FedPD: A Federated Learning Framework with Adaptivity to Non-IID Data

Xinwei Zhang†, Mingyi Hong†, Sairaj Dhople†, Wotao Yin‡ and Yang Liu#

***Abstract*—Federated Learning (FL) is popular for communication-efficient learning from distributed data. To utilize data at different clients without moving them to the cloud, algorithms such as the Federated Averaging (FedAvg) have adopted a computation then aggregation model, in which multiple local updates are performed using local data before aggregation. These algorithms fail to work when faced with practical challenges, e.g., the local data being non-identically independently distributed. In this paper, we first characterize the behavior of the FedAvg algorithm, and show that without strong and unrealistic assumptions on the problem structure, it can behave erratically. Aiming at designing FL algorithms that are provably fast and require as few assumptions as possible, we propose a new algorithm design strategy from the primal-dual optimization perspective. Our strategy yields algorithms that can deal with non-convex objective functions, achieves the best possible optimization and communication complexity (in a well-defined sense), and accommodates full-batch and mini-batch local computation models. Importantly, the proposed algorithms are *communication efficient*, in that the communication effort can be reduced when the level of heterogeneity among the local data also reduces. To the best of our knowledge, this is the first algorithmic framework for FL that achieves all the above properties.**

I. Introduction

Federated learning (FL)—a distributed machine learning approach proposed in [1]—has gained popularity for applications involving learning from distributed data. In FL, a cloud server (the "server") can communicate with distributed data sources (the "agents"). The goal is to train a global model that works well for all the distributed data, but without requiring the agents to reveal too much local information. Since its inception, the broad consensus on FL's implementation appears to involve a generic "local update" strategy to save communication efforts. The basic communication pattern "computation and aggregation" (CTA) protocol involves the following steps: S1) the server sends the global model $\mathbf{x}$ to the agents; S2) the agents update their local models $\mathbf{x}_i$'s based on their local data for several iterations; S3) the server aggregates $\mathbf{x}_i$'s to obtain a new global model $\mathbf{x}$. The CTA protocol is popular, partly because transmitting local gradients and other statistics to the server is undesirable. For instance, it has been shown that local gradient information can leak private data [2], [3], [4] and increase the cost when applying privacy preserving methods.

Even though the FL paradigm has attracted significant attention from both academia and industry, and many algorithms such as Federated Averaging (FedAvg) have been proposed [5], [6], [7], [8], several attributes are not clearly established. In particular, the commonly adopted local update strategy poses significant theoretical and practical challenges to designing effective FL algorithms. This work attempts to provide a deeper understanding of FL by raising and resolving key theoretical questions, as well as by developing an effective algorithmic framework with several desirable features.

Problem Formulation. Consider the following problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) &\triangleq \frac{1}{N}\sum_{i=1}^N f_i(\mathbf{x}), \\ f_i(\mathbf{x}) &\triangleq w_i \sum_{\xi_i\in\mathcal{D}_i} F(\mathbf{x};\xi_i), \end{aligned} \tag{1}$$

where ξ_i denotes one sample in data set $\mathcal{D}_i$ stored on the i-th agent; $F:\mathbb{R}^d\to\mathbb{R}$ is the "loss function" for data point ξ_i; and $w_i>0$ is a "weight coefficient" (a common choice is $w_i=1/|\mathcal{D}_i|$ [9]). We assume that the loss function takes the same form across different agents, and furthermore, we denote $M:=\sum_{i=1}^N|\mathcal{D}_i|$ to be the total number of samples. One can also consider a related setting, where each $f_i(\mathbf{x})$ represents the expected loss [8]

$$f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\in\mathcal{P}_i} F(\mathbf{x};\xi_i), \tag{2}$$

where $\mathcal{P}_i$ denotes the data distribution on the i-th agent. Throughout the paper, we will make the following blanket assumptions for problem (1):

A 1. *Each $f_i(\cdot)$, as well as $f(\cdot)$ in* (1) *is L–smooth:*

$$\begin{aligned} &\|\nabla f_i(\mathbf{x})-\nabla f_i(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \\ &\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \ \forall\, \mathbf{x},\mathbf{y}\in\mathbb{R}^d, i=1,\ldots,N. \end{aligned}$$

A 2. *The objective of problem* (1) *is lower bounded: $f(\mathbf{x})\ge c>-\infty$, $\forall\ \mathbf{x}\in\mathbb{R}^d$.*

In addition to these standard assumptions, state-of-the-art efforts on analysis of FL algorithms oftentimes invoke a number of more *restrictive* assumptions.

A 3. (Bounded Gradient Dissimilarity (BGD)) *[11] The gradients ∇f_i's are upper bounded (by a constant $G>0$ and $B\ge 0$)*

$$\frac{1}{N}\sum_{i=1}^N \|\nabla f_i(\mathbf{x})\|^2 \le G^2 + D^2\|\nabla f(\mathbf{x})\|^2, \quad \forall\ \mathbf{x}\in\mathbb{R}^d. \tag{3}$$

Let us comment on the two special cases of this assumption. 1) When $D=0$: this assumption is the so-called bounded gradient (BG) assumption, and it indicates the local gradients

† University of Minnesota, email: {zhan6234,mhong,sdhople}@umn.edu; ‡ University of California, Los Angeles, email: wotaoyin@math.ucla.edu; # Webank, Co. Ltd, email: yangliu@webank.com.

TABLE I: Convergence rates of FL algorithms, measured by total rounds of communication (RC), number of local updates (LC) and number of samples (SC), before reaching ϵ-stationary solution. CTA refers to CTA protocol, LP refers solving local problem to certain accuracy, BGD refers to bounded gradient dissimilarity, and CVX refers to convexity, NC is non-convex, μSC means μ-Strongly Convex. p is the function of $\mathcal{O}(\frac{\epsilon}{G^2})$ illustrated in Fig. 1.

Algorithm	CVX	BGD	CTA	LP	RC (T)	LC (QT)	SC
FedAvg [10]	μSC	(G,0)	✓	×	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$
FedAvg [11]	μSC	(G,D)	✓	×	$\mathcal{O}\left(1/\epsilon^{1/2}\right)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon)$
Local-GD [12]	C	-	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(Q/\epsilon^{3/2})$
FedAvg [5]	NC	(G,0)	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
FedAvg [11]	NC	(G,D)	✓	×	$\mathcal{O}(1/\epsilon^{3/2})$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
VRL-SGD [13]	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
F-SVRG [14]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}((M+Q)/\epsilon)$
SCAFFOLD [11]	NC	-	×	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^{2})$	$\mathcal{O}(1/\epsilon^{2})$
FedSplit [15]	μSC	-	✓	✓	$\mathcal{O}(\log(1/\epsilon))$	$\mathcal{O}(Q\log(1/\epsilon))$	$\mathcal{O}(QB\log(1/\epsilon))$
FedProx [16]	NC	(0,D)	✓	✓	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(QB/\epsilon)$
Fed-PD	NC	-	✓	✓	$\mathcal{O}\left(1/\epsilon\right)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(QB/\epsilon)$
Fed-PD	NC	(G,1)	✓	✓	$\mathcal{O}((1-p)/\epsilon)$	$\mathcal{O}(Q(1-p)/\epsilon)$	$\mathcal{O}(QB(1-p)/\epsilon)$
Fed-PD (VR)	NC	-	✓	×	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(Q/\epsilon)$	$\mathcal{O}(M+\sqrt{M}/\epsilon)$

are upper bounded by some constant. In early works, the BG assumption is used to bound the deviation between the agents after multiple local updates, which is critical for analyzing FL algorithms; 2) When $G = 0$ and $D = 1$: this assumption indicates that the local functions have the same gradient for all $\mathbf{x}$, or equivalently the distribution of local data is homogeneous.

Finally, we mention that our objective is to understand the FL algorithm from an optimization perspective. So we say that a solution $\mathbf{x}$ is an ϵ-stationary solution if the following holds:

$$\|\nabla f(\mathbf{x})\|^2 \leq \epsilon. \tag{4}$$

We are interested in finding the *minimum* system resources required, such as the number of local updates, the number of times local data are transmitted to the server, and the number of times local samples $F(\mathbf{x};\xi_i)$'s are accessed, before computing an ϵ-solution (4). These quantities are referred to as *local computation*, *communication complexity*, and *sample complexities*, respectively. Below, we list four questions to be addressed in this work.

Q1 (local updates). What are the best local update directions for the agents to take to achieve the best overall system performance (stability, sample complexity, etc.)?

Q2 (global aggregation). Can we use more sophisticated processing in the aggregation step to improve system performance (sample or communication complexity)?

Q3 (communication efficiency). What is the minimum communication (at each round and in total) to achieve a desired solution accuracy?

Q4 (assumptions). What is the best performance that a CTA type algorithm can achieve while relying on a minimum set of assumptions?

Although these questions are not directly related to data privacy—another important aspect of FL—we argue that answering these fundamental questions can provide a much-needed understanding of the FL approach. A few recent works have touched upon those questions. Still, to our knowledge, none of them have provided a thorough investigation of the questions listed above.

Related Works. We discuss existing algorithms in FL by roughly classifying them based on two considerations: 1) Communication protocol: whether the algorithm follows the CTA protocol, i.e., only transfer the models during the communication, or transfer more information; 2) Local update strategy: whether the local agents solve a local problem to a certain accuracy, or just perform certain fixed steps of local update. The results are summarized in Table I. It is pertinent to consider how these algorithms address questions Q1–Q4.

To answer Q1, let us review the local steps used for state-of-the-art algorithms. The well-known FedAvg algorithm performs multiple local (stochastic) GD steps to minimize the local loss function between two aggregation steps; see Algorithm 1 below.

Algorithm 1 FedAvg Algorithm

Initialize: $\mathbf{x}_i^0 \triangleq \mathbf{x}^0, i = 1, \ldots, N$
for $r = 0, \ldots, T-1$ *(stage)* **do**
  for $q = 0, \ldots, Q-1$ *(iteration)* **do**
    for $i = 1, \ldots, N$ in parallel **do**
      Local update: $\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta \nabla F(\mathbf{x}_i^{r,q}; \xi_i^{r,q})\ \forall i$
    end for
  end for
  Global averaging: $\mathbf{x}^{r+1} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^{r,Q}$
  Update agents' $\mathbf{x}_i^{r+1,0} = \mathbf{x}^{r+1},\ i = 1, \ldots, N$
end for

However, in most cases, successive local GD steps lead to sub-optimal communication complexity [12], [17]. By using correction terms, FedSplit [15] greatly reduces the communication complexity in the convex setting; VRL-SGD [13] and SCAFFOLD [11] also reduce the communication complexity in certain non-convex settings, but VRL-SGD requires some bounded variance assumption, which essentially implies that the (stochastic) gradients are bounded. Additionally, SCAFFOLD needs to communicate both the local models and the local gradients, which doubles the communication overhead.

For Q2, although most algorithms use simple averaging, F-SVRG [14] and SCAFFOLD break the CTA protocol. F-SVRG shows an improvement in sample complexity, and SCAFFOLD improves the dependence on agent number N compared with VRL-SGD. However, there is little discussion

on whether other types of linear processing are helpful or the CTA protocol is enough for FL algorithms.

