This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3198176

FedBCD: A Communication-Efficient Collaborative Learning Framework for Distributed Features

Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, *Member, IEEE,* and Qiang Yang, *Fellow, IEEE*

***Abstract*—We introduce a novel federated learning framework allowing multiple parties having different sets of attributes about the same user to jointly build models without exposing their raw data or model parameters. Conventional federated learning approaches are inefficient for cross-silo problems because they require the exchange of messages for gradient updates at every iteration, and raise security concerns over sharing such messages during learning. We propose a *Federated Stochastic Block Coordinate Descent (FedBCD)* algorithm, allowing each party to conduct multiple local updates before each communication to effectively reduce communication overhead. Under a practical security model, we show that parties cannot infer others' exact raw data (*"deep leakage"*) from collections of messages exchanged in our framework, regardless of the number of communication to be performed. Further, we provide convergence guarantees and empirical evaluations on a variety of tasks and datasets, demonstrating significant improvement inefficiency.**

***Index Terms*—Federated Learning, Data Privacy, Federated Stochastic Block Coordinate Descent, cross-silo federated learning, distributed features**

I. Introduction

FEDERATED and collaborative learning has emerged to be an attractive solution to the data silo and privacy problem. While distributed learning (DL) frameworks [1] originally aims at parallelizing computing power and distributes data identically across multiple servers, federated learning (FL) [2], [3] focuses on data locality, non-IID distribution and privacy. In most of the existing federated learning frameworks, data are distributed by samples thus share the same set of attributes. However, a different scenario is cross-organizational federated learning problems where parties share the same users but have different set of features. For example, a local bank and a local retail company in the same city may have large overlap in user base and it is beneficial for these parties to build collaborative learning models with their respective features.

Feature-partitioned collaborative learning problems have been studied in the setting of both DL [4], [5], [6] and FL [7], [8], [9], [10]. However, existing architectures have not sufficiently addressed the communication and privacy problem especially in communication-sensitive scenarios where data are geographically distributed and data locality and privacy are of paramount significance (i.e., in a FL setting). In these approaches, *per-iteration* communication are often required, since the update of algorithm parameters needs contributions from all parties. In sample-partitioned FL[2], it is demonstrated that multiple local updates can be performed with federated averaging (FedAvg), reducing the number of communication round effectively. Whether it is feasible to perform such multiple local update strategy over distributed features is not clear. In addition, recent attacks to FL [11] show that sharing gradients during training processes may leak raw data. Privacy concerns in the distributed-feature settings are yet to be addressed to prevent inference over the messages exchanged.

In this paper, we propose a collaborative learning framework for distributed features named *Federated stochastic block coordinate descent (FedBCD)*, where parties only share an inner product of model parameters and raw data per sample during each communication, and can continuously perform local model updates (in either a parallel or sequential manner) without per-iteration communication. While FedAvg applies to sample-partitioned FL scenarios where complete sets of model parameters are averaged after multiple local updates, FedBCD targeted the feature-partitioned FL scenarios where subsets of model parameters and features perform multiple local gradient updates independently to reduce communication overhead. Therefore FedBCD does not apply to the sample-partitioned FL scenario. In the proposed framework, all raw data and model parameters stay local, and each party does not learn other parties' data or model parameters either before or after the training. There is no loss in performance of the collaborative model as compared to the model trained in a centralized manner. In our paper, we demonstrate experimentally that the communication cost can be significantly reduced by adopting FedBCD. Compared with the existing distributed (stochastic) coordinate descent methods [12], [13], [14], [15], we show for the first time that when the number of local updates, mini-batch size and learning rates are selected appropriately, the FedBCD converges to a $\mathcal{O}(1/\sqrt{T})$ accuracy with $\mathcal{O}(\sqrt{T})$ rounds of communications despite performing multiple local updates using staled information. We further provide security guarantees for exchanging transmitted data under a mild and practical security protocol, removing the hard constraint for data encryption. We show that it is not possible to infer raw data values not only from a single round of communication, but from the **collections of all exchanged messages through the learning process regardless of how many iterations are performed**. Our framework is applicable to parties with arbitrary local sub-models (e.g.neural networks) as long as

Yang Liu and Qiang Yang are the corresponding authors. Email: liuy03@air.tsinghua.edu.cn, qyang@cse.ust.hk. Yang Liu and Xinwei Zhang are co-first authors.

Yang Liu is with Institute for AI Industry Research, Tsinghua University. Yan Kang, Liping Li, and Qiang Yang are with Department of Artificial Intelligence, Webank, Shenzhen, China. Xinwei Zhang and Mingyi are with the University of Minnesota. Qiang Yang and Tianjian Chen are also affiliated with Hong Kong University of Science and Technology.

they connect at a final linear layer (e.g. linear and logistic regression).

II. Related Work

Traditional distributed learning adopts a parameter server architecture [1] to enable a large amount of computation nodes to train a shared model by aggregating locally-computed updates. The issue of privacy in DL framework is considered in [16]. FL [2] adopted a `FedAvg` algorithm which runs Stochastic Gradient Descent (SGD) for multiple local updates in parallel to achieve better communication efficiency. The authors of [17] studied the `FedAvg` algorithm under the parallel restarted SGD framework and analyzed the convergence rate and communication savings under IID settings. In [18], the convergence of the `FedAvg` algorithm under non-IID settings was investigated. All the work above consider the sample-partitioned scenario.

Feature-partitioned learning architectures have been developed for models including trees [9], linear and logistic regression [7], [8], [4], [6], and neural networks [10], [5]. Distributed Coordinate Descent [12] used balanced partitions and decoupled computation at the individual coordinate in each partition; Distributed Block Coordinate Descent [13] assumes feature partitioning is given and performs synchronous block updates of variables which is suited for MapReduce systems with high communication cost settings. These approaches require *synchronization at every iteration*. *Asynchronous* BCD [14] and *Asynchronous* ADMM algorithms [15] tries to tame various kinds of asynchronicity using strategies such as small stepsize, and careful update scheduling, and the design objective is to ensure that the algorithm can still behave reasonably under non-ideal computing environment. [19] propose a dynamic diffusion approach which requires communications between parties with its related neighbours. Our approach tries to address the expensive communication overhead problem in the primal domain in FL scenario by systematically adopting BCD with sufficient number of local updates guided by theoretical convergence guarantees. We assume only one party has labels and all the other parties communicate only with that party to minimize information exchanges in the system.

To prevent data leakage in federated learning, privacy-preserving techniques, such as Homomorphic Encryption (HE) [20], Secure Multi-party Computation (SMPC) [21] are typically applied to transmitted data [7], [10], [9], adding expensive communication overhead to the architectures. Differential Privacy (DP) is also a commonly-adopted approach, but such approaches suffer from precision loss [5], [6]. Hybrid approaches [22], [23] are also proposed to achieve better trade-off between accuracy and security. [24] adds noise to data but requires an anonymization network to cancel the noise. [25] propose a LANN-SVD algorithm implemented in distributed settings but it applies to single-layer neural networks only. We prove that the proposed learning protocol is secure under a practical and heuristic security model for a variety of models, leading to much more efficient solutions and providing a trade off between security and efficiency.

III. Problem Definition

Suppose K data parties collaboratively train a machine learning model based on a dataset with N data samples $\mathcal{D} \triangleq \{\xi_i\}_{i=1}^N$, where the samples consist of the feature and the label $\xi \triangleq (\mathbf{x}, y)$. The feature vector $\mathbf{x}_i \in \mathbb{R}^{1\times d}$ are distributed among K parties $\{\mathbf{x}_{i,k} \in \mathbb{R}^{1\times d_k}\}_{k=1}^K$, where d_k is the feature dimension of party k. We assume one that part K holds the labels of all the data. Let us denote the data set as $\mathcal{D}_k \triangleq \{\mathbf{x}_{i,k}\}_{i=1}^N$, for $k \in [K-1]$, $\mathcal{D}_K \triangleq \{\mathbf{x}_{i,K}, y_{i,K}\}_{i=1}^N$.Then the collaborative training problem can be formulated as

$$\min_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta};\mathcal{D}) \triangleq \frac{1}{N}\sum_{i=1}^N f(\theta_1,\ldots,\theta_K;\xi_i) + \lambda\sum_{k=1}^K \gamma(\theta_k) \quad (1)$$

where $\theta_k \in \mathbb{R}^{d_k}$ denotes the training parameters of the kth party; $\mathbf{\Theta} = [\theta_1;\ldots;\theta_K]$; $f(\cdot)$ and $\gamma(\cdot)$ denotes the loss function and regularizer and λ is the hyperparatemer; For a wide range of models such as linear and logistic regression, and support vector machines, the loss function has the following form:

$$f(\theta_1,\ldots,\theta_K;\xi_i) = f\left(\sum_{k=1}^K \mathbf{x}_{i,k}\theta_k, y_{i,K}\right) \quad (2)$$

The objective is for each party k to find its θ_k without sharing its data $\mathcal{D}_k$ or parameter θ_k to other parties.

IV. The proposed FedBCD Algorithms

If a mini-batch $\mathcal{S} \subset \mathcal{D}$ of S data points is sampled, the stochastic partial gradient w.r.t. θ_k is given by

$$g_k(\mathbf{\Theta};\mathcal{S}) \triangleq \nabla_k f(\mathbf{\Theta};\mathcal{S}) + \lambda\nabla\gamma(\theta_k). \quad (3)$$

Let $H_i^k \triangleq \mathbf{x}_{i,k}\theta_k$ and $H_i \triangleq \sum_{k=1}^K H_i^k$, then for the loss function in equation (2), we have

$$\nabla_k f(\mathbf{\Theta};\mathcal{S}) = \frac{1}{S}\sum_{\xi_i\in\mathcal{S}} \frac{\partial f(H_i, y_{i,K})}{\partial H_i}(\mathbf{x}_{i,k})^T \quad (4)$$

To compute $\nabla_k f(\mathbf{\Theta};\mathcal{S})$ locally, each party $k \in [K-1]$ sends $I_{\mathcal{S}}^{k,K} \triangleq \{H_i^k\}_{i\in\mathcal{S}}$ to party K, who then calculates $I_{\mathcal{S}}^{K,q} \triangleq \{\frac{\partial f(H_i,y_{i,K})}{\partial H_i}\}_{i\in\mathcal{S}}$ and sends to other parties. $I^{q,k}(\cdot)$ is the collection of the information required from party q to k. And finally all parties can compute gradient updates with equation (4).

