Towards Optimal Communication Complexity in Distributed Non-Convex Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of distributed stochastic non-convex optimization with intermittent communication. We consider the full participation setting where M machines work in parallel over R communication rounds, as well as the partial participation setting where m machines are sampled each round. We propose and analyze a new algorithm that improves existing methods by requiring fewer and lighter variance reduction operations. We also present lower bounds, showing our algorithm is either *optimal* or *almost optimal* in most settings. Numerical experiments demonstrate the superior performance of our algorithm.

1 Introduction

We consider the following distributed optimization problem with M machines:

$$\min_{x \in \mathbb{R}^d} F(x) := \frac{1}{M} \sum_{m=1}^{M} F_m(x), \tag{1.1}$$

where F_m, which denotes the objective on machine m, is a non-convex function for all m, as is the average objective F. We want to solve this problem in the *intermittent communication* (IC) setting [1, 2] where the machines work in parallel and are allowed to make K oracle calls between two communication rounds for R consecutive rounds. The IC setting has been widely studied [3, 4, 5, 6, 7, 8, 9, 10, 11, 2, 12] over the past decade, and many recent studies have focused on the problem with non-convex and heterogeneous objectives [13, 14, 15] which are common in cross-device federated learning (FL) [16, 17]. Towards this end, several algorithms [18, 19, 20, 21, 22, 23], all involving *local updates* (à la local-SGD [3, 16]), have been proposed and analyzed under assumptions bounding the heterogeneity of machines' objectives. Although these algorithms demonstrate promising empirical performances, it remains elusive whether these algorithms provably dominate the embarrassingly parallelizable alternatives, i.e., *mini-batch* variants of the optimal sequential algorithms [24, 25, 26] (a.k.a. centralized algorithms).

Until very recently, the situation was similar even in the simpler *homogeneous* setting where F_m's are all identical and convex, and Woodworth et al. [2] showed that the optimal algorithm often does not require local updates at all. Even when F_m's are not identical, for high levels of heterogeneity, accelerated mini-batch SGD [27] is optimal [28]. Should we expect something similar in the non-convex setting? Or, can we prove that in some regime *local-update* algorithms improve over the naive centralized baselines?

In this paper, we start by noting that in the absence of any heterogeneity assumption (c.f., Section 2), centralized algorithms already have the best worst-case convergence guarantee. Thus, only when the heterogeneity is low can the *local-update* algorithms potentially have an advantage. This was the motivation behind some of the recent works [18, 21, 22]. However, in the absence of any lower

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

bound that explicitly depends on the heterogeneity parameter (such as in [15, 29]), it is not possible to definitively claim such an improvement. To alleviate this, **(C.1)** we provide novel communication complexity lower bounds, under the assumption that F_m's have bounded first-order or second-order heterogeneity (see Section 2). We show that centralized algorithms [24, 25, 26] can never achieve this optimal communication complexity, and most of the existing *local-update* algorithms cannot attain it either.

Table 1: Comparison of convergence rate for different algorithms in the intermittent communication setting. ζ and τ corresponds to the first and second order heterogeneity (see Section 2) of the problem. Note that $\tau \leq 2L$ and can be much smaller than L. The results suppress only and all numerical constants. * Although BVR-L-SGD and Algorithm 1 have the same fast convergence rate, BVR-L-SGD requires each client to compute large batch gradients for many rounds of communications and is thus less communication efficient in practice (see Section 3 for a detailed comparison). S stands for a stochastic oracle ($\sigma \geq 0$) and E stands for an exact oracle ($\sigma = 0$); MIMELITEMVR uses both.

Method	Convergence Rate, i.e. $\mathbb{E}\left\|\nabla F(\widehat{x})\right\|^2 \leq$	Oracle
Full Participation Setting		
SCAFFOLD/Mini-batch SGD [18]	$\frac{\Delta L}{R} + \left(\frac{\sigma^2 \Delta L}{MKR}\right)^{1/2}$	S
Mini-batch STORM ([26], Theorem D.1)	$\frac{\Delta L}{R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{MKR}\right)^{2/3}$	S
Lower Bound (Centralized) Theorem 2.1	$\frac{\Delta L}{R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{MKR}\right)^{2/3}$	-
STEM [20]	$(\Delta L + \sigma^2 + \zeta^2)\left(\frac{1}{R} + \frac{1}{(MKR)^{2/3}}\right)$	S
BVR-L-SGD [22]* CE-LSGD, Theorem 3.1	$\frac{\Delta \tau}{R} + \frac{\Delta L}{\sqrt{K}R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{MKR}\right)^{2/3}$	S
CE-LGD Theorem 3.1	$\frac{\Delta \tau}{R} + \frac{\Delta L}{KR}$	E
Lower Bound Theorem 3.2	$\min\left\{\frac{\Delta \tau}{R}, \frac{\zeta^2}{R}\right\} + \frac{\Delta L}{KR} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{MKR}\right)^{2/3}$	-
Partial Participation Setting		
FEDAVG [13]	$\frac{\Delta L}{R} + \left(\frac{\sigma^2 \Delta L}{mKR}\right)^{1/2} + \left(\frac{(\zeta\sqrt{K}+\sigma)\Delta L}{\sqrt{K}R}\right)^{2/3}$	S
Mini-batch STORM (Theorem 3.3)	$\frac{\Delta L}{R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{m\sqrt{K}R}\right)^{2/3} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta L}{mR}\right)^{2/3}$	S
Lower Bound (Centralized) Theorem B.6	$\frac{\Delta L}{R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma \Delta L}{MKR}\right)^{2/3} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta L}{mR}\right)^{2/3}$	-
MIMEMVR [21]	$\frac{\Delta \tau}{R} + \frac{\Delta L}{KR} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta \tau}{\sqrt{m}R}\right)^{2/3}$	E
MIMELITEMVR [21]	$\frac{\Delta \tau}{R} + \frac{\Delta L}{KR} + \frac{\zeta^2+\sigma^2}{R} + \left(\frac{(\zeta+\sigma)\Delta \tau}{R}\right)^{2/3}$	S+E
CE-LSGD Theorem 3.3	$\frac{\Delta \tau}{R} + \frac{\Delta L}{\sqrt{K}R} + \frac{\sigma^2}{mKR} + \left(\frac{\sigma \Delta L}{mKR}\right)^{2/3} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta \tau}{mR}\right)^{2/3} + \left(\frac{\Delta(\sigma\tau + L\zeta)}{m\sqrt{K}R}\right)^{2/3}$	S
CE-LGD Theorem 3.3	$\frac{\Delta \tau}{R} + \frac{\Delta L}{KR} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta \tau}{mR}\right)^{2/3}$	E
Lower Bound Theorem 3.4	$\min\left\{\frac{\Delta \tau}{R}, \frac{\zeta^2}{R}\right\} + \frac{\Delta L}{KR} + \frac{\sigma^2}{mKR} + \left(\frac{\sigma \Delta L}{mKR}\right)^{2/3} + \frac{\zeta^2}{mR} + \left(\frac{\zeta \Delta L}{mKR}\right)^{2/3}$	-

