Characterizing the Accuracy-Communication-Privacy Trade-off in Distributed Stochastic Convex Optimization

Sudeep Salgia* Carnegie Mellon
Nikola Pavlovic† Cornell
Yuejie Chi* Carnegie Mellon
Qing Zhao† Cornell

January 6, 2025

Abstract

We consider the problem of differentially private stochastic convex optimization (DP-SCO) in a distributed setting with M clients, where each of them has a local dataset of N i.i.d. data samples from an underlying data distribution. The objective is to design an algorithm to minimize a convex population loss using a collaborative effort across M clients, while ensuring the privacy of the local datasets. In this work, we investigate the accuracy-communication-privacy trade-off for this problem. We establish matching converse and achievability results using a novel lower bound and a new algorithm for distributed DP-SCO based on Vaidya's plane cutting method. Thus, our results provide a complete characterization of the accuracy-communication-privacy trade-off for DP-SCO in the distributed setting.

Keywords: distributed convex optimization, differential privacy, communication complexity, trade-offs

Contents

*Department of Electrical and Computer Engineering, Carnegie Mellon University; {ssalgia,yuejiec}@andrew.cmu.edu.
†Department of Electrical and Computer Engineering, Cornell University; {np358,qz16}@cornell.edu.

1 Introduction

We consider the problem of distributed stochastic convex optimization where M clients, with the aid of a central server, aim to collaboratively minimize a convex function of the form

$$\mathcal{L}(x) = \mathbb{E}_{z\sim\mathcal{P}}[\ell(x; z)] \tag{1}$$

using their local datasets consisting of N i.i.d. samples from the distribution $\mathcal{P}$. Here $x \in \mathcal{X}$ denotes the decision variable where $\mathcal{X}$ is a convex, compact set and $\ell(x; z)$ denotes the loss at point x using the datum z. We study this problem under the additional constraint of ensuring differential privacy [Dwork et al., 2006] of the local datasets at each client. This problem arises in numerous settings and represents a typical scenario for Federated Learning (FL) [McMahan et al., 2017], which has emerged as the de facto approach for collaboratively training machine learning models using a large number of devices coordinated through a central server [Kairouz et al., 2021, Wang et al., 2021].

Designing efficient algorithms for differentially private distributed stochastic convex optimization, also referred to as distributed DP-SCO, requires striking a careful balance between the primary objective of minimizing the optimization error and two competing desiderata — communication cost and privacy.

Communication cost. There is a natural tension between the accuracy and the communication cost of a distributed learning algorithm, as achieving a lower optimization error entails the clients sharing more information, which results in higher communication costs. Communication between the participating clients and the coordinating server is well-known to be the primary bottleneck in distributed learning, particularly in the scenario where clients have bandwidth constraints [Tang et al., 2020, Zhao et al., 2023]. The overall communication cost of a distributed SCO algorithm consists of two parts — the frequency of communication and the size of the message in each communication round. There has been a substantial effort towards designing communication-efficient algorithms in both non-private and private settings, either by reducing the frequency of communication [Gorbunov et al., 2021, Karimireddy et al., 2020, Khaled et al., 2020, Li et al., 2020, 2022a, Liu et al., 2022, McMahan et al., 2017, Zhao et al., 2021], or by using compression/quantization strategies to minimize the message sizes [Agarwal et al., 2018, Ding et al., 2021, Hönig et al., 2022, Jhunjhunwala et al., 2021, Konečný et al., 2016, Li et al., 2022b, Suresh et al., 2017, Wang et al., 2020b, 2024, Zong et al., 2021].

Privacy. Often in various applications, the local data at participating agents contains sensitive information that should remain private and not become publicly available during the learning process. It has been shown that preventing the transfer of the actual data during the learning process is not sufficient to guarantee privacy of the local data and can leak private information during the training process [Zhu et al., 2019]. Thus, it is desirable to provide formal guarantees to protect the private data [Geyer et al., 2017, Kairouz et al., 2021, Wang et al., 2021]. In this work, we consider sophisticated privacy preserving techniques like differential privacy (DP) [Dwork et al., 2006] to ensure the privacy of the local data. In a seminal work, Abadi et al. [2016] proposed the DP-SGD algorithm where they combined SGD with DP techniques to provide formal guarantees on the privacy of the dataset for training deep networks. Since then, numerous optimization algorithms have been proposed that ensure the privacy of the local dataset using DP. At a high level, DP ensures the privacy of the local datasets by introducing uncertainty into the output of the algorithm, which makes it difficult for an adversary to discern private information. This injection of additional uncertainty results in a natural trade-off between privacy and the accuracy of differentially private algorithms.

Fundamental accuracy-communication-privacy trade-off. Existing studies largely focus on designing algorithms that aim to balance accuracy with one of the two desiderata which provides only a partial picture of the three-way trade-off among accuracy, communication, and privacy for the problem of distributed DP-SCO. Moreover, there lack studies that characterize the converse region of this three-way trade-off, leaving the question of the optimality of existing results open. In this work, we aim to study and characterize this three-way trade-off from first principles to provide a fresh perspective and new insights into this fundamental problem.

1.1 Main results

We consider the problem of distributed stochastic convex optimization with M clients, each with a local dataset of N points, under the constraint of $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differential privacy (See Section 2 for the precise definition). In this work, we provide a complete characterization of the accuracy-communication-privacy trade-off for this problem. The accuracy of an optimization algorithm refers to the sub-optimality gap or the excess risk and is measured as $\mathcal{L}(\widehat{x}) - \min_{x\in\mathcal{X}} \mathcal{L}(x)$, where $\widehat{x}$ denotes the output of the algorithm. We summarize the main results of our work below.

- *Lower bound on the accuracy-communication-privacy trade-off:* We derive a novel lower bound on the accuracy of a distributed SCO algorithm as a function of its communication cost. Specifically, we establish that the error rate of any distributed SCO algorithm is at least $\Omega\left(\sqrt{\frac{d^2}{MN\cdot\mathsf{CC}}}\right)$ (ignoring other terms), where CC is the communication cost of the algorithm, measured as the total number of bits transmitted by each agent on average (See Section 2 for the precise definition) and d is the dimension of the decision variable. This implies that any algorithm with order-optimal accuracy incurs a communication cost of $\Omega(d^2)$ bits. This is the *first* result that tightly characterizes the lower bound of communication complexity for *any* distributed optimization algorithm for general convex functions. When combined with existing lower bounds on the accuracy-privacy trade-off, the proposed bound characterizes the converse region of the accuracy-communication-privacy trade-off. In particular, our lower bound implies that the accuracy for any DP-SCO algorithm is at least $\Omega\left(\sqrt{\frac{d^2}{MN\cdot\min\{\mathsf{CC}, dN\varepsilon_{\mathsf{DP}}^2\}}}\right)$, where CC is the communication cost of the algorithm and $\varepsilon_{\mathsf{DP}}$ is the differential privacy parameter. We establish the lower bound by showing that solving a convex optimization problem is at least as hard as solving d mean estimation problems. In contrast, existing lower bounds rely on the straightforward reduction of convex optimization to estimation of an unknown vector in d dimensions. This is the first result that establishes that convex optimization is *significantly* harder than mean estimation, tightening existing lower bounds.

- *Achieving the optimal accuracy-communication-privacy trade-off:* We propose a new distributed DP-SCO algorithm, called CHARTER, that achieves the optimal accuracy-communication-privacy trade-off as dictated by the lower bound. In particular, we show that CHARTER is an $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differentially private algorithm that achieves an excess risk of $\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{MN}} + \frac{\sqrt{d}}{\sqrt{MN}\varepsilon_{\mathsf{DP}}}\right)$ and incurs a communication cost of $\widetilde{\mathcal{O}}(d^2)$.[1] To the best of our knowledge, this is the first algorithm to achieve optimal accuracy for the problem of distributed DP-SCO. This is also the first algorithm to achieve order-optimal communication cost even in the non-private setting. Our proposed algorithm, CHARTER, departs from the family of gradient descent methods and builds upon the classical plane cutting methods [Anstreicher, 1997, Vaidya, 1996]. This paradigm shift is the key piece of the puzzle that allows us to achieve the optimal three-way trade-off, particularly along the dimension of communication complexity. The primary observation here is that the gradient descent family adopts an optimization framework that is married to the function landscape. The over-reliance on the function landscape requires more frequent communication to counter the noisy updates, particularly when the landscape is flatter. On the other hand, our plane-cutting-based method adopts a geometric perspective akin to binary search methods which allows for constant progress independent of the function landscape thereby reducing the need for frequent communication.