For Q3, a number of recent works show that a total of $O(1/\epsilon)$ aggregation steps are needed for non-convex problems to achieve ϵ-solution (4). However, bounded variance assumption and local statistics are needed. It is not clear if this achieves the best communication complexity.

As for Q4, the algorithms typically require either bounded variance assumption, or some BGD assumption, or both to achieve a good performance. FedSplit shows a possible optimal performance under the strongly convex setting, but the best performance under the non-convex setting remains unknown.

Main Contributions. First, we address Q1-Q4 and provide an in-depth examination of the CTA protocol. We show that algorithms following the CTA protocol that are based on successive local gradient updates, the best possible communication efficiency is $\mathcal{O}(1/\epsilon)$; neither additional local processing nor general linear processing can improve this rate.

We then propose a meta-algorithm called Federated Primal-Dual (FedPD), which follows the CTA protocol and can be implemented in several different forms with desirable properties. In particular, *i)* it achieves convergence under only Assumptions A1– A2, *ii)* it achieves the best possible optimization and communication complexity when data is non-i.i.d., and *iii)* the communication pattern of the proposed algorithm can be adapted to the degree of non-i.i.d.-ness of the local data. That is, we show that under A3, communication saving and data heterogeneity interestingly exhibit a linear-logarithmic relationship; see Fig. 1 for an illustration. To our knowledge, this is the first algorithm for FL that achieves all the above properties.

TABLE II: Summary of notation used in the paper

N, i	total number, and index of clients
M, B, b	total number, batch size and index of samples
T, r	total number and index of communication rounds
Q, q	total number and index of local updates
$\mathbf{x}_0^r$	global model at communication round r
$\mathbf{x}_{0,i}^r$	i^{th} client's estimated global model at round r
$\mathbf{x}_i^{r,q}$	i^{th} client's model at round r and step q
$\mathbf{x}_{0,i}^{r,+}$	the model i^{th} client send to server after round r

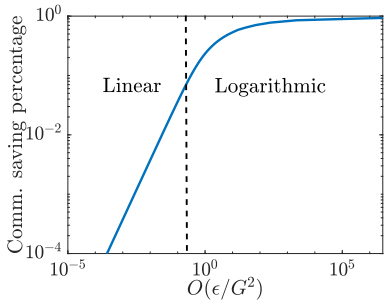


Fig. 1: Relation of the percentage of comm. savings, accuracy ϵ, heterogeneity G. Details in Section IV.

II. Properties of CTA Protocols

In this section, we formally address questions raised in the previous section about the CTA protocol.

A. Communication Lower Complexity Bounds

We first address Q2–Q3 under the CTA protocol. Specifically, for problems satisfying A1–A2, does performing multiple local updates or using different ways to combine local models reduce communication complexity? We show below that under the CTA protocol, such a saving is impossible.

Consider the following generic CTA protocol. Let t denote the index for communication rounds. Between two rounds $t-1$ and t, each agent performs Q local updates. Denote $x_i^{t-1,q}$ to be the q-th local update. Then, $x_i^{t-1,Q}$'s are sent to the server, combined through a (possibly time-varying) function $V^t(\cdot) : \mathbb{R}^{Nd} \to \mathbb{R}^d$, and sent back. The agents then generate a new iterate by combining the received message with past gradients using a (possibly time-varying) function $W_i^t(\cdot)$:

$$x^t = V^t(\{x_i^{t-1,Q}\}_{i=1}^N),\ x_i^{t,0} = x^t,\ \ \forall\ i \in [N], \tag{5a}$$

$$x_i^{t,q} \in W_i^t\left(\{x_i^{r,k}, \{\nabla F(x_i^{r,q}; \xi_i)\}_{\xi_i \in D_i}\}_{k \in [q-1], r \in [t]}\right),$$
$$\forall\ q \in [Q],\ \forall\ i \in [N]. \tag{5b}$$

We focus on the case where the $V^t(\cdot)$'s and $W_i^t(\cdot)$'s are *linear* operators, which implies that $x_i^{t,q}$ can use all past iterates and (sample) gradients for its update. Clearly, (5) covers both the local-GD and local-SGD versions of FedAvg as special cases.

In the following, we provide an informal statement of the result. The formal statement and the full proof are given in Theorem 4, which due to space limitation is relegated to the supplementary material F.

Claim II.1. (Informal) *Consider any algorithm A that belongs to the class described in* (5)*, with $V^t(\cdot)$ and $W_i^t(\cdot)$'s being linear and possibly time-varying operators. Then, there exists a non-convex problem instance satisfying Assumptions 1–2 such that for any $Q > 0$, algorithm A takes at least $\mathcal{O}(1/\epsilon)$ communication rounds to reach an ϵ-stationary solution satisfying* (4)*.*

Remark 1. The proof technique is related to those developed from both classical and recent works that characterize lower bounds for first-order methods, in both centralized [18], [19] and decentralized [20], [21] settings. The main technical difference is that our processing model (5) additionally allows local processing iterations, and there is a central aggregator. In the proof, we construct problem instances in which f_i's are non-i.i.d. (i.e., G in assumption A3 grows with the total number of iterations T, and $D = 1$). Then we show that it is necessary to aggregate (thus communicate) to make any progress. On the other hand, it is obvious that in another extreme case where the data are homogeneous (i.e., $G = 0$, $D = 1$), only $\mathcal{O}(1)$ communication rounds are needed. ■

B. Local Update Strategy and Bounded Gradient

We now address Q1 and Q4. We consider the FedAvg Algorithm and show that when using (stochastic) gradient as the local update direction, the bounded gradient assumption A3 is critical to ensure performance.

Claim II.2. *Fix any constant $\eta > 0$, $Q > 1$ for FedAvg. There exists a problem that satisfies A1 and A2 but fails to satisfy A3, on which FedAvg diverges to infinity.*

Due to space limitation, the proof of the above result is relegated to Appendix A.

Remark 2. A recent work [12] has shown that FedAvg with *constant* stepsize $\eta > 0$ can only converge to a neighborhood of the global minimizer for *convex* problems. Beyond that, our result indicates that when f_i's are *non-convex*, FedAvg can perform much worse without the BGD assumption. Even if $Q = 2$ and there exists a solution such that $\sum_{i=1}^N \|f_i(\hat{\mathbf{x}})\|^2 = 0$, FedAvg (with constant stepsize η) diverges and the iteration can go to ∞. ■

The above result suggests that, despite its popularity, the pure local (stochastic) gradient direction is not compatible with the CTA protocol. This motivates the design of local update strategies that allow the agents to work together properly.

III. The FedPD Framework

In this section, we propose a meta-algorithm called Federated Primal-Dual (FedPD), which is an efficient algorithm following the CTA protocol. Among many of its features, the FedPD achieves the communication lower bound mentioned in the previous section without requiring additional assumptions such as BGD (3). Further, we show that for problems satisfying the BGD assumption (3), the proposed algorithm can effectively reduce communication overhead.

Our algorithm is based on the following *global consensus* reformulation of the original problem (1):

$$\min_{\mathbf{x}_0, \mathbf{x}_i} \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}_i), \quad \text{s.t. } \mathbf{x}_i = \mathbf{x}_0, \ \forall i \in [N]. \tag{6}$$

To present our algorithm, let us define the augmented Lagrangian (AL) function of (6) as

$$\mathcal{L}(\mathbf{x}_{0:N}, \boldsymbol{\lambda}) \triangleq \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i(\mathbf{x}_0, \mathbf{x}_i, \lambda_i),$$

$$\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_0, \lambda_i) \triangleq f_i(\mathbf{x}_i) + \langle \lambda_i, \mathbf{x}_i - \mathbf{x}_0 \rangle + \frac{1}{2\eta} \|\mathbf{x}_i - \mathbf{x}_0\|^2.$$

Fixing $\mathbf{x}_0$, the AL is separable over all local pairs $\{(\mathbf{x}_i, \lambda_i)\}$. The key technique in the design is to specify *how* each local AL $\mathcal{L}_i(\cdot)$ should be optimized, and *when* to perform model aggregation.

Federated primal-dual algorithm (FedPD) can be easily implemented in the FL setting, while capturing the main idea of the classical primal-dual based algorithm; see Algorithm 2. In particular, its update rules share a similar pattern as the Alternating Direction Method of Multipliers (ADMM), but it does not specify how the local models are updated. Instead, an *oracle* $\text{Oracle}_i(\cdot)$ is used as a placeholder for local processing, and we will see that careful instantiations of these oracles lead to algorithms with different properties. Importantly, we introduce a critical constant $p \in [0, 1)$, which determines the frequency at which the aggregation and communication steps are skipped. By using FedPD, we can see that at each communication round, only the local models are exchanged. In Algorithm 3 and Algorithm 4 we provide different oracles for FedPD. It is worth noting that Oracle II is based on the idea of variance reduction, and it can achieve a lower sample complexity compared with those in Oracle I. Below, we provide more discussion about these proposed local oracles.

Algorithm 2 Federated Primal-Dual Algorithm

Input: $\mathbf{x}^0, \eta, p, T, Q_1, \ldots, Q_N$
Initialize: $\mathbf{x}_0^0 = \mathbf{x}^0$,
for $r = 0, \ldots, T-1$ **do**
  for $i = 1, \ldots, N$ in parallel do local updates **do**
    $\mathbf{x}_i^{r+1} = \text{Oracle}_i(\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r), Q_i)$
    $\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r)$ #Dual updates
    $\mathbf{x}_{0,i}^{r+} = \mathbf{x}_i^{r+1} + \eta \lambda_i^{r+1}$
  end for
  With probability $1-p$ do global communication:
    Global Communicate:
    $\mathbf{x}_0^{r+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{0,i}^{r+}$
    $\mathbf{x}_{0,i}^{r+1} = \mathbf{x}_0^{r+1}, \ i = 1, \ldots, N$
  With probability p skip global communication:
    Local Update: $\mathbf{x}_{0,i}^{r+1} \triangleq \mathbf{x}_{0,i}^{r+}$
end for

Algorithm 3 Oracle Choice I

Input: $\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r), Q_i$
Initialize: $\mathbf{x}_{i,0}^r = \mathbf{x}_i^r$,
Option I (GD)
for $q = 0, \ldots, Q_i - 1$ **do**
$\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta_1 \nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i^{r,q}, \mathbf{x}_{0,i}^r, \lambda_i^r)$
end for
Option II (SGD)
for $q = 0, \ldots, Q_i - 1$ **do**
  $\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta_1 (h_i(\mathbf{x}_i^{r,q}; \xi_i^{r,q}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r,q} - \mathbf{x}_{0,i}^r))$
end for
Output: $\mathbf{x}_i^{r+1} \triangleq \mathbf{x}_i^{r,Q_i}$

In Algorithm 3, the stochastic gradient is defined as

$$h_i(\mathbf{x}_i^{r,q}; \xi_i^{r,q}) \triangleq \nabla F(\mathbf{x}_i^{r,q}; \xi_i^{r,q}), \text{ with } \xi_i^{r,q} \sim \mathcal{D}_i, \tag{7}$$

where $\sim$ denotes uniform sampling. Further, for both options, Q_i's are chosen so that the local problems are solved accurately enough. Specifically, for GD (Option I) we need to ensure that we run the inner iterations long enough such that the following holds:

$$\left\|\nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r)\right\|^2 \le \epsilon_1. \tag{8}$$

Similarly, for SGD (Option II), we need to assume that the following holds:

$$\mathbb{E}\left\|\nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r)\right\|^2 \le \epsilon_1. \tag{9}$$

Note that in Algorithm 2 we provide two ways for solving this subproblem by using GD and SGD, but any other solver that achieves (8) can be used. Despite the simplicity of the local updates, we will show that using Oracle I makes FedPD adaptive to the non-i.i.d. parameter G.

