:

$$\|\nabla r_k^0(w_t^k,w_{t_0})\|^2 \leq m^2E^2H^4G^2\eta_{t_0}^2. \quad (10)$$

We next bound B_2, which can be rewritten as

$$\begin{aligned}
&\mathsf{E}\|\nabla F_k^0(w_t^k,\xi_t^k,w_{t_0})-\nabla F_k(w_t^k,\xi_t^k)\|^2\\
=&\mathsf{E}\|\nabla f_k(w_t^k,\xi_t^k)-\nabla f_k(w_t^k,\xi_t^k)+\nabla r_k^0(w_t^k,w_{t_0})-\nabla r_k(w_t^k)\|^2\\
=&\mathsf{E}\|\nabla r_k^0(w_t^k,w_{t_0})-\nabla r_k(w_t^k)\|^2\\
\le&4m\sum_{j\in\mathcal{B}_k}\mathsf{E}\|(\phi(\varpi_t^k,x_k)-\phi(w_{t_0},x_j))\nabla\phi(\varpi_t^k,x_k)-\\
&(\phi(\varpi_t^k,x_k)-\phi(\varpi_t^k,x_j))(\nabla\phi(\varpi_t^k,x_k)-\nabla\phi(\varpi_t^k,x_j)\|^2\\
=&4m\sum_{j\in\mathcal{B}_k}\mathsf{E}\|(\phi(\varpi_t^k,x_j)-\phi(\varpi_{t_0},x_j))\nabla\phi(\varpi_t^k,x_k)+\\
&(\phi(\varpi_t^k,x_k)-\phi(\varpi_t^k,x_j))\nabla\phi(\varpi_t^k,x_j)\|^2\\
\le&8m^2\mathsf{E}(\|(\phi(\varpi_t^k,x_j)-\phi(\varpi_{t_0},x_j))\nabla\phi(\varpi_t^k,x_k)\|^2+\\
&\mathsf{E}\|(\phi(\varpi_t^k,x_k)-\phi(\varpi_t^k,x_j))\nabla\phi(\varpi_t^k,x_j)\|^2)\\
\le&8m^2H^2(\tau^2+\mathsf{E}\|\phi(\varpi_t^k,x_j)-\phi(\varpi_{t_0},x_j)\|^2). \qquad (11)
\end{aligned}$$

Since ϕ is a convex mapping, therefore:

$$\begin{aligned}
&\mathsf{E}\|\phi(\varpi_t^k,x_j)-\phi(\varpi_{t_0},x_j)\|^2\\
\le&\mathsf{E}\|\varpi_t^k-\varpi_{t_0}\|^2\cdot\|\nabla\phi(\tilde{w}_t^k,x_j)\|^2\le E^2\eta_{t_0}^2G^2H^2. \qquad (12)
\end{aligned}$$

By combining (6)-(12), it follows that:

$$\begin{aligned}
&\mathsf{E}\|v_{t+1}^0-v_{t+1}\|^2\le\sum_k 2p_kE^2\eta_{t_0}^2(G^2+G'^2+2GG^0)+\\
&4\eta_t^2\sum_k p_k(4G^2+4\lambda^2m^2E^2G'^2H^4\eta_{t_0}^2+4\lambda^2m^2E^2H^4G^2\eta_{t_0}^2+\\
&8m^2H^2(\tau^2+E^2\eta_{t_0}^2G^2H^2)\le\eta_{t_0}^2C_1+\eta_{t_0}^4C_2.
\end{aligned}$$

where the last inequality is from $\eta_{t_0}\le\eta_t$. □

By Lemma 1 and 3, we have the convergence result for rFedAvg+.

Theorem 1. *Let Assumptions A1 - A6 hold and $L,\mu,\sigma_k,G,G^0,\tau$ be defined therein. Choose $\kappa=\frac{L}{\mu}$, $\gamma=max(8\kappa,E)$, $v^0=2v+\frac{8C_1}{\mu^2}+\frac{32C_2}{\mu^4}$ and the learning rate $\eta_t=\frac{2}{\mu(\gamma+t)}$. Then for rFedAvg+:*

$$\mathsf{E}[F(w_t^0)-F^*]\le\frac{L}{2}\frac{v^0}{t+\gamma-E}.$$

Proof. Let $\Gamma_t=E\|w_t^0-w^*\|^2$. From Lemma 1 and Lemma 3, it follows that:

$$\begin{aligned}
\Gamma_t&=\mathsf{E}\|v_t^0-v_t+v_t-w^*\|^2\le2\mathsf{E}(\|v_t^0-v_t\|^2+\|v_t-w^*\|^2)\\
&\le\frac{2v}{t+\gamma}+\frac{8C_1}{\mu^2(t+\gamma-E)^2}+\frac{32C_2}{\mu^4(t+\gamma-E)^4}\\
&\le\frac{v^0}{t+\gamma-E}.
\end{aligned}$$

Then by the L-smoothness of $F(\cdot)$:

$$\mathsf{E}[F(w_t^0)-F^*]\le\frac{L}{2}\Gamma_t\le\frac{L}{2}\frac{v^0}{t+\gamma-E}.$$

□

For rFedAvg, the regularization $r_{l_k}^0$ is computed with each client's local model parameter $\varpi_{cE}^{k_{l_k}}$ rather than the global parameter w_{t_0}.