1.2 Related work

DP-ERM. The problem of empirical risk minimization, or ERM for short, aims to minimize the population loss $\mathcal{L}(x)$ by minimizing the sample loss function $\widehat{\mathcal{L}}(x) = \frac{1}{N}\sum_{n=1}^{N} \ell(x; z_n)$ for a given dataset $\mathcal{D} = \{z_n\}_{n=1}^{N}$. The problem of DP-ERM has been extensively studied in the centralized setting and the upper and lower bounds on the accuracy of DP-ERM are well-known [Bassily et al., 2014, Chaudhuri and Monteleoni, 2008, Chaudhuri et al., 2011, Iyengar et al., 2019, Jain et al., 2012, Ullman, 2015, Wang et al., 2017]. The problem of DP-ERM has also received significant attention in the distributed setting [Ding et al., 2021, Huang et al.,

[1] Here, $\widetilde{\mathcal{O}}(\cdot)$ denotes the order up to logarithmic factors.

2015, Jayaraman et al., 2018, Li et al., 2022b, Murata and Suzuki, 2023, Phuong and Phong, 2022, Triastcyn et al., 2021, Wang et al., 2020b, Zhang et al., 2020]. However, solutions of ERM are known to result in poor generalization. In particular, it has been shown that solutions of ERM lead to a sub-optimal error of $\Omega(\sqrt{d/N})$ for the problem of SCO, across a large class of functions [Feldman, 2016]. Consequently, these results are necessarily sub-optimal for the problem of DP-SCO.

DP-SCO. The gap between DP-ERM and DP-SCO was first addressed in the centralized setting by Bassily et al. [2019], where the authors propose a new algorithm that leverages the uniform stability of SGD [Bousquet and Elisseeff, 2002] and achieves the order-optimal accuracy of $\mathcal{O}\left(\frac{1}{\sqrt{N}} + \frac{\sqrt{d}}{N\varepsilon_{\mathsf{DP}}}\right)$ for the problem of SCO. Since then, there have been a series of studies [Arora et al., 2022, Asi et al., 2021, Bassily and Sun, 2023, Bassily et al., 2021, Choquette-Choo et al., 2024, Feldman et al., 2020, Han et al., 2022, Kulkarni et al., 2021, Liu and Asi, 2024, Song et al., 2021, Wang et al., 2020a, 2023] that further analyze the problem of DP-SCO and propose efficient algorithms with optimal performance for a wide range of scenarios in the centralized setting. However, these results do not have an analogous version for the distributed setting. In the non-private setting, the problem of distributed SCO has been extensively studied numerous algorithms have been proposed that achieve the order-optimal accuracy of $\mathcal{O}\left(1/\sqrt{MN}\right)$ [Khaled et al., 2020, Reisizadeh et al., 2020, Woodworth et al., 2020a,b].

Communication-efficient algorithms. As mentioned earlier, there is an extensive line of work that focuses on designing communication-efficient algorithms both in non-private and private (DP-ERM) settings. The best-known bound on the communication complexity of distributed SCO algorithms is $\mathcal{O}(d\sqrt{MN})$ [Haddadpour et al., 2021, Reisizadeh et al., 2020].

There is a line of work that studies lower bounds on communication complexity for various distributed learning problems like mean estimation, distribution estimation, and linear bandits [Barnes et al., 2020b, Braverman et al., 2016, Duchi et al., 2014, Salgia and Zhao, 2023]. For the problem of convex optimization, Korhonen and Alistarh [2021] derive a lower bound of $\Omega(d)$ by a reduction to mean estimation. Tsitsiklis and Luo [1987] derive a lower bound of $\Omega(d)$ and they conjecture a lower bound of $\Omega(d^2)$. They also partially prove their conjecture for a restricted class of communication models. Vempala et al. [2020] also derive a lower bound of $\Omega(d^2)$ for the problem of linear regression in the non-stochastic setting where each client only has access to a partial set of features. For the gradient descent family of algorithms without acceleration, a lower bound of $\Omega(d\sqrt{MN})$ on the communication complexity was shown by Arjevani and Shamir [2015], Huang et al. [2022], Woodworth et al. [2018]. Similar results for distributed learning over general networks for the gradient descent family of algorithms under the span assumption were obtained in Scaman et al. [2017, 2019]. In this work, we establish a lower bound of $\Omega(d^2)$ for the general stochastic convex optimization that holds for *all* algorithms. Our bound improves upon the best-known bound of $\Omega(d)$ and also resolves the conjecture in Tsitsiklis and Luo [1987]. Arjevani and Shamir [2015] also derived a $\Omega(d^2)$ lower bound for the class of algorithms that perform empirical risk minimization of quadratic functions with a single round of communication, where analogous result for a more general class of algorithms that allow for multiple rounds of communication and operate over general convex functions was left as an open question. The proposed lower bound in this work also addresses this open question.

Notation: The notations $f(x) = \mathcal{O}(g(x))$ and $f(x) \lesssim g(x)$ both imply that the relation $f(x) \leq Cg(x)$ holds for all x for some constant $C > 0$, independent of x. Similarly, $f(x) = \Omega(g(x))$ and $f(x) \gtrsim g(x)$ both imply that the relation $f(x) \geq cg(x)$ holds for all x for some constant $c > 0$, independent of x. We use $\widetilde{\mathcal{O}}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to denote the corresponding relations above up to logarithmic factors. For $n \in \mathbb{N}$, we use the shorthand $[n] := \{1, 2, \ldots, n\}$. For any event $\mathcal{E}$, we use $\mathcal{E}^c$ to denote its complement. For any two vectors $v, w \in \mathbb{R}^d$, $\langle v, w \rangle$ denotes the standard inner product and $\|v\|_2 = \sqrt{\langle v, v \rangle}$ denotes the ℓ_2-norm of vector v.

2 Problem Formulation

Stochastic convex optimization. We consider a distributed learning setup which consists of a single central server and M clients. Each client $m \in \{1, 2, \ldots, M\}$ has access to a local dataset $\mathcal{D}_m = \{z_{m,n}\}_{n=1}^N \in$

$\mathcal{Z}^N$ consisting of N i.i.d. data samples from a distribution $\mathcal{P}_m$ that takes values in a set $\mathcal{Z}$. The objective of the clients is to collaboratively minimize the function:

$$\min_{x\in\mathcal{X}} \mathcal{L}(x) := \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{z\sim\mathcal{P}_m}\left[\ell(x,z)\right], \tag{2}$$

over an input domain $\mathcal{X}$ using their local datasets $\mathcal{D}_m$. Here, $\mathcal{X} \subset \mathbb{R}^d$ is a convex, compact set, and $\ell : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ denotes the loss function of interest. Let $R := \sup\{\|x - y\|_2 \mid x, y \in \mathcal{X}\}$ denote the diameter of the set $\mathcal{X}$. Before moving forward, we outline below some definitions and assumptions that are commonly used in the SCO literature.

Definition 1. *A function f is called L-Lipschitz over $\mathcal{X}$ if for all $x, x' \in \mathcal{X}$, $\|f(x) - f(x')\|_2 \le L\|x - x'\|_2$.*

Definition 2. *Let f be a convex function over a domain $\mathcal{X} \subset \mathbb{R}^d$. The subgradient of f at a point $x \in \mathcal{X}$, denoted by $\partial f(x)$, is given by*

$$\partial f(x) = \{c \in \mathbb{R}^d : f(y) - f(x) \ge c^\top (y - x), \quad \forall y \in \mathcal{X}\}.$$

Assumption 1. *The population loss function $\mathcal{L}(x)$ is convex. For all $x \in \mathcal{X}$ and $c \in \partial\mathcal{L}(x)$, $\|c\|_2 \le 1$.*

Let $\partial\ell(x; z)$ denote the noisy estimate of the sub-gradient at x evaluated using the data point z. We would like to point out that we slightly abuse the notation here for ease of presentation; $\partial\ell(x; z)$ does not necessarily correspond to a subgradient of ℓ as it is not assumed to be a convex function.