Alternatively, when instantiating the local oracle using Algorithm 4, the original local problems are not required to solve to ϵ_1 accuracy. Instead, we successively optimize a linearized AL function:

$$\tilde{\mathcal{L}}_i^r(\mathbf{x}_i) \triangleq \tilde{f}_i(\mathbf{x}_i; \mathbf{x}_i^{r,q}) + \left\langle \lambda_r^i, \mathbf{x}_i - \mathbf{x}_{0,i}^r \right\rangle + \frac{1}{2\eta} \left\|\mathbf{x}_i - \mathbf{x}_{0,i}^r\right\|^2.$$

Algorithm 4 Oracle Choice II

Input: $\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r), Q, I, B$
Initialize: $\mathbf{x}_i^{r,0} = \mathbf{x}_i^r$,
if $r \mod I = 0$ **then** $g_i^{r,0} = \nabla f_i(\mathbf{x}_i^{r,0})$
else $g_i^{r,0} = g_i^{r-1,Q}$
end if
for $q = 0, \ldots, Q-1$ **do**
$\mathbf{x}_i^{r,q+1} = \arg\min_{\mathbf{x}_i} \tilde{\mathcal{L}}_i(\mathbf{x}_i, \mathbf{x}_{0,i}^r, \lambda_i^r; \mathbf{x}_i^{r,q}, g_i^{r,q})$
$g_i^{r,q+1} = g_i^{r,q} + \frac{1}{B}\sum_{b=1}^{B}(h_i(\mathbf{x}_i^{r,q+1}; \xi_{i,b}^{r,q}) - h_i(\mathbf{x}_i^{r,q}; \xi_{i,b}^{r,q}))$
end for
Output: $\mathbf{x}_i^{r+1} \triangleq \mathbf{x}_i^{r,Q}$, $g_i^{r,Q}$

In the above expression, we linearize $f_i(\mathbf{x}_i)$ at inner iteration $\mathbf{x}_i^{r,q}$ as

$$\tilde{f}_i^r(\mathbf{x}_i; \mathbf{x}_i^{r,q}) \triangleq f(\mathbf{x}_i^{r,q}) + \langle g_i^{r,q}, \mathbf{x}_i - \mathbf{x}_i^r\rangle + \frac{1}{2\gamma}\left\|\mathbf{x}_i - \mathbf{x}_i^{r,q}\right\|^2,$$

where γ is a constant and $g_i^{r,q}$ is an approximation of $\nabla f_i(\mathbf{x}_i^{r,q})$. The optimizer has a closed-form expression:

$$\mathbf{x}_i^{r,q+1} = \frac{\eta}{\eta+\gamma}\mathbf{x}_i^{r,q} + \frac{\gamma}{\eta+\gamma}\mathbf{x}_{0,i}^r - \frac{\eta\gamma}{\eta+\gamma}(g_i^{r,q} + \lambda_i^r).$$

In Oracle II, an agent i first decides whether to compute the full gradient $\nabla f_i(\mathbf{x}_i^{r,0})$, or to keep using the previous estimate $g_i^{r-1,Q}$. Then Q local steps are performed; each requires B local data samples. In this scheme, Q can be chosen as *any* positive integer.

It is important to note that this oracle does not simply apply the variance reduction (VR) technique (such as F-SVRG) to solve the subproblem of optimizing $\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_{0,i}^r, \lambda_i^r)$. That is, it is *not* a variation of Oracle I. Instead, the VR technique is applied to the entire primal-dual iteration, and the full gradient evaluation $\nabla f_i(\mathbf{x}_i^{r,0})$ is only needed every I iteration r. Later we will see that if I is large enough, then there is an $\mathcal{O}(\sqrt{M})$ reduction of sample complexity.

IV. Convergence & Complexity Analysis

In this section, we first provide a basic convergence analysis of FedPD without assuming A3 (or effectively, with G in (3) being infinity). Then, we show that with Assumption A3, FedPD allows some communication rounds to be skipped when the local functions become similar (that is, when G become smaller). We refer the readers to Appendix B for detailed proof of Theorem 1 and 3. Due to space limitation the proof of Theorem 2 is included in the supplemental material.

A. Analysis Without the BGD Assumption

We first characterize the convergence of FedPD with different oracles, without assuming the BGD assumption A3.

Theorem 1. *Suppose A1 –A2 hold. Define* $D_0 := f(\mathbf{x}_0^0) - f(\mathbf{x}^\star)$. *Consider FedPD with Oracle I, where* Q_i*'s are selected such that* (8) *holds if Option I is used, and* (9) *holds if Option II is used. Set* $0 < \eta < \frac{\sqrt{5}-1}{4L}$, $p = 0$. *Then we have:*

$$\frac{1}{T}\sum_{r=0}^{T}\|\nabla f(\mathbf{x}_0^r)\|^2 \le \frac{C_2}{T}D_0 + C_4\epsilon_1,$$

where C_2, C_4 *are constants independent of* T, G, p.

Theorem 2. *Suppose A1–A2 hold. Consider FedPD with Oracle II. Choose* $p = 0$, $\eta \in \big(0, \frac{1}{3(Q+\sqrt{QI/B})L}\big)$, *and* $\gamma > \frac{5\eta}{B\sqrt{L}}$. *Then, the following holds (where* $C_9 > 0$ *is a constant):*

$$\frac{1}{T}\sum_{r=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_0^r)\|^2 \le \frac{C_9}{T}(f(\mathbf{x}_0^0) - f(\mathbf{x}^\star)). \tag{10}$$

Remark 3. For Oracle I to achieve ϵ accuracy, we need to set the communication round $T = C_2D_0/\epsilon$ and local accuracy $\epsilon_1 = \epsilon/C_4$. As the local AL is strongly convex with respect to $\mathbf{x}_i$, optimizing it to ϵ accuracy requires $Q_i = \mathcal{O}(\log(\epsilon))$ iterations for GD and $Q_i = \mathcal{O}(1/\epsilon)$ for SGD [22]. ■

Remark 4. Suppose Oracle II runs for T communication rounds, the total number of full gradient evaluation is $T/I + 1$, each uses M samples. Meanwhile, the total number of mini-batch stochastic gradient evaluation is TQ, each uses $2B$ samples per node. So the total sample complexity is $\mathcal{O}(M + MT/I + 2TQBN)$. Therefore, we choose $I = \sqrt{M}, B = I/QN = \sqrt{M}/QN$, then the sample complexity of Algorithm 4 is $\mathcal{O}(M + \frac{\sqrt{M}}{\epsilon})$. ■

B. Analysis with the BGD Assumption

In this subsection, we analyze how the additional assumption A3 can affect the proposed algorithm. Towards this end, let us consider the following $(G, 1)$-BGD assumption (which is equivalent to A3 with $D = 1$).

A 4. (G,1)-BGD *The local functions are called* $(G, 1)$*-BGD if either one of the equivalent conditions below holds:*

$$\begin{aligned}&\|\nabla f_i(\mathbf{x}) - \nabla f_j(\mathbf{x})\| \le G,\ \forall\ \mathbf{x} \in \mathbb{R}^d,\ \forall\ i \ne j,\\ &\text{or } \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\| \le G\ \forall\ \mathbf{x} \in \mathbb{R}^d,\ \forall\ i.\end{aligned} \tag{11}$$

Theorem 3. *Suppose A1 –A2 and A4 holds. Consider FedPD with Oracle I, where* Q_i*'s are selected such that* (8) *holds if Option I is used, and* (9) *holds if Option II is used. Set* $0 < \eta < \frac{\sqrt{5}-1}{4L}$, $0 \le p < 1$. *Then we have:*

$$\frac{1}{T}\sum_{r=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x}_0^r)\|^2 \le \frac{C_2}{T}D_0 + C_4\epsilon_1 + \eta^2(1-p)(N-1)C_5$$
$$\times (1 - C_3^{\frac{1}{(1-p)}})^2 p^2 \frac{(1+L\eta)^2(2L\eta + p(3+L\eta))^2}{N(1-2L\eta-p(1+L\eta))^2}(G^2 + \epsilon_1). \tag{12}$$

Here $C_2, C_4, C_5 > 0$ *are constants independent of* T, G, p; $C_3 := \frac{p(1+L\eta)+L\eta}{1-L\eta} \ge 0$.

Remark 5. **(Communication reduction).** Note that since $0 \le p < 1$, the total communication rounds is given by $T(1-p)$. To achieve the ϵ accuracy, we need to chose $T = C_2D_0/\epsilon$, $\epsilon_1 = \epsilon/C_4$, and need to chose p appropriately so that the last term in (12) is also smaller than ϵ. This implies that $T = C_2D_0/\epsilon$ and the following shall hold

$$C(p) \triangleq \eta^2(1-p)(N-1)C_5(1 - C_3^{1/(1-p)})^2p^2$$
$$\times \frac{(1+L\eta)^2(2L\eta + p(3+L\eta))^2}{N(1-2L\eta-p(1+L\eta))^2} \le \frac{\epsilon}{3G^2}.$$

The above relation implies that p and $\frac{\epsilon}{G^2}$ should be related by $(1-p)T = \mathcal{O}(\frac{G-\sqrt{\epsilon}}{G\epsilon})$ when $G^2 \in (\mathcal{O}(\epsilon), \infty)$; further,

TABLE III: The relation between p and $\frac{\epsilon}{G^2}$ with fixed $\eta = \frac{\sqrt{5}-1}{8L}$.