Working with these limitations in mind, our main contribution is **(C.2)** a new algorithm **CE-LSGD** that we show to be **min-max optimal when equipped with exact gradient oracles** and near-optimal when provided with stochastic gradient oracles (c.f., Section 2). Our algorithm, like many other *local-update* algorithms, uses variance-reduction techniques [24, 26] but requires both fewer and

lighter "heavy-batch" operations compared to the existing methods (see discussion in Section 3). **(C.3)** We also analyze CE-LSGD in the partial client participation setting, which is of particular interest in cross-device federated learning (FL)[17] where there is an extremely large number of clients. Not only does CE-LSGD improve over the best-known communication complexity, but it is the only algorithm that *doesn't require exact oracle* queries for variance reduction and still manages to be nearly optimal. Furthermore, if **endowed with exact oracles, CE-LSGD is almost min-max optimal even in the partial participation setting**. Thus our results provably demonstrate optimality of local update methods, at least in some regimes. Even in simpler convex settings, we don't know of any local update method (exact or stochastic) known to be min-max optimal in the heterogeneous setting [30, 15]. We summarize our results and the comparison to some important baselines in Tables 1 and 2. **(C.4)** As an auxiliary contribution, we provide a variant of our algorithm which uses stochastic hessian vector product oracles and is thus useful for settings where only a single copy of the model can be stored on the edge device. We also empirically compare our method against centralized and *local-update* algorithms, demonstrating faster convergence and better communication efficiency.

Table 2: Comparison of optimal communication and oracle complexity required by different algorithms to attain $\mathbb{E}\|\nabla F(\widehat{x})\|_2^2 \leq \epsilon$. ζ and τ are the heterogeneity (see Section 2) of the problem. $\tau \leq 2L$ and can be much smaller than L. The results suppress only numerical constants and assume that $\epsilon^{1/2} \preceq \min\{(\sigma/M) \cdot (\tau/L), \Delta L/\sigma, \Delta\tau/\zeta, \zeta/m\}$, i.e., ϵ is small enough. The first inequality ensures we are in the green regime described in figure 1 and guarantees that $\Delta LM/\epsilon \preceq \sigma\Delta L/\epsilon^{3/2}$; the second inequality guarantees that $\sigma^2/\epsilon \preceq \sigma\Delta L/\epsilon^{3/2}$; the third inequality guarantees that $\zeta^2/m\epsilon \preceq \zeta\Delta\tau/m\epsilon^{3/2}$; and the fourth inequality guarantees that $\Delta L/\epsilon \leq \zeta\Delta L/m\epsilon^{3/2}$.

Methods	Communication Complexity (R)	Oracle Complexity (N)
Full Participation Setting		
SCAFFOLD/Mini-batch SGD [18]	$\frac{\Delta L}{\epsilon}$	$\frac{\sigma^2\Delta L}{\epsilon^2}$
Mini-batch STORM ([26], Theorem D.1)	$\frac{\Delta L}{\epsilon}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}}$
Lower Bound (Centralized) (Theorem 2.1)	$\frac{\Delta L}{\epsilon}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}}$
STEM [20]	$\frac{\Delta L+\sigma^2+\zeta^2}{\epsilon}$	$\frac{(\Delta L)^{3/2}+\sigma^3+\zeta^3}{\epsilon^{3/2}}$
BVR-L-SGD [22] CE-LSGD, Theorem 3.1	$\frac{\Delta\tau}{\epsilon}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}}$
Lower Bound Theorem 3.2	$\min\left\{\frac{\Delta\tau}{\epsilon}, \frac{\zeta^2}{\epsilon}\right\}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}}$
Partial Participation Setting		
FedAvg [13]	$\frac{\zeta\Delta L}{\epsilon^{3/2}}$	$\frac{\sigma^2\Delta L}{\epsilon^2}$
Mini-batch STORM (Theorem D.1)	$\frac{\zeta\Delta L}{m\epsilon^{3/2}}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}} \cdot \left(1+\frac{\sigma}{\zeta}\right)$
Lower Bound (Centralized) (Theorem 2.1)	$\frac{\zeta\Delta L}{m\epsilon^{3/2}}$	$\frac{\sigma\Delta L}{\epsilon^{3/2}}$
MimeMVR [21]	$\frac{\zeta\Delta\tau}{m^{1/2}\epsilon^{3/2}}$	Uses Exact Oracle
MimeLiteMVR [21]	$\frac{\zeta^2+\sigma^2}{\epsilon} + \frac{(\zeta+\sigma)\Delta\tau}{\epsilon^{3/2}}$	Uses Exact Oracle
CE-LSGD Theorem 3.3	$\frac{\zeta\Delta\tau}{m\epsilon^{3/2}}$	$\frac{\zeta\Delta L}{\epsilon^{3/2}} \cdot \frac{L}{\tau} + \frac{\sigma\Delta L}{\epsilon^{3/2}} \cdot \left(1+\frac{\sigma\tau}{\zeta L}\right)$
Lower Bound Theorem 3.4, [31, 32]	$\min\left\{\frac{\Delta\tau}{\epsilon}, \frac{\zeta^2}{\epsilon}\right\} + \frac{\zeta^2}{m\epsilon}$	$\frac{\zeta\Delta L}{\epsilon^{3/2}} + \frac{\sigma\Delta L}{\epsilon^{3/2}}$