Theorem 2. *Let Assumptions A1 - A6 hold and $L,\mu,\sigma_k,G,G^0,\tau$ be defined therein. Choose $\kappa=\frac{L}{\mu}$, $\gamma=max(8\kappa,E)$, $v^0=2v+\frac{8C_1}{\mu^2}+\frac{32C_3}{\mu^4}$ and the learning rate $\eta_t=\frac{2}{\mu(\gamma+t)}$. Then for rFedAvg:*

$$\mathsf{E}[F(w_t^0)-F^*]\le\frac{L}{2}\frac{v^0}{t+\gamma-E}.$$

where

$$C_3=\sum_k 64p_km^2E^2H^4(4G^2+G'^2+2\lambda^2(2G^2+3G'^2)).$$

Proof. The only difference is to replace $\phi(w_{t_0},x_j)$ with $\phi(v_{t_0}^{0j},x_j)$. (w_j is the local parameter in client j)
Similar to Lemma 2, we can get:

$$\mathsf{E}\|\tilde{v}_{t_0}^{0j}-w_{t_0}\|^2\le E^2\eta_{t_0}^2\Gamma_{t_0}G'^2\le4E^2\eta_{t_0}^2G'^2 \qquad (13)$$

$$\mathsf{E}\|\tilde{w}_t^{0k}-w_{t_0}\|^2\le4E^2\eta_{t_0}^2\Gamma_{t_0}G'^2\le16E^2\eta_{t_0}^2G'^2. \qquad (14)$$

By combining (13) and (14), we can bound $\|\nabla r_k^0(w_t^{0k},v_{t_0}^{0j})\|^2$:

$$\begin{aligned}
&\mathsf{E}\|\nabla r_k^0(w_t^{0k},\tilde{v}_{t_0}^{0j})\|^2\le m^2H^4\mathsf{E}\|w_t^{0k}-v_{t_0}^{j^\theta}\|^2\\
&\le m^2H^4\mathsf{E}(\|w_t^{0k}-w_{t_0}\|^2+\|v_{t_0}^{0j}-w_{t_0}\|^2)\\
&\le20m^2E^2G'^2H^4\eta_{t_0}^2. \qquad (15)
\end{aligned}$$

In a similar way:

$$\mathsf{E}\|\nabla r_k^0(w_t^k,v_{t_0}^{0j})\|^2\le4m^2E^2H^4\eta_{t_0}^2(4G^2+G'^2). \qquad (16)$$

Since ϕ is a convex mapping, therefore:

$$\begin{aligned}
&\mathsf{E}\|\phi(\varpi_t^k,x_j)-\phi(\tilde{v}_{t_0}^{0j},x_j)\|^2\\
\le&\mathsf{E}\|\varpi_t^k-\tilde{v}_{t_0}^{0j}\|^2\cdot\|\nabla\phi(\tilde{w}_t^k,x_j)\|^2\le4E^2H^2\eta_{t_0}^2(4G^2+G'^2). \qquad (17)
\end{aligned}$$

By combining (6)-(8),(11),(15)-(17), it follows that:

$$\begin{aligned}
&\mathsf{E}\|v_{t+1}^0-v_{t+1}\|^2\le\sum_k 2p_kE^2\eta_{t_0}^2(G^2+G'^2+2GG^0)+\\
&4\eta_t^2\sum_k p_k(4G^2+80\lambda^2m^2E^2G'^2H^4\eta_{t_0}^2+16\lambda^2m^2E^2H^4\eta_{t_0}^2\\
&(4G^2+G'^2)+8m^2H^2(\tau^2+4E^2\eta_{t_0}^2H^2(4G^2+G'^2))\\
&\le\eta_{t_0}^2C_1+\eta_{t_0}^4C_3.
\end{aligned}$$

Let $\Gamma_t=E\|w_t^0-w^*\|^2$. From Lemma 1, it follows that:

$$\begin{aligned}
\Gamma_t&=\mathsf{E}\|v_t^0-v_t+v_t-w^*\|^2\le2\mathsf{E}(\|v_t^0-v_t\|^2+\|v_t-w^*\|^2)\\
&\le\frac{2v}{t+\gamma}+\frac{8C_1}{\mu^2(t+\gamma-E)^2}+\frac{32C_3}{\mu^4(t+\gamma-E)^4}\\
&\le\frac{v^0}{t+\gamma-E}. \qquad (18)
\end{aligned}$$

Then by the L-smoothness of $F(\cdot)$:

$$\mathsf{E}[F(w_t^0)-F^*]\le\frac{L}{2}\Gamma_t\le\frac{L}{2}\frac{v^0}{t+\gamma-E}.$$

□

Remarks. From Theorem 1 and Theorem 2, rFedAvg+ and rFedAvg have a convergence rate of $O(1/T)$, which is the

same as FedAvg, yet the constant terms are larger. rFedAvg+ has a smaller constant term than rFedAvg (*i.e.*, $C_2 < C_3$), which verifies the effectiveness of double synchronization in rFedAvg+. Our assumptions can be loosen by non-convexity of f_k, which is explored in [8]. As we mainly focus on the convergence of r_k, we do not discuss assumptions of vanilla FedAvg (*i.e.*, f_k). Although our analysis is based on strongly convex objectives and full participation of clients, we empirically show that our methods outperform FedAvg with non-convex models *e.g.* neural networks, and our method also work with partial participation.

VI. Evaluation

This section presents the evaluations on standard benchmarks with non-IID data in both cross-device and cross-silo settings.

A. Experimental settings

Compared Methods. We compare the following methods.