Assumption 2. *For all $m \in \{1, 2, \ldots, M\}$ and $x \in \mathcal{X}$, and $z \in \mathcal{P}_m$, $\ell(x; z)$ is σ_f^2-sub-Gaussian random variable and $\partial\ell(x; z)$ is a σ_g^2-sub-Gaussian random vector such that $\mathbb{E}[\partial\ell(x; z)] \in \partial\mathcal{L}(x)$. This implies that for all $v \in \mathbb{R}^d$ with $\|v\|_2 \le 1$ and $\lambda \in \mathbb{R}$, $\mathbb{E}[\exp(\lambda \langle v, \partial\ell(x; z) - \mathbb{E}_z[\partial\ell(x; z)]\rangle)] \le \exp\left(\lambda^2\sigma_g^2/2d\right)$ and $\mathbb{E}[\exp(\lambda(\ell(x; z) - \mathbb{E}_z[\ell(x; z)]))] \le \exp(\lambda^2\sigma_f^2/2)$. Consequently, for all m and all $x \in \mathcal{X}$, $\mathbb{E}_{z\sim\mathcal{P}_m}[\|\partial\ell(x; z) - \mathbb{E}[\partial\ell(x; z)]\|^2] \le \sigma_g^2$.*

Assumptions 1 and 2 are imposed on the behavior of the population loss function and the distribution of the samples, which is a relatively milder requirement than on the sample loss for each data point. The assumption on the gradient norm in Assumption 1 can be relaxed to any $L > 0$ using an appropriate scaling. For simplicity, we consider the case of $L = 1$. For simplicity of notation, throughout the rest of the paper, we use $\partial\mathcal{L}(x)$ to denote an element of the set $\partial\mathcal{L}(x)$.

Accuracy. Let $\widehat{x}_{\mathscr{A}}$ denote the output of an algorithm $\mathscr{A}$ when run on a loss function $\mathcal{L}$. The excess risk of the algorithm $\mathscr{A}$ on $\mathcal{L}$ is given as

$$\mathsf{ER}(\mathscr{A}; \mathcal{L}) = \mathcal{L}(\widehat{x}_{\mathscr{A}}) - \min_{x\in\mathcal{X}} \mathcal{L}(x). \tag{3}$$

We measure the performance of an algorithm $\mathscr{A}$ using $\mathsf{ER}(\mathscr{A})$, where

$$\mathsf{ER}(\mathscr{A}) := \sup_{\mathcal{L}\in\mathcal{F}} \mathbb{E}[\mathsf{ER}(\mathscr{A}; \mathcal{L})] \tag{4}$$

denotes the worst-case expected excess risk over the functions in $\mathcal{F}$, the family of convex, 1-Lipschitz functions. Here, the expectation is taken over the randomness in the datasets $\{\mathcal{D}_m\}_{m=1}^M$ and the algorithm $\mathscr{A}$. For a prescribed error $\delta_{\mathsf{Err}} \in (0, 1)$, we analogously define $\mathsf{ER}(\mathscr{A}; \delta_{\mathsf{Err}})$ which corresponds to a bound on $\sup_{\mathcal{L}\in\mathcal{F}} \mathsf{ER}(\mathscr{A}; \mathcal{L})$ that holds with probability $1 - \delta_{\mathsf{Err}}$.

Communication cost. We adopt the commonly used communication model where the clients can communicate only via the server. Each client can upload messages to the server which the server can broadcast to all other clients. This is commonly referred to as the blackboard model of communication [Barnes et al., 2020a, Braverman et al., 2016]. The communication cost of an algorithm $\mathscr{A}$ is measured as

$$\mathsf{CC}(\mathscr{A}) = \frac{1}{M}\sum_{m=1}^{M} C_m(\mathscr{A}), \tag{5}$$

where $C_m(\mathscr{A})$ denotes the number of bits uploaded by client m during a run of algorithm $\mathscr{A}$. We focus only on the upload communication costs in this work, as often they are the communication bottleneck.

Differential privacy. To formally define differentially private algorithms, we use the following notion of indistinguishability.

Definition 3. *For a given $\varepsilon > 0$ and $\delta \in (0, 1)$, two distributions P and Q with a common support are said to be (ε, δ) indistinguishable (denoted as $P \sim_{(\varepsilon,\delta)} Q$) if the following relation holds for all events O in the probability space:*

$$e^{-\varepsilon}(P(O) - \delta) \leq Q(O) \leq e^{\varepsilon} P(O) + \delta.$$

Let $\{\mathcal{D}_m, \mathcal{D}'_m\}_{m=1}^M$ be a collection of pairs of *neighboring datasets* such that for all $m \in \{1, 2, \ldots, M\}$, $\mathcal{D}_m$ and $\mathcal{D}'_m$ differ on at most one data point. We refer to such datasets as neighboring datasets. We call an algorithm $\mathscr{A}$ to be (ε, δ) differentially private [Dwork et al., 2006], if for all collections of neighboring datasets $\{\mathcal{D}_m, \mathcal{D}'_m\}_{m=1}^M$, we have $\mathscr{A}(\{\mathcal{D}_m\}_{m=1}^M) \sim_{(\varepsilon,\delta)} \mathscr{A}(\{\mathcal{D}'_m\}_{m=1}^M)$, where the probability is taken over the randomness in $\mathscr{A}$.

3 Lower Bound

In this section, we investigate the converse region of the accuracy-communication-privacy trade-off. The following theorem characterizes the lower bound on the worst-case accuracy (i.e., the excess population risk) of any distributed DP-SCO algorithm as a function of communication complexity and privacy guarantees.

Theorem 1. *Consider the distributed SCO problem outlined in Eqn. (2) over a domain with diameter R, where the underlying data distributions satisfy Assumption 2. The excess risk of any $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differentially private algorithm $\mathscr{A}$ for the problem of distributed SCO satisfies*

$$\mathit{ER}(\mathscr{A}) \gtrsim R \cdot \max\left\{ \min\left\{ \sqrt{\frac{\sigma_g^2}{N}}, \sqrt{\frac{\sigma_g^2 d^2}{MN\mathsf{CC}(\mathscr{A})}}, \frac{1}{\sqrt{d}} \right\}, \ \min\left\{ \sqrt{\frac{\sigma_g^2}{MN}}, \frac{1}{\sqrt{d}} \right\}, \ \frac{\sqrt{d}}{\sqrt{MN}\varepsilon_{\mathsf{DP}}} \right\}.$$

The proof is deferred to Appendix A. The above theorem provides a lower bound on the accuracy of any differentially private algorithm that solves the DP-SCO problem as a function of its communication cost and privacy parameter. This is the *first* information-theoretic, algorithm-independent lower bound on the accuracy-communication trade-off of a distributed SCO algorithm, both in non-private and private settings. Several comments on the theorem are in order.

Accuracy-Communication trade-off. An immediate corollary of the above theorem is a lower bound on the accuracy-communication trade-off in the non-private setting, i.e., $\varepsilon_{\mathsf{DP}} = \infty$ with $N = \Omega(\sigma_g^2 d)$, which reads (up to scaling of R and σ_g^2)

$$\mathsf{ER}(\mathscr{A}) \gtrsim \max\left\{ \min\left\{ \frac{1}{\sqrt{N}}, \sqrt{\frac{d^2}{MN\mathsf{CC}(\mathscr{A})}} \right\}, \frac{1}{\sqrt{MN}} \right\}.$$

Our lower bound also exhibits the well-known inverse relation between accuracy and communication that has been derived for other distributed learning problems [Braverman et al., 2016, Duchi et al., 2014]. Moreover, it tightens the existing lower bound from $\Omega(d)$ to $\Omega(d^2)$. No algorithm $\mathscr{A}$ will achieve the excess risk $\mathsf{ER}(\mathscr{A}) \lesssim \frac{1}{\sqrt{MN}}$ — the optimal rate in the centralized setting — unless the communication cost satisfies

$$\mathsf{CC}(\mathscr{A}) \gtrsim d^2.$$

This reflects our intuitive belief about the inherent hardness of general convex optimization compared to other problems like mean estimation. We would like to emphasize that this result holds only for general convex functions and not strongly convex functions. For the case of strongly convex functions, the $\Omega(d)$ bound is tight, as shown by matching upper bounds [Haddadpour et al., 2019, Reisizadeh et al., 2020, Salgia et al., 2024, Spiridonoff et al., 2020].