Range of p	C_3	$C(p)$	p as function of $\frac{\epsilon}{G^2}$	Relation
$[0, \frac{1-2L\eta}{1+L\eta})$	< 1	$\approx 12\eta^2 p^2$	$\sqrt{\frac{1}{36\eta^2}\frac{\epsilon}{G^2}}$	**Linear**
$[\frac{1-2L\eta}{1+L\eta}, 1)$	≥ 1	$\approx 14\eta^2 C_3^{2/(1-p)}$	$1 - 2/\log(\frac{1}{42\eta^2}\frac{\epsilon}{G^2})$	**Log**

$p \to 1$ at a log-rate when $G^2 \to 0$; see Table III for details. These results characterize the relation between communication saving and the homogeneity of the local problems. ■

C. Connection with Other Algorithms

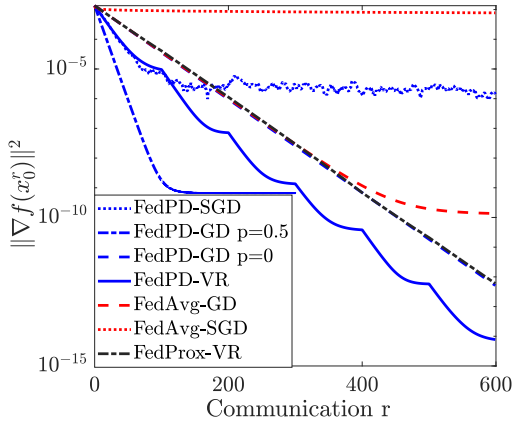
Before we close this section, we discuss the relation of FedPD with a few existing algorithms. In FedProx [16] the agents optimize the following local objective: $f_i(\mathbf{x}_i) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{x}_0^r\|^2$. FedProx algorithm fails to converge to the global stationary solution. In contrast, FedPD introduces extra local dual variables $\{\lambda_i\}$ that record the gap between the local model $\mathbf{x}_i$ and the global model $\mathbf{x}_0$ which help the global convergence. FedDANE [23] also proposes a way of designing the subproblem by using the global gradient, but this violates the CTA protocol. Compared with these two algorithms, the proposed FedPD has weaker assumptions, and it achieves better sample and/or communication complexity. In SCAFFOLD [11], the clients perform the following update:

$$\mathbf{x}_i^{r,q+1} = \mathbf{x}_i^{r,q} - \eta(g_i^{r,q} - c_i^r + c^r),$$
$$c_i^{r+1} = c_i^r - c + \frac{1}{K\eta}(\mathbf{x}_0^r - \mathbf{x}_i^{r,Q}),$$

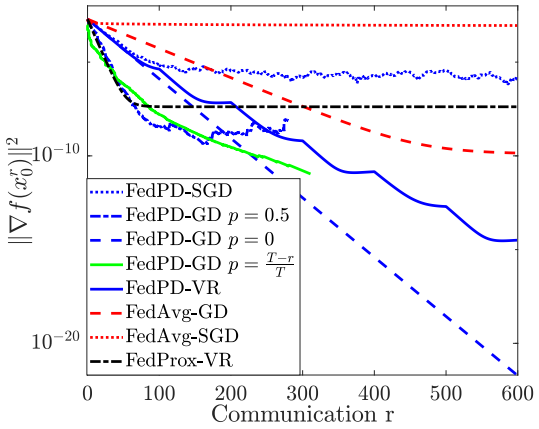
and the server performs the following step:

$$\mathbf{x}_0^{r+1} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^{r,Q}, \quad c^{r+1} = \frac{1}{N}\sum_{i=1}^N c_i^{r+1}.$$

Compared to the update with FedPD, we can observe that $c - c_i$'s play the same role as the dual variables λ_i's in FedPD. But SCAFFOLD requires the clients to send the c_i's to the server which breaks the CTA protocol and doubles the communicated information. However, our result showed that even without the extra communication, FedPD can achieve the same convergence rate by adopting a more sophisticated local update direction.



(a) Stationary gap of FedAvg, FedProx and FedPD; weakly non-i.i.d. data.



(b) Stationary gap of FedAvg, FedProx and FedPD; strongly non-i.i.d. data.

Fig. 2: The convergence result of the algorithms on penalized logistic regression with weakly and strongly non-i.i.d. data with respect to the number of communication rounds.

V. Numerical Experiments

In the first experiment, we show the convergence of the proposed algorithms on synthetic data with FedAvg and FedProx as baselines. We use the non-convex penalized logistic regression [24] as the loss function.

In the experiment, we use two ways to generate the data. In the first case (referred to as the "weakly non-i.i.d" case), the features and the labels on the agents are randomly generated, so the local data sets are not very non-i.i.d. In the second case (referred to as the "strong non-i.i.d." case), we first generate the feature vector $\mathbf{a}$'s following the standard Normal distribution, then we generate the local model $\mathbf{x}_i$ on the i^{th} agent by using uniform distribution in the range of $[-10, 10]$ for each component. Then we compute the label b's according to the local models and the features, and then add noise following the standard normal distribution. In this case, the agents' data distribution is more non-i.i.d compared to the first case. In both cases, there are 400 samples on each agent with total 100 agents.

We run FedPD with Oracle I (FedPD-SGD and FedPD-GD) and Oracle II (FedPD-VR). For FedPD-SGD, we set $Q = 600$, and for FedPD-GD and FedPD-VR we set $Q = 8$. For FedPD-GD we set $p = 0$ and $p = 0.5$, where in the latter case, the agents skip half of the communication rounds. For FedPD-VR, we set mini-batch size $B = 1$ and gradient computation frequency $I = 20$. For comparison, we also run FedAvg with local GD/SGD and FedProx. For FedAvg with GD, $Q = 8$, and for FedAvg with SGD, $Q = 600$. For FedProx, we solve the local problem using variance reduction for $Q = 8$ iterations. The total number of iterations T is set as 600 for all algorithms.

Figure 2 shows the results with respect to the number of communication rounds. In Fig. 2(a), we compare the

convergence of the tested algorithms on weakly non-i.i.d. data set. It is clear that FedProx and FedPD with $p = 0$ (i.e., no communication skipping) are comparable. Meanwhile, FedAvg with local GD will not converge to the stationary point with a constant stepsize when local update step $Q > 1$. By skipping half of the communication, FedPD with local GD can still achieve a similar error as FedAvg, but using fewer communication rounds. In Fig. 2(b), we compare the convergence results of different algorithms with the strongly non-i.i.d. data set. We can see that the algorithms using stochastic solvers become less stable compared with the case when the data sets are weakly non-i.i.d. Further, FedPD-VR and FedPD-GD with $p = 0$ are still able to converge to the global stationary point while FedProx will achieve a similar error as the FedAvg with local GD.

We have included more details on the experimental results and additional experiments in Appendix D.

VI. Conclusion

We study federated learning under the CTA protocol. We explore a number of theoretical properties of this protocol and design a meta-algorithm called FedPD, which contains various algorithms with desirable properties, such as achieving the best communication/computation complexity and adapting its communication pattern with data heterogeneity.

References

[1] "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen, "iDLG: Improved deep leakage from gradients," 2020.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, "How to backdoor federated learning," *arXiv preprint arXiv:1807.00459*, 2018.

[4] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu, "A framework for evaluating gradient leakage attacks in federated learning," *arXiv preprint arXiv:2004.10397*, 2020.

[5] Hao Yu, Sen Yang, and Shenghuo 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*, 2019, vol. 33, pp. 5693–5700.

[6] Jianyu Wang and Gauri Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient sgd algorithms," *arXiv preprint arXiv:1808.07576*, 2018.

[7] Sebastian Urban Stich, "Local sgd converges fast and communicates little," *ICLR 2019 - International Conference on Learning Representations*, p. 17, 2019.

[8] Hao Yu, Rong Jin, and Sen Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, California, USA, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 7184–7193, PMLR.

[9] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, "Federated learning: Challenges, methods, and future directions," *arXiv preprint arXiv:1908.07873*, 2019.

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

[11] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," *arXiv preprint arXiv:1910.06378*, 2019.

[12] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik, "First analysis of local GD on heterogeneous data," *arXiv preprint arXiv:1909.04715*, 2019.

[13] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng, "Variance reduced local SGD with lower communication complexity," *arXiv preprint arXiv:1912.12844*, 2019.

[14] Shicong Cen, Huishuai Zhang, Yuejie Chi, Wei Chen, and Tie-Yan Liu, "Convergence of distributed stochastic variance reduced methods without sampling extra data," *arXiv preprint arXiv:1905.12648*, 2019.

[15] Reese Pathak and Martin J Wainwright, "Fedsplit: An algorithmic framework for fast federated optimization," *arXiv preprint arXiv:2005.05238*, 2020.

[16] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith, "On the convergence of federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[17] Filip Hanzely and Peter Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[18] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Springer, 2004.

[19] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points i," *Mathematical Programming*, Jun 2019.

[20] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," *arXiv preprint arXiv:1702.08704*, 2017.

[21] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," *IEEE Transactions on Signal processing*, July 2019, accepted for publication.

[22] K. Yuan, B. Ying, S. Vlaski, and A. H. Sayed, "Stochastic gradient descent with finite samples sizes," in *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2016, pp. 1–6.

[23] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, "FedDANE: A federated newton-type method," *arXiv preprint arXiv:2001.01920*, 2020.

[24] Anestis Antoniadis, Irène Gijbels, and Mila Nikolova, "Penalized likelihood regression for generalized linear models with non-quadratic penalties," *Annals of the Institute of Statistical Mathematics*, vol. 63, no. 3, pp. 585–615, 2011.

[25] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

Appendix A
Proof of Claim II.2

Proof. We consider the following problem with $N = 2$, which satisfies both Assumptions 1 and 2, with $f(\mathbf{x}) = 0,\ \forall\ \mathbf{x}$. It is easy to show that A3 is not satisfied.

$$f_1(\mathbf{x}) = \mathbf{x}^2, \quad f_2(\mathbf{x}) = -\mathbf{x}^2. \tag{13}$$

Each local iteration of the FedAvg is given by

$$\mathbf{x}_1^{r+1} = (1-\eta^r)\mathbf{x}_1^r, \quad \mathbf{x}_2^{r+1} = (1+\eta^r)\mathbf{x}_2^r. \tag{14}$$

For simplicity, let us define $\mathbf{y} = [\mathbf{x}_1, \mathbf{x}_2]^T$, and define the matrix $\mathbf{D}_r = [1-\eta^r, 0; 0, 1+\eta^r]$. Then running Q rounds of the FedAvg algorithm starting with $r = kQ$ for some non-negative integer $k \geq 0$, can be expressed as

$$\mathbf{y}^{(k+1)Q} = \prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{y}^{kQ+1},\ \mathbf{y}^{kQ+1} = \frac{1}{2}\mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ}\mathbf{y}^{kQ}.$$

Therefore, overall we have

$$\mathbf{y}^{(k+1)Q} = \frac{1}{2}\prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ}\mathbf{y}^{kQ}. \tag{15}$$

In particular, pick $\eta^r = \frac{1}{\sqrt{r}}$ when $r \neq kQ+1$ and $\eta^{kQ+1} = 1/2$. Then for $Q > 1$, we can show that the matrix $\frac{1}{2}\prod_{r=kQ+1}^{(k+1)Q-1} \mathbf{D}_r \mathbf{1}\mathbf{1}^T \mathbf{D}_{kQ}$ has an eigenvalue given below:

$$\lambda = \frac{1}{4}\prod_{r=kQ+2}^{(k+1)Q-1}(1-\frac{1}{\sqrt{r}})(1-\frac{1}{\sqrt{kQ}}) + \frac{3}{4}\prod_{r=kQ+2}^{(k+1)Q-1}(1+\frac{1}{\sqrt{r}})(1+\frac{1}{\sqrt{kQ}}) > 1$$

This indicates that the algorithm will diverge. ■

APPENDIX B
PROOFS FOR RESULTS IN SECTION III

A. Proof of Theorem 1 and Theorem 3

First, let us assume that when the GD option in Oracle I is used, Q_i is large enough such that the following holds:

$$\left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r,Q_i}, \mathbf{x}_0^r, \lambda_i^r)\right\|^2 = \left\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_0^r, \lambda_i^r)\right\|^2 \leq \epsilon_1. \quad (16)$$

Similarly, when the SGD option is used, then Q_i is chosen such that the following holds true:

$$\mathbb{E}[\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r,Q_i}, \mathbf{x}_0^r, \lambda_i^r)\|^2] = \mathbb{E}[\|\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}_i^{r+1}, \mathbf{x}_0^r, \lambda_i^r)\|^2] \leq \epsilon_1. \quad (17)$$

The difference does not significantly change the proofs and the results. So throughout the proof of this theorem, we use (16) as the condition.