FedAvg [1]: the vanilla Federated Averaging algorithm.
FedProx [6]: it focuses on tackling the non-IID data and partial participation problem together in FL.
Scaffold [8]: it uses variance reduction to correct the client drifts in FedAvg on non-IID data.
q-FedAvg [7]: it aims at fairness in FL in heterogeneous networks but it also works with non-IID data.
rFedAvg: our proposed Algorithm 1.
rFedAvg+: our proposed Algorithm 2

Datasets. We compare the performance of different methods on 4 datasets: MNIST [14], CIFAR10 [13], Sent140 [15] and FEMNIST [16]. MNIST and CIFAR10 are image classification benchmarks commonly used in FL [1], [5], [11]. Sent140 is a naturally non-IID dataset for sentiment analysis which contains millions of tweets with annotated sentiment based on the emoticons they use. Another representative naturally non-IID dataset, Federated Extended MNIST (FEMNIST) is also included. It is built by partitioning the data in Extended MNIST based on the writer of the digit/character . Following [32], we test different settings of non-IIDness. The label distribution skewness is simulated on MNIST and CIFAR10 while Sent140 and FEMNIST are naturally feature distribution skewed and quantity skewed. We divide MNIST and CIFAR10 following [8]: we first allocate to each client s% IID data, then sort the remaining $(100 - s)$% data according to the label and allocate them evenly to the clients. We use three settings, Similarity $s = 0\%$ (totally non-IID), Similarity $s = 10\%$ (relatively non-IID) and Similarity $s = 100\%$ (IID) to measure the non-IID-ness. For Sent140 and FEMNIST, we sample 500 users directly from the dataset as the non-IID setting, and randomly shuffle the subset and evenly allocate it to the 500 clients to simulate the IID setting. We evaluate the cross-device and cross-silo settings with different number of clients N, local steps E and sample ratio SR.

Cross-Device Setting: $N = 500, E = 10, SR = 0.2$.
Cross-Silo Setting: $N = 20, E = 5, SR = 1.0$.

Note that $SR = 1.0$ refers to full client participation and $SR = 0.2$ refers to partial participation (20% of clients in each round).

Models and Common Training Hyperparameters. For MNIST and CIFAR10, we use the same CNN structure as [1] where the dimension of the last FC layer is 512. The local optimizer is SGD with learning rate 0.1 (0.01 for FedProx on cross-device settings otherwise it will not converge). The batch size is 100 for cross-silo settings and 32 for cross-device settings. For Sent140, we use 2-layer LSTM + 1-layer FC (dimension of output vector is 256) with pre-trained word vectors. The local optimizer is RMSProp with learning rate 0.01 and batch size 10.

Algorithm-Specific Hyperparameters. The setup of additional hyperparameters for each algorithm are summarized below.

FedAvg: it has no extra hyperparameters.
FedProx: for MNIST and CIFAR10, we set $\mu = 1.0$. For Sent140, we set $\mu = 0.01$ otherwise it can hardly converge.
Scaffold: we set $\eta_g = 1.0$ on all the datasets.
q-FedAvg: we set $q = 1.0$ on MNIST and CIFAR10 and $q = 10^{-4}$ on Sent140 (larger q results in divergence).
rFedAvg: we set $\lambda = 10^{-4}$ on MNIST, $\lambda = 10^{-5}$ on CIFAR10 and $\lambda = 0.1$ on Sent140, as λ also works as normalization coefficient and is related to the dimension and values of feature vectors. The MMD regularizer is calculated on the last FC layer for all datasets.
rFedAvg+: the settings are the same as rFedAvg.

Experimental Environment. We implement all the methods with PyTorch 1.8.0. The experiments were conducted on five Intel(R) Xeon(R) Platinum 8269CY 3.10GHz CPUs each with 4 cores. The code is open-sourced on github [1].

Evaluation Metrics. We use the train loss and test accuracy as evaluation metrics. We also record the training time per round to compare the efficiency of rFedAvg and rFedAvg+.

B. Results

We will show the detailed comparing results, the parameter study and other evaluations respectively.

1) Results on MNIST: The results on MNIST are shown in Fig. 2, Fig. 3 Tab. I and Tab. II. We omit the curves with similarity 100% as they are similar to figures with similarity 10% We record the accuracy for 60 communication rounds. We observe from Fig. 2a and Fig. 3a that rFedAvg and rFedAvg+ can converge faster and more stable than the baselines in average while Scaffold and FedAvg are also competitive in the cross-device 0% similarity setting. FedProx and q-FedAvg perform relatively worse and they also have larger variances. In the cross-silo 0% similarity setting (Fig. 2b, Fig. 3b), the gaps between the methods (except FedProx) become smaller but we can still find from Tab. I that rFedAvg+ can perform the best. With similarity = 10%, all the algorithms can have better

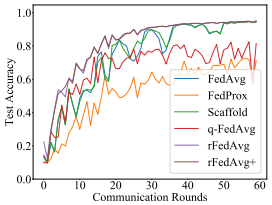
[1] https://github.com/BUAA-BDA/rfedavg

TABLE I: Test accuracy on the three datasets in cross-silo setting. The best performance is marked in bold.