Comparison with existing SGD-based lower bounds. Several existing studies [Arjevani and Shamir, 2015, Woodworth et al., 2018, 2021] have tightly characterized the communication complexity of the gradient descent family of algorithms. In particular, Woodworth et al. [2021] show that in order to achieve an accuracy of $\Theta(1/\sqrt{MN})$, SGD and accelerated SGD require $\Theta(\sqrt{MN})$ and $\Theta((MN)^{1/4})$ rounds of communication respectively, where in each round each agent transmits a d-dimensional vector to the server. These results are not directly comparable with those obtained above as they only hold for smooth functions, i.e., gradient is also a Lipschitz function. On the other hand, the lower bound derived in this work allows for non-smooth and even non-differentiable functions. Moreover, the lower bounds on gradient descent algorithms [Arjevani and Shamir, 2015, Woodworth et al., 2021], where the bounds are derived using the optimization dynamics, only hold in the regime $d = \widetilde{\Omega}((MN)^{5/4})$. The lower bound in Theorem 1 is algorithm agnostic and is based on achieving statistical efficiency using information-theoretic tools. As is typical of statistical bounds, the above theorem results in non-trivial bounds in the data-rich regime, i.e., $MN \gtrsim d$. While a thoroughly fair comparison is not possible between our results and existing ones, the lower bound in Theorem 1 suggests that (non-accelerated) SGD-based algorithms that are commonly used in real-world applications incur sub-optimal communication costs in the regime $\sqrt{MN} \gtrsim d$.

Privacy-Communication trade-off. The above theorem suggests that up to an extent, privacy and communication work in tandem with each other, i.e., reducing communication allows to one achieve stronger privacy guarantees and higher privacy requirements allow for reduced communication costs. Such a behavior echoes a similar result obtained in Chen et al. [2020, 2024] for the case of distributed mean estimation. This trade-off between privacy and communication for DP-SCO, however, is evident only in the very high privacy regime. Specifically, note that the privacy term will be larger than the communication term only when $\varepsilon_{\mathsf{DP}} = \Omega(1/\sqrt{Nd})$. Consequently, in the very-high privacy regime, i.e., $\varepsilon_{\mathsf{DP}} = \Omega(1/\sqrt{Nd})$, the privacy requirements allow for communication costs that scale as $o(d^2)$. However, for typical use cases, i.e., $\varepsilon_{\mathsf{DP}} = \Theta(1)$, this part of the trade-off is not relevant.

High-level proof idea. We establish our lower bound by considering the behavior of any algorithm on a specifically chosen convex function. In particular, we consider a function of the form $\max_{i=1,\dots,d}\{a_i^\top x - b_i\}$ for appropriately chosen vectors $\{a_i\}_{i=1}^d$ and scalars $\{b_i\}_{i=1}^d$. Similar constructions that take the form of a maximum over linear functions have been used in previous studies to establish other lower bounds for convex optimization [Feldman, 2016, Nemirovskii and Yudin, 1983]. We establish the bound using a two-step reduction: *(i)* we first show that optimizing this function is equivalent to learning at least $\Omega(d)$ vectors from the set $\{a_i\}_{i=1}^d$; *(ii)* we then show that learning these vectors is equivalent to solving $\Omega(d)$ mean estimation problems. We arrive at the final bound by combining these observations with existing bounds on the mean estimation problem [Braverman et al., 2016, Duchi et al., 2014]. We would like to point out that while the current bound is derived only for 1-Lipschitz functions, the analysis can be extended in a straightforward manner to allow for L-Lipschitz functions. In such a case, the privacy related term in the lower bound in Theorem 1 gets scaled by a factor of L, while the rest remain as is.

4 Algorithm

In this section, we explore the achievability frontier of the accuracy-communication-privacy trade-off. One of the key challenges in designing an optimal algorithm is to bridge the existing sub-optimality gap along the communication complexity frontier. In order to address this challenge, we revisit one of the classic convex optimization approaches — Vaidya's plane cutting method [Vaidya, 1996].

Plane cutting methods. The philosophy of plane-cutting methods is based on the fundamental definition of convex functions. In particular, we know that the gradient of a convex function f satisfies the relation $0 \geq f(x^\star) - f(x) \geq \langle \partial f(x), x^\star - x \rangle$, where $x^\star \in \arg\min_x f(x)$. This implies the gradient of a convex function allows us to construct a separating hyperplane to identify which half of the domain contains the minimizer — a key observation exploited in plane-cutting methods. In each iteration, they obtain the gradient at a carefully chosen point, which allows them to eliminate a constant fraction of the domain. Thus in $\mathcal{O}(d\log(N))$ iterations, they can arrive within a radius of $1/\sqrt{N}$ around the minimizer.

Algorithm design. The above key property of plane-cutting methods allows us to bridge the communication sub-optimality gap in distributed SCO. Specifically, we build upon the plane-cutting methods by replacing the deterministic gradients with an estimate computed by the clients. This plane-cutting-based framework allows us to transform the original SCO problem into estimating the gradient at $\widetilde{\mathcal{O}}(d)$ points. Note that this is precisely the reduction that characterizes the lower bound on the communication complexity, thereby resulting in an order-optimal communication complexity.

While the plane cutting approach allows us to achieve the optimal communication complexity, we lose the uniform stability of SGD based approaches which has been shown to be crucial to achieve the optimal accuracy-privacy trade-off. Thus, to address the accuracy-privacy trade-off, we carefully design our gradient estimation routine to guarantee both privacy and generalization.

Next, we first provide an overview of the plane-cutting method used in this work followed by a description of our proposed algorithm, CHARTER.[2]

4.1 Vaidya's Plane Cutting Method

Vaidya's Plane Cutting method is a classical convex optimization algorithm proposed by Vaidya [1996] to minimize a convex function $f(x)$ over a given convex, compact set $\mathcal{K}$. We first introduce some notation and then provide a general description of the algorithm.

Let $P = \{x \in \mathbb{R}^d : Ax \geq b\}$ be a bounded d-dimensional polyhedron, where $A \in \mathbb{R}^{p\times d}$ and $b \in \mathbb{R}^p$. The volumetric barrier of the set P is defined as

$$V(x) := \frac{1}{2}\log(\det(H(x))), \qquad \text{where } H(x) = \sum_{i=1}^{p} \frac{a_i a_i^\top}{(a_i^\top x - b_i)^2}.$$

Here, $a_i^\top$ is the i^{th} row of the matrix A and $\det(B)$ denotes the determinant of the matrix B. The minimizer of the function $V(x)$ in the interior of P is referred to as the volumetric center of the set P. Lastly, for all $i \in \{1, 2, \ldots, p\}$, we define

$$\sigma_i(x) := \frac{a_i^\top (H(x))^{-1} a_i}{(a_i^\top x - b_i)^2}.$$

Vaidya's method proceeds by generating a sequence of pairs $(A_k, b_k) \in \mathbb{R}^{p_k\times d} \times \mathbb{R}^{p_k}$ such that the corresponding polyhedrons contain the solution of the problem, i.e., minimizer of the function f. Here p_k denotes the number of constraints used to describe the polyhedron constructed during the k^{th} iteration. The initial polyhedron (A_0, b_0) is taken to be a unit hypercube. For simplicity, we assume that $\mathcal{K}$ corresponds to this hypercube. The algorithm can be easily modified to the case where (A_0, b_0) is a bounding hypercube of the set $\mathcal{K}$. Vaidya's method also uses two hyperparameters $\eta, \gamma \in (0, 1)$ which are numerical constants independent of problem parameters. The parameter γ is used to control the total number of constraints in any given iteration by eliminating constraints that are less important. The parameter η, together with γ, determines the rate of progress in each iteration.

At the beginning of each iteration $k \geq 0$, the learner determines the approximate volumetric center x_k and calculates $\{\sigma_i(x_k)\}_{i=1}^{p_k}$. The next polyhedron characterized by the pair (A_{k+1}, b_{k+1}) is obtained from the current result by either adding or removing a constraint. In particular,

- if, $\sigma_i(x_k) = \min_{1\leq j\leq p_k} \sigma_j(x_k) < \gamma$, then (A_{k+1}, b_{k+1}) is obtained by eliminating the i^{th} row from (A_k, b_k);
- otherwise, the algorithm first determines a $\beta_k \in \mathbb{R}$, such that

$$\frac{c_k^\top (H(x_k))^{-1} c_k}{(c_k^\top x - \beta_k)^2} = \frac{1}{2}\sqrt{\eta\gamma},$$

  where $c_k \in -\partial f(x_k)$ (which is the subgradient of f at x_k) and then adds the constraint $(c_k^\top, \beta_k)$ to (A_k, b_k) to obtain (A_{k+1}, b_{k+1}).