Notation. We use $\mathcal{B}$ to denote the index set and $|\mathcal{B}|$ to denote its cardinality. For $x \in \mathbb{R}^d$, we use $\|x\|$ to denote its ℓ_2-norm. For $A \in \mathbb{R}^{d\times d}$, $\|A\|$ denotes the operator norm. $[n]$ denotes the set $\{1, 2, \ldots, n\}$. We use $\approx, \preceq, \succeq$ to denote equality and inequality up to numerical constants.

2 Our Setting and the Centralized Baselines

In this section, we introduce some definitions and assumptions that will be used in our analysis. Our goal is to find an ϵ-approximate stationary point of F, i.e., a point $x \in \mathbb{R}^d$ such that $\mathbb{E}[\|\nabla F(x)\|^2] \leq \epsilon$, where the expectation is w.r.t. any randomness in the choice of x. We consider client objectives in the class $\mathcal{F}(L)$ of differentiable and L-smooth functions. We also make assumptions that relate the functions of different clients to one another. These are typically known as assumptions on the *"heterogeneity"* of the problem, and we consider two classes of problems.

Definition 1. *Assume $\{F_m \in \mathcal{F}(L)\}_{m=1}^M$ are first-order ζ-heterogeneous, i.e., $\sup_{x\in\mathbb{R}^d} \sum_{m=1}^M \|\nabla F_m(x) - \nabla F(x)\|^2/M \leq \zeta^2$. And for all $\Delta \geq 0$, $F(0) - \inf_{x\in\mathbb{R}^d} F(x) \leq \Delta$, i.e., the average objective has bounded sub-optimality at zero. Then we say that $\{F_m\}_{m\in[M]} \in \boldsymbol{\mathcal{F}_M^1(L, \Delta, \zeta)}$.*

Definition 2. *Assume twice-differentiable $\{F_m \in \mathcal{F}(L)\}_{m=1}^M$ are second-order τ-heterogeneous, i.e., $\sup_{m\in[M],x\in\mathbb{R}^d} \|\nabla^2 F_m(x) - \nabla^2 F(x)\| \leq \tau$. And for all $\Delta \geq 0$, $F(0) - \inf_{x\in\mathbb{R}^d} F(x) \leq \Delta$, i.e., the average objective has bounded sub-optimality at zero. Then we say that $\{F_m\}_{m\in[M]} \in \boldsymbol{\mathcal{F}_M^2(L, \Delta, \tau)}$.*

We assume that each machine has access to the following multi-point oracle [32] [Section 5.3, 2].

Definition 3. *Given a function $G \in \mathcal{F}(L, \Delta)$, $\mathcal{O}_G^{n,L,\sigma} : (\mathbb{R}^d)^n \times \mathcal{Z} \to (\mathbb{R})^n \times (\mathbb{R}^d)^n$ is a multi-point stochastic first order oracle if for some distribution $\mathcal{D}$ on $\mathcal{Z}$, and for all $x_1, \ldots, x_n \in \mathbb{R}^d$, the oracle samples a random seed $z \sim \mathcal{D}$ and returns estimators $\mathcal{O}_G^{n,L,\sigma}(x_1, \ldots, x_n, z) = (\{f(x_i; z)\}_{i\in[n]}, \{g(x_i; z)\}_{i\in[n]})$ such that $\forall i \in [n]$, $\mathbb{E}_{z\sim\mathcal{D}}(f(x_i; z), g(x_i; z)) = (G(x_i), \nabla G(x_i))$ and $\mathbb{E}_{z\sim\mathcal{D}} \|g(x_i; z) - \nabla G(x_i)\|^2 \leq \sigma^2$. Furthermore, the unbiased gradients satisfy mean smoothness, i.e., for all $x, y \in \mathbb{R}^d$, $\mathbb{E}_{z\sim\mathcal{D}} [\|g(x; z) - g(y; z)\|] \leq L \|x - y\|$.*

Finally, we consider a generalization of zero respecting algorithms denoted by $\mathcal{A}_{ZR}$ (see Appendix A) in the intermittent communication (IC) setting. This class captures a very wide variety of distributed optimization algorithms, including mini-batch SGD, accelerated mini-batch SGD, local SGD, and all the variance-reduction algorithms. Algorithms which are not distributed zero-respecting are those whose iterates have components in directions about which the algorithm has no information, meaning that in some sense, it is just "wild guessing". We also denote the class of centralized algorithms in $\mathcal{A}_{ZR}$ by $\mathcal{A}_{ZR}^{cent}$ (see Appendix A). This includes algorithms such as mini-batch SARAH [24] and mini-batch STORM [26]. We first present a lower bound result applicable to centralized algorithms.