For an arbitrary loss function, let us define the collection of information needed to compute $\nabla_k f(\mathbf{\Theta};\mathcal{S})$ as

$$I_{\mathcal{S}}^{-k} \triangleq \{I_{\mathcal{S}}^{q,k}\}_{q\neq k}. \quad (5)$$

where the stochastic gradients (3) can be computed as the following:

$$\begin{aligned} g_k(\mathbf{\Theta};\mathcal{S}) &= \nabla_k f(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S}) + \lambda\nabla\gamma(\theta_k) \\ &\triangleq g_k(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S}). \end{aligned} \quad (6)$$

Therefore, the overall stochastic gradient is given as

$$g(\mathbf{\Theta};\mathcal{S}) \triangleq [g_1(I_{\mathcal{S}}^{-1}, \theta_1; \mathcal{S}); \cdots; g_K(I_{\mathcal{S}}^{-K}, \theta_K; \mathcal{S})]. \quad (7)$$

A direct approach to optimize (1) is to use the vanilla stochastic gradient descent (SGD) algorithm given below

$$\theta_k \leftarrow \theta_k - \eta g_k(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S}), \quad \forall\ k, \tag{8}$$

which requires communication of intermediate results *at every iteration*. This could be very inefficient, especially when K is large or the task is communicationally heavy. This vanilla SGD algorithm (termed *FedSGD*) converges with a rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$, regardless of the choice of K [26]. Since each iteration requires one round of communication among all the parties, T rounds of communication is required to achieve an error of $\mathcal{O}(\frac{1}{\sqrt{T}})$. In our proposed *FedBCD*, each party performs Q (with $Q \geq 1$) consecutive local gradient updates before communicating the intermediate results among each other either in parallel (*FedBCD-p*) or sequentially (*FedBCD-s*), see Figure 1(b). Note that when $Q = 1$, FedBCD-p reduces to FedSGD.

Algorithm 1: FedBCD-p: Federated Stochastic Block Coordinate Descent

Input: learning rate η, communication frequency Q
Output: Model parameters θ_1, $\theta_2 ... \theta_K$
Party 1,2..K initialize θ_1, θ_2, ...θ_K.
for *each iteration* $r = 1, 2, \ldots$ **do**
  if $r \bmod \mathrm{Q} = 0$ **then**
    Randomly sample a mini-batch $\mathcal{S} \subset \mathcal{D}$;
    Exchange($\{1, 2...K\}, \mathcal{S}$);
  end
  for *party* $k \in [K]$*, in parallel* **do**
    k computes $g_k(I_{\mathcal{S}}^{-k}, \theta_k^r; \mathcal{S})$ using (6) and updates $\theta_k^{r+1} \leftarrow \theta_k^r - \eta g_k(I_{\mathcal{S}}^{-k}, \theta_k^r; \mathcal{S})$;
  end
end
Exchange(U,$\mathcal{S}$): *# U is the set of party IDs*
  if *equation* (2) *holds* **then**
    each party $k \in U$ and $k \neq K$ in parallel computes and sends $I_{\mathcal{S}}^{k,K}$ to party K;
    party K computes and sends $I_{\mathcal{S}}^{K,k}$ to party $k \in U$;
  else
    each party $k \in U$ in parallel computes and sends $I_{\mathcal{S}}^{k,q}$ to party $q \in U$;
  end

Because $I_{\mathcal{S}}^{-k}$ is the intermediate information obtained from the *most recent synchronization*, $g_k(I_{\mathcal{S}}^{-k}, \theta_k; \mathcal{S})$ may contain *staled* information so it may no longer be an unbiased estimate of the true partial gradient $\nabla_k \mathcal{L}(\mathbf{\Theta})$. On the other hand, during the Q local updates no inter-party communication is required. Therefore, one could expect that there will be some interesting trade-off between communication efficiency and computational efficiency. These trade-offs will be analyzed in our theoretical results, and illustrated in our numerical experiments.

V. Convergence Analysis

In this section, we perform convergence analysis of the FedBCD algorithm. Our analysis will be focused on Algorithm 1 and the sequential version can be analyzed use similar techniques. Let r denote the iteration index, in which each iteration one round of local update is performed; Let r_0 denote the latest iteration before r in which synchronization has been performed, and the intermediate information $I_{\mathcal{S}}^{-k}$'s are exchanged. Let $\mathbf{y}_k^r$ denote the "local vector" that node k uses to compute its local gradient at iteration r, that is

$$g_k(\mathbf{y}_k^r; \mathcal{S}) = g_k([\mathbf{\Theta}_{-k}^{r_0}, \theta_k^r]; \mathcal{S}]) \tag{9}$$

where $[\mathbf{v}_{-k}, w]$ denotes a vector $\mathbf{v}$ with its kth element replaced by w. Note that by Algorithm 1, each node k always updates the kth element of $\mathbf{y}_k^r$, while the information about $\mathbf{\Theta}_{-k}^{r_0}$ is obtained by the most recent synchronization step. Further, we use the "global" variable $\mathbf{\Theta}^r$ to collect the most updated parameters at each iteration of each node, where $\mathbf{y}_{k,j}$ denotes the jth element of $\mathbf{y}_k$:

$$\mathbf{\Theta}^r = [\theta_1^r; \ldots; \theta_K^r] \triangleq [\mathbf{y}_{1,1}^r; \ldots; \mathbf{y}_{K,K}^r]. \tag{10}$$

Note that $\{\mathbf{\Theta}^r\}$ is only a sequence of "virtual" variables, it is never explicitly formed in the algorithm.

We make the following assumptions to the problem and the algorithm.

A1: Uniform Sampling. We assume that $\mathcal{S}_l$ is a mini-batch of size S, and each sample in $\mathcal{S}_l$ is sampled i.i.d from $\mathcal{D}$ for all ℓ.

A2: Bounded Variance. Assume that the variance of the stochastic gradient satisfies the following:

$$\mathbb{E}_\xi \left\| g_k(\mathbf{\Theta}; \xi) - \nabla_k \mathcal{L}(\mathbf{\Theta}) \right\|^2 \leq \sigma^2, \forall\ \mathbf{\Theta}.$$

A3: Lipschitz Gradient. Assume that the loss function satisfies the following:

$$\|\nabla \mathcal{L}(\mathbf{\Theta}_1) - \nabla \mathcal{L}(\mathbf{\Theta}_2)\| \leq L \left\| \mathbf{\Theta}_1 - \mathbf{\Theta}_2 \right\|, \forall\ \mathbf{\Theta}_1, \mathbf{\Theta}_2.$$
$$\mathbb{E}_\xi \left[\| g_k(\mathbf{\Theta}_1; \xi) - g_k(\mathbf{\Theta}_2; \xi) \| \right] \leq L_k \left\| \mathbf{\Theta}_1 - \mathbf{\Theta}_2 \right\|, \forall\ \mathbf{\Theta}_1, \mathbf{\Theta}_2.$$

Note that the second assumption of A3, which we refer to as the averaged gradient Lipschitz smooth condition, is stronger than directly assuming Lipschitz smoothness, but it is still a rather standard assumption in SGD analysis; see, e.g., [27], [28].

Based on the above assumption, we have the following convergence result for Algorithm 1.

Theorem 1: Suppose assumptions A1-A3 hold. Running Algorithm 1 for T iterations, with the stepsize parameter chosen as $\eta \leq \frac{1}{L + 2Q^2 \sum_{k=1}^K L_k^2 / L}$, $Q \geq \sqrt{S/K}$, we have:

$$\begin{aligned} \frac{1}{T} \sum_{r=0}^{T-1} \mathbb{E} \left\| \nabla \mathcal{L}(\mathbf{\Theta}^r) \right\|^2 &\leq \frac{2}{\eta T} \mathbb{E}[\mathcal{L}(\mathbf{\Theta}^0) - \mathcal{L}(\mathbf{\Theta}^T)] \\ &+ \left(\frac{2\eta K C_1 + 2\eta^2 Q^2 K C_2}{S} \right) \cdot \sigma^2, \end{aligned} \tag{11}$$

where $C_1 = 6 + 72L^2 + 120L^4$, $C_2 = C_1 \cdot (\sum_{k=1}^K L_k^2 + \max_k\{L_k^2\})$ are two positive constants.

Remark 1. It is non-trivial to find an unbiased estimator for the local stochastic gradient $g_k(\mathbf{y}_k^r; \mathcal{S})$ because after each synchronization step, each agent k performs Q *deterministic steps* based on the same data set $\mathcal{S}$ while fixing all the rest of the variable blocks at $\mathbf{\Theta}_{-k}^{r_0}$. This is significantly different from FedAvg-type algorithms, where at each iteration a new mini-batch is sampled at each node. The proof for Theorem 1 is provided in Appendix A. ■

Remark 2. Let us discuss how to choose T, η, and Q so that ϵ accuracy is achieved, such that the following holds:

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\right\|^2 = \mathcal{O}(\epsilon).$$

First, it is clear that the following choices are valid:

$$T \geq \frac{4(\mathcal{L}(\mathbf{\Theta}^0) - \underline{\mathcal{L}})}{\eta\epsilon},\ \eta \leq \frac{\epsilon S}{8KC_1},\ Q \leq \sqrt{\frac{\epsilon S}{8K\eta^2 C_2}},$$

where $\underline{\mathcal{L}} = \inf_{\mathbf{\Theta}}\{\mathcal{L}(\mathbf{\Theta})\}$ denotes the lower bound of $\mathcal{L}(\cdot)$. Then, to understand the precise relation between the communication and computation, let us fix the stepsize as $\eta = \frac{\epsilon S}{8KC_1}$. Then the total number of required local updates T and the local communication rounds Q can be chosen as

$$T = \frac{32KC_1(\mathcal{L}(\mathbf{\Theta}^0) - \underline{\mathcal{L}})}{S\epsilon^2},\ Q = \sqrt{\frac{8KC_1^2}{\epsilon S C_2}}.$$

Therefore, the total number of communication required is

$$\frac{T}{Q} = \mathcal{O}\left(\frac{K^{1/2}}{S^{1/2}\epsilon^{3/2}}\right).$$ ■

Remark 3. The previous remark indicates that with any fixed S and K, we can choose $\eta = \mathcal{O}(\epsilon), Q = \frac{1}{\epsilon^{1/2}}$; it also shows that the convergence speed of the algorithm is $\mathcal{O}(\frac{1}{\epsilon^2})$ in terms of the total number of local updates, and $\mathcal{O}(\frac{1}{\epsilon^{3/2}})$ in terms of the total number of communication rounds. To the best of our knowledge, it is the first time that such rates have been proven for any algorithms with multiple local steps designed for the feature-partitioned federated learning problem. ■