[2]The algorithm is named CHARTER because it is based on using private planes for the plane-cutting method.

Vaidya's method has been studied in great detail since it was proposed by Vaidya. We refer the reader to Anstreicher [1997, 1998], Jiang et al. [2020], Lee et al. [2015], Ye [1996] and references therein for additional details about the implementation and hyperparameter choices. Vaidya's method has also been studied for non-private, stochastic convex optimzation in the centralized setting [Feldman et al., 2021, Gladin et al., 2021, 2022, Mehrotra, 2000]. We extend results in these studies to a distributed setting with differential privacy. Also, CHARTER offers an improved statistical complexity over these existing studies.

4.2 The CHARTER algorithm

The proposed algorithm builds upon the classical plane cutting methods, while incorporating the elements of stochasticity and privacy. The algorithm consists of two stages, the learning stage and the verification stage. Before the start of the algorithm, each client splits their dataset into two parts, namely $\mathcal{D}_m^{(1)}$ and $\mathcal{D}_m^{(2)}$ consisting of $2N/3$ and $N/3$ samples respectively.[3]

The learning stage. The first stage of the algorithm generates a sequence of iterates $\{x_0, x_1, \ldots, x_K\}$ using K iterations of the Vaidya's method, where K is a parameter of the algorithm. In the k^{th} iteration, the cutting plane is constructed using an estimate of $\partial\mathcal{L}(x_{k-1})$ which is computed collaboratively by the clients.

In order to collaboratively estimate the gradient at a given point x, each client m first computes

$$\widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x) := \frac{3K}{N} \sum_{z \in \mathcal{D}_m^{(1,k)}} \mathsf{clip}(\partial\ell(x; z); G_0) \cdot \mathbb{1}\{z \notin \cup_{j=1}^{k-1} \mathcal{D}_m^{(1,j)}\}, \tag{6}$$

which is an estimate of $\partial\mathcal{L}(x)$ based on the local data at the client m. Here $\mathcal{D}_m^{(1,k)}$ is a subset of size $N/3K$[4] drawn randomly from the set $\mathcal{D}_m^{(1)}$ during the k^{th} iteration. The clip is the standard clipping function, where $\mathsf{clip}(y, G) = y \cdot \min\{1, G/\|y\|_2\}$. Note that, in order to ensure that the estimated gradient is an independent sample of $\partial\mathcal{L}(x)$, we only use the samples that have not been seen before, as denoted by the indicator function $\mathbb{1}\{\cdot\}$. This is crucial to guarantee generalization. This, however, introduces a bias in the estimate (denoted by the superscript b), which we correct for in a later step. The non-private estimate is then privatized using the Gaussian mechanism to obtain,

$$\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,b}}(x) := \widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x) + \mathcal{N}(0, \sigma_0^2 I_d). \tag{7}$$

After privatizing, we debias the gradient estimate by dividing the privatized estimate by $3KT_{k,m}/N$, to obtain

$$\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,u}}(x) := \frac{N}{3KT_{k,m}} \cdot \widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,b}}(x). \tag{8}$$

Here $T_{k,m} = |\mathcal{D}_m^{(1,k)} \setminus \bigcup_{j=1}^{k-1} \mathcal{D}_m^{(1,j)}|$ is the number of unseen elements in $\mathcal{D}_m^{(1,k)}$. We use this two step procedure to estimate the gradient to ensure both *privacy* and *generalization*. Specifically, in order to ensure generalization, we need to ensure that we use an independent estimate of $\partial\mathcal{L}(x)$ in each iteration. A straightforward way to guarantee that is to randomly sample from the subset of samples not seen so far. However, this does not allow us leverage privacy amplification guarantees through subsampling as the randomness in the algorithm becomes dependent across different calls to the dataset. Thus, to address this issue we always sample from the entire dataset, which helps us obtain optimal privacy dependence through subsampling and composition [Balle et al., 2018, Dwork et al., 2015, Kairouz et al., 2015, Steinke, 2022]. To obtain generalization, we drop the previously seen samples in constructing our estimate to ensure independence of the samples. We carefully choose our batch sizes to ensure that $3KT_{k,m}/N = \Theta(1)$ holds with high probability for all iterations so that the utility of the algorithm worsens by no more than a constant factor when we debias the gradient after privatization. We would like to point out that $T_{k,m}$ is independent of the *actual* value of the samples. As a result, the debiasing step is effectively a post-processing step and thus maintains the privacy of the estimate. Lastly, we quantize the privatized estimate to obtain

$$\widehat{\partial\mathcal{L}}_m(x) := \mathcal{Q}(\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,u}}(x); D_0, J_0). \tag{9}$$

[3] Without loss of generality, we assume N is divisible by 3.

[4] The batch size can be set to $\lceil N/3K \rceil$ to ensure it is an integer. We ignore the divisibility issue for the ease of presentation.

Here, $\mathscr{Q}$ is the standard stochastic quantization routine [Suresh et al., 2017] that separately clips each coordinate to within the interval $[-D_0, D_0]$ and quantizes it using J_0 bits. Specifically, the quantizer first splits the interval $[-D_0, D_0]$ into $2^{J_0} - 1$ intervals of equal length where $-D_0 = r_1 < r_2 \ldots < r_{2^{J_0}} = D_0$ correspond to end points of the intervals. Each coordinate of input vector w is then separately quantized as follows. The value of the p-th coordinate, $\mathscr{Q}(w)[p]$, is set to be r_{j_p-1} with probability $\frac{r_{j_p} - w[p]}{r_{j_p} - r_{j_p-1}}$ and to r_{j_p} with the remaining probability, where $j_p := \min\{j : r_j < w[p] \leq r_{j+1}\}$. It is straightforward to note that each coordinate of $\mathscr{Q}(w)$ can be represented using J_0 bits and has an error of at most $2D_0 \cdot 2^{-J_0}$.

Finally, each client transmits the quantized version $\widehat{\partial \mathcal{L}}_m(x)$ to the server, where it evaluates

$$\widehat{\partial \mathcal{L}}(x) = \frac{1}{M} \sum_{m=1}^{M} \widehat{\partial \mathcal{L}}_m(x) \tag{10}$$

and sends it back to the clients to be used in the Vaidya's plane cutting method.

The verification stage. In the second stage, each client uses their local dataset $\mathcal{D}_m^{(2)}$ to estimate the value of $\mathcal{L}(x)$ for all the $K+1$ iterates, $\{x_0, x_1, \ldots, x_K\}$, generated during the learning stage. The values are computed using a similar three step procedure as used in the learning stage, i.e., estimation, privatization and quantization. In particular, for each $x \in \{x_0, x_1, \ldots, x_K\}$, each agent computes

$$\widehat{\mathcal{L}}_m^{\mathsf{NonPriv}}(x) := \frac{3}{N} \sum_{z \in \mathcal{D}_m^{(2)}} \ell(x; z) \cdot \mathbb{1}\{|\ell(x; z)| \leq G_1\}, \tag{11}$$

$$\widehat{\mathcal{L}}_m^{\mathsf{Priv}}(x) := \widehat{\mathcal{L}}_m^{\mathsf{NonPriv}}(x) + \mathcal{N}(0, \sigma_1^2), \tag{12}$$

$$\widehat{\mathcal{L}}_m(x) := \mathscr{Q}(\widehat{\mathcal{L}}_m^{\mathsf{Priv}}(x); D_1, J_1). \tag{13}$$

The local estimates $\{\widehat{\mathcal{L}}_m(x_k)\}_{k=0}^K$ are sent to the server, where they are averaged, and the index

$$k^\star := \arg\min_k \frac{1}{M} \sum_{m=1}^{M} \widehat{\mathcal{L}}_m(x_k) \tag{14}$$

is returned by the server. The output of the algorithm is set to $x_{k^\star}$. A pseudocode of the algorithm is presented in Algorithm 1.

4.3 Setting the parameters

The desired performance of the algorithm is obtained by carefully choosing the parameters in both stages. Let $\varepsilon_{\mathsf{DP}} > 0$ and $\delta_{\mathsf{DP}} \in (0, 1)$ denote the privacy parameters and let $\delta_{\mathsf{Err}} \in (0, 1)$. We follow the following choices of parameters.