Throughout the proof, we denote the expectation taken on the communication r^{th} iteration to the $r+1^{\text{th}}$ iteration conditioning on all the previous knowledge as $\mathbb{E}_{r+1}$. Using these notations, define the error between different nodes as

$$\triangle^r \triangleq [\triangle\mathbf{x}_0^r; \triangle\mathbf{x}^r], \text{ with} \quad (18)$$

$$\triangle\mathbf{x}_0^r \triangleq \max_{i,j}\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r\right\|, \ \triangle\mathbf{x}^r \triangleq \max_{i,j}\left\|\mathbf{x}_i^r - \mathbf{x}_j^r\right\|. \quad (19)$$

Here, $\triangle\mathbf{x}_0^r$ denotes the maximum difference of estimated center model among all the nodes and $\triangle\mathbf{x}^r$ denotes the maximum difference of local models among all nodes.

From the termination condition that generates $\mathbf{x}_i^{r+1}$ (given in (16)), we have

$$\nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^{r+1} = \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) = \mathbf{e}_i^{r+1}, \quad (20)$$

where $\left\|\mathbf{e}_i^{r+1}\right\|^2 \leq \epsilon_1$, and the first equality holds because of the update rule of λ_i. Furthermore, from the update step of λ_i^{r+1}, we can explicitly write down the following expression

$$\lambda_i^{r+1} = \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) = -\nabla f_i(\mathbf{x}_i^{r+1}) + \mathbf{e}_i^{r+1}.$$

The main lemmas that we need are outlined below. Their proofs can be found in Appendix Sec. C.

Lemma 1. *Suppose A1 holds true. Consider FedPD with Algorithm 4 (Oracle I) as the update rule. When the local problem is solved such that* (16) *is satisfied, we have*

$$\begin{aligned}&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)\\ &\leq -\frac{1-2L\eta}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 - \frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2\\ &+ \eta\left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2 + \frac{\epsilon_1}{2L}.\end{aligned} \quad (21)$$

Then we derive a key lemma about how the error propagates if the communication step is skipped.

Lemma 2. *Suppose A1 and A4 hold. Consider FedPD with Algorithm 4 (Oracle I) as the update rule. When the local problem is solved such that* (16) *is satisfied, the difference between the local models $\mathbf{x}_i^r$'s and the local copies of the global models $\mathbf{x}_{0,i}^r$'s is bounded by*

$$\mathbb{E}_{r+1}\triangle^{r+1} \leq \frac{1}{1-L\eta}(A\triangle^r + \eta B(G + 2\sqrt{\epsilon_1}), \quad (22)$$

where we have defined

$$A \triangleq \begin{bmatrix} p(1+L\eta) & pL\eta(1-L\eta) \\ 1 & L\eta \end{bmatrix},$$

which is a rank one matrix with eigenvalues $(0, L\eta + p(1+L\eta))$ and $B = [p(3+L\eta), 2]^T$.

Next, we define a virtual sequence $\{\overline{\mathbf{x}}_0^r\}$ where $\overline{\mathbf{x}}_0^r \triangleq \frac{1}{N}\sum_{i=1}^N \mathbf{x}_{0,i}^r$ which is the average of the local $\mathbf{x}_{0,i}^r$. We know that $\mathbf{x}_{0,i}^r = \mathbf{x}_0^r$ when $r \mod R = 1$ (i.e., when the communication and aggregation step is performed). Next, we bound the error between the local AL and the global AL evaluated at the virtual sequence.

Lemma 3. *Suppose A1 holds. Consider FedPD with Algorithm 4 (Oracle I) as the update rule. When the local problem is solved such that* (16) *is satisfied, the difference between local AL and the global AL is bounded as below:*

$$\begin{aligned}&\frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \overline{\mathbf{x}}_0^{r+1}, \lambda_i^{r+1})]\\ &\geq -\frac{(N-1)}{2N\eta}(\triangle\mathbf{x}_0^{r+1})^2.\end{aligned} \quad (23)$$

Lastly we bound the objective function using the global AL.

Lemma 4. *Under A1 and A2, consider FedPD with Algorithm 4 (Oracle I) as the update rule. When the local problem is solved to ϵ_1 accuracy satisfying* (16)*, the difference between the original loss and the augmented Lagrangian is bounded.*

$$f(\mathbf{x}_0^r) \leq \mathcal{L}(\mathbf{x}_{0:N}^r, \boldsymbol{\lambda}^r) - \frac{1-2L\eta}{N\eta}\sum_{i=1}^N\left\|\mathbf{x}_i^r - \mathbf{x}_0^r\right\|^2 + \frac{\epsilon_1}{2L}. \quad (24)$$

Now we are ready to prove Theorem 1 and Theorem 3.

B. Proof of Theorem 1 and Theorem 3

First, for notational simplicity, let us define the following:

$$\mathcal{L}_i^r \triangleq \mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r),\ \mathcal{L}_i^{r+1} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+1}, \lambda_i^{r+1}) \quad (25)$$
$$\mathcal{L}_i^{r+} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^{r+}, \lambda_i^{r+1}),\ \bar{\mathcal{L}}_i^{r+1} \triangleq \mathcal{L}_i(\mathbf{x}_i^{r+1}, \bar{\mathbf{x}}_0^{r+1}, \lambda_i^{r+1}).$$

Notice that from the optimality condition (20), the following holds:

$$\left\|\lambda_i^r - \lambda_i^{r-1}\right\|^2 \leq 2L^2 \left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\|^2 + 4\epsilon_1. \quad (26)$$

Then we bound the gradients of $\mathcal{L}(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)$.

$$\begin{aligned}
&\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\| = \left\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r)\right\| \\
&\overset{(20)}{=} \Big\|\nabla f_i(\mathbf{x}_i^r) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r) - \nabla f_i(\mathbf{x}_i^{r+1}) - \lambda_i^r \\
&-\frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) + \mathbf{e}_i^{r+1}\Big\| \leq \frac{1+L\eta}{\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\| + \sqrt{\epsilon_1}.
\end{aligned} \quad (27)$$

Note that when no aggregation has been performed at iteration r, then $\mathbf{x}_{0,i}^r = \mathbf{x}_i^r + \eta\lambda_i^r$, so the following holds

$$\|\nabla_{\mathbf{x}_0}\mathcal{L}_i^r\| = \left\|\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r)\right\| = 0. \quad (28)$$

When aggregation has been performed at iteration r, then $\mathbf{x}_{0,i}^r = \frac{1}{N}\sum_{j=1}^N(\mathbf{x}_j^r + \eta\lambda_j^r),\ \forall i$, so we have

$$\|\nabla_{\mathbf{x}_0}\mathcal{L}(\mathbf{x}_{0:N}^r, \boldsymbol{\lambda}^r)\| = \left\|\frac{1}{N}\sum_{i=1}^N(\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^r - \mathbf{x}_{0,i}^r))\right\| = 0. \quad (29)$$

Further by using the definition of $\mathcal{L}_i^r$ and the dual update step, we have:

$$\begin{aligned}
\|\nabla_{\lambda_i}\mathcal{L}_i^r\| &= \left\|\mathbf{x}_i^r - \mathbf{x}_{0,i}^r\right\| \\
&\leq \left\|\mathbf{x}_i^r - \mathbf{x}_{0,i}^{r-1}\right\| + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\| \\
&\leq \eta\left\|\lambda_i^r - \lambda_i^{r-1}\right\| + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\| \\
&\leq \eta(L\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\| + 2\sqrt{\epsilon_1}) + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|.
\end{aligned} \quad (30)$$

From (28), we know that $\|\nabla_{\mathbf{x}_0}\mathcal{L}_i^r\| = 0$. So we see that the size of the full gradient $\nabla\mathcal{L}_i^r$ can be expressed by:

$$\left\|\nabla\mathcal{L}_i(\mathbf{x}_i^r, \mathbf{x}_{0,i}^r, \lambda_i^r)\right\|^2 = \|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2 + \|\nabla_{\lambda_i}\mathcal{L}_i^r\|^2 \quad (31)$$
$$\leq (\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\| + \|\nabla_{\lambda_i}\mathcal{L}_i^r\|)^2. \quad (32)$$

Then we have

$$\begin{aligned}
\|\nabla\mathcal{L}_i^r\|^2 &\leq \Bigg(\frac{1+L\eta}{\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\| + \sqrt{\epsilon_1} \\
&+ \eta(L\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\| + 2\sqrt{\epsilon_1}) + \left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|\Bigg)^2 \\
&\leq C_6\left(\left\|\mathbf{x}_{0,i}^{r-1} - \mathbf{x}_{0,i}^r\right\|^2\right. \\
&\left.+ \left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \left\|\mathbf{x}_i^r - \mathbf{x}_i^{r-1}\right\|^2 + \epsilon_1\right),
\end{aligned} \quad (33)$$

where $C_6 \geq 3\max\{(\frac{1+L\eta}{\eta})^2, (1+2\eta)^2, L^2\eta^2\}$. Apply (26) to Lemma 1 we obtain

$$\begin{aligned}
&\frac{1 - 2L\eta - 4L^2\eta^2}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2 \\
&-\frac{1+8L\eta}{2L}\epsilon_1 \leq \mathcal{L}_i^r - \mathcal{L}_i^{r+}.
\end{aligned} \quad (34)$$

Notice that when communication is not performed, we have $\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+1}\right\|^2 \leq \left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2$. When communication is performed, the following holds:

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+1}\right\|^2 \\
&= \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2 + \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^{r+1}\right\|^2 \\
&\leq \frac{2}{N}\sum_{i=1}^N\left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+}\right\|^2 + \frac{N-1}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2,
\end{aligned} \quad (35)$$

where the last inequality holds due to the use of Jensen's inequality, and the definition of $\triangle\mathbf{x}_0^{r+1}$ in (18). It follows that summing both sides of (34) over i, we have

$$\begin{aligned}
&\frac{1 - 2L\eta - 4L^2\eta^2}{2\eta}\sum_{i=1}^N\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \frac{N(1+8L\eta)}{2L}\epsilon_1 \\
&+ \sum_{i=1}^N(\frac{1}{4\eta}\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 - \frac{N-1}{4\eta}(\triangle\mathbf{x}_0^{r+1})^2) \\
&\leq \sum_{i=1}^N\left(\mathcal{L}_i^r - \mathcal{L}_i^{r+}\right) + \frac{N(1+8L\eta)}{L}\epsilon_1.
\end{aligned} \quad (36)$$