Remark 4. Compared with the existing distributed stochastic coordinate descent methods [12], [13], [14], [15] that requires $\mathcal{O}(1/\epsilon^2)$ communication/computation update to achieve $\mathcal{O}(\epsilon)$ accuracy, our results are different. It shows that, despite using stochastic gradients and performing multiple local updates using staled information, only $\mathcal{O}(1/\epsilon^{3/2})$ communication rounds are requires (out of total for $\mathcal{O}(1/\epsilon^2)$ iterations) to achieves $\mathcal{O}(\epsilon)$ accuracy. Compare with vanilla BCD, FedBCD saves communication by having multiple local updates. ■

Remark 5. If we consider the impact of the number of nodes K and the batch size S, then from Remark 1 we have $T = \mathcal{O}(\frac{K}{\epsilon^2 S})$ and $\frac{T}{Q} = \frac{K^{1/2}}{\epsilon^{3/2}S^{1/2}}$. This indicates that the proposed algorithm has a slow down w.r.t the number of parties involved and a speed up w.r.t the batch size. In practice, the factor of K is mild assuming that the total number of parties involved is usually not large and we can always pick larger batch size $S > K$ to cancel the impact of K. ■

VI. Security Analysis

Here we aim to find out whether one party can learn other party's data ($\mathbf{x}_i^k$) from collections of messages exchanged ($G^{k,q}(I_S^k)$) during training. Whereas previous research studied data leakage from exposing complete set of model parameters or gradients, of dimension d_k [11], [29], [30], in our protocol model parameters are kept private, and only the intermediate results (such as inner product of model parameters and feature), which is of reduced dimension, 1 in the case of the linear model, are exposed. Thus, the gradients exchanged from party K to others are also the gradients with respect to this intermediate message, of reduced dimension, not the model parameters themselves. Therefore the previous leakage attack do not apply in our scenario.

In the following discussion, we assume that we use the ℓ_2-norm square regularizer in (1) $\gamma(\theta_k) = \frac{1}{2}\left\|\theta_k\right\|^2$.

Security Definition Let $\mathcal{S}_r$ be the set of data point sampled at the rth iteration and i_r denotes the ith sample of the rth iteration. $H_{i_r}^k$ is the contribution of the ith sample from the kth party to other parties. At the $(r+1)$th iteration, we update weight variables according to equation (4)

$$\theta_k^{r+1} = \theta_k^r - \eta^r\left(\frac{1}{S}\sum_{\xi_{i_r}\in\mathcal{S}_r} g(H_{i_r}, y_{i_r,K})(\mathbf{x}_{i_r,k})^T + \lambda\theta_k^r\right) \tag{12}$$

The security definition is that for any party k with undisclosed dataset $\mathcal{D}_k$ and training parameters θ_k following FedBCD, there exists infinite solutions for $\{\mathbf{x}_{i_r,k}\}_{i_r\in\mathcal{S}_{r,r=0,\ldots,T}}$ that yield the same set of contributions $\{H_{i_r}^k\}_{r=0,\ldots,T}$. That is, *one can not determine party k's data $\mathbf{x}_{i,k}$ uniquely from its exchanged messages of $\{H_r^k\}_{r=0,\cdots,T}$ regardless the value of T, the total number of iterations.*

With only one iteration ($T = 1$), such a security definition is inline with prior security definitions proposed in privacy-preserving machine learning and secure multiple computation (SMC), such as [31], [32], [33], [7], [34]. Here the assumption is that no prior information about other parties is available, so it is impossible to infer the exact raw data $\mathbf{x}_{i,k}$. Note under this heuristic security definition, when some prior knowledge about the data is known, an adversary may be able to eliminate some alternative solutions or certain derived statistical information may be revealed [32], [33]. However to infer the exact raw data $\mathbf{x}_{i,k}$ one needs additional prior information about the data, which is not always available. In our work, we assume zero knowledge about other parties. Our main focus is to propose a general framework for performing much efficient feature-partitioned collaborative learning with lossless accuracy based on a practical and heuristic security model, which tradeoffs between privacy and efficiency and allows much more efficient solutions.

Over multiple iterations, the observations by other parties *are iterative outputs from FedBCD algorithm and are all correlated based on equation (12).* That is, parties obtain $T-1$ times more information. Although it is easy to show security of $\mathbf{x}_{i,k}$ by sending only one round of $H_{i_r}^k$ due to the reduced dimensionality, it is unclear whether raw data will be leaked after thousands or millions of rounds of iterative communications. To our best knowledge, we are the first to provide proof for the multiple-iteration scenario (see Proof of Theorem 2 in Appendix).

Theorem 2: For K-party collaborative learning framework following (2) with $K \geq 2$, the FedBCD Algorithm is secured for party k if k's feature dimension is greater than 1, i.e., $d_k \geq 2$.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3198176

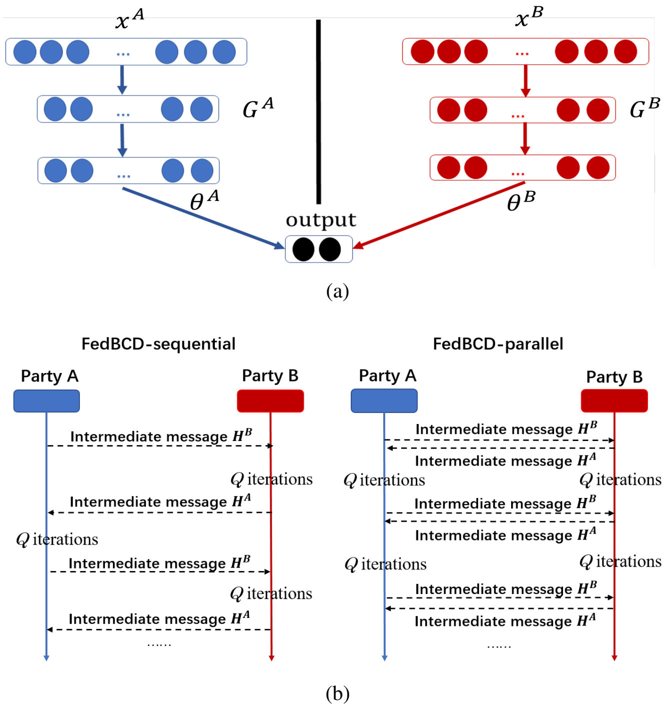


Fig. 1. Illustration of a 2-party collaborative learning framework (a) with neural network(NN)-based local model. (b) *FedBCD-s* and *FedBCD-p* algorithms

The security proof can be readily extended to collaborative systems where parties have arbitrary local sub-models (such as neural networks) but connect at the final prediction layer with loss function (2) (see Figure 1(a)). Let G^k be the local transformation on $\mathbf{x}_{i,k}$ and is unknown to parties other than k. We choose G^k to be the identity map, i.e. $G^k(\mathbf{x}_{i,k}) = \mathbf{x}_{i,k}$, then the problem reduces to Theorem 2.

VII. Experiments

A. Datasets and Models

MIMIC-III. We compile a subset of the MIMIC-III [35] database containing more than 31 million clinical events that correspond to 17 clinical variables and get the final training and test sets of 17,903 and 3,236 ICU stays, respectively. For each variable we compute six different sample statistic features on seven different subsequences of a given time series, obtaining $17\times7\times6 = 714$ features. We focus on the in-hospital mortality prediction task based on the first 48 hours of an ICU stay We partition each sample vertically by its clinical features. In a practical situation, clinical variables may come from different hospitals or different departments in the same hospital and can not share their data due to the patients personal privacy. This task is referred to as MIMIC-LR.

NUS-WIDE. The NUS-WIDE dataset [36] consists of 634 low-level images features extracted from Flickr images as well as their associated tags and ground truth labels. We assign image features to one party and textual tag features to another party. The objective is to perform a federated transfer learning (FTL) task studied in [10]. Each party utilizes a neural network having one hidden layer with 64 units to learn feature representation from their raw inputs. Then, the feature representations of both sides are fed into a final federated layer. This task is refered to as NUS-FTL.

MNIST. We partition each MNIST [37] image with shape $28 \times 28 \times 1$ vertically into two parts (each part has shape $28 \times 14 \times 1$). Each party uses a local CNN sub-model (two 3x3 convolution layers with 64 channels, followed by a fully connected layer with 256 units) to learn feature representation, which then are fed into a logistic regression layer with 512 parameters for a binary classification task. We refer this task as MNIST-CNN.

Default-Credit. We partition the features into 15 demographic features and 18 payment features, which often happens when banks leverage alternative data for user credit risk prediction. We perform a FTL task as described above but with homomorphic encryption applied. We refer to this task as Credit-FTL.

For all experiments, we adopt a decay learning rate strategy with $\eta^r = \frac{\eta^0}{\sqrt{r+1}}$, where η^0 is optimized for each experiment. We fix the batch size to 64 and 256 for MIMIC-LR and MNIST-CNN respectively. Note although not considered in our experiments, in real-world settings, some features may be overlapping or distributed feature selection [38], [39] may need to be performed.

B. Evaluation Metric

For all dataset, we consider the training loss (loss for short) $\mathcal{L}(\mathbf{\Theta}; \mathcal{D})$ and Area Under Curve (AUC) as the performance metrics. The training loss is defined by (1) and evaluated using training dataset. AUC is the area under the receiver operating characteristics (ROC) curve, which represents the relationship between false-positive rate and true-positive rate for different probability thresholds of model predictions. This area is bounded to 1. The perfect AUC score is 1 and the worst is 0, meaning the model gives wrong prediction for every sample. AUC is a preferred metric especially in classification problems with imbalanced samples. Our objective is to minimize the training loss as shown in (1) and maximize the AUC. As loss is minimized during training, the performance of the model, indicated by AUC, is also maximized. In experiments, we demonstrate the loss and AUC as a function of communication rounds during training to compare the convergence rate of different algorithms.

C. Results and Discussion

FedBCD-p vs FedBCD-s. We first study the impact of varying local iterations on the communication efficiency of both FedBCD-p and FedBCD-s algorithms based on MIMIC-LR and MNIST-CNN (Figure 2). We observe similar convergence for FedBCD-s and FedBCD-p for various values of Q. However, for the same communication round, the running time of FedBCD-s doubles that of FedBCD-p due to sequential execution. As the number of local iteration increases, we observe that the required number of communication rounds reduce dramatically (Table I). Therefore, by reasonably increasing the number of local iteration, we can take advantage of the parallelism on participants and save the overall communication costs by reducing the number of total communication rounds required.