- The number of iterations is set to $K := \left\lceil (4d/\gamma) \log\left(\frac{d\sqrt{MN}}{\gamma \sigma_g}\right) \right\rceil$, where γ is the parameter of Vaidya's method.
- The clipping radii are set to $G_0 := 1 + \sigma_g \sqrt{2 \log(4MN)}$ and $G_1 := R + \sigma_f \sqrt{2 \log(4MN)}$.
- The privacy noise parameters are set to $\sigma_0^2 := \frac{1080 G_0^2 \log^2(2.5/\delta_{\mathsf{DP}}) K}{N^2 \varepsilon_{\mathsf{DP}}^2}$ and $\sigma_1^2 := \frac{40 G_1^2 \log^2(2.5K/\delta_{\mathsf{DP}}) K}{N^2 \varepsilon_{\mathsf{DP}}^2}$.
- The quantization parameters are set to $D_0 := G_0 + \sigma_0 \sqrt{32 \log\left(\frac{40MKd}{\delta_{\mathsf{Err}}}\right)}$, $D_1 := G_1 + \sigma_1 \sqrt{2 \log\left(\frac{16MK}{\delta_{\mathsf{Err}}}\right)}$, $J_0 := \left\lceil \log_2\left(\frac{2D_0 N \varepsilon_{\mathsf{DP}}}{\sqrt{d} + \sigma_g \varepsilon_{\mathsf{DP}} \sqrt{N}}\right) \right\rceil$ and $J_1 := \left\lceil \log_2\left(\frac{2D_1 N \varepsilon_{\mathsf{DP}}}{R\sqrt{d} + \sigma_f \varepsilon_{\mathsf{DP}} \sqrt{N}}\right) \right\rceil$.

Algorithm 1: CHARTER: At client m

Input: Initial point x_0
Divide the local dataset into $\mathcal{D}_m^{(1)}$ and $\mathcal{D}_m^{(2)}$
`// Set the parameters as described in Sec. 4.3`
`//Learning Stage`
for $k = 0, 1, \ldots, K$ **do**
  Sample a subset $\mathcal{D}_m^{(1,k)}$ of size $N/3K$ uniformly at random from $\mathcal{D}_m$
  Compute the estimate $\widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k)$ using Eqn. (6)
  Compute $\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,b}}(x_k)$ using Eqn. (7)
  Compute $\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,u}}(x_k)$ using Eqn. (8)
  Quantize the current estimate using Eqn. (9) to obtain $\widehat{\partial\mathcal{L}}_m(x_k)$
  Transmit $\widehat{\partial\mathcal{L}}_m(x_k)$ to the server and receive $\widehat{\partial\mathcal{L}}(x_k)$
  Use Vaidya's Method with $\widehat{\partial\mathcal{L}}(x_k)$ to compute x_{k+1}
end for
`// Verification Stage`
for $k = 0, 1, \ldots, K$ **do**
  Evaluate $\widehat{\mathcal{L}}_m(x_k)$ using Eqns. (11), (12) and (13)
end for
Transmit $\{\widehat{\mathcal{L}}_m(x_k)\}_{k=0}^K$ to the server and receive $k^\star$
return $x_{k^\star}$

4.4 Performance guarantees

The following theorem characterizes the performance of the proposed algorithm.

Theorem 2. *Assume that Assumptions 1 and 2 hold and the domain $\mathcal{X}$ is a hypercube. If* CHARTER *is run with the choice of parameters described in Section 4.3 with $N = \Omega(d\log(KM))$ samples at each agent, then for any given privacy parameters $\varepsilon_{\mathsf{DP}} \in (0, 1.5/\sqrt{K})$ and $\delta_{\mathsf{DP}} \in (0,1)$, and error probability $\delta_{\mathsf{Err}} \in (0,1)$,*

- CHARTER *is $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differentially private;*
- *The error rate of* CHARTER *satisfies*

$$ER(\text{CHARTER}\,; \delta_{\mathsf{Err}}) = \widetilde{\mathcal{O}}\left(\frac{R\sigma_g + \sigma_f}{\sqrt{MN}} + (R(1+\sigma_g) + \sigma_f) \cdot \frac{\sqrt{d}}{N\varepsilon_{\mathsf{DP}}\sqrt{M}}\right);$$

- *The communication cost of* CHARTER *satisfies*

$$CC(\text{CHARTER}) = Kd(J_0 + J_1) = \widetilde{\mathcal{O}}(d^2).$$

A proof of the above theorem can be found in Appendix B. For the case of general L-Lipschitz functions, the term $(1+\sigma_g)$ gets updated to $(L+\sigma_g)$. A few implications of the theorem are in order.

Optimal Accuracy-Communication-Privacy Trade-off. As shown by the above theorem, CHARTER is differentially private, achieves the optimal accuracy, including linear speedup w.r.t. the number of clients, and order-optimal communication complexity (for $\varepsilon_{\mathsf{DP}} \geq \sqrt{d/N}$) as dictated by the lower bound derived in the previous section. Thus, CHARTER is the *first* algorithm to achieve order-optimal performance on all the three fronts for distributed, differentially private stochastic optimization of general convex functions. Together with our lower bound, it provides tight characterization of the frontier for $\varepsilon_{\mathsf{DP}} \leq \frac{1.5}{\sqrt{K}}$. This constraint on the privacy parameter is a consequence of using privacy amplification by subsampling without replacement which holds only for $\varepsilon_{\mathsf{DP}} < 1$ [Balle et al., 2018],. We believe this can be resolved by utilizing a different privacy amplification scheme. We leave the extension to future work. We would also like to point out that we assume $\mathcal{X}$ to be a hypercube only for convenience. The result extends immediately to general convex bodies by appropriately incorporating the change in Vaidya's method.

Beyond the three-way trade-off. In addition to achieving optimal performance along all the three desiderata, CHARTER also possesses several other desirable properties. Theorem 2 holds for general, convex, Lipschitz functions without any assumption on smoothness of the function, as is required by numerous existing studies. Moreover, in terms of gradient complexity, CHARTER requires only $\mathcal{O}(N)$ gradient computations,[5] which improves upon the current state of the art for distributed algorithms [Murata and Suzuki, 2023] and matches that in the single agent setting [Choquette-Choo et al., 2024]. Furthermore, note that we do not require the data distribution to be identical for all the clients. For each point x_k, we use an unbiased estimate of the gradient based on data from *all* the clients. As a result, the gradient estimated at the server is always an unbiased estimate of the true gradient of $\mathcal{L}(x)$, even when the client distributions are different. Thus, CHARTER also lends itself to scenarios with heterogeneous data distribution. Lastly, CHARTER also allows seamless integration with client sampling. In particular, if at each communication round only a $s \in (0, 1)$ fraction of clients are available, CHARTER offers a similar error rate guarantee with M replaced with sM.

5 Discussions and Future Work

Our approach presents a departure from the existing SGD family of algorithms and adopts a different philosophical outlook toward the optimization problem. Specifically, SGD adopts a "function landscape" based optimization approach, where it moves down the the function landscape until it reaches the bottom of the valley, or equivalently the minimum value. On the other hand, CHARTER adopts a more geometric perspective to the optimization problem where the objective is to eliminate regions of the domain that do not contain the minimizer, reminiscent of the classic bisection algorithms in one dimension [Frazier et al., 2019, Vakili et al., 2019]. While both approaches offer similar accuracy and privacy performances, the key difference is reflected in their communication complexities. The "function landscape" based approach is inherently tied to the steepness of the function valley. When the valley is wide, over reliance on the local steps taken by the agents precludes the algorithm from determining a useful descent direction, resulting in a bias (also referred to as the client drift) that leads to a sub-optimal performance. In order to remedy this and prevent excessive reliance on local steps, SGD-based algorithms need to communicate frequently, which results in high communication complexity. On the other hand, the geometric perspective to optimization avoids this pitfall and can eliminate sub-optimal regions at a constant rate, thereby requiring less frequent communication. A similar conclusion in the context of adapting to function regularity was noted in Vakili et al. [2019].

While this geometric perspective offers an improved communication complexity, it comes at the cost of increased computation complexity. Specifically, Vaidya's method has a computation complexity of $\mathcal{O}(d^3+\mathsf{grad})$, where grad denotes the overall computational complexity of evaluating all the gradients [Jiang et al., 2020]. This order of scaling prevents the application of our proposed approach to high-dimensional problems. Given the inherent necessity of adopting a geometric perspective to achieve optimal communication complexity, this suggests that the three-way trade-off is likely a four-way trade-off with computational complexity as the fourth axis. An interesting future direction is to explore if and how computational complexity poses a bottleneck in achieving the optimal accuracy-communication-privacy trade-off.