Theorem 2.1 (Centralized Lower Bound)**.** *For all $\tau, \Delta, \zeta, \sigma \geq 0$, and $2L \geq \tau$, every algorithm $A \in \mathcal{A}_{ZR}^{cent}$ optimizing a problem in $\mathcal{F}_M^1(L, \Delta, \zeta) \cup \mathcal{F}_M^2(L, \Delta, \tau)$, with access to an oracle $\mathcal{O}_{F_m}^{2,L,\sigma}$ over $R \succeq 1$ communication rounds must output x_R^A such that,*

$$\mathbb{E}\left[\left\|\nabla F(x_R^A)\right\|^2\right] \succeq \frac{\Delta L}{R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma\Delta L}{MKR}\right)^{2/3}.$$

The proof of this theorem follows the known oracle complexity lower bounds [31, 32] and is included in Appendix B. This theorem shows that when heterogeneity is large i.e., $\tau \approx L$, mini-batch SARAH/STORM already achieve the optimal communication and oracle complexity (see Table 1). Therefore, most existing methods including FEDAVG[16], SCAFFOLD [18] and FEDPAGE [19] can not outperform these centralized baselines as their analyses do not benefit from small heterogeneity. At the same time, this highlights the limitation of the centralized baselines, which also do not improve with lower heterogeneity. Certain existing *local-update* algorithms such as MIMEMVR [21] and BVR-L-SGD [22] can indeed improve upon centralized algorithms in the low heterogeneity regime. In the next section, we quantify this improvement, and show that our algorithm strictly dominates the centralized baselines and almost matches our lower bound for algorithms in $\mathcal{A}_{ZR}$.

3 Our Algorithm and Min-max Optimality

In this section we present our communication-efficient algorithm abbreviated CE-LSGD and illustrate it in Algorithm 1. Note that for $m \in [M]$, we use the notation $\nabla F_{m,\mathcal{B}^m}(x) := \sum_{l \in \mathcal{B}^m} g(x; z_l \sim \mathcal{D}_m)/|\mathcal{B}^m|$ to denote the stochastic mini-batch gradient obtained by querying $\mathcal{O}^{2,L,\sigma}_{F_m}$ for $|\mathcal{B}^m|$ many times, and $\nabla F_{m,z}(x)$ to denote a single stochastic gradient $g(x; z \sim \mathcal{D}_m)$.

At each iteration of Algorithm 1, we need **two rounds** of communication, i.e., two back and forth communications between the server and all clients. Our method uses the extra round of communication, i.e., line 4 to line 9, to update the variance reduced gradient v_r using the current and previous server models, x_r, x_{r-1} respectively. In the following discussion, we will use the iteration number R and communication complexity of Algorithm 1 interchangeably.

Algorithm 1 Communication Efficient Local Stochastic Gradient Descent (CE-LSGD)

input Initialization x_0, communication round R, local steps K, step size η, momentum $\beta \in [0, 1]$, initial batch size b_0

1: Let $x_{-1} = x_0$
2: **for** $r = 0, 1, \ldots, R-1$ **do**
3: **if** $r = 0$ set $\beta = 1, K = 1, B = b_0$ **else** set β, K as the input values, $B = K$
4: **Communicate (send)** (x_r, x_{r-1}) to clients
5: **on client** $m \in [M]$ **do**
6: Compute $\nabla F_{m,\mathcal{B}^m_r}(x_r)$ and $\nabla F_{m,\mathcal{B}^m_r}(x_{r-1})$, where $|\mathcal{B}^m_r| = B$
7: **Communicate (rec)** $\left(\nabla F_{m,\mathcal{B}^m_r}(x_r), \nabla F_{m,\mathcal{B}^m_r}(x_{r-1})\right)$ to the server
8: **end on client**
9: $v_r = \frac{1}{M}\sum_{m=1}^{M} \nabla F_{m,\mathcal{B}^m_r}(x_r) + (1-\beta)\left(v_{r-1} - \frac{1}{M}\sum_{m=1}^{M} \nabla F_{m,\mathcal{B}^m_r}(x_{r-1})\right)$
10: **Communicate (send)** (x_r, v_r) to client $\widetilde{m}_r$, where $\widetilde{m}_r \sim Unif\left([M]\right)$
11: **on client** $\widetilde{m}$ **do**
12: $w^{\widetilde{m}_r}_{r+1,1} := w^{\widetilde{m}_r}_{r+1,0} := x_r, v^{\widetilde{m}_r}_{r,0} := v_r$
13: **for** $k = 1, \ldots, K$ **do**
14: $v^{\widetilde{m}_r}_{r,k} = \nabla F_{\widetilde{m}_r, z^{\widetilde{m}_r}_{r,k}}(w^{\widetilde{m}_r}_{r+1,k}) + v^{\widetilde{m}_r}_{r,k-1} - \nabla F_{\widetilde{m}_r, z^{\widetilde{m}_r}_{r,k}}(w^{\widetilde{m}_r}_{r+1,k-1})$
15: $w^{\widetilde{m}_r}_{r+1,k+1} = w^{\widetilde{m}_r}_{r+1,k} - \eta v^{\widetilde{m}_r}_{r,k}$
16: **end for**
17: **Communicate (rec)** $\left(w^{\widetilde{m}_r}_{r+1,K+1}\right)$ to the server
18: **end on client**
19: Let $x_{r+1} = w^{\widetilde{m}_r}_{r+1,K+1}$
20: **end for**

output Choose $\widetilde{x}$ uniformly from $\{w^{\widetilde{m}_r}_{r,k}\}_{r \in [R], k \in [K]}$

At the core of our proposed method is the variance reduction term v_r and the local gradient estimator $v^m_{r,k}$ (lines 9 and 14 in Algorithm 1). The construction of the local gradient estimator is motivated by the variance reduction technique of SARAH [24, 25]. Intuitively, the estimation error between the proposed local gradient estimator $v^m_{r,k}$ and the full gradient $\nabla F(w^m_{r+1,k})$, i.e., $\mathbb{E}\|v^m_{r,k} - \nabla F(w^m_{r+1,k})\|$, can be decomposed into two dominating terms: $\mathbb{E}\|v_r - \nabla F(x_r)\|^2$ and $\tau^2 K \sum_{k=1}^{K} \mathbb{E}\|w^m_{r+1,k} - w^m_{r+1,k-1}\|^2$. The first term is the estimation error between the variance reduction term v_r and the full gradient $\nabla F(x_r)$. Since v_r is updated based on the momentum-based variance reduction technique [26], this estimation error is dominated by $L^2\mathbb{E}\|x_r - x_{r-1}\|^2$, which approaches zero as the algorithm converges. Similarly, the second term $\mathbb{E}\|w^m_{r+1,k} - w^m_{r+1,k-1}\|^2$ approaches zero as the algorithm converges and the τ factor controls the benefit we can obtain from small heterogeneity. Intuitively, for smaller values of τ, we can make more local updates and the algorithm converges faster. We note that, our method reduces to mini-batch STORM if we choose the number of local updates K to be one (see Appendix D). Our formal convergence result is as follows:

Theorem 3.1 (Simplified Statement)**.** *Suppose* $\{F_m\}_{m \in [M]} \in \mathcal{F}^2_M(L, \Delta, \tau)$ *for* $L, \Delta, \tau \geq 0, \tau \leq 2L$ *then,*

(a) if each client $m \in [M]$ has a stochastic oracle $\mathcal{O}_{F_m}^{2,L,\sigma}$, and assuming $\frac{\Delta L}{R} \preceq \frac{\sigma^2}{\sqrt{MK}}$, then the output $\widetilde{x}$ of Algorithm 1 using $\beta = \max\left\{\frac{1}{R}, \frac{(\Delta L)^{2/3}(MKb)^{1/3}}{\sigma^{4/3}R^{2/3}}\right\}$, $b_0 = KR$, and $\eta = \min\left\{\frac{1}{L}, \frac{1}{K\tau}, \frac{(\beta M)^{1/2}}{LK^{1/2}}\right\}$ satisfies,

$$\mathbb{E}\|\nabla F(\widetilde{x})\|^2 \preceq \frac{\Delta\tau}{R} + \frac{\Delta L}{\sqrt{K}R} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma\Delta L}{MKR}\right)^{2/3};$$

(b) if each client $m \in [M]$ has a deterministic oracle $\mathcal{O}_{F_m}^{2,L,0}$, then the output $\widetilde{x}$ of Algorithm 1 using $\beta = 1$ and $\eta = \min\left\{\frac{1}{L}, \frac{1}{K\tau}\right\}$ satisfies,

$$\mathbb{E}\|\nabla F(\widetilde{x})\|^2 \preceq \frac{\Delta\tau}{R} + \frac{\Delta L}{KR}.$$

In Appendix C, we derive this result by carefully tuning β, b_0. We show that the convergence rate attained by our algorithm is *almost optimal* by proving the following lower bound result.

Theorem 3.2. *For all $L, \sigma, \tau, \Delta, \zeta \geq 0,\ \tau \leq 2L,\ \zeta \leq \sqrt{\Delta L}$, every algorithm $A \in \mathcal{A}_{zr}$, optimizing a problem in $\mathcal{F}_M^1(L, \Delta, \zeta) \cup \mathcal{F}_M^2(L, \Delta, \tau)$ with $K > 0$ intermittent accesses to two-point first-order oracles $\{\mathcal{O}_{F_m}^{2,L,\sigma}\}_{m\in[M]}$ on all the machines, outputs x_R^A after $R \geq c_1$ rounds such that,*

$$\mathbb{E}\left[\left\|\nabla F(x_R^A)\right\|^2\right] \succeq \min\left\{\frac{\zeta^2}{R}, \frac{\Delta\tau}{R}\right\} + \frac{\Delta L}{KR} + \frac{\sigma^2}{MKR} + \left(\frac{\sigma\Delta L}{MKR}\right)^{2/3}.$$

We can make two observations by comparing the upper and lower bounds for problems in $\mathcal{F}_M^2(L, \Delta, \tau)$. First, in the deterministic setting ($\sigma = 0$), our upper bound matches the lower bound; **hence CE-LSGD is min-max optimal**. Thus, our result improves over all the existing results in this setting, including MIMEMVR [21]. Second, in the stochastic setting ($\sigma > 0$), our algorithm's upper bound is optimal except for the second term in Theorem 3.1, which has a $\Delta L/(\sqrt{K}R)$ factor as opposed to the $\Delta L/(KR)$ term in the lower bound. In our analysis in Appendix C we highlight what causes this gap. We conjecture that the lower bound is tight, and like the deterministic setting, our algorithm is also min-max optimal in the stochastic setting.

Our construction for Theorem 3.2 uses the non-convex hard instance proposed by Carmon et al.[31] and splits it across different machines to get a communication complexity lower bound. This idea has been used previously to give lower bounds in the heterogeneous setting [33, 15, 34]. We prove the result in Appendix B. From looking at Table 1, we can note that BVR-L-SGD [22] also attains a similar upper bound as our method.[1] In Appendix C, we show that with deterministic oracle BVR-L-SGD also attains the min-max optimal rate. This is not surprising, knowing that several variance-reduced algorithms [25, 26, 24] are simultaneously optimal even in the sequential setting. Still, our method requires fewer and lighter variance reduction operations, which leads to better scalability from the algorithmic design perspective. In the next section, we carefully examine the difference between these methods.

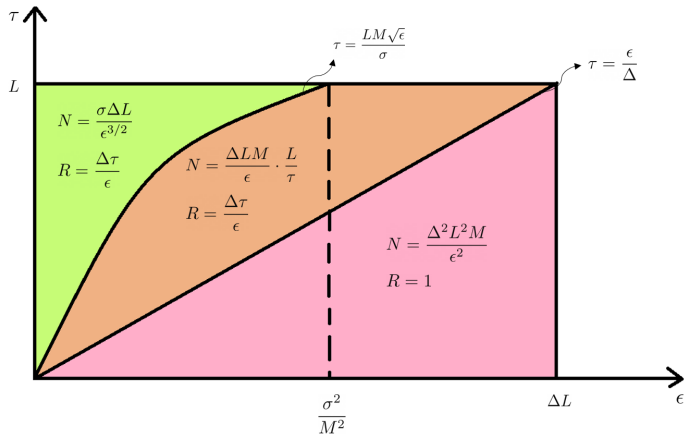
3.1 The Perspective of Reducing Communication