This article has been accepted for publication in IEEE Transactions on Signal Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSP.2022.3198176

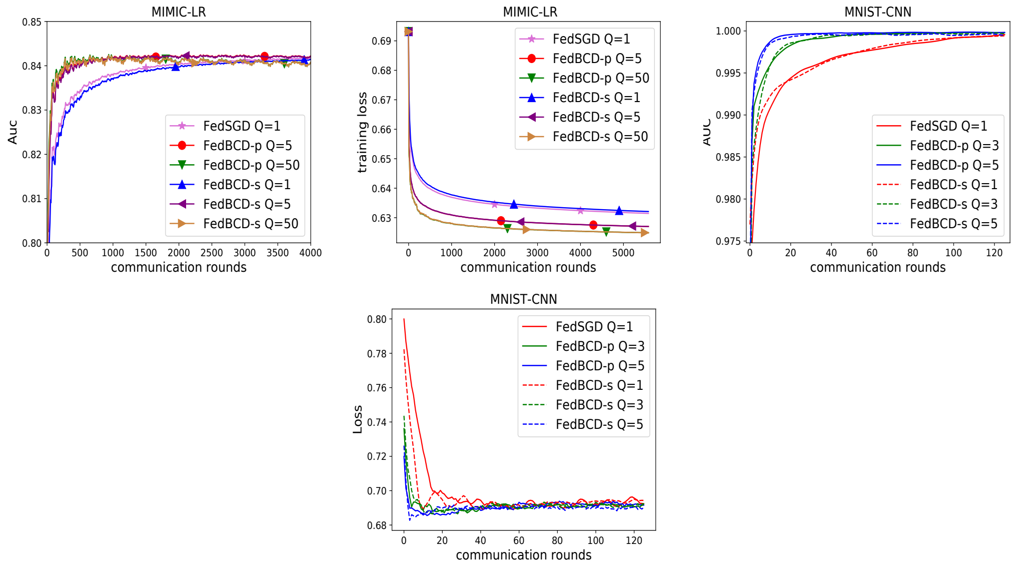


Fig. 2. Comparison of AUC and training loss in MIMIC-LR, MNIST-CNN, NUS-FTL with varying Q local iterations.

Impact of Q. Theorem 1 suggests that as Q grows the required number of communication rounds may first decrease and then increase again, and eventually the algorithm may not converge to optimal solution. To further investigate the relationship between the convergence rate and the local iteration Q, we evaluate FedBCD-p algorithm on NUS-FTL with a large range of Q. The results are shown in Figure 2 and Figure 3(a), which illustrate that FedBCD-p achieves the best AUC with the least number of communication rounds when $Q = 15$. For each target AUC, there exists an optimal Q. This manifests that one needs to carefully select Q to achieve the best communication efficiency, as suggested by Theorem 1.

Figure 3(b) shows that for very large local iteration $Q = 25, 50$ and 100, the FedBCD-p cannot converge to the AUC of 83.7%. This phenomenon is also supported by Theorem 1, where if Q is too large the right hand side of (11) may not go to zero. Next we further address this issue by making the algorithm less sensitive in choosing Q.

Proximal Gradient Descent. [40] proposed adding a proximal term to the local objective function to alleviate potential divergence when local iteration is large. Here, we explore this idea to our scenario. We rewrite (9) as follows:

$$g_k(\mathbf{y}_k^r;\xi_i) = g_k([\mathbf{\Theta}_{-k}^{r_0}, \theta_k^r];\xi_i]) + \mu(\theta_k^r - \theta_k^{r_0}) \qquad (13)$$

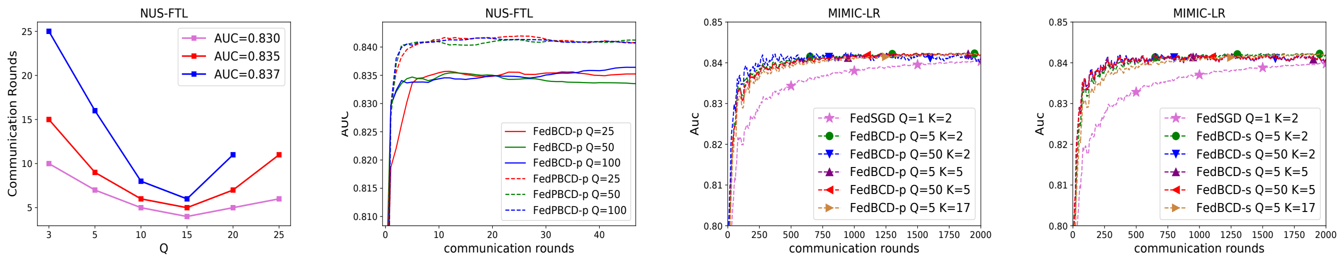
where $\mu(\theta_k^r - \theta_k^{r_0})$ is the gradient of the proximal term $\frac{\mu}{2}||\theta_k^r - \theta_k^{r_0}||^2$, which exploits the initial model $\theta_k^{r_0}$ of party k to limit the impact of local updates by restricting the locally updated model to be close to $\theta_k^{r_0}$. We denote the proximal version of FedBCD-p as FedPBCD-p. We then apply FedPBCD-p with $\mu = 0.1$ to NUS-FTL for $Q = 25$, 50 and 100 respectively. Figure 3(b) illustrates that if Q is too large, FedBCD-p fails to converge to optimal solutions whereas the FedPBCD-p converges faster and is able to reach at a higher test AUC than FedBCD-p.

Increasing number of Parties. In this section, we increase the number of parties to five and seventeen and conduct experiments for MIMIC-LR task. We partition data by clinical variables with each party having all the related features of the same variable. We adopt a decay learning rate strategy with $\frac{\eta^0}{\sqrt{(r+1)K}}$ according to Theorem 1. The results are shown in Figure 3(c) and 3(d). We can see that the proposed method still performs well when we increase the local iterations for multiple parties. As we increase the number of parties to five and seventeen, FedBCD-p is slightly slower than the two-party case, but the impact of node K is very mild, which verifies the theoretical analysis in Remark 3.

Implementation with HE. In this section, we investigate the efficiency of FedBCD-p algorithm with homomorphic encryption (HE) applied. Using HE to protect transmitted information ensures higher security but it is extremely computationally expensive to perform computations on encrypted data. In such a scenario, carefully selecting Q may reduce communication rounds but may also introduce computational overhead because the total number of local iterations may increase ($Q\times$ number of communication rounds). We integrated the FedBCD-p algorithm into the current FTL implementation on FATE [41] and simulate two-party learning on two machines with Intel Xeon Gold model with 20 cores, 80G memory and 1T hard disk. The experimental results are summarized in Table II. It shows that FedBCD-p with larger Q costs less communication rounds and total training time to reach a specific AUC with a mild increase in computation time but more than 70 percents reduction in communication round from FedSGD to $Q = 10$.

VIII. Conclusions and Future Work

In this paper, we propose a federated learning framework for distributed features, in which parties perform more than one local update of gradients before communication. We provide proof that exact raw data are not exposed in such protocol and the relaxed privacy constraint leads to much more efficient solutions. Our approach significantly reduces the number of communication rounds and the total communication overhead. We theoretically prove that the algorithm achieves global



(a) (b) (c) (d)

Fig. 3. (a) Communication round vs Q. (b) Comparison between FedBCD-p and FedPBCD-p for large local iterations. Comparison of AUC in MIMIC-III dataset with varying Q and number of parties K. (c) FedBCD-p; (d) FedBCD-s.

	MIMIC-LR AUC 84%		MNIST-CNN AUC 99.7%	
Algo.	Q	rounds	Q	rounds
FedSGD	1	334	1	46
FedBCD	5	71	3	16
	50	52	5	8
FedBCD-s	1	407	1	48
	5	74	3	15
	50	52	5	9

TABLE I

NUMBER OF COMMUNICATION ROUNDS TO REACH A TARGET AUC FOR FEDBCD-P, FEDBCD-S AND FEDSGD ON MIMIC-LR AND MNIST-CNN RESPECTIVELY.

Credit-FTL						
AUC	Algo.	Q	R	comp.	comm.	total
70%	FedSGD	1	17	11.33	11.34	22.67
	FedBCD	5	4	13.40	2.94	16.34
		10	2	10.87	2.74	13.61
75%	FedSGD	1	30	20.50	20.10	40.60
	FedBCD	5	8	26.78	5.57	32.35
		10	4	23.73	2.93	26.66
80%	FedSGD	1	46	32.20	30.69	62.89
	FedBCD	5	13	43.52	9.05	52.57
		10	7	41.53	5.12	46.65

TABLE II

NUMBER OF COMMUNICATION ROUNDS, COMPUTATION, COMMUNICATION AND TOTAL TRAINING TIME (MINS) TO REACH TARGET AUC FOR FEDSGD VERSUS FEDBCD-P.

convergence with a decay learning rate and proper choice of local updates. The approach is supported by our extensive experimental evaluations. In the future, we plan to investigate ways to further improve communication efficiency of such approaches for more complex and asynchronized collaborative systems.

APPENDIX A
CONVERGENCE ANALYSIS

In this section, we provide the proof of the convergence result Theorem 1.