Another interesting direction is to iron out the small sub-optimality region for the communication cost in the high-privacy regime. We believe this can be remedied using more sophisticated quantization schemes that combine privacy and quantization [Chen et al., 2024]. The proposed scheme in Chen et al. [2024] uses at most a single bit for each dimension. However, in our setting it might be necessary to use multiple bits to represent the values in each dimension which precludes a direct adaptation of that approach. Designing quantization schemes that allow for multiple-bit representation while ensuring privacy is another direction worth exploring.

Acknowledgement

The work of S. Salgia and Y. Chi is supported in part by the grants NSF CNS-2148212, ECCS-2318441, ONR N00014-19-1-2404 and AFRL FA8750-20-2-0504, and in part by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

[5] We refer to the computation of the gradient at a single data point as a unit computation. For the function evaluation, we compute $K = \widetilde{\mathcal{O}}(d)$ scalar values, which is computationally equivalent to $\widetilde{\mathcal{O}}(1)$ gradient evaluations.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

A. Agarwal, M. J. Wainwright, P. Bartlett, and P. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, volume 22, 2009.

N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, volume 31, 2018.

K. M. Anstreicher. On vaidya's volumetric cutting plane method for convex programming. *Mathematics of Operations Research*, 22(1):63–89, 1997.

K. M. Anstreicher. Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9(1):190–206, 1998.

Y. Arjevani and O. Shamir. Communication complexity of distributed convex learning and optimization. *Advances in neural information processing systems*, 28, 2015.

R. Arora, R. Bassily, C. Guzmán, M. Menart, and E. Ullah. Differentially private generalized linear models revisited. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, volume 35, pages 22505–22517, 2022.

H. Asi, V. Feldman, T. Koren, and K. Talwar. Private stochastic convex optimization: Optimal rates in l1 geometry. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, pages 393–403. PMLR, 2021.

B. Balle, G. Barthe, and M. Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, volume 31, 2018.

R. Bardenet and O.-A. Maillard. Concentration inequalities for sampling without replacement, 2015.

L. P. Barnes, W.-N. Chen, and A. Özgür. Fisher information under local differential privacy. *IEEE Journal on Selected Areas in Information Theory*, 1(3):645–659, 2020a.

L. P. Barnes, Y. Han, and A. Ozgur. Lower bounds for learning distributions under communication constraints via fisher information. *Journal of Machine Learning Research*, 21(236):1–30, 2020b.

R. Bassily and Z. Sun. User-level private stochastic convex optimization with optimal rates. In *Proceedings of the 40th International Conference on Machine Learning, ICML*, pages 1838–1851. PMLR, 2023.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE 55th annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

R. Bassily, V. Feldman, K. Talwar, and A. Guha Thakurta. Private stochastic convex optimization with optimal rates. *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, 32, 2019.

R. Bassily, C. Guzmán, and M. Menart. Differentially private stochastic optimization: New results in convex and non-convex settings. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, volume 34, pages 9317–9329, 2021.

O. Bousquet and A. Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2: 499–526, 2002.

M. Braverman, A. Garg, T. Ma, H. L. Nguyen, and D. P. Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 1011–1020, 2016.

K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, volume 21, 2008.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

W.-N. Chen, P. Kairouz, and A. Ozgur. Breaking the communication-privacy-accuracy trilemma. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, volume 33, pages 3312–3324, 2020.

W.-N. Chen, D. Song, A. Ozgur, and P. Kairouz. Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation. *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*, 36, 2024.

C. A. Choquette-Choo, A. Ganesh, and A. Thakurta. Optimal rates for dp-sco with a single epoch and large batches, 2024.

J. Ding, G. Liang, J. Bi, and M. Pan. Differentially private and communication efficient collaborative learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 7219–7227, 2021.

J. C. Duchi, M. I. Jordan, M. J. Wainwright, and Y. Zhang. Optimality guarantees for distributed statistical estimation, 2014.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th annual ACM Symposium on Theory of Computing*, pages 117–126, 2015.

V. Feldman. Generalization of ERM in stochastic convex optimization: The dimension strikes back. *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, 29, 2016.

V. Feldman, T. Koren, and K. Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

V. Feldman, C. Guzman, and S. Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *Mathematics of Operations Research*, 46(3):912–945, 2021.

P. I. Frazier, S. G. Henderson, and R. Waeber. Probabilistic bisection converges almost as quickly as stochastic approximation. *Mathematics of Operations Research*, 44(2):651–667, 2019.

R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective, 2017.

E. Gladin, A. Sadiev, A. Gasnikov, P. Dvurechensky, A. Beznosikov, and M. Alkousa. Solving smooth min-min and min-max problems by mixed oracle algorithms. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 19–40. Springer, 2021.

E. L. Gladin, A. V. Gasnikov, and E. S. Ermakova. Vaidya's method for convex stochastic optimization problems in small dimension. *Mathematical Notes*, 112(1):183–190, 2022.

E. Gorbunov, F. Hanzely, and P. Richtárik. Local sgd: Unified theory and new efficient methods. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3556–3564. PMLR, 2021.

F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. R. Cadambe. Local SGD with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 2350–2358. PMLR, 2021.

Y. Han, Z. Liang, Z. Liang, Y. Wang, Y. Yao, and J. Zhang. Private streaming sco in l-p geometry with applications in high dimensional online decision making. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8249–8279. PMLR, 2022.

W. Hoeffding. *Probability inequalities for sums of bounded random variables*. Springer, 1994.

R. Hönig, Y. Zhao, and R. Mullins. DAdaQuant: Doubly-adaptive quantization for communication-efficient federated learning. In *Proceedings of the 39th International Conference on Machine Learning, ICML*, volume 162, pages 8852–8866. PMLR, 2022.

X. Huang, Y. Chen, W. Yin, and K. Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. *Advances in Neural Information Processing Systems*, 35:18955–18969, 2022.

Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *Proceedings of the 16th International Conference on Distributed Computing and Networking*, ICDCN '15, 2015.

R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

B. Jayaraman, L. Wang, D. Evans, and Q. Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, volume 31, 2018.

D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar. Adaptive quantization of model updates for communication-efficient federated learning. In *Proceedings of the 46th IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 3110–3114. IEEE, 2021.

H. Jiang, Y. T. Lee, Z. Song, and S. C.-w. Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 944–953, 2020.

P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pages 1376–1385. PMLR, 2015.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. D'Oliveira, H. Eichner, S. El Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecní, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. *Advances and open problems in federated learning*, volume 14. Now Publishers Inc, 2021.

S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume PartF16814, pages 5088–5099, 2020. ISBN 9781713821120.

A. Khaled, K. Mishchenko, and P. Richtárik. Tighter Theory for Local SGD on Identical and Heterogeneous Data. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 4519–4529. PMLR, 2020.

J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated Learning: Strategies for Improving Communication Efficiency, 2016. URL http://arxiv.org/abs/1610.05492.

J. H. Korhonen and D. Alistarh. Towards tight communication lower bounds for distributed optimisation. *Advances in Neural Information Processing Systems*, 34:7254–7266, 2021.

J. Kulkarni, Y. T. Lee, and D. Liu. Private non-smooth erm and sco in subquadratic steps. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, volume 34, pages 4053–4064, 2021.

Y. T. Lee, A. Sidford, and S. C.-w. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065. IEEE, 2015.

D. Levy, Z. Sun, K. Amin, S. Kale, A. Kulesza, M. Mohri, and A. T. Suresh. Learning with user-level privacy. *Advances in Neural Information Processing Systems*, 34:12466–12479, 2021.

B. Li, S. Cen, Y. Chen, and Y. Chi. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *Journal of Machine Learning Research*, 21(180):1–51, 2020.

B. Li, Z. Li, and Y. Chi. Destress: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization. *SIAM Journal on Mathematics of Data Science*, 4(3):1031–1051, 2022a.

Z. Li, H. Zhao, B. Li, and Y. Chi. SoteriaFL: A unified framework for private federated learning with communication compression. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, volume 35, pages 4285–4300, 2022b.

D. Liu and H. Asi. User-level differentially private stochastic convex optimization: Efficient algorithms with optimal rates. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pages 4240–4248. PMLR, 2024.

Y. Liu, X. Zhang, Y. Kang, L. Li, T. Chen, M. Hong, and Q. Yang. Fedbcd: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing*, 70: 4277–4290, 2022. doi: 10.1109/TSP.2022.3198176.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017.