Taking the expectation over the randomness on the communication step, we obtain the following:

$$\begin{aligned}
&\mathbb{E}_{r+1}\frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^r - \mathcal{L}_i^{r+}] \\
&= \frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^r - \mathcal{L}_i^{r+1}] + \mathbb{E}_{r+1}\frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^{r+1} - \mathcal{L}_i^{r+}] \\
&\overset{(a)}{=} \frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^r - \mathcal{L}_i^{r+1}] + \frac{1}{N}\sum_{i=1}^N[(1-p)\bar{\mathcal{L}}_i^{r+1} + p\mathcal{L}_i^{r+} - \mathcal{L}_i^+] \\
&= \frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^r - \mathcal{L}_i^{r+1}] + (1-p)\frac{1}{N}\sum_{i=1}^N[\bar{\mathcal{L}}_i^{r+1} - \mathcal{L}_i^{r+}] \\
&\overset{(b)}{\leq} \frac{1}{N}\sum_{i=1}^N[\mathcal{L}_i^r - \mathcal{L}_i^{r+1}] + (1-p)\frac{N-1}{2\eta N}(\triangle\mathbf{x}_0^{r+1})^2
\end{aligned} \quad (37)$$

where (a) expands the expectation on p, and use the fact that with probability p, $\mathbf{x}_{0,i}^{r+1} = \mathbf{x}_{0,i}^{r+}$, and with probability $(1-p)$ $\mathbf{x}_0^{r+1}$ will be updated; in (b) we apply Lemma 3 to the last term.

Combining (36) and (37), we have

$$\begin{aligned}
&\min\left\{\frac{1 - 2L\eta - 4L^2\eta^2}{2\eta}, \frac{1}{2\eta}, \frac{1+8L\eta}{2L}\right\} \\
&\times \frac{1}{N}\sum_{i=1}^N\mathbb{E}_{r+1}\left[\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 + \epsilon_1\right] \\
&\leq \frac{1}{N}\sum_{i=1}^N\left[\mathcal{L}_i^r - \mathcal{L}_i^{r+1}\right] \\
&+ \frac{1+8L\eta}{L}\epsilon_1 + (1-p)\frac{(N-1)}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2.
\end{aligned} \quad (38)$$

Combining (33), (36) and (38), define $C_7 = 2C_6/\min\{\frac{1-2L\eta-4L^2\eta^2}{2\eta}, \frac{1}{2\eta}, \frac{1+8L\eta}{2L}\}$ and sum up the iterations, we have

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2\\
&\overset{(33)(36)}{\leq} \frac{2C_6}{N}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left[\left\|\mathbf{x}_{0,i}^r-\mathbf{x}_{0,i}^{r+}\right\|^2+\left\|\mathbf{x}_i^r-\mathbf{x}_i^{r+1}\right\|^2\right.\\
&\left.+(1-p)\frac{(N-1)}{\eta N}(\triangle\mathbf{x}_0^{r+1})^2+\epsilon_1\right]\\
&\overset{(38)}{\leq} \frac{C_7}{N}\sum_{r=0}^{T}\sum_{i=1}^{N}(\mathcal{L}_i^r-\mathcal{L}_i^{r+1})\\
&\quad+\frac{C_7(1+8L\eta)}{L}\epsilon_1+(1-p)C_7\sum_{r=0}^{T}\frac{N-1}{N\eta}\mathbb{E}(\triangle\mathbf{x}_0^{r+1})^2.
\end{aligned}\tag{39}$$

Next we bound the last term in the above inequality. By iteratively applying Lemma 2 from $\tau=0$ to r and use the fact that $G^0=0$, we have

$$\begin{aligned}
&\mathbb{E}\triangle\mathbf{x}_0^{r+1}\overset{(18)}{=}[1,0]\times\mathbb{E}\triangle^{r+1}\\
&\leq[1,0]\times\sum_{\tau=0}^{r}(\frac{A}{1-L\eta})^\tau\eta\frac{[p(3+L\eta),2]^T}{1-L\eta}(G+2\sqrt{\epsilon_1}).
\end{aligned}\tag{40}$$

From Lemma 2 we have:

$$\lambda\left(\frac{1}{1-L\eta}A\right)=\frac{p(1+L\eta)+L\eta}{1-L\eta}\triangleq C_8.$$

So by squaring both side of (40), we have

$$\begin{aligned}
&\mathbb{E}(\triangle\mathbf{x}_0^{r+1})^2\\
&\leq\left\|[1,0]\sum_{\tau=1}^{r}(\frac{A}{1-L\eta})^\tau\eta\frac{[p(3+L\eta),2]^T}{1-L\eta}(G+2\sqrt{\epsilon_1})\right\|^2\\
&\leq\mathbb{E}\left(\frac{1-C_8^{r+1}}{1-C_8}\right)^2(2p\eta(1+L\eta)(2L\eta+p(3+L\eta)))^2\\
&\times(G^2+\epsilon_1)\\
&=4p^2\eta^2(1+L\eta)^2(2L\eta+p(3+L\eta))^2\\
&\times\left(\frac{1-C_8^{1/(1-p)}}{1-C_8}\right)^2(G^2+\epsilon_1).
\end{aligned}\tag{41}$$

Substitute (41) into (39) and divide both sides by T we have

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2\\
&\leq\frac{C_7}{T}\left(\mathcal{L}(\mathbf{x}_0^0,\mathbf{x}_i^0,\lambda_i^0)-\mathcal{L}(\mathbf{x}_i^T,\mathbf{x}_{0,i}^T,\lambda_i^T)\right)+\frac{C_7(1+8L\eta)}{L}\epsilon_1\\
&+\eta^2(1-p)(N-1)C_7(1-C_8^{1/(1-p)})^2p^2\\
&\times\frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-C_8)^2}(G^2+\epsilon_1).
\end{aligned}\tag{42}$$

From the initial conditions we have $\mathcal{L}(\mathbf{x}_0^0,\mathbf{x}_i^0,\lambda_i^0)=f(\mathbf{x}_0^0)$ and apply Lemma 4 we obtain

$$\begin{aligned}
&\frac{1}{NT}\sum_{i=1}^{N}\sum_{r=0}^{T}\mathbb{E}\left\|\nabla\mathcal{L}_i^r\right\|^2\\
&\leq\frac{C_7(f(\mathbf{x}_0^0)-f(\mathbf{x}_0^T))}{T}+\frac{C_7(1+8L\eta)}{L}\epsilon_1\\
&+\eta^2(1-p)(N-1)C_7(1-C_8^{1/(1-p)})^2p^2\\
&\times\frac{(1+L\eta)^2(2L\eta+p(3+L\eta))^2}{N(1-C_8)^2}(G^2+\epsilon_1).
\end{aligned}\tag{43}$$

Finally we bound $\left\|\nabla f(\mathbf{x}_0^r)\right\|^2$ by

$$\begin{aligned}
&\left\|\nabla f(\mathbf{x}_0^r)\right\|^2\leq\ 2\left\|\nabla f(\mathbf{x}_0^r)-\frac{1}{N}\sum_{i=1}^{N}\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2\\
&\leq\frac{4}{N}\sum_{i=1}^{N}\left\|\nabla f_i(\mathbf{x}_0^r)-\nabla f_i(\mathbf{x}_i^r)\right\|^2\\
&\quad+4\left\|\frac{1}{N\eta}\sum_{i=1}^{N}(\eta\lambda_i^r+\mathbf{x}_i^r-\mathbf{x}_{0,i}^r)\right\|^2+\frac{2}{N}\sum_{i=1}^{N}\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2\\
&\overset{(a)}{\leq}\frac{4L^2}{N}\sum_{i=1}^{N}\|\mathbf{x}_0^r-\mathbf{x}_i^r\|^2+\frac{2}{N}\sum_{i=1}^{N}\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2\\
&=\frac{4L^2}{N}\sum_{i=1}^{N}\|\nabla_{\lambda_i}\mathcal{L}_i^r\|^2+\frac{2}{N}\sum_{i=1}^{N}\|\nabla_{\mathbf{x}_i}\mathcal{L}_i^r\|^2\leq\ \frac{4L^2}{N}\sum_{i=1}^{N}\|\nabla\mathcal{L}_i^r\|^2,
\end{aligned}\tag{44}$$

where in (a) we use the same argument in (28) and (29).

Therefore, set $p=0$ Theorem 1 is proved, and when $p\neq0$, Theorem 3 is proved. During the proof, we need all $C_2,\ldots,C_7,C_8>0$, therefore, $0<\eta<\frac{\sqrt{5}-1}{4L}$.

Finally, let us note that if the local problems are solved with SGD, then the local problem needs to be solved such that the condition (17) holds true. As no other information of the local solvers except error term $\mathbf{e}_i^r$ is used in the proof, the proofs and results of FedPD with SGD as local solver will not change much, except that all the results hold in expectation. Therefore we skip the proof for the SGD version.

C. Constants used in the proofs

In this subsection we list all the constants $C_2,\ldots,C_8$ used in the proof of Theorem 1 and Theorem 3.

$$\begin{aligned}
&C_2\geq4L^2C_7,\qquad C_3=C_8,\qquad C_4\geq\frac{C_2(1+8L\eta)}{L},\\
&C_5=8C_2,\qquad C_6\geq3\max\{(\frac{1+L\eta}{\eta})^2,(1+2\eta)^2,L^2\eta^2\},\\
&C_7=2C_6/\min\{\frac{1-2L\eta-4L^2\eta^2}{2\eta},\frac{1}{2\eta},\frac{1+8L\eta}{2L}\},\\
&C_8=\frac{p(1+L\eta)+L\eta}{1-L\eta},
\end{aligned}$$

we can see that when $0<\eta<\frac{\sqrt{5}-1}{4L}$, all the terms are positive.