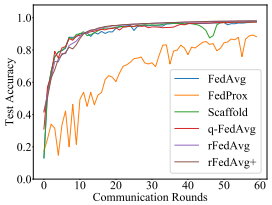
Method	MNIST			CIFAR10			Sent140	
	Sim 0%	Sim 10%	Sim 100%	Sim 0%	Sim 10%	Sim 100%	Non-IID	IID
FedAvg	97.07 ± 0.34	98.85 ± 0.04	99.28 ± 0.01	45.22 ± 1.00	$\mathbf{60.44 \pm 0.08}$	67.39 ± 0.08	72.60 ± 0.35	75.75 ± 0.26
FedProx	85.74 ± 5.61	96.06 ± 0.07	98.13 ± 0.04	21.35 ± 0.33	49.22 ± 0.63	63.31 ± 0.22	50.55 ± 1.97	61.41 ± 2.35
Scaffold	97.35 ± 0.11	98.86 ± 0.04	99.28 ± 0.01	44.56 ± 0.82	60.38 ± 0.12	67.80 ± 0.01	72.89 ± 0.54	75.90 ± 0.25
q-FedAvg	97.28 ± 0.05	98.79 ± 0.04	99.25 ± 0.01	32.24 ± 0.04	58.69 ± 0.05	68.69 ± 0.14	50.67 ± 0.01	60.07 ± 5.36
rFedAvg	97.88 ± 0.07	98.97 ± 0.08	99.30 ± 0.02	46.12 ± 0.82	60.22 ± 0.14	67.91 ± 0.02	$\mathbf{73.26 \pm 0.32}$	74.61 ± 0.37
rFedAvg+	$\mathbf{98.02 \pm 0.03}$	$\mathbf{99.15 \pm 0.01}$	99.31 ± 0.02	$\mathbf{47.41 \pm 0.32}$	60.36 ± 0.10	68.01 ± 0.02	72.85 ± 0.23	75.69 ± 0.34

TABLE II: Test accuracy on the three datasets in cross-device setting. The best performance is marked in bold.

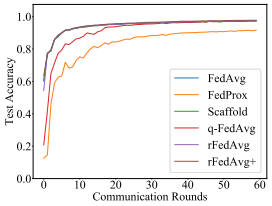
Method	MNIST			CIFAR10			Sent140	
	Sim 0%	Sim 10%	Sim 100%	Sim 0%	Sim 10%	Sim 100%	Non-IID	IID
FedAvg	94.38 ± 0.26	97.37 ± 0.06	98.42 ± 0.05	42.31 ± 0.88	54.78 ± 0.25	58.36 ± 0.13	72.88 ± 0.26	77.22 ± 0.28
FedProx	69.36 ± 5.78	91.46 ± 0.16	96.35 ± 0.06	24.85 ± 2.79	43.25 ± 0.44	49.73 ± 0.22	70.93 ± 0.56	72.74 ± 0.64
Scaffold	94.37 ± 0.19	97.38 ± 0.07	98.42 ± 0.05	42.33 ± 0.91	54.83 ± 0.34	58.36 ± 0.13	67.63 ± 0.82	69.16 ± 0.57
q-FedAvg	74.42 ± 10.17	97.27 ± 0.16	98.40 ± 0.04	32.09 ± 0.82	49.29 ± 0.18	50.37 ± 0.08	69.89 ± 0.80	67.81 ± 0.37
rFedAvg	94.66 ± 0.40	97.45 ± 0.04	98.43 ± 0.01	43.09 ± 1.85	$\mathbf{55.08 \pm 0.33}$	58.50 ± 0.08	75.69 ± 0.78	76.73 ± 0.37
rFedAvg+	$\mathbf{94.72 \pm 0.26}$	$\mathbf{97.73 \pm 0.04}$	98.44 ± 0.04	$\mathbf{44.14 \pm 0.97}$	54.90 ± 0.34	58.48 ± 0.15	$\mathbf{76.38 \pm 0.21}$	77.74 ± 0.16



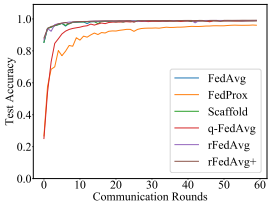
(a) Cross-device Sim 0%



(b) Cross-silo Sim 0%

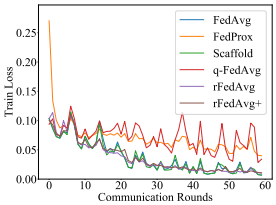


(c) Cross-device Sim 10%

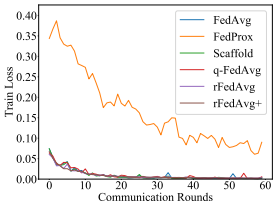


(d) Cross-silo Sim 10%

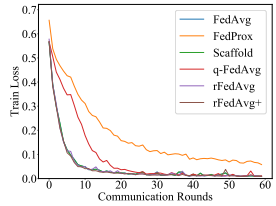
Fig. 2: Accuracy curves on MNIST.



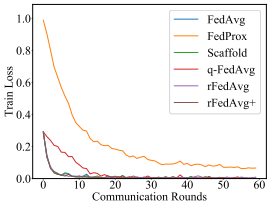
(a) Cross-device Sim 0%



(b) Cross-silo Sim 0%



(c) Cross-device Sim 10%

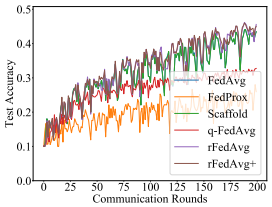


(d) Cross-silo Sim 10%

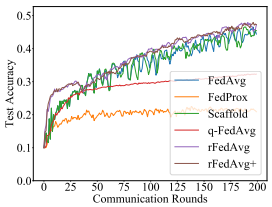
Fig. 3: Loss curves on MNIST.

performance while the advantages of rFedAvg and rFedAvg+ become smaller. With similarity = 100%, all the algorithms perform nearly the same. Overall, we find that with higher non-IID-ness (smaller similarity), the proposed methods can perform relatively better both in cross-silo and cross-device settings, which proves its effectiveness in dealing with non-IID data in FL. However, we also observe that even in the worst case (cross-device similarity 0%) most of the methods can still achieve a test accuracy of nearly 95%. It indicates that the non-IID problem is not severe on MNIST even with extreme data division.