So far, we have looked at convergence rates in the intermittent communication model, where K, R is fixed. However, another perspective is reducing the communication complexity to the minimum possible with the minimum required oracle complexity. Both these complexities can be expressed in terms of ϵ using the convergence guarantees we showed, where we want to attain an ϵ-approximate stationary point. This view is often more useful when communication rounds comprise the bulk of the required physical time. This scenario is common in FL, where devices become available intermittently, which delays the synchronous updates. With this in mind, we summarize the communication and oracle complexities attained by both our method and BVR-L-SGD [22] in Figure 1 when optimizing with stochastic oracles. Notice that the figure has three different regimes based on the relative scaling of τ versus ϵ. We focus on the green regime characterized by $\epsilon^{1/2} \in (0, \tau\sigma/(LM)]$. This regime is of most practical interest in deep learning, where modern over-parameterized models often drive ϵ to really small values. And when ϵ is small enough, this regime covers a wide range of values of τ.[2].

[1] Note that the authors in [22] don't show how to get this rate, but we do so in Appendix C.

[2] We talk about the other regimes while giving the full statement of Theorem 3.1 in Appendix C

Figure 1: Illustration of the best communication complexity R and oracle complexity N that our method can obtain for different ϵ and τ. Green regime: Our method can obtain the optimal communication and oracle complexities. Orange regime: Our method can obtain the optimal communication using a larger oracle complexity. Red regime: Our method only needs one round of communication using a larger oracle complexity.



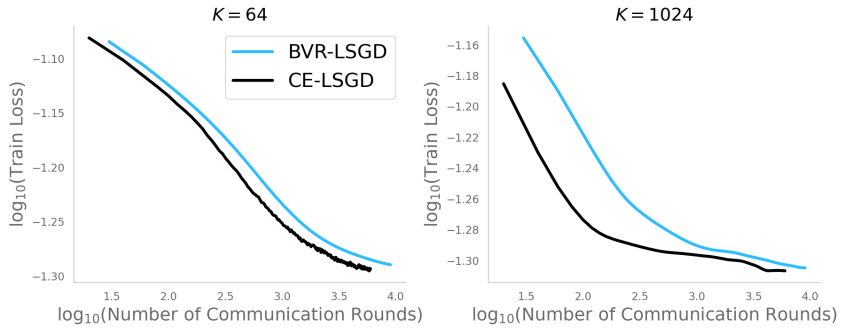


Figure 2: Training loss of CE-LSGD and BVR-L-SGD on CIFAR-10 data-set versus the number of communication rounds in the intermittent communication setting with different local-updates K. We use $M = 10$ machines, and synthetically generate heterogeneous data-sets (see Section 4) with $q = 0.1$. All oracle queries use a mini-batch of size $b = 16$, i.e., each machine has Kb oracle queries between two communication rounds. We note that our method has a faster convergence in all the settings, which highlights its communication efficiency. Fixed step-sizes η for both the methods were tuned in $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ (to obtain best loss) following [22], our method set the momentum $\beta = 0.3$, $b_{max}^{our} = K$, while $b_{max}^{BVR} = 5000$ according to [22].

In the green regime, both CE-LSGD and BVR-L-SGD require $K = \sigma L/(\tau M \epsilon^{1/2})$ local steps to achieve the optimal communication and oracle complexities. However, BVR-L-SGD requires multiple heavy-batch stochastic gradient computations on each machine with batch size b_{max}. In particular, for BVR-L-SGD, we have $\rho_{\text{BVR}} = b_{max}/K = \sigma\tau/(L\epsilon^{1/2})$, which suggests that for $S = L\Delta/\sigma\sqrt{\epsilon}$ communication rounds, it requires each machine to compute ρ_{BVR} times heavier batch stochastic gradients compared to the other communication rounds. As for CE-LSGD, we have $b_0 = \sigma^3/(L\Delta M\epsilon^{1/2})$, which gives us $\rho_{\text{our}} = b_0/K = \sigma^2\tau/(L^2\Delta)$. This suggests that our method only requires each machine to compute ρ_{our} times larger batch stochastic gradient, and that too only once. Furthermore, $\rho_{\text{our}}/\rho_{\text{BVR}} = \sigma\epsilon^{1/2}/(L\Delta) \leq 1$. Thus, the size of our large batch gradient is also smaller than the one for BVR-L-SGD and **our method has fewer and lighter heavy-batch operations**.

Suppose one implements both these methods in the intermittent communication model, i.e., by breaking the large batch computation across multiple rounds, with local budget $K = \sigma L/(\tau M \epsilon^{1/2})$. In that case, the effective communication complexity of both methods is $\Delta\tau/\epsilon$, and this subtle difference gets washed away. However, in Figure 2, we show that this equivalence up to numerical constants doesn't hold in practice, where our method converges faster than BVR-L-SGD. In Table 2, we summarize the communication and oracle complexities attained by different algorithms in the green regime.

3.2 The Partial Participation Setting

In settings such as cross-device federated learning [17], there are often tens of thousands of clients (think of android mobile users) and it is not feasible to consider training on all of the clients synchronously. It is more natural to consider a partial sampling of clients for each communication round. More formally, we can re-state our distributed optimization problem as follows:

$$\min_{x\in\mathbb{R}^d} F(x) := \mathbb{E}_{n\sim\mathcal{P}}\left[F_n(x)\right], \tag{3.1}$$

where $\mathcal{P}$ is a probability distribution on the clients, and we assume at each communication round, we can sample m clients independently from $\mathcal{P}$. This is w.l.o.g. because in settings like cross-device FL, there is an extremely large number of devices, and most devices participate only once. We also need to modify the IC setting: during each communication round, $S_r \sim \mathcal{P}^m$ clients participate, and each queries their oracle K times. This setting has been considered in several works [18, 21]. We can adapt Algorithm 1 to this setting by communicating with only m clients at each round and using m_0 clients for the first round to initialize the variance reduction term (see Appendix E). Specifically, we can guarantee the following convergence rate where $\mathcal{F}^1_{\mathcal{P}}(L,\Delta,\zeta)$ and $\mathcal{F}^2_{\mathcal{P}}(L,\Delta,\tau)$ are natural generalizations of $\mathcal{F}^1_M(L,\Delta,\zeta)$ and $\mathcal{F}^2_M(L,\Delta,\tau)$ to the partial participation setting, formally defined in Appendix A.