APPENDIX C
PROOF FOR LEMMA 1– LEMMA 3

A. Proof of Lemma 1

We divide the left hand side (LHS) of (21), i.e., $\mathcal{L}_i^{r+} - \mathcal{L}_i^r$, into the sum of three parts (where $\mathcal{L}_i^{r+}, \mathcal{L}_i^r$ are defined in (25)):

$$\begin{aligned}\mathcal{L}_i^{r+} - \mathcal{L}_i^r &= \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r \\ &+ \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) \\ &+ \mathcal{L}_i^{r+} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}).\end{aligned} \tag{45}$$

We bound the first difference by first applying A1 to $-f(\cdot)$:

$$-f_i(\mathbf{x}_i^r) \le -f_i(\mathbf{x}_i^{r+1}) + \left\langle -\nabla f_i(\mathbf{x}_i^{r+1}), \mathbf{x}_i^r - \mathbf{x}_i^{r+1} \right\rangle + \frac{L}{2}\left\|\mathbf{x}_i^r - \mathbf{x}_i^{r+1}\right\|^2,$$

and obtain the following series of inequalities:

$$\begin{aligned}&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) - \mathcal{L}_i^r \\ &\le \left\langle \nabla f_i(\mathbf{x}_i^{r+1}), \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{L}{2}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 \\ &+ \left\langle \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 - \frac{1}{2\eta}\left\|\mathbf{x}_i^r - \mathbf{x}_{0,i}^r\right\|^2 \\ &\overset{(a)}{=} \left\langle \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle + \frac{L}{2}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 \\ &+ \frac{1}{2\eta}\left\langle \mathbf{x}_i^{r+1} + \mathbf{x}_i^r - 2\mathbf{x}_{0,i}^r, \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle \\ &\overset{(b)}{=} \left\langle \nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r), \mathbf{x}_i^{r+1} - \mathbf{x}_i^r \right\rangle \\ &+ \frac{L}{2}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 - \frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 \\ &\overset{(c)}{\le} \frac{1}{2L}\left\|\nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r)\right\|^2 \\ &+ \frac{L}{2}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 - \frac{1 - L\eta}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 \\ &\overset{(d)}{\le} -\frac{1 - 2L\eta}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_i^r\right\|^2 + \frac{\epsilon_1}{2L}.\end{aligned} \tag{46}$$

In the above derivation, in (a) we use the fact that $\|a\|^2 - \|b\|^2 = \langle a+b, a-b\rangle$ when vector a, b has the same length to the last two terms; in (b) we split the last term into $2\mathbf{x}_i^{r+1} - 2\mathbf{x}_{0,i}^r$ and $-\mathbf{x}_i^{r+1} + \mathbf{x}_i^r$; in (c) we use the fact that $\langle a, b\rangle \le \frac{L}{2}\|a\|^2 + \frac{1}{2L}\|b\|^2$); in (d) we apply the fact that $\mathbf{x}_i^{r+1}$ is the inexact solution; see (20).

Then we bound the second difference in (45) by the following:

$$\begin{aligned}&\mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^r) \\ &= \left\langle \lambda_i^{r+1} - \lambda_i^r, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\ &\overset{(a)}{=} \left\langle \lambda_i^{r+1} - \lambda_i^r, \eta(\lambda_i^{r+1} - \lambda_i^r) \right\rangle = \eta\left\|\lambda_i^{r+1} - \lambda_i^r\right\|^2,\end{aligned} \tag{47}$$

where (a) directly comes from the update rule of λ_i^{r+1}.

Further we bound the third difference in (45) by the following:

$$\begin{aligned}&\mathcal{L}_i^{r+} - \mathcal{L}_i(\mathbf{x}_i^{r+1}, \mathbf{x}_{0,i}^r, \lambda_i^{r+1}) \\ &= \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+} \right\rangle - \left\langle \lambda_i^{r+1}, \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r \right\rangle \\ &+ \frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+}\right\|^2 - \frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r\right\|^2 \\ &\overset{(a)}{=} \left\langle \lambda_i^{r+1}, \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle + \frac{1}{2\eta}\left\langle 2\mathbf{x}_i^{r+1} - 2\mathbf{x}_{0,i}^{r+} + \mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r, \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle \\ &= \left\langle \frac{1}{\eta}(\eta\lambda_i^{r+1} + \mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^{r+}), \mathbf{x}_{0,i}^r - \mathbf{x}_{0,i}^{r+} \right\rangle - \frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2 \\ &\overset{(b)}{=} -\frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+} - \mathbf{x}_{0,i}^r\right\|^2,\end{aligned} \tag{48}$$

where, in (a), we use the same reasoning as in (46) (a) and (b); in (b) we apply the update rule of $\mathbf{x}_{0,i}^{r+}$ in the FedPD algorithm, which implies that the first term becomes zero.

Finally we sum up (46), (47), (48) and Lemma 1 is proved.

B. Proof of Lemma 2

First we derive the relation between $\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\|$ for arbitrary $i \ne j$ and $\triangle^r$ by using the definition of ϵ_1 (20):

$$\begin{aligned}&\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\| \\ &\overset{(20)}{=} \left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r - \eta(\nabla f_i(\mathbf{x}_i^{r+1}) + \lambda_i^r - \mathbf{e}_i^{r+1} - \nabla f_j(\mathbf{x}_j^{r+1}) - \lambda_j^r + \mathbf{e}_j^{r+1})\right\| \\ &\le \left\|\mathbf{x}_{0,i}^r - \mathbf{x}_{0,j}^r\right\| + \eta\left\|\nabla f_i(\mathbf{x}_i^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1})\right\| \\ &+ \eta\left\|\lambda_i^r - \lambda_j^r\right\| + \eta(\left\|\mathbf{e}_i^{r+1}\right\| + \left\|\mathbf{e}_j^{r+1}\right\|) \\ &\overset{(a)}{\le} \triangle\mathbf{x}_0^r + \eta\left\|\nabla f_i(\mathbf{x}_i^{r+1}) - \nabla f_i(\mathbf{x}_j^{r+1}) + \nabla f_i(\mathbf{x}_j^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1})\right\| \\ &+ \eta\left\|\lambda_i^r - \lambda_j^r\right\| + 2\eta\sqrt{\epsilon_1} \\ &\overset{(b)}{\le} \triangle\mathbf{x}_0^r + L\eta\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\| + \eta\left\|\nabla f_i(\mathbf{x}_j^{r+1}) - \nabla f_j(\mathbf{x}_j^{r+1})\right\| \\ &+ \eta\left\|\lambda_i^r - \lambda_j^r\right\| + 2\eta\sqrt{\epsilon_1} \\ &\overset{(c)}{\le} \triangle\mathbf{x}_0^r + L\eta\left\|\mathbf{x}_i^{r+1} - \mathbf{x}_j^{r+1}\right\| + \eta G + \eta\left\|\lambda_i^r - \lambda_j^r\right\| + 2\eta\sqrt{\epsilon_1} \\ &\overset{(d)}{=} \frac{1}{1 - L\eta}\triangle\mathbf{x}_0^r + \frac{\eta}{1 - L\eta}G + \frac{\eta}{1 - L\eta}\left\|\lambda_i^r - \lambda_j^r\right\| + \frac{2\eta}{1 - L\eta}\sqrt{\epsilon_1},\end{aligned} \tag{49}$$

where in (a) we plug the definition of $\triangle\mathbf{x}_0^r$ and $\mathbf{e}_i^{r+1}$; in (b) we use A1; (c) comes form A4; in (d) we move the second term to the left and divide both side by $1 - L\eta$.

Then we bound the difference $\left\|\lambda_i^r - \lambda_j^r\right\|$ by plugging in the expression of λ_i^r in (20), and note that $\lambda_i^r + \frac{1}{\eta}(\mathbf{x}_i^{r+1} - \mathbf{x}_{0,i}^r) = \lambda_i^{r+1}$, we have:

$$\begin{aligned}&\left\|\lambda_i^r - \lambda_j^r\right\| \\ &= \left\|-\nabla f_i(\mathbf{x}_i^r) + \mathbf{e}_i^r + \nabla f_j(\mathbf{x}_j^r) - \mathbf{e}_j^r\right\| \\ &\overset{(a)}{\le} \left\|\nabla f_i(\mathbf{x}_i^r) - \nabla f_i(\mathbf{x}_j^r)\right\| + \left\|\nabla f_i(\mathbf{x}_j^r) - \nabla f_j(\mathbf{x}_j^r)\right\| + 2\sqrt{\epsilon_1} \\ &\overset{(b)}{\le} L\left\|\mathbf{x}_i^r - \mathbf{x}_j^r\right\| + G + 2\sqrt{\epsilon_1} \\ &\overset{(c)}{\le} L\triangle\mathbf{x}^r + G + 2\sqrt{\epsilon_1},\end{aligned} \tag{50}$$

where (a) and (b) follow the same argument in (a), (b) and (c) of (49) ; in (c) we plug in the definition of $\triangle\mathbf{x}^r$.

Next we bound the difference $\left\|\mathbf{x}_{0,i}^{r+1} - \mathbf{x}_{0,j}^{r+1}\right\|$. With probability $1 - p$ the aggregation step has just been done at iteration r, $\mathbf{x}_{0,i}^{r+1} = \mathbf{x}_{0,j}^{r+1}$.With probability p, they are

not equal, then we take expectation with communication probability p, and obtain

$$\begin{aligned}&\mathbb{E}_{r+1}\left\|\mathbf{x}_{0,i}^{r+1}-\mathbf{x}_{0,j}^{r+1}\right\|\\&=p\left\|\mathbf{x}_i^{r+1}-\mathbf{x}_j^{r+1}+\eta(\lambda_i^{r+1}-\lambda_j^{r+1})\right\|\\&\leq p\left\|\mathbf{x}_i^{r+1}-\mathbf{x}_j^{r+1}\right\|+p\eta\left\|\lambda_i^{r+1}-\lambda_j^{r+1}\right\|\\&\overset{(a)}{\leq} p(1+L\eta)\triangle\mathbf{x}^{r+1}+p\eta(G+2\sqrt{\epsilon_1}),\end{aligned}\tag{51}$$

where in (a) we plug in the definition of $\triangle\mathbf{x}^{r+1}$ and (50). As these relations hold true for arbitrary (i,j) pairs, they are also true for the maximum of $\left\|\mathbf{x}_i^{r+1}-\mathbf{x}_j^{r+1}\right\|$ and $\left\|\mathbf{x}_{0,i}^{r+1}-\mathbf{x}_{0,j}^{r+1}\right\|$.

Therefore stacking (49) and (51) and plug in (50), we have

$$\begin{aligned}\triangle\mathbf{x}^{r+1}\leq&\frac{1}{1-L\eta}(L\eta\triangle\mathbf{x}^r+\triangle\mathbf{x}_0^r)+\frac{2\eta}{1-L\eta}(G+2\sqrt{\epsilon_1}),\\\mathbb{E}_{r+1}\triangle\mathbf{x}_0^{r+1}\leq&p\frac{1+L\eta}{1-L\eta}(L\eta\triangle\mathbf{x}^r+\triangle\mathbf{x}_0^r)+p\frac{\eta(3+L\eta)}{1-L\eta}(G+2\sqrt{\epsilon_1}).\end{aligned}\tag{52}$$

Rewrite it into matrix form then we complete the proof of Lemma 2.