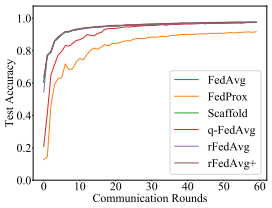
2) Results on CIFAR10: The results on CIFAR10 are shown in Fig. 4, Fig. 5, Tab. I and Tab. II. We record the results for 200 rounds. We can find roughly that the non-IID division of CIFAR10 can lead to about 30% of accuracy loss comparing with IID division, which means CIFAR10 is more appropriate for non-IID evaluations. In the totally non-IID cases (similarity 0%), rFedAvg+ performs best both in cross-device and cross-silo settings and leads other methods by over 2%. FedAvg is still competitive and shows obvious advantages over FedProx and q-FedAvg. We also observe that the baselines' curves of test accuracy oscillate violently especially in cross-device settings while those of rFedAvg and rFedAvg+ look more stable with higher averages. With the similarity increasing to 10% and 100%, the accuracies of all methods rise very fast and the proposed methods have less obvious advantages or are even outperformed by FedAvg in the cross-silo similarity 10% setting. The results are aligned with [5] that only a small part of shared IID data can bring considerable improvement of performance. However, such strategy should not be allowed in FL according to the privacy requirements. Therefore, the proposed methods can be meaningful especially in totally non-



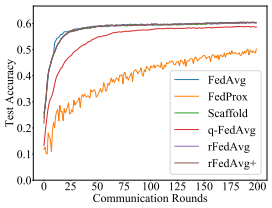
(a) Cross-device Sim 0%



(b) Cross-silo Sim 0%

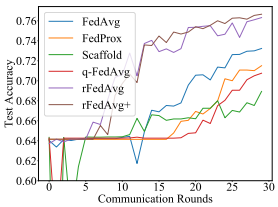


(c) Cross-device Sim 10%

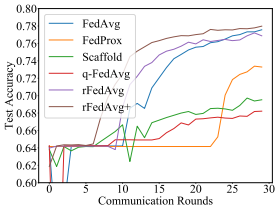


(d) Cross-silo Sim 10%

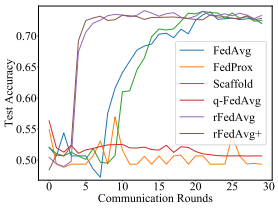
Fig. 4: Accuracy curves on CIFAR10.



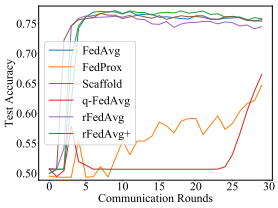
(a) Cross-device non-IID accuracy



(b) Cross-device IID accuracy

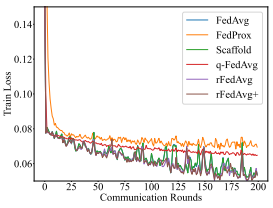


(c) Cross-silo non-IID accuracy

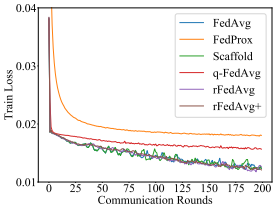


(d) Cross-silo IID accuracy

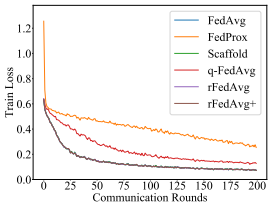
Fig. 6: Accuracy curves on Sent140.



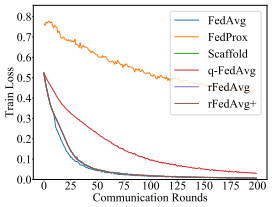
(a) Cross-device Sim 0%



(b) Cross-silo Sim 0%

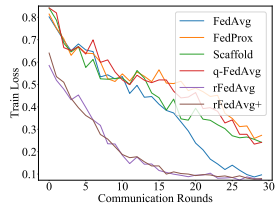


(c) Cross-device Sim 10%

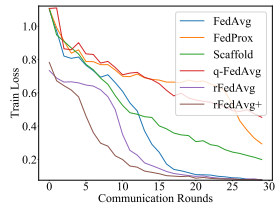


(d) Cross-silo Sim 10%

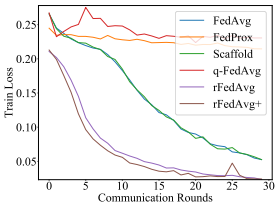
Fig. 5: Loss curves on CIFAR10.



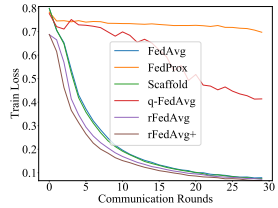
(a) Cross-device non-IID



(b) Cross-device IID



(c) Cross-silo non-IID



(d) Cross-silo IID

Fig. 7: Loss curves on Sent140.