Before we begin the proof, we find the following relations useful:

$$\begin{aligned} &\|a+b\|^2 = \|a-c+c-b\|^2 \\ &\le (1+\alpha)\|a-c\|^2 + (1+\frac{1}{\alpha})\|c-b\|^2, \forall \alpha > 0. \end{aligned} \tag{14}$$

For notation simplicity, let us define the stacked stochastic gradient as iteration r as:

$$\mathbf{G}^r \triangleq [g_1(I_{\mathcal{S}_l}^{-1}, \theta_1^r; \mathcal{S}_l); \cdots; g_K(I_{\mathcal{S}_l}^{-K}, \theta_K^r; \mathcal{S}_l)]. \tag{15}$$

Proof of Theorem 1: First apply Lipschitz condition of $\mathcal{L}$, we have:

$$\begin{aligned} &\mathcal{L}(\mathbf{\Theta}^{r+1}) - \mathcal{L}(\mathbf{\Theta}^r) \le \langle \nabla\mathcal{L}(\mathbf{\Theta}^r), \mathbf{\Theta}^{r+1} - \mathbf{\Theta}^r \rangle \\ &\quad + \frac{L}{2}\|\mathbf{\Theta}^{r+1} - \mathbf{\Theta}^r\|^2 \\ &\overset{(a)}{=} -\eta \langle \nabla\mathcal{L}(\mathbf{\Theta}^r), \mathbf{G}^r \rangle + \frac{L\eta^2}{2}\|\mathbf{G}^r\|^2 \\ &\overset{(b)}{=} -\frac{\eta}{2}\left(\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 + \|\mathbf{G}^r\|^2 - \|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbf{G}^r\|^2\right) \\ &\quad + \frac{L\eta^2}{2}\|\mathbf{G}^r\|^2 \\ &= -\frac{\eta}{2}\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 - \frac{\eta}{2}(1-\eta L)\|\mathbf{G}^r\|^2 + \frac{\eta}{2}\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbf{G}^r\|^2, \end{aligned} \tag{16}$$

where (a) applies the update rule of Algorithm 1; (b) use the fact that $\langle a, b\rangle = \frac{1}{2}\left(\|a\|^2 + \|a\|^2 - \|a-b\|^2\right)$. For simplicity, we use $\mathbb{E}^{r_0}$ to denote the expectation conditioned on all the past histories of the algorithm up to iteration r^0. Taking expectation, we have:

$$\begin{aligned} &\mathbb{E}^{r_0}[\mathcal{L}(\mathbf{\Theta}^{r+1}) - \mathcal{L}(\mathbf{\Theta}^r)] \le -\frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 \\ &\quad - \frac{\eta}{2}(1-\eta L)\mathbb{E}^{r_0}\|\mathbf{G}^r\|^2 + \frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbf{G}^r\|^2 \\ &\overset{(a)}{=} -\frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 + \frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbf{G}^r\|^2 \\ &\quad - \frac{\eta}{2}(1-\eta L)(\|\mathbb{E}^{r_0}\mathbf{G}^r\|^2 + \mathbb{E}^{r_0}\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2) \\ &\overset{(b)}{\le} -\frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 \\ &\quad - \frac{\eta}{2}(1-\eta L)(\|\mathbb{E}^{r_0}\mathbf{G}^r\|^2 + \mathbb{E}^{r_0}\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2) \\ &\quad + \frac{\eta}{2}\left((1+\frac{1}{\eta L})\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbb{E}^{r_0}\mathbf{G}^r\|^2\right. \\ &\quad \left. +(1+\eta L)\mathbb{E}^{r_0}\|\mathbb{E}^{r_0}\mathbf{G}^r - \mathbf{G}^r\|^2\right) \\ &= -\frac{\eta}{2}\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 - \frac{\eta}{2}(1-\eta L)\|\mathbb{E}^{r_0}\mathbf{G}^r\|^2 \\ &\quad + \eta^2 L \underbrace{\mathbb{E}^{r_0}\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2}_{\text{Term 1}} \\ &\quad + \frac{1+\eta L}{2L}\underbrace{\mathbb{E}^{r_0}\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbb{E}^{r_0}\mathbf{G}^r\|^2}_{\text{Term 2}}, \end{aligned} \tag{17}$$

where (a) uses the fact that $\mathbb{E}(X)^2 = \mathbb{E}(X^2) + \mathbb{E}(X - \mathbb{E}(X))^2$; (b) uses (14) with $\alpha = \eta L$. Next, we bound Term 1 and Term 2 in the above inequality separately.

A. Bound of Term 1

1) Let us first bound $\mathbb{E}^{r_0}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2]$.

First, we denote the variables updated using minibatch $\mathcal{S}_l$ as $\mathbf{\Theta}^r(\mathcal{S}_l)$, using sample ξ as $\mathbf{\Theta}^r(\xi)$ starting from $\mathbf{\Theta}^{r_0}$. That is, we have the following update rules:

$$\begin{aligned}\mathbf{\Theta}_k^{r_0+1}(\mathcal{S}_l) &\triangleq \mathbf{\Theta}_k^{r_0} - \eta g_k(\mathbf{\Theta}^{r_0};\mathcal{S}_l), \\ \mathbf{\Theta}_k^{r_0+\tau}(\mathcal{S}_l) &\triangleq \mathbf{\Theta}_k^{r_0+\tau-1}(\mathcal{S}_l) - \eta g_k(\mathbf{y}_k^{r_0+\tau-1}(\mathcal{S}_l);\mathcal{S}_l),\end{aligned} \tag{18}$$

where $\mathbf{y}_k^r(\mathcal{S}_l) \triangleq [\mathbf{\Theta}_{-k}^{r_0}, \theta_k^r(\mathcal{S}_l)]$ is the model used for updating the parameters of party k, and

$$\begin{aligned}\mathbf{\Theta}_k^{r_0+1}(\xi) &\triangleq \mathbf{\Theta}_k^{r_0} - \eta g_k(\mathbf{\Theta}^{r_0};\xi), \\ \mathbf{\Theta}_k^{r_0+\tau}(\xi) &\triangleq \mathbf{\Theta}_k^{r_0+\tau-1}(\xi) - \eta g_k(\mathbf{y}_k^{r_0+\tau-1}(\xi);\xi),\end{aligned} \tag{19}$$

where $\mathbf{y}_k^r(\xi) \triangleq [\mathbf{\Theta}_{-k}^{r_0}, \theta_k^r(\xi)]$. Additionally, we have $\mathbf{y}_k^{r_0} = \mathbf{\Theta}^{r_0}, \forall k \in [K]$. Further let us define $r \triangleq r_0 + \tau$.

Using the above notations, we can rewrite $\mathbb{E}^{r_0}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2]$ as:

$$\begin{aligned}&\mathbb{E}^{r_0}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\mathbf{G}^r\|^2] \\ &= \sum_{k=1}^K \mathbb{E}_{\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\mathcal{S}_l) - \mathbb{E}_{\mathcal{S}_l} g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\mathcal{S}_l)\right\|^2\right] \\ &\overset{(a)}{\leq} \sum_{k=1}^K \underbrace{\mathbb{E}_{\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\mathcal{S}_l) - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right\|^2\right]}_{\triangleq A_{\tau,k}}\end{aligned} \tag{20}$$

where (a) uses the fact that $\mathbb{E}(X-\mathbb{E}(X))^2 \leq \mathbb{E}(X-Y)^2$ for all constant Y. Then we can bound $A_{\tau,k}$ by the following terms $B_{\tau,k}, \sigma_{\tau,k}^2$:

$$\begin{aligned}&A_{\tau,k} \\ &\overset{(14)}{\leq} 2\,\mathbb{E}_{\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\mathcal{S}_l) - \mathbb{E}_{\xi'\in\mathcal{S}_l} g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi')\right\|^2\right] \\ &\quad + 2\,\mathbb{E}_{\mathcal{S}_l}\left[\left\|\mathbb{E}_{\xi'\in\mathcal{S}_l} g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right\|^2\right] \\ &\overset{(a)}{\leq} 2\underbrace{\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\xi') - g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi')\right\|^2\right]}_{\triangleq B_{\tau,k}} \\ &\quad + \frac{2}{S^2}\mathbb{E}_{\mathcal{S}_l}\left[\left\|\sum_{\xi'\in\mathcal{S}_l} g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right\|^2\right] \\ &= 2B_{\tau,k} \\ &\quad + \frac{2}{S^2}\mathbb{E}_{\mathcal{S}_l}\sum_{\xi'\in\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right\|^2\right] \\ &\quad + \frac{2}{S^2}\mathbb{E}_{\mathcal{S}_l}\sum_{\xi'\neq\xi''\in\mathcal{S}_l}\Big[\Big\langle g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi), \\ &\qquad g_k(\mathbf{y}_k^{r_0+\tau}(\xi'');\xi'') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\Big\rangle\Big] \\ &\overset{(b)}{=} 2B_{\tau,k} \\ &\quad + \frac{2}{S}\underbrace{\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right\|^2\right]}_{\triangleq \sigma_{\tau,k}^2},\end{aligned} \tag{21}$$

where in (a) we use the fact that $g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\mathcal{S}_l) = \mathbb{E}_{\xi'\in\mathcal{S}_l} g_k(\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l);\xi')$ and apply Jensen's inequality to the first term and break the expectation; in (b) we use assumption A1 that $\xi \in \mathcal{S}_l$ are i.i.d sampled from $\mathcal{D}$ so that the last terms are all zero. We then bound $B_{\tau,k}$ and $\sigma_{\tau,k}^2$ separately by recursion. First, the term $B_{\tau,k}$ can be bounded as below:

$$\begin{aligned}B_{\tau,k} &\overset{(a)}{\leq} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l}\left[\left\|\mathbf{y}_k^{r_0+\tau}(\mathcal{S}_l) - \mathbf{y}_k^{r_0+\tau}(\xi')\right\|^2\right] \\ &\overset{(b)}{=} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l}\Bigg[\Bigg\|\mathbf{\Theta}^{r_0} - \eta\sum_{\tau_1=0}^{\tau-1} g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l) \\ &\qquad -\mathbf{\Theta}^{r_0} + \eta\sum_{\tau_1=0}^{\tau-1} g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi');\xi')\Bigg\|^2\Bigg] \\ &\overset{(c)}{\leq} \eta^2 L_k^2\tau\sum_{\tau_1=0}^{\tau-1}\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l} \\ &\qquad \left[\left\|g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l) - g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi');\xi')\right\|^2\right] \\ &\overset{(14)}{\leq} 2\eta^2 L_k^2\tau\sum_{\tau_1=0}^{\tau-1}\mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l} \\ &\qquad \left[\left\|g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l) - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi);\xi)\right\|^2\right. \\ &\qquad \left.+\left\|\mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi);\xi) - g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi');\xi')\right\|^2\right] \\ &= 2\eta^2 L_k^2\tau\sum_{\tau_1=0}^{\tau-1}(A_{\tau_1,k} + \sigma_{\tau_1,k}^2),\end{aligned} \tag{22}$$

where (a) applies block Lipschitz assumption A3.2; in (b) we expand the updates to $\mathbf{\Theta}^{r_0}$; (c) applies Cauchy-–Schwarz inequality.