Theorem 3.3. *Suppose for all n in support of $\mathcal{P}$, $F_n \in \mathcal{F}^1_{\mathcal{P}}(L,\Delta,\zeta)\cap\mathcal{F}^2_{\mathcal{P}}(L,\Delta,\tau)$ then,*

(a) if each client n has a stochastic oracle $\mathcal{O}^{2,L,\sigma}_{F_n}$, and assuming that $\frac{\Delta\tau}{R}+\frac{\Delta L}{\sqrt{K}R}\preceq\frac{\sigma^2}{\sqrt{m}K}+\frac{\zeta^2}{\sqrt{m}}$, the output $\widetilde{x}$ of Algorithm 1 using $b_0=K$, $m_0=mR$, $\beta=\max\left\{\frac{1}{R},\left(\frac{\Delta(\tau+L/\sqrt{K})\sqrt{m}}{R(\sigma^2/K+\zeta^2)}\right)^{2/3}\right\}$, and $\eta = \min\left\{\frac{1}{L},\frac{1}{K\tau},\frac{1}{\sqrt{K}L},\frac{\sqrt{\beta m}}{\sqrt{K}L},\frac{\sqrt{\beta m}}{\tau K}\right\}$ satisfies

$$\mathbb{E}\|\nabla F(\widetilde{x})\|^2 \preceq \frac{\Delta\tau}{R}+\frac{\Delta L}{\sqrt{K}R}+\frac{\sigma^2}{mKR}+\left(\frac{\sigma\Delta L}{mKR}\right)^{2/3}+\frac{\zeta^2}{mR}+\left(\frac{\zeta\Delta\tau}{mR}\right)^{2/3}+\left(\frac{\Delta(\sigma\tau+L\zeta)}{m\sqrt{K}R}\right)^{2/3};$$

(b) if each client n has a deterministic oracle $\mathcal{O}^{2,L,0}_{F_n}$, and assuming that $\frac{\Delta\tau}{R}\preceq\frac{\zeta^2}{\sqrt{m}}$, then the output $\widetilde{x}$ of Algorithm 1 using $m_0=mR$, $\beta=\max\left\{\frac{1}{R},\left(\frac{\Delta\tau\sqrt{m}}{\zeta^2R}\right)^{2/3}\right\}$, and $\eta=\min\left\{\frac{1}{L},\frac{1}{K\tau},\frac{\sqrt{\beta m}}{\tau K}\right\}$, satisfies

$$\mathbb{E}\|\nabla F(\widetilde{x})\|^2 \preceq \frac{\Delta\tau}{R}+\frac{\Delta L}{KR}+\frac{\zeta^2}{mR}+\left(\frac{\zeta\Delta\tau}{mR}\right)^{2/3}.$$

As far as we know, CE-LSGD is the only one in the partial participation setting that uses just stochastic oracles for implementing variance reduction with local updates. In tables 2 and 1 we show that with a deterministic oracle, CE-LGD attains a strictly faster convergence rate than the best-known result MIMEMVR [21], which also uses a deterministic oracle. More specifically, CE-LGD's communication complexity $\zeta\Delta\tau/m\epsilon^{3/2}$ improves over the communication complexity of $\zeta\Delta\tau/\sqrt{m}\epsilon^{3/2}$ for MIME-MVR. Furthermore, we have the following lower bound result to show that the convergence rate of our algorithm is *almost optimal*.

Theorem 3.4. *For all $L,\sigma,\tau,\Delta,\zeta\geq 0$, $\tau\leq 2L$, $\zeta\leq\sqrt{\Delta L}$ every algorithm $A\in\mathcal{A}_{zr}$ optimizing a problem in $\mathcal{F}^1_{\mathcal{P}}(L,\Delta,\zeta)\cup\mathcal{F}^2_{\mathcal{P}}(L,\Delta,\tau)$ with $K>0$ intermittent accesses to two-point first-order oracles $\{\mathcal{O}^{2,L,\sigma}_{F_n}\}_{n\in support(\mathcal{P})}$ on all the machines outputs x^A_R after $R\succeq 1$ rounds such that,*

$$\mathbb{E}\left[\left\|\nabla F(x^A_R)\right\|^2\right] \succeq \min\left\{\frac{\Delta\tau}{R},\frac{\zeta^2}{R}\right\}+\frac{\Delta L}{KR}+\frac{\sigma^2}{mKR}+\left(\frac{\sigma\Delta L}{mKR}\right)^{2/3}+\frac{\zeta^2}{mR}+\left(\frac{\zeta\Delta L}{mKR}\right)^{2/3}.$$

Note that in the deterministic setting (i.e., $\sigma=0$), the only gap is in the last term of CE-LGD's upper bound. We **conjecture that CE-LGD is optimal even in the partial partipation setting, and our lower bound can be improved.** There are more gaps in the stochastic setting, and we hope to remove them in future work.

4 Numerical Experiments

We evaluate the performance of our method by optimizing a two-layer fully connected network for multi-class classification on the CIFAR-10 [35] data-set. Since we are in the heterogeneous setting, we need to artificially generate a data-set. We follow the same data processing procedure as in [22]. We first make sure that all the ten classes in CIFAR-10 have the same number of samples (roughly around 5000), and assign $q \times 100\%$ of class m's samples to client $m \in [10]$ where q is chosen from $\{0.1, 0.35, 0.6, 0.85\}$. For each class m, we evenly split the remaining $(1 - q) \times 100\%$ samples to the other 9 clients except client m. Thus, q controls the heterogeneity of our data-set, with small q corresponding to small heterogeneity.