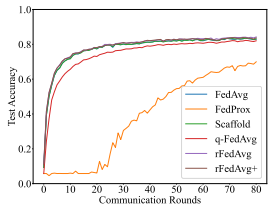
IID settings where no IID data can be shared.

3) Results on Sent140: The results on Sent140 are shown in Fig. 6, Fig. 7, Tab. I and Tab. II. We record the results for 30 rounds. In cross-device settings with non-IID data, we can observe from Fig. 6a and Fig. 7a that the advantages of rFedAvg and rFedAvg+ are obvious comparing with the baselines. They can lead by over 3% according to Tab. II and the superiority in convergence speed is also very obvious accroding to Fig. 6a. On IID data, the performance of FedAvg approaches the proposed methods closely. In cross-silo settings, We can find that rFedAvg and rFedAvg+ still outperform the baselines on non-IID data significantly while FedProx and q-FedAvg can hardly converge. The reasons that they perform bad on Sent140 may be that they are only designed for SGD and do not support other optimizers well. But the proposed rFedAvg and rFedAvg+ can still work well with RMSProp, which verifies that our methods have better compatibility.

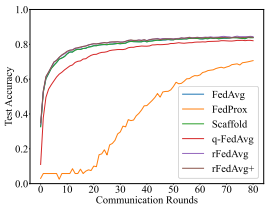
4) Results on FEMNIST: The results on FEMNIST are shown in Fig. 8. We evaluate 2 settings: with 100 clients and 500 clients respectively and record the results for 80 rounds. In the figures, low cost refers to $SR = 0.1, E = 10$ and high cost refers to $SR = 0.2, E = 20$. We can observe that the proposed rFedAvg performs the best among all baselines, while rFedAvg+ also shows competitive performance with both 100 clients and 500 clients.

5) Parameter Study: We vary the hyperparameters including λ, N, E and SR to show the results. The experiments are conducted on CIFAR10 with non-IID division (similarity 0%).

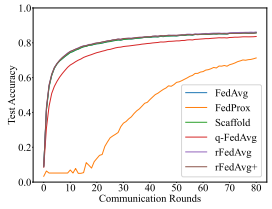
Impact of λ. From Fig. 9a, we observe that with too small or too large λ, the optimization may be negative and the accuracy can even be lower than FedAvg. The reason is that the MMD loss is usually over 100 in the first few rounds due to the high dimension of v while the training loss is smaller than 0.1.



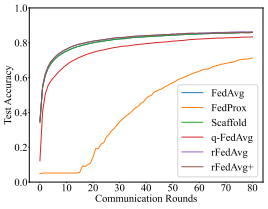
(a) 100 clients (low cost)



(b) 100 clients (high cost)

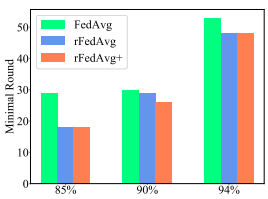


(c) 500 clients (low cost)

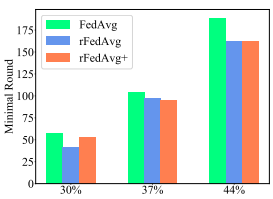


(d) 500 clients (high cost)

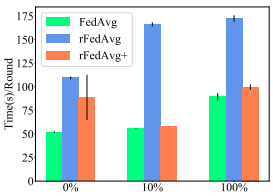
Fig. 8: Accuracy curves on FEMNIST with 100/500 clients.



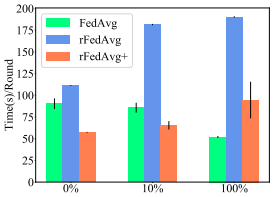
(a) Minimal rounds needed on MNIST



(b) Minimal rounds needed on CIFAR10

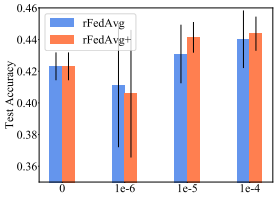


(c) Training time on MNIST

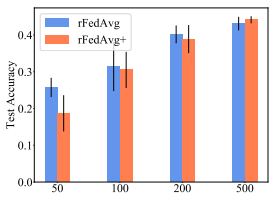


(d) Training time on CIFAR10

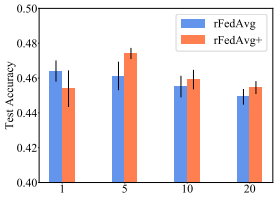
Fig. 10: Efficiency evaluation.



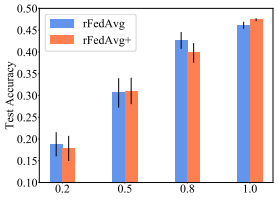
(a) Test accuracy varying λ



(b) Test accuracy varying N



(c) Test accuracy varying E



(d) Test accuracy varying SR

Fig. 9: Parameter study.

Therefore it is necessary to find an appropriate λ for different settings (*e.g.*, $\lambda = 10^{-5}$ in this setting).

Impact of N. Results varying the number of clients N are shown in Fig. 9b. We find that with smaller N, the accuracy decreases very fast, even to below 0.3 when $N = 50$. The reason is that the clients' data become more unbiased with smaller N while the sample ratio SR remains 0.2. The results are also consistent with those in Fig. 9d which vary SR with the same N. But there also exists a dividing crest of accuracy, *e.g.*, the threshold is $SR = 0.2, N = 200$ in this case. The benefits to the accuracy become negligible when $N \cdot SR$ exceeds the threshold. Therefore it is important to find this threshold especially in cross-device and non-IID settings, to reduce the surplus participants and unnecessary communication cost.