We then bound $\sigma_{\tau,k}^2$. First, note that when $\tau = 0$, we have

$$\sigma_{0,k}^2 = \mathbb{E}_{\mathcal{S}_l}\mathbb{E}_{\xi'\in\mathcal{S}_l}\left[\|g_k(\mathbf{y}_k^{r_0};\xi') - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0};\xi)\|^2\right] \overset{\text{A2}}{\leq} \sigma^2 \tag{23}$$

For the general case when $\tau \geq 1$, we have:

$$\begin{aligned}\sigma_{\tau,k}^2 &= \mathbb{E}_{\xi'\in\mathcal{D}}\Big[\Big\|g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi') \\ &\quad + g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi') - \mathbb{E}_{\xi\in\mathcal{D}}[g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi)] \\ &\quad + \mathbb{E}_{\xi\in\mathcal{D}}[g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi)] - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\Big\|^2\Big] \\ &\overset{(14)}{\leq} 3\,\mathbb{E}_{\xi'\in\mathcal{D}}\left[\left\|g_k(\mathbf{y}_k^{r_0+\tau}(\xi');\xi') - g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi')\right\|^2\right] \\ &\quad + 3\,\mathbb{E}_{\xi'\in\mathcal{D}}\Big[\Big\|g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi') \\ &\qquad - \mathbb{E}_{\xi\in\mathcal{D}}[g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi)]\Big\|^2\Big] \\ &\quad + 3\,\mathbb{E}_{\xi'\in\mathcal{D}}\Big[\Big\|\mathbb{E}_{\xi\in\mathcal{D}}[g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi)] \\ &\qquad - \mathbb{E}_{\xi\in\mathcal{D}} g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\Big\|^2\Big] \\ &\overset{(a)}{\leq} 3\,\mathbb{E}_{\xi'\in\mathcal{D}} L_k^2\left[\left\|\mathbf{y}_k^{r_0+\tau}(\xi') - \mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')]\right\|^2\right] + 3\sigma^2 \\ &\quad + 3\,\mathbb{E}_{\xi'\in\mathcal{D}}\Big[\Big\|\mathbb{E}_{\xi\in\mathcal{D}}\left[g_k(\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')];\xi)\right. \\ &\qquad \left.-g_k(\mathbf{y}_k^{r_0+\tau}(\xi);\xi)\right]\Big\|^2\Big] \\ &\overset{(b)}{\leq} 3L_k^2\,\mathbb{E}_{\xi'\in\mathcal{D}}\left[\left\|\mathbf{y}_k^{r_0+\tau}(\xi') - \mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')]\right\|^2\right] + 3\sigma^2 \\ &\quad + 3L_k^2\,\mathbb{E}_{\xi\in\mathcal{D}}\left[\left\|\mathbb{E}_{\xi''\in\mathcal{D}}[\mathbf{y}_k^{r_0+\tau}(\xi'')] - \mathbf{y}_k^{r_0+\tau}(\xi)\right\|^2\right]\end{aligned}$$

$$\begin{aligned}
&\stackrel{(c)}{=} 6\eta^2 L_k^2 \mathbb{E}_{\xi'\in\mathcal{D}} \left[\left\| \sum_{\tau_1=0}^{\tau-1} g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi');\xi') \right.\right. \\
&\qquad \left.\left. - \mathbb{E}_{\xi''\in\mathcal{D}} \left[\sum_{\tau_1=0}^{\tau-1} g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi'');\xi'')\right]\right\|^2\right] + 3\sigma^2 \\
&\stackrel{(d)}{\le} 6\eta^2 L_k^2 \tau \sum_{\tau_1=0}^{\tau-1} \mathbb{E}_{\xi'\in\mathcal{D}} \left[\left\| g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi');\xi') \right.\right. \\
&\qquad \left.\left. - \mathbb{E}_{\xi''\in\mathcal{D}}\, g_k(\mathbf{y}_k^{r_0+\tau_1}(\xi'');\xi'')\right\|^2\right] + 3\sigma^2 \\
&= 6\eta^2 L_k^2 \tau \sum_{\tau_1=0}^{\tau-1} \sigma_{\tau_1,k}^2 + 3\sigma^2, \qquad (24)
\end{aligned}$$

where (a) applies assumption A3.2 to the first term, A2 to the second term; in (b) we have merged the two expectations, and applied assumption A3.2 to the last term; notice the expectation on ξ and ξ' are independent for the first and the third term, so in (c) we merge the first and the third terms and recursively apply (19) to $\mathbf{y}_k^{r_0+\tau}(\xi'), \mathbf{y}_k^{r_0+\tau}(\xi'')$ until $\mathbf{\Theta}^{r_0}$, and cancel $\mathbf{\Theta}^{r_0}$; (d) applies Cauchy-–Schwarz inequality.

At this point, we have the following relations:

$$\begin{aligned}
&\mathbb{E}^{r_0}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2] \le \sum_{k=1}^{K} A_{\tau,k}, \\
&A_{\tau,k} \le 2B_{\tau,k} + \frac{2\sigma_{\tau,k}^2}{S} \le 4\eta^2 L_k^2 \tau \sum_{\tau_1=0}^{\tau-1} (A_{\tau_1,k} + \sigma_{\tau_1,k}^2) + \frac{2\sigma_{\tau,k}^2}{S}, \\
&\sigma_{0,k}^2 \le \sigma^2,\ \sigma_{\tau,k}^2 \le 6\eta^2 L_k^2 \tau \sum_{\tau_1=0}^{\tau-1} \sigma_{\tau_1,k}^2 + 3\sigma^2.
\end{aligned}$$

Notice that $\tau \le Q$. By choosing $6\eta^2 L_k^2 Q^2 \le 1$, which implies that $\eta \le \frac{1}{\sqrt{6}QL_k}$, and by recursively substituting the terms, we have the following bounds:

$$\begin{aligned}
&\sigma_{\tau,k}^2 \le \left[3 + 18(\tau^2-\tau)\eta^2 L_k^2 + 36\tau^3\eta^4 L_k^4\right]\sigma^2 \\
&A_{\tau,k} \le \left[\frac{6}{S} + \left(12 + \frac{60}{S}\right)\tau^2\eta^2 L_k^2 + \left(40 + \frac{80}{S}\right)\tau^4\eta^4 L_k^4\right]\cdot\sigma^2 \\
&\mathbb{E}^{r_0}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2 \le \left[\frac{6K}{S} + \left(12 + \frac{60}{S}\right) Q^2\eta^2 \sum_{k=1}^{K} L_k^2 \right. \\
&\qquad \left. + \left(40 + \frac{80}{S}\right) Q^4\eta^4 \sum_{k=1}^{K} L_k^4\right]\cdot\sigma^2. \qquad (25)
\end{aligned}$$

This completes bounding term $\mathbb{E}[\|\mathbf{G}^r - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2]$.

B. Proof of Term 2

2) Then, let us bound $\mathbb{E}^{r_0}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2$. We have the following series of relations:

$$\begin{aligned}
&\mathbb{E}^{r_0}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2 \\
&= \sum_{k=1}^{K} \mathbb{E}_{\mathcal{S}_l} \left\|\nabla_k\mathcal{L}(\mathbf{\Theta}^r(\mathcal{S}_l)) - \mathbb{E}_{\mathcal{S}_l'}\, g_k(\mathbf{y}_k^r(\mathcal{S}_l');\mathcal{S}_l')\right\|^2 \\
&\stackrel{(a)}{\le} \sum_{k=1}^{K} \mathbb{E}_{\mathcal{S}_l}\,\mathbb{E}_{\mathcal{S}_l'} \left\|g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\mathcal{S}_l') - g_k(\mathbf{y}_k^r(\mathcal{S}_l');\mathcal{S}_l')\right\|^2 \\
&\stackrel{(b)}{\le} \sum_{k=1}^{K} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\,\mathbb{E}_{\mathcal{S}_l'} \left\|\mathbf{\Theta}^r(\mathcal{S}_l) - \mathbf{y}_k^r(\mathcal{S}_l')\right\|^2 \\
&\stackrel{(c)}{=} \sum_{k=1}^{K} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\,\mathbb{E}_{\mathcal{S}_l'} \left[\left\|\theta_k^r(\mathcal{S}_l) - \theta_k^r(\mathcal{S}_l')\right\|^2 + \sum_{j\ne k}\left\|\theta_j^r(\mathcal{S}_l) - \theta_j^{r_0}\right\|^2\right] \\
&\stackrel{(d)}{=} \eta^2 \sum_{k=1}^{K} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\,\mathbb{E}_{\mathcal{S}_l'} \left[\left\|\sum_{\tau=r_0}^{r-1} \left(g_k(\mathbf{y}_k^\tau(\mathcal{S}_l);\mathcal{S}_l) - g_k(\mathbf{y}_k^\tau(\mathcal{S}_l');\mathcal{S}_l')\right)\right\|^2 \right. \\
&\qquad \left. + \sum_{j\ne k}\left\|\sum_{\tau=r_0}^{r-1} g_j(\mathbf{y}_j^\tau(\mathcal{S}_l);\mathcal{S}_l)\right\|^2\right] \\
&\stackrel{(e)}{\le} \eta^2\tau \sum_{\tau_1=0}^{\tau-1}\sum_{k=1}^{K} L_k^2\,\mathbb{E}_{\mathcal{S}_l}\,\mathbb{E}_{\mathcal{S}_l'} \left[\sum_{j\ne k}\left\|g_j(\mathbf{y}_j^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l)\right\|^2 \right. \\
&\qquad \left. + \left\|g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l) - g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l');\mathcal{S}_l')\right\|^2\right] \\
&\stackrel{(f)}{=} \eta^2\tau \sum_{\tau_1=0}^{\tau-1}\sum_{k=1}^{K} (L_k^2 + \sum_{j=1}^{K} L_j^2)\,\mathbb{E}_{\mathcal{S}_l}\left\|g_k(\mathbf{y}_k^{r_0+\tau_1}(\mathcal{S}_l);\mathcal{S}_l)\right\|^2 \\
&\stackrel{(g)}{\le} \eta^2\tau \left(\sum_{k=1}^{K} L_k^2 + \max_k\{L_k^2\}\right) \sum_{\tau_1=0}^{\tau-1} \mathbb{E}^{r_0}[\|\mathbf{G}^{r_0+\tau_1}\|^2] \\
&= \eta^2\tau \left(\sum_{k=1}^{K} L_k^2 + \max_k\{L_k^2\}\right) \times \\
&\quad \sum_{\tau_1=0}^{\tau-1} \mathbb{E}^{r_0}\left[\|\mathbf{G}^{r_0+\tau_1} - \mathbb{E}^{r_0}\,\mathbf{G}^{r_0+\tau_1}\|^2 + \|\mathbb{E}^{r_0}\,\mathbf{G}^{r_0+\tau_1}\|^2\right], \qquad (26)
\end{aligned}$$

where (a) uses the fact that $\nabla_k\mathcal{L}(\mathbf{\Theta}^r(\mathcal{S}_l)) = \mathbb{E}_{\mathcal{S}_l'}\, g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\mathcal{S}_l')$ and applies Jensen's inequality, that is

$$\begin{aligned}
&\left\|\mathbb{E}_{\mathcal{S}_l'}\, g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\mathcal{S}_l') - \mathbb{E}_{\mathcal{S}_l'}\, g_k(\mathbf{y}_k^r(\mathcal{S}_l');\mathcal{S}_l')\right\|^2 \\
&\le \mathbb{E}_{\mathcal{S}_l'}\,\|g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\mathcal{S}_l') - g_k(\mathbf{y}_k^r(\mathcal{S}_l');\mathcal{S}_l')\|^2;
\end{aligned}$$