C. Proof of Lemma 3

Let us first recall that the definition of local AL is given below:

$$\mathcal{L}_i(\mathbf{x}_i,\mathbf{x}_0,\lambda_i)\triangleq\ f_i(\mathbf{x}_i)+\langle\lambda_i,\mathbf{x}_i-\mathbf{x}_0\rangle+\frac{1}{2\eta}\left\|\mathbf{x}_i-\mathbf{x}_0\right\|^2.$$

Similar to (48), we have

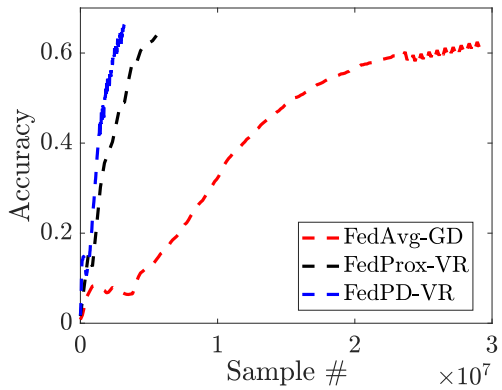
$$\begin{aligned}\mathcal{L}_i^{r+}-\bar{\mathcal{L}}_i^{r+1}=&\left\langle\lambda_i^{r+1},\mathbf{x}_i^{r+1}-\mathbf{x}_{0,i}^{r+}\right\rangle-\left\langle\lambda_i^{r+1},\mathbf{x}_i^{r+1}-\bar{\mathbf{x}}_0^{r+1}\right\rangle\\&+\frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1}-\mathbf{x}_{0,i}^{r+}\right\|^2-\frac{1}{2\eta}\left\|\mathbf{x}_i^{r+1}-\bar{\mathbf{x}}_0^{r+1}\right\|^2\\\overset{(a)}{=}&-\frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+}-\bar{\mathbf{x}}_0^{r+1}\right\|^2\\\overset{(b)}{=}&-\frac{1}{2\eta}\left\|\mathbf{x}_{0,i}^{r+}-\frac{1}{N}\sum_{j=1}^N\mathbf{x}_{0,j}^{r+}\right\|^2\\=&-\frac{1}{2\eta}\left\|\frac{1}{N}\sum_{j=1}^N(\mathbf{x}_{0,i}^{r+}-\mathbf{x}_{0,j}^{r+})\right\|^2\\\overset{(c)}{\geq}&-\frac{1}{2\eta N}\sum_{j\neq i}\left\|\mathbf{x}_{0,i}^{r+}-\mathbf{x}_{0,j}^{r+}\right\|^2\\\overset{(d)}{\geq}&-\frac{N-1}{2\eta N}(\triangle\mathbf{x}_0^{r+1})^2,\end{aligned}\tag{53}$$

where (a) follows the same argument in (48); in (b),we plug in the definition of $\bar{\mathbf{x}}_0^{r+1}$; in (c) we use Jensen's inequality and we bound the term with $\triangle\mathbf{x}_0^{r+1}$. Then the lemma is proved.

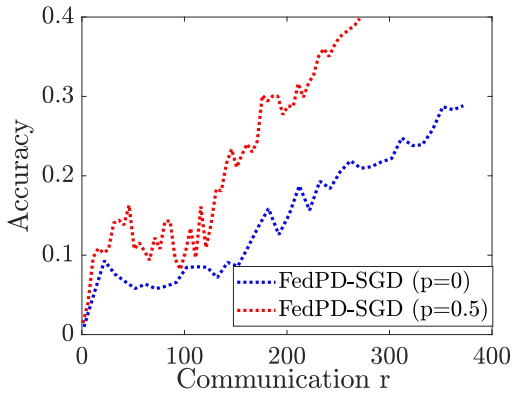
D. Proof of Lemma 4

Applying A1, we have:

$$\begin{aligned}f_i(\mathbf{x}_0^r)\leq&\ f_i(\mathbf{x}_i^r)+\langle\nabla f_i(\mathbf{x}_i^r),\mathbf{x}_0^r-\mathbf{x}_i^r\rangle+\frac{L}{2}\left\|\mathbf{x}_0^r-\mathbf{x}_i^r\right\|^2\\\overset{(20)}{=}&\ \mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_0^r,\lambda_i^r)-\langle\mathbf{e}_i^r,\mathbf{x}_0^r-\mathbf{x}_i^r\rangle-\frac{1-L\eta}{2\eta}\left\|\mathbf{x}_0^r-\mathbf{x}_i^r\right\|^2\\\leq&\ \mathcal{L}_i(\mathbf{x}_i^r,\mathbf{x}_0^r,\lambda_i^r)+\frac{\epsilon_1}{2L}-\frac{1-2L\eta}{2\eta}\left\|\mathbf{x}_0^r-\mathbf{x}_i^r\right\|^2.\end{aligned}\tag{54}$$



(a) The testing accuracy of FedAvg-GD, FedProx-VR and FedPD-VR with respect to the number of samples.



(b) The testing accuracy of FedPD-SGD with $R=1$ and $R=2$ with respect to the number of communications.

Fig. 3: The convergence result of the algorithms on training neural network for handwriting character classification.

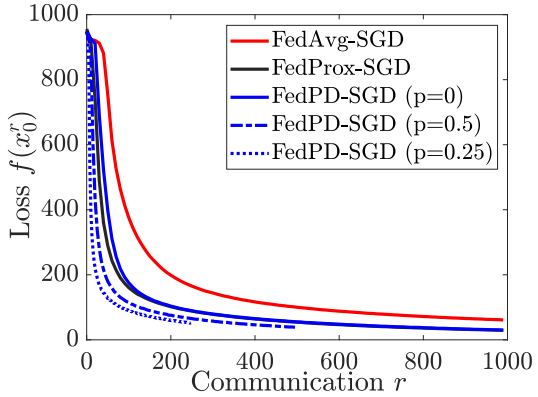
Taking an average over N agents we are able to prove Lemma 4.

Appendix D
Additional Numerical Results

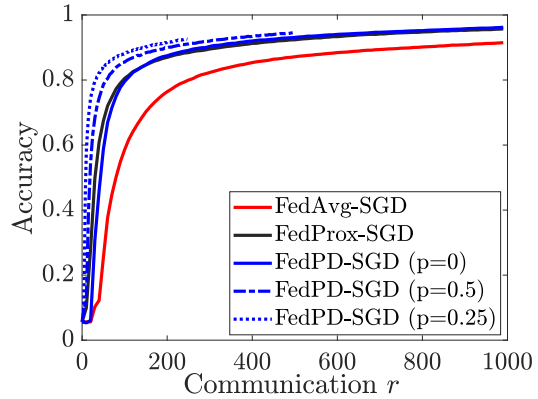
A. Handwritten Character Classification

In the second experiment, we compare FedPD with FedAvg and FedProx on the FEMNIST data set [25]. The FEMNIST data set collects the handwritten characters, including numbers 1–10 and the upper- and lower-case letters A–Z and a–z, from different writers and is separated by the writers, therefore the data set naturally preserves non-i.i.d-ness.

The entire data set contains 805,000 samples collected from 3,550 writers. In our experiments, we use the data collected from 100 writers with an average of 300 samples per writer and the size of the whole data set is 29,214. We set the number of agent $N=90$, the first ten agents are assigned with data from two writers, and the rest of the agents are assigned with data form one writer. Therefore, the data distribution is neither i.i.d. nor balanced. We use the neural network given in [25] as the training model, which consists of 2 convolutional layers and two fully connected layers. The output layer has 62

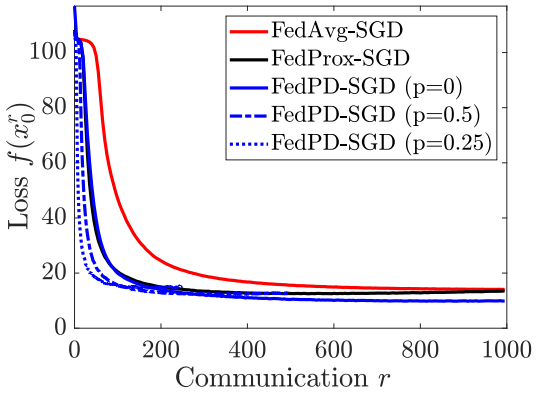


(a) The loss value of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

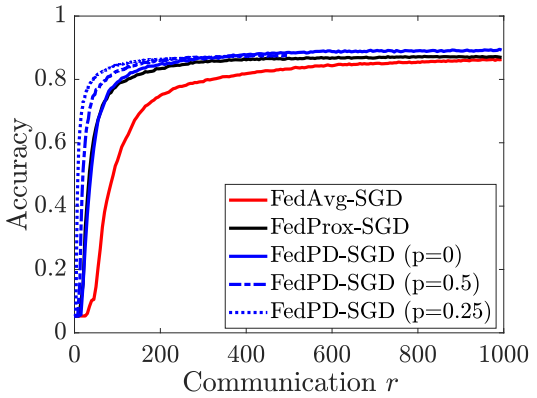


(b) The training accuracy of of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

Fig. 4: The convergence results of the algorithms on training neural networks on the federated handwritten characters classification problem.



(a) The testing loss value of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.



(b) The testing accuracy of of FedAvg-SGD, FedProx-SGD and FedPD-SGD with respect to the number of communication rounds.

Fig. 5: The convergence results of the algorithms on training neural networks on the federated handwritten characters classification problem with test data set.

neurons that matches the number of classes in the FEMNIST data set.

The numerical results shown in Fig. 3 in the main text were generated by running MATLAB codes on Amazon Web Services (AWS), with Intel Xeon E5-2686 v4 CPUs. In the training phase, we train the CNN model with FedAvg, FedProx and FedPD. In Fig. 3(a), for FedAvg, we use gradient descent for $Q = 8$ local update steps between each communication rounds; to solve the local problem for FedProx, we use SARAH with $Q = 20$ local steps; we use FedPD with Oracle II, computing full gradient every $I = 20$ communication rounds and perform $Q = 2$ local steps between two communication rounds. The hyper-parameters we use for FedAvg is $\eta = 0.005$; for FedProx we use $\rho = 1$ and stepsize $\eta = 0.01$; for FedPD we use $\eta = 100$ and $\gamma = 400$. In Fig. 3(b), we use FedPD with Oracle I, with $Q = 20$, $\eta = 100$ and $\gamma = 400$ and the mini-batch size 2. We set the communication saving to $p = 0$ and $p = 0.5$.

The results shown in Fig. 4 were generated by running Python codes (using the the PyTorch package[1]) with AMD EPYC 7702 CPUs and an NVIDIA V100 GPU.

In the training phase, we train with FedProx, FedAvg and FedPD with a total $T = 1000$ outer iterations. The local problems are solved with SGD for $Q = 300$ local iterations and the mini-batch size in evaluating the stochastic gradient is 2. The stepsize choice for FedAvg, FedProx and FedPD are 0.001, 0.01 and 0.01, the hyper-parameter of FedProx is $\rho = 1$ and for FedPD $\eta = 1$. In the experiment, we set the communication saving for FedPD to be $p = 0$, $p = 0.5$ and $p = 0.25$. Note that we also tested FedAvg with larger stepsize 0.01, but the algorithm becomes unstable, and its performance degrages significantly. As shown in Fig. 4 and 5, FedAvg is slower than FedPD and FedProx, while FedProx has similar performance as FedPD when $R = 1$. Further, we can see that as the frequency of communication of FedPD decreases, the final accuracy decreases and the final loss increases. However, the drop of accuracy is not significant, so FedPD is able to achieve a better performance with the same number of communication rounds.

[1]PyTorch: An Imperative Style, High-Performance Deep Learning Library, https://pytorch.org/