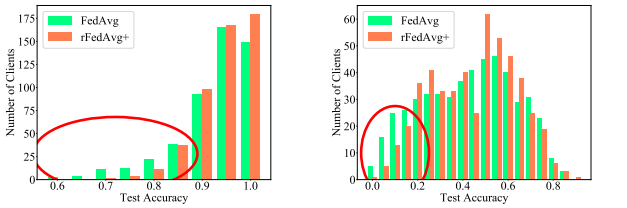
TABLE III: Size of δ (B).

Model	Cross-Silo		Cross-Device	
	CNN	RNN	CNN	RNN
rFedAvg	56160	35680	280800	178400
rFedAvg+	2808	1784	2808	1784

Impact of E. We also vary the number of local steps E in Fig. 9c with the same number of communication rounds $C = 200$. We observe that the accuracy will sightly decrease with larger E except for rFedAvg+ with $E = 1$. The possible reason is that a bit more local steps may help reduce the variance of the estimated average of v. However, with the same rounds of SGD (*e.g.*, $E \cdot C = 200$), the convergence rate will decrease significantly with bigger E (accuracy < 0.3 when $E \geq 5$, we omit the results due to the space limitation), which is consistent with the theoretical results.

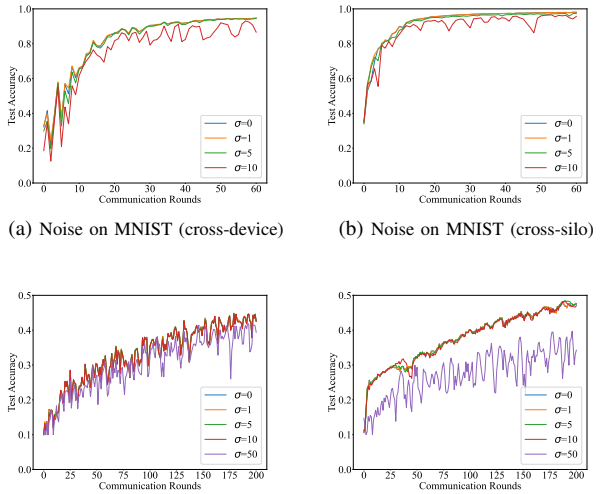
Impact of SR. The findings in Fig. 9d are similar to those in Fig. 9b. With the same N and smaller sample ratio SR, the accuracy can obviously get worse. Therefore, in cross-silo and non-IID settings, it is reasonable to set large SR or even to avoid using client sampling to ensure the learning performance, as the communication cost may not be the primary problem when N is small.

6) Efficiency Evaluation: We first compare the minimal communication rounds needed for achieving different levels of accuracy and the results are shown in Fig. 10a and Fig. 10b. The settings are both cross-device with non-IID data. We can observe that rFedAvg and rFedAvg+ needs fewer rounds to converge to some specific levels than FedAvg, which verifies that the proposed methods are more efficient in communication rounds. We further compare the training times. The results are shown in Fig. 10c and Fig. 10d. We can observe that the



(a) Performance of clients on MNIST (b) Performance of clients on CIFAR10

Fig. 11: Fairness evaluation.



(a) Noise on MNIST (cross-device) (b) Noise on MNIST (cross-silo)

(c) Noise on CIFAR10 (cross-device) (d) Noise on CIFAR10 (cross-silo)

Fig. 12: Privacy evaluation.

average training time of rFedAvg+ (orange bar) is generally half of rFedAvg (blue bar), and also close to FedAvg (green bar). With the similarity of 10%, the training time of rFedAvg+ can be reduced by $2/3$. To further evaluate the communication cost, we compare the memory used for δ in rFedAvg and rFedAvg+. The results are shown in Tab. III. We can observe that the size of δ in rFedAvg+ does not expand with the client number as rFedAvg does, which saves the communciation overhead significantly. The results verify that the efficiency optimization in rFedAvg+ is effective.

7) Fairness Evaluation: We also evaluate the fairness of the proposed methods. We can observe from Fig. 11 that on both MNIST and CIFAR10 the test accuracy on the worst clients (*i.e.* the red circles in the figures) can be generally higher in rFedAvg+ than in FedAvg. This result verifies that our methods can improve not only the overall performance but also the performance on the worst clients, which corresponds to better fairness.

8) Privacy Evaluation: We also evaluate rFedAvg+ under privacy preservation. Following [43], we insert Gaussian noise into the intermediate regularization variable δ with noise standard deviation σ_2: $\tilde{\delta}_i \leftarrow \delta_i + \frac{1}{L}\mathcal{N}(0, \sigma_2^2 C_0^2 I)$, where L is the batch size, σ_2 is the noise parameters, C_2 is the clipping constant. The results are shown in Fig. 12. We can observe that with $\sigma_2 \leq 5$, the curves are almost overlapped and the performance is hardly affected. But with larger magnitude of σ_2, the performance may be damaged. It means that our approach is compatible with certain level of privacy preservation.

C. Summary of Results

In summary, the proposed methods, rFedAvg and rFedAvg+ can generally outperform FedAvg by $0.95\% - 2.19\%$ in cross-silo settings and by $0.34\% - 3.50\%$ in cross-device settings with non-IID data. We further show that rFedAvg+ can run about twice as fast as rFedAvg in cross-device settings therefore the efficiency optimization techniques are useful. The proposed methods can also improve the performance on the worst clients, which can make the global model more fair. Moreover, our methods are robust with certain level of differentially private Gaussian noise, which means that privacy can still be preserved with regularization.