(b) uses Jensen's inequality that

$$\begin{aligned}
&\|g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\mathcal{S}_l') - g_k(\mathbf{y}_k^r(\mathcal{S}_l');\mathcal{S}_l')\|^2 \\
&\le \mathbb{E}_{\xi\in\mathcal{S}_l'}\,\|g_k(\mathbf{\Theta}^r(\mathcal{S}_l);\xi) - g_k(\mathbf{y}_k^r(\mathcal{S}_l');\xi)\|^2
\end{aligned}$$

and applies assumption A3.2; (c) applies the definition of $\mathbf{y}_k^r(\mathcal{S}_l)$ that $\mathbf{y}_k^r(\mathcal{S}_l) \triangleq [\mathbf{\Theta}_{-k}^{r_0}, \theta_k^r(\mathcal{S}_l)]$; in (d) we expand the update steps until r^0; (e) applies Cauchy-Schwarz inequality; in (f) we reorder the sum and apply the i.i.d. assumption A1 to $\mathcal{S}_l, \mathcal{S}_l'$; in (g) we plug in the definition of $\mathbf{G}$. This completes bounding the term $\mathbb{E}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r) - \mathbb{E}\,\mathbf{G}^r\|^2$.

C. Proof of Main Result

Main result: Substitute the last term in (17) with (26) and let $\tau = r - r_0$, we have:

$$\begin{aligned}
&\mathbb{E}^{r_0}[\mathcal{L}(\mathbf{\Theta}^{r+1}) - \mathcal{L}(\mathbf{\Theta}^r)] \le -\frac{\eta}{2}\,\mathbb{E}^{r_0}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 \\
&\quad - \frac{\eta}{2}(1-\eta L)\,\|\mathbb{E}^{r_0}\,\mathbf{G}^r\|^2 + \eta^2 L\,\mathbb{E}^{r_0}\,\|\mathbf{G}^r - \mathbb{E}^{r_0}\,\mathbf{G}^r\|^2 \\
&\quad + \frac{1+\eta L}{2L}\eta^2 \left(\sum_{k=1}^{K} L_k^2 + \max_k\{L_k^2\}\right)\tau \sum_{\tau_1=0}^{\tau-1} \|\mathbb{E}^{r_0}\,\mathbf{G}^{r_0+\tau_1}\|^2 \\
&\quad + \frac{1+\eta L}{2L}\eta^2 \left(\sum_{k=1}^{K} L_k^2 + \max_k\{L_k^2\}\right)\tau \\
&\qquad \times \sum_{\tau_1=0}^{\tau-1} \mathbb{E}^{r_0}\,\|\mathbf{G}^{r_0+\tau_1} - \mathbb{E}^{r_0}\,\mathbf{G}^{r_0+\tau_1}\|^2 \\
&\le -\frac{\eta}{2}\,\mathbb{E}^{r_0}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2 - \frac{\eta}{2}(1-\eta L)\,\|\mathbb{E}^{r_0}\,\mathbf{G}^r\|^2 \\
&\quad + \frac{1+\eta L}{2L}\eta^2 \left(\sum_{k=1}^{K} L_k^2 + \max_k\{L_k^2\}\right)\tau \sum_{\tau_1=0}^{\tau-1} \|\mathbb{E}^{r_0}\,\mathbf{G}^{r_0+\tau_1}\|^2
\end{aligned}$$

$$+\eta^2\left(L+\eta\frac{1+\eta L}{2L}\left(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\}\right)\tau^2\right)\times\sigma^2$$
$$\times\left[\frac{6K}{S}+\left(12+\frac{60}{S}\right)Q^2\eta^2\sum_{k=1}^{K}L_k^2+\left(40+\frac{80}{S}\right)Q^4\sum_{k=1}^{K}\eta^4L_k^4\right],$$

where in the second inequality, we set $\eta\le\frac{1}{\sqrt{6}QL_k}$ and plug in (25). Average over $r=0,\ldots,T-1$ and reorganize the terms, we obtain:

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2\le\frac{2}{\eta T}\,\mathbb{E}[\mathcal{L}(\mathbf{\Theta}^0)-\mathcal{L}(\mathbf{\Theta}^T)]$$
$$-\left(1-\eta\left(L+\frac{1+\eta L}{2L}\left(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\}\right)Q^2\right)\right)\mathbb{E}\,\|\mathbb{E}^{r_0}\,\mathbf{G}^r\|^2$$
$$+2\eta\left(L+\eta\frac{1+\eta L}{2L}\left(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\}\right)Q^2\right)$$
$$\times\left[\frac{6K}{S}+\left(12+\frac{60}{S}\right)Q^2\eta^2\sum_{k=1}^{K}L_k^2+\left(40+\frac{80}{S}\right)Q^4\eta^4\sum_{k=1}^{K}L_k^4\right]\cdot\sigma^2.$$

Let

$$\left(1-\eta\left(L+\frac{1+\eta L}{2L}\left(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\}\right)Q^2\right)\right)\ge 0,$$

($\eta\le\frac{1}{L+2Q^2\sum_{k=1}^{K}L_k^2/L}$), then we have

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2\le\frac{2}{\eta T}\,\mathbb{E}[\mathcal{L}(\mathbf{\Theta}^0)-\mathcal{L}(\mathbf{\Theta}^T)] \tag{27}$$
$$+2\eta\cdot\left(L+\eta\frac{1+\eta L}{2L}\left(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\}\right)Q^2\right)\cdot\sigma^2$$
$$\times\left[\frac{6K}{S}+\left(12+\frac{60}{S}\right)Q^2\eta^2\sum_{k=1}^{K}L_k^2+\left(40+\frac{80}{S}\right)Q^4\eta^4\sum_{k=1}^{K}L_k^4\right].$$

Further let $Q\ge\sqrt{S/K}$, we have $\frac{1+\eta L}{2L}\le 1$ and

$$\left(12+\frac{60}{S}\right)Q^2\eta^2\sum_{k=1}^{K}L_k^2+\left(40+\frac{80}{S}\right)Q^4\eta^4\sum_{k=1}^{K}L_k^4$$
$$\le\left(72L^2+120L^4\right)\frac{K}{S},$$

therefore we have:

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\,\|\nabla\mathcal{L}(\mathbf{\Theta}^r)\|^2\le\frac{2}{\eta T}\,\mathbb{E}[\mathcal{L}(\mathbf{\Theta}^0)-\mathcal{L}(\mathbf{\Theta}^T)]$$
$$+\left(\frac{2\eta KC_1+2\eta^2Q^2KC_2}{S}\right)\cdot\sigma^2,$$

where $C_1=6+72L^2+120L^4$, $C_2=C_1\cdot(\sum_{k=1}^{K}L_k^2+\max_k\{L_k^2\})$. This completes the proof of Theorem 1. ■

Appendix B
Proof of Theorem 2

We first show that the conclusion holds for the case when $k<K$.

Let $\mathbf{x}_{i_j}^k$ denotes the ith sample of the data set $\mathcal{S}_j$ sampled at jth iteration. With initial weight $\theta_k^0\in R^{d_k}$, we first show that if we can find infinite number of non-identity orthogonal matrix $U\in R^{d_k\times d_k}$ such that

$$\theta_k^0=U^T\theta_k^0. \tag{28}$$

then for any $\{\mathbf{x}_{i_r,k}\}_{i\in\mathcal{S}_r,r=0,\ldots,T}$ that yields observations $\{H_{i_r}^k\}_{r=0,\ldots,T}$, we can construct another set of data

$$\tilde{\mathbf{x}}_{i_r,k}:=\mathbf{x}_{i_r,k}U \tag{29}$$

where U is chosen to satisfy condition (28), to produce the same exchanged values $\{H_{i_r}^k\}_{r=0,\ldots,T}$.

Let $\{\tilde{H}_{i_r}^k\}$ be observations generated by $\{\tilde{\mathbf{x}}_{i_r,k}\}$, and $\{\tilde{\theta}_k^r\}$ be weight variables with

$$\tilde{\theta}_k^0=U^T\theta_k^0. \tag{30}$$

That means for all $r=0,\ldots,T$,

$$\tilde{H}_{i_r}^k = H_{i_r}^k \tag{31}$$
$$\tilde{\theta}_k^r = U^T\theta_k^r. \tag{32}$$

Proof We adopt recursive proof here. First, it is easy to verify (31) for $r=0$, since,

$$\begin{aligned}H_{i_0}^k &= \mathbf{x}_{i_0,k}UU^T\theta_k^0\\ &= (\mathbf{x}_{i_0,k}U)(U^T\theta_k^0)\\ &= \tilde{\mathbf{x}}_{i_0,k}\tilde{\theta}_k^0\\ &= \tilde{H}_{i_0}^k,\end{aligned}$$

From equation (7), we define

$$g_{i_r}:=g(\mathbf{\Theta};\xi_{i_r}) \tag{33}$$

Now assuming that condition (31) and (32) hold for $r\le\tau$. That is,

$$g_{i_r} = \tilde{g}_{i_r} \tag{34}$$
$$\tilde{\theta}_k^r = U^T\theta_k^r \tag{35}$$

Then (32) holds for $r=\tau+1$ because

$$\tilde{\theta}_k^{\tau+1} \tag{36}$$
$$= \tilde{\theta}_k^{\tau}-\eta(\frac{1}{S}\sum_{i\in\mathcal{S}_\tau}\tilde{g}_i(\tilde{\mathbf{x}}_{i,k})^T+\lambda\tilde{\theta}_k^{\tau}) \tag{37}$$
$$= U^T\theta_k^{\tau}-\eta(\frac{1}{S}\sum_{i\in\mathcal{S}_\tau}g_i(\mathbf{x}_{i,k}U)^T+\lambda U^T\theta_k^{\tau}) \tag{38}$$
$$= U^T(\theta_k^{\tau}-\eta(\frac{1}{S}\sum_{i\in\mathcal{S}_\tau}g_i(\mathbf{x}_{i,k})^T+\lambda\theta_k^{\tau})) \tag{39}$$
$$= U^T\theta_k^{\tau+1} \tag{40}$$

where (38) follows from (34) and (35). Note if Q local updates are performed, locally we have

$$\theta_k^{r,q+1}-\theta_k^{r,q}=(1-\eta\lambda)(\theta_k^{r,q}-\theta_k^{r,q-1}) \tag{41}$$

where $\theta_k^{r,q}$ denotes the qth local update of rth iteration. it is thus easy to show that

$$\tilde{\theta}_k^{\tau+1,q}=U^T\theta_k^{\tau+1,q} \tag{42}$$

Next we show (31) holds for $r=\tau+1$.

$$\tilde{H}_{i_{\tau+1}}^k = \tilde{\mathbf{x}}_{i_{\tau+1},k}\tilde{\theta}_k^{\tau+1} \tag{43}$$
$$= x_{i_{\tau+1},k}UU^T\theta_k^{\tau+1} \tag{44}$$
$$= x_{i_{\tau+1},k}\theta_k^{\tau+1} \tag{45}$$
$$= H_{i_{\tau+1}}^k \tag{46}$$

The proof is completed for $k < K$.