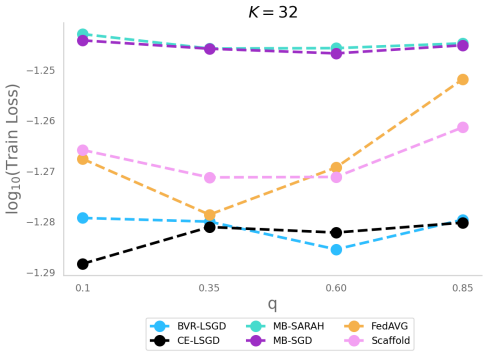


Figure 3: Comparing CE-LSGD to centralized and local-update methods, for fixed $K = 32$ and varying heterogeneity controlled by q on CIFAR-10 [35] data-set. Like Figure 2 we use mini-batch size $b = 16$ for each oracle query. Thus each method makes Kb oracle queries every round per machine. All the methods for different q are tuned separately, following a similar hyper-parameter search as in Figure 2.

We perform two different experiments. In the first experiment, we directly compare our method, i.e., CE-LSGD, with BVR-L-SGD in the intermittent communication setting (see Figure 2). We observe that while both the methods converge to a similar quality of solution eventually, our method is more communication efficient. In the second experiment, we compare our method with BVR-L-SGD [22] as well as FedAvg [16], SCAFFOLD [18], MB-SARAH [24] and MB-SGD [5] for the same number of updates/iterations. The last two methods are centralized baselines, and we use the local computation to compute a mini-batch stochastic gradient. We again observe that CE-LSGD and BVR-L-SGD have comparable performance which is better than all the other methods.

5 Discussion and Open Problems

In this paper, we provide a new communication-efficient local update algorithm CE-LSGD and analyze it in the full and partial client participation setting with intermittent communication. In the deterministic setting, i.e., with access to exact oracles, our algorithm is the optimal and almost optimal for full and partial participation respectively. Moreover, when equipped with stochastic oracles our algorithm attains the best-known convergence guarantees to our knowledge in both models of participation. Our lower bound results provide a much-needed baseline to measure algorithmic developments in non-convex distributed optimization and help us characterize CE-LGD's optimality.

In Appendix F, we provide an extension of CE-LSGD which uses a stochastic Hessian vector product oracle [12, 36] instead of a multi-point oracle, and recovers similar optimal communication complexity. This is relevant for memory-constrained online settings where it might not be feasible to preserve several copies of a model on the client device for making simultaneous queries for variance reduction algorithms.

Our work leaves some open questions. In the deterministic partial participation setting, is our lower bound loose? We believe this is most likely the situation. In the stochastic full participation setting, does CE-LSGD attain the min-max optimal rate? There is a $1/\sqrt{K}$ gap in our optimization term for any model of participation which vanishes in the deterministic setting. Thus we don't know if the gap is *"real"* or a result of our analysis. We discuss this further in appendix C. Finally there are several gaps in the stochastic partial participation setting. We believe some of these can be alleviated by improving the deterministic lower bound, but others seem imply that our analysis is loose. Finally we don't know if we can use a simple single machine baseline in the red regime in Figure 1? We are not able to show this with only assumptions on τ. However, in Appendix C, we discuss this regime when ζ is also small and show how a single machine baseline can be used here. In general, it is unclear if the set of problems in $\mathcal{F}_M^1(L, \Delta, \zeta)$ are indeed simpler than $\mathcal{F}_M^2(L, \Delta, \tau)$. We hope to tackle these questions in our future work.

References

[1] Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. *Advances in neural information processing systems*, 31, 2018.

[2] Blake E Woodworth, Brian Bullins, Ohad Shamir, and Nathan Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pages 4386–4437. PMLR, 2021.

[3] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. Parallelized stochastic gradient descent. *Advances in neural information processing systems*, 23, 2010.

[4] Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. *Advances in neural information processing systems*, 24, 2011.

[5] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1), 2012.

[6] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. *Advances in Neural Information Processing Systems*, 26, 2013.

[7] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[8] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. PMLR, 2014.

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

[10] Aymeric Dieuleveut and Kumar Kshitij Patel. Communication trade-offs for local-sgd with large step size. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pages 10334–10343. PMLR, 2020.

[12] Brian Bullins, Kshitij Patel, Ohad Shamir, Nathan Srebro, and Blake E Woodworth. A stochastic newton algorithm for distributed convex optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

[13] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

[14] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[15] Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020.

[16] H Brendan McMahan, Eider Moore, Daniel Ramage, S Hampson, and B Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data (2016). *arXiv preprint arXiv:1602.05629*, 2016.

[17] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. corr. *arXiv preprint arXiv:1912.04977*, 2019.

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

[19] Haoyu Zhao, Zhize Li, and Peter Richtárik. Fedpage: A fast local stochastic gradient method for communication-efficient federated learning. *arXiv preprint arXiv:2108.04755*, 2021.

[20] Prashant Khanduri, Pranay Sharma, Haibo Yang, Mingyi Hong, Jia Liu, Ketan Rajawat, and Pramod Varshney. Stem: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020.

[22] Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. *arXiv preprint arXiv:2102.03198*, 2021.

[23] Rudrajit Das, Anish Acharya, Abolfazl Hashemi, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061*, 2020.

[24] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621. PMLR, 2017.

[25] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

[26] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

[27] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[28] Blake Woodworth. The minimax complexity of distributed optimization. *arXiv preprint arXiv:2109.00534*, 2021.

[29] Margalit R Glasgow, Honglin Yuan, and Tengyu Ma. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 9050–9090. PMLR, 2022.

[30] Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang. On the unreasonable effectiveness of federated averaging with heterogeneous data. *arXiv preprint arXiv:2206.04723*, 2022.

[31] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

[32] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

[33] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *Advances in neural information processing systems*, 28, 2015.

[34] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*, 2020.

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

[36] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*, pages 242–299. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]