VII. Conclusion

We study the non-IID problem of federated learning in this paper. We reveal that FedAvg can suffer from inconsistent distributions and high discrepancy of local models when learning on non-IID data despite its efficiency in communication. Inspired by domain adaptation, we propose a distribution regularization for FL on non-IID data to reduce the discrepancy of data distributions between clients. To further reduce communication cost when learning with the distribution regularization, we devise two novel communication-efficient distributed learning algorithms named rFedAvg and rFedAvg+. We theoretically establish their convergence for strongly convex objectives. Finally, we conduct extensive experiments on 4 datasets with both CNN and LSTM as learning models. The results show that the proposed algorithms have obvious advantages over the state-of-the-arts on non-IID data in terms of communication rounds and test accuracy.

The data heterogeneity problem will still be a major challenge for federated learning in the future. Solutions from a single perspective like regularization cannot fully address the problem especially when the data is extremely non-IID. As a future direction, adaptive participant selection and personalized federated learning can be combined with a centralized training framework, to improve the generalization of the global model and the personalization performance of local models simultaneously.

Acknowledgment

We are grateful to anonymous reviewers for their constructive comments. This work is partially supported by the National Key Research and Development Program of China under Grant No. 2018AAA0101100, the National Science Foundation of China (NSFC) under Grant No. U21A20516 and 62076017, the Funding Grant No. 22-TQ23-14-ZD-01-001, WeBank Scholars Program and Didi Collaborative Research Program. Yongxin Tong is the corresponding author.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017, pp. 1273–1282.

[2] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM TIST*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[4] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *CoRR*, vol. abs/1806.00582, 2018.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *MLSys*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds., 2020.

[7] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in *ICLR*, 2020.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5132–5143.

[9] Y. Deng, M. M. Kamani, and M. Mahdavi, "Distributionally robust federated averaging," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates Inc., 2020.

[10] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*. IEEE, 2020, pp. 1698–1707.

[11] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2020.

[12] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates Inc., 2020.

[13] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Project Report, Stanford*, 2009.

[16] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," *CoRR*, vol. abs/1812.01097, 2018.

[17] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *AAAI*. Palo Alto, CA, USA: AAAI, 2020, pp. 6283–6290.

[18] Y. Wang, Y. Tong, D. Shi, and K. Xu, "An efficient approach for cross-silo federated learning to rank," in *ICDE*, 2021, pp. 1128–1139.

[19] Y. Tong, Y. Yuan, Y. Cheng, L. Chen, and G. Wang, "Survey on spatiotemporal crowdsourced data management techniques," *J. Softw.*, vol. 28, no. 1, pp. 35–58, 2017.

[20] B. Du, Y. Tong, Z. Zhou, Q. Tao, and W. Zhou, "Demand-aware charger planning for electric vehicle sharing," in *KDD*, 2018, pp. 1330–1338.

[21] Y. Tong, L. Wang, Z. Zhou, L. Chen, B. Du, and J. Ye, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *SIGMOD*, 2018, pp. 773–788.

[22] J. She, Y. Tong, L. Chen, and T. Song, "Feedback-aware social event-participant arrangement," in *SIGMOD*, 2017, pp. 851–865.

[23] Y. Tong, X. Pan, Y. Zeng, Y. Shi, C. Xue, Z. Zhou, X. Zhang, L. Chen, Y. Xu, K. Xu, and W. Lv, "Hu-fu: Efficient and secure spatial queries over data federation," *PVLDB*, vol. 15, no. 6, pp. 1159–1172, 2022.

[24] Y. Wang, Y. Tong, Z. Zhou, Z. Ren, Y. Xu, G. Wu, and W. Lv, "Fed-ltd: Towards cross-platform ride hailing via federated learning to dispatch," in *SIGKDD*, 2022, pp. 4079–4089.

[25] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[26] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, 2020.

[27] Y. Li, Y. Yuan, Y. Wang, X. Lian, Y. Ma, and G. Wang, "Distributed multimodal path queries," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3196–3210, 2022.

[28] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," *CoRR*, vol. abs/1911.00972, 2019.

[29] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora, "Fetchsgd: Communication-efficient federated learning with sketching," in *ICML*, vol. 119. New York, NY, USA: ACM, 2020, pp. 8253–8265.

[30] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates Inc., 2020.

[31] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates Inc., 2020.

[32] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *ICDE*. IEEE, 2022, pp. 965–978.

[33] J. Blitzer, R. T. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *EMNLP*, D. Jurafsky and É. Gaussier, Eds. Stroudsburg, PA, USA: ACL, 2006, pp. 120–128.

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[35] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *AAAI*, D. Fox and C. P. Gomes, Eds., 2008, pp. 677–682.

[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[37] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *ICCV*. Piscataway, NJ, USA: IEEE Press, 2013, pp. 2200–2207.

[38] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.

[39] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, F. R. Bach and D. M. Blei, Eds., vol. 37. New York, NY, USA: ACM, 2015, pp. 97–105.

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *CVPR*. Piscataway, NJ, USA: IEEE Press, 2017, pp. 2962–2971.

[41] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. New York, NY, USA: ACM, 2018, pp. 1994–2003.

[42] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," in *ICLR*, 2020.

[43] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 308–318.