Next we show the conclusion holds for $k = K$. Similarly for any $\{\mathbf{x}_{i_r,K}\}_{r=0,\ldots,T}$ and $\{y_{i_r,K}\}$, we construct a different solution as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{i_r,K} &:= \mathbf{x}_{i_r,K} U && (47) \\ \tilde{y}_{i_r,K} &:= y_{i_r,K}. && (48) \end{aligned}$$

where $U \in R^{d_k \times d_k}$ is a non-identity orthogonal matrix satisfying (29) for $k = K$. Therefore we have

$$\begin{aligned} H_{i_r}^K &= g(H_{i_r}, y_{i_r,K}) && (49) \\ &= g(\tilde{H}_{i_r}, \tilde{y}_{i_r,K}) && (50) \\ &= \tilde{H}_{i_r}^K && (51) \end{aligned}$$

which means the constructed output is identical to the original output.

Finally, we only need to show that we can find infinite number of non-identity orthogonal matrix $U \in R^{d_k \times d_k}$ to satisfy (28). This proof is provided in the following Lemma.

Lemma 1: For any vector $\theta^0 \in R^{d_k}$. There exists infinite many of non-identity orthogonal matrix $U \in R^{d_k \times d_k}$ such that

$$\begin{aligned} UU^T &= I && (52) \\ U\theta^0 &= \theta^0 && (53) \end{aligned}$$

Proof. First we construct a orthogonal U_1 satisfying (52) and (53) for

$$\theta^0 := e_1 = (1, 0, \cdots, 0)^T \in R^{d_k} \quad (54)$$

Then we complete the proof by generalizing the construction for an arbitrary $\theta^0 \in R^{d_k}$.

With $\theta^0 = e_1$, we construct U_1 in the following way

$$U_1 := \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & V \end{array} \right] \quad (55)$$

where $V \in R^{(d_k-1)\times(d_k-1)}$ is any non-identity orthogonal matrix with $d_k > 2$, i.e.,

$$VV^T = I. \quad (56)$$

Condition (52) is satisfied since

$$\begin{aligned} U_1 U_1^T &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & V \end{array} \right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & V^T \end{array} \right] && (57) \\ &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & VV^T \end{array} \right] && (58) \\ &= I, && (59) \end{aligned}$$

and condition (53) is satisfied trivially, i.e.,

$$U_1 e_1 = \left[\begin{array}{c|c} 1 & 0, \cdots, 0 \end{array} \right]^T = e_1 \quad (60)$$

For any arbitrary θ^0, we apply the *Householder transformation* to "rotate" it to the basis vector e_1, i.e.,

$$\theta^0 = \|\theta^0\|_2 P e_1 \quad (61)$$

where P is the *Householder transformation* operator such as

$$\begin{aligned} P &= P^T && (62) \\ PP^T &= PP = I && (63) \end{aligned}$$

Therefore from U_1 defined in (55) we can construct U by

$$U = PU_1P. \quad (64)$$

Finally, we verifies that U satisfies condition (52)) and (53)):

$$\begin{aligned} UU^T &= PU_1P(PU_1^TP) && (65) \\ &= PU_1(PP)U_1^TP && (66) \\ &= PU_1U_1^TP && (67) \\ &= PP = I && (68) \\ U\theta^0 &= PU_1P\theta^0 && (69) \\ &= \|\theta^0\|_2 P(e_1U_1) && (70) \\ &= \|\theta^0\|_2 Pe_1 && (71) \\ &= \theta^0 && (72) \end{aligned}$$

where (70) holds since from (61) we have

$$P\theta^0 = \|\theta^0\|_2 e_1$$

■

ACKNOWLEDGMENT

This work was partially supported by the National Key Research and Development Program of China under Grant No. 2018AAA0101100.

REFERENCES

[1] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. aurelio Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, “Large scale distributed deep networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1223–1231. [Online]. Available: http://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, “Federated learning of deep networks using model averaging,” *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D’Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, “Advances and open problems in federated learning,” *CoRR*, vol. abs/1912.04977, 2019. [Online]. Available: http://arxiv.org/abs/1912.04977

[4] C. Gratton, V. D., R. Arablouei, and S. Werner, “Distributed ridge regression with feature partitioning,” 10 2018, pp. 1423–1427.

[5] Y. Hu, D. Niu, J. Yang, and S. Zhou, “Fdml: A collaborative machine learning framework for distributed features,” in *KDD*, 2019.

[6] Y. Hu, P. Liu, L. Kong, and D. Niu, “Learning privately over distributed features: An ADMM sharing approach,” *CoRR*, vol. abs/1907.07735, 2019. [Online]. Available: http://arxiv.org/abs/1907.07735

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *CoRR*, vol. abs/1902.04885, 2019. [Online]. Available: http://arxiv.org/abs/1902.04885

[8] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, “Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption,” *CoRR*, vol. abs/1711.10677, 2017.

[9] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, “Secureboost: A lossless federated learning framework,” *CoRR*, vol. abs/1901.08755, 2019. [Online]. Available: http://arxiv.org/abs/1901.08755

[10] Y. Liu, T. Chen, and Q. Yang, "Secure federated transfer learning," *CoRR*, vol. abs/1812.03337, 2018. [Online]. Available: http://arxiv.org/abs/1812.03337

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 14 774–14 784. [Online]. Available: http://papers.nips.cc/paper/9617-deep-leakage-from-gradients.pdf

[12] P. Richtárik and M. Takáč, "Distributed coordinate descent method for learning with big data," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2657–2681, Jan. 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2946645.3007028

[13] D. Mahajan, S. S. Keerthi, and S. Sundararajan, "A distributed block coordinate descent method for training l1regularized linear classifiers," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3167–3201, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?id=3122009.3176835

[14] Z. Peng, Y. Xu, M. Yan, and W. Yin, "Arock: an algorithmic framework for asynchronous parallel coordinate updates," 2015, preprint, arXiv:1506.02396.

[15] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild!: A lock-free approach to parallelizing stochastic gradient dea distributed, asynchronous and incremental algorithm for nonconvex optimization: An admm based approachvia a scent," 2011, online at arXiv:1106.57320v2.

[16] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 1310–1321. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813687

[17] H. Yu, S. Yang, and S. Zhu, "Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5693–5700.

[18] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[19] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 977–992, 2019.

[20] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation, Academia Press*, pp. 169–179, 1978.

[21] A. C. Yao, "Protocols for secure computations," in *SFCS*, 1982, pp. 160–164.

[22] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, and R. Zhang, "A hybrid approach to privacy-preserving federated learning," *CoRR*, vol. abs/1812.03224, 2018. [Online]. Available: http://arxiv.org/abs/1812.03224

[23] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[24] V. Hartmann and R. West, "Privacy-preserving distributed learning with secret gradient descent," *CoRR*, vol. abs/1906.11993, 2019. [Online]. Available: http://arxiv.org/abs/1906.11993

[25] O. Fontenla-Romero, B. Guijarro-Berdiñas, B. Pérez-Sánchez, and M. Gómez-Casal, "LANN-DSVD: A privacy-preserving distributed algorithm for machine learning," in *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*, 2018. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-140.pdf

[26] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[27] H. Wang and A. Banerjee, "Randomized block coordinate descent for online and stochastic optimization," *arXiv preprint arXiv:1407.0107*, 2014.

[28] H. Eichner, T. Koren, B. McMahan, N. Srebro, and K. Talwar, "Semi-cyclic stochastic gradient descent," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1764–1773.

[29] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 603–618. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134012

[30] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706, 2018.

[31] Q. Li, Z. Wen, and B. He, "Practical federated gradient boosting decision trees," in *34th AAAI Conference on Artificial Intelligence (AAAI-20)*, 2020.

[32] W. Du, Y. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *SIAM Proceedings Series*, M. Berry, U. Dayal, C. Kamath, and D. Skillicorn, Eds., 2004, pp. 222–233.

[33] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 639–644. [Online]. Available: http://doi.acm.org/10.1145/775047.775142

[34] O. L. Mangasarian, E. W. Wild, and G. M. Fung, "Privacy-preserving classification of vertically partitioned data via random kernels," *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 3, pp. 12:1–12:16, Oct. 2008. [Online]. Available: http://doi.acm.org/10.1145/1409620.1409622

[35] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[36] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *CIVR*, 2009.

[37] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[38] V. Bolón-Canedo, K. Sechidis, N. Sánchez-Maroño, A. Alonso-Betanzos, and G. Brown, "Insights into distributed feature ranking," *Information Sciences*, vol. 496, pp. 378 – 398, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518307588

[39] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, "Centralized vs. distributed feature selection methods based on data complexity measures," *Knowledge-Based Systems*, vol. 117, pp. 27 – 45, 2017, volume, Variety and Velocity in Data Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116303537

[40] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization for heterogeneous networks," *arXiv preprint arXiv:1812.06127*, Jul. 2019. [Online]. Available: https://arxiv.org/abs/1812.06127

[41] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-815.html