S. Mehrotra. *Volumetric center method for stochastic convex programs using sampling*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät ..., 2000.

T. Murata and T. Suzuki. Diff2: Differential private optimization via gradient differences for nonconvex distributed learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 25523–25548. PMLR, 2023.

A. S. Nemirovskii and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

T. T. Phuong and L. T. Phong. Distributed differentially-private learning with communication efficiency. *Journal of Systems Architecture*, 128:102555, 2022.

A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 2021–2031. PMLR, 2020.

S. Salgia and Q. Zhao. Distributed linear bandits under communication constraints. In *Proceedings of the 40th International Conference on Machine Learning, ICML*, pages 29845–29875. PMLR, 2023.

S. Salgia, T. Gabay, Q. Zhao, and K. Cohen. A communication-efficient adaptive algorithm for federated learning under cumulative regret. *IEEE Transactions on Signal Processing*, 2024.

K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pages 3027–3036. PMLR, 2017.

K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal convergence rates for convex distributed optimization in networks. *Journal of Machine Learning Research*, 20(159):1–31, 2019.

S. Song, T. Steinke, O. Thakkar, and A. Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

A. Spiridonoff, A. Olshevsky, and I. C. Paschalidis. Local SGD With a Communication Overhead Depending Only on the Number of Workers, 2020.

T. Steinke. Composition of differential privacy & privacy amplification by subsampling, 2022.

A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 7, pages 5119–5128, 2017. ISBN 9781510855144.

Z. Tang, S. Shi, X. Chu, W. Wang, and B. Li. Communication-efficient distributed deep learning: A comprehensive survey, 2020.

A. Triastcyn, M. Reisser, and C. Louizos. Dp-rec: Private & communication-efficient federated learning, 2021.

J. N. Tsitsiklis and Z.-Q. Luo. Communication complexity of convex optimization. *Journal of Complexity*, 3 (3):231–243, 1987.

J. Ullman. Private multiplicative weights beyond linear queries. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems*, pages 303–312, 2015.

P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical programming*, 73(3):291–341, 1996.

S. Vakili, S. Salgia, and Q. Zhao. Stochastic gradient descent on a tree: An adaptive and robust approach to stochastic convex optimization. In *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 432–438. IEEE, 2019.

S. S. Vempala, R. Wang, and D. P. Woodruff. The communication complexity of optimization. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1733–1752. SIAM, 2020.

D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, 30, 2017.

D. Wang, H. Xiao, S. Devadas, and J. Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10081–10091. PMLR, 2020a.

J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, et al. A field guide to federated optimization, 2021.

L. Wang, R. Jia, and D. Song. D2p-fed: Differentially private federated learning with efficient communication, 2020b.

L. Wang, B. Jayaraman, D. Evans, and Q. Gu. Efficient privacy-preserving stochastic nonconvex optimization. In *Uncertainty in Artificial Intelligence*, pages 2203–2213. PMLR, 2023.

Y. Wang, X. Cao, S. Jin, and M.-Y. Chow. A novel privacy enhancement scheme with dynamic quantization for federated learning, 2024.

B. Woodworth, K. K. Patel, and N. Srebro. Minibatch vs local SGD for heterogeneous distributed learning. In *Advances in Neural Information Processing Systems*, 2020a.

B. Woodworth, K. K. Patel, S. U. Stich, Z. Dai, B. Bullins, H. Brendan McMahan, O. Shamir, and N. Srebro. Is local SGD better than minibatch SGD? In *37th International Conference on Machine Learning, ICML*, pages 10265–10274, 2020b. ISBN 9781713821120.

B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, 31, 2018.

B. E. Woodworth, B. Bullins, O. Shamir, and N. Srebro. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Conference on Learning Theory*, pages 4386–4437. PMLR, 2021.

Y. Ye. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. *Mathematical Programming*, 78(1):85–104, 1996.

X. Zhang, M. Fang, J. Liu, and Z. Zhu. Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed sgd approach. In *Proceedings of the 21st International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 261–270, 2020.

H. Zhao, Z. Li, and P. Richtárik. Fedpage: A fast local stochastic gradient method for communication-efficient federated learning, 2021.

Z. Zhao, Y. Mao, Y. Liu, L. Song, Y. Ouyang, X. Chen, and W. Ding. Towards efficient communications in federated learning: A contemporary survey. *Journal of the Franklin Institute*, 2023.

L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, 32, 2019.

H. Zong, Q. Wang, X. Liu, Y. Li, and Y. Shao. Communication reducing quantization for federated learning with local differential privacy mechanism. In *IEEE/CIC International Conference on Communications in China (ICCC)*, pages 75–80. IEEE, 2021.

A Proof of Theorem 1

In order to establish the lower bound, we focus on bounding the accuracy as a function of the communication cost without the constraint on privacy. The proof consists of three main steps.

- Constructing the "hard" instance: We first construct a function of interest on which we analyze the performance of a distributed SCO algorithm. This is a typical step in establishing lower bounds, where the function of interest is chosen to reflect the inherent hardness of the problem.
- Reduction to mean estimation: In the second step, we show that optimizing the above function is at least as hard as solving $\Omega(d)$ mean estimation problems. This step allows us to reduce the original convex optimization problem to a set of simpler problems for which we understand the accuracy communication trade-off.
- Establishing the final bound: The final bound is then established by combining the above reduction with techniques developed for the mean estimation problem.

A.1 Constructing the instance of interest

Let $\{a_1, a_2, \ldots, a_d\}$ be any orthonormal basis of $\mathbb{R}^d$. Let $\mathbf{b} = (b_1, b_2, \ldots, b_d) \in \{-1, 1\}^d$. Throughout the proof, we set $\mathcal{X} = \mathcal{B}(1)$, the unit ball in $\mathbb{R}^d$ centered at the origin. For all $i \in \{1, 2, \ldots, d\}$, we define the following function:

$$f_i(x) = \left| a_i^\top x - \frac{b_i}{\sqrt{d}} \right|.$$

The results in the lower bound can be extended immediately to a ball of radius $R/2$ by replacing b_i with $Rb_i/2$ throughout the proof. For simplicity of notation we present the proof with $R = 2$.

We will consider the following objective function for the analysis:

$$f(x) = \alpha \cdot \max_{i=1,2,\ldots,d} f_i(x), \tag{15}$$

where $\alpha \in (0, 1]$ is a parameter, whose value is chosen later. Note that $\{f_i(x)\}_{i=1}^d$ is a collection of convex, 1-Lipschitz functions. Since taking the maximum operation preserves this property, $f(x)$ is also a convex, 1-Lipschitz function. Let $\mathcal{F}'$ denote the class of functions of the above form corresponding to different choices of the orthonormal basis $\{a_1, a_2, \ldots, a_d\}$ and the vector $\mathbf{b}$.

It is straightforward to note that $f(x) \geq 0$ for all $x \in \mathcal{X}$ and $f(x)$ has a unique minimizer $x^\star$ with $f(x^\star) = 0$ where

$$x^\star := \frac{1}{\sqrt{d}} \sum_{i=1}^{d} a_i b_i. \tag{16}$$

We consider the following model for the subgradient observations. In particular, the subgradient of any randomly drawn data point z is distributed as

$$\partial f(x; z) \sim \alpha \cdot a_{i(x)} \cdot s_{i(x)}(x) + \mathcal{N}(0, (\sigma^2/d) \cdot I_d), \tag{17}$$

where for all $j \in \{1, 2, \ldots, d\}$, $s_j(x)$ is defined as:

$$s_j(x) := \begin{cases} +1 \text{ if } a_j^\top x - \frac{b_j}{\sqrt{d}} \geq 0, \\ -1 \text{ otherwise,} \end{cases} \tag{18}$$

and $i(x)$ is given by:

$$i(x) := \min \{j \in \{1, 2, \ldots, d\} \mid f(x) = f_j(x)\}. \tag{19}$$

In other words, $s_j(x)$ determines the sign of the $a_j^\top x - \frac{b_j}{\sqrt{d}}$ and $i(x)$ is the smallest index from the set of the functions that achieve the maximum value at x. It is straightforward to note that this distribution satisfies Assumption 2 with $\sigma_g = \sigma$. Let $\mathcal{A}$ denote the class of algorithms that can optimize functions in $\mathcal{F}'$ using observations of the form (17).

For analytical convenience, we adopt the framework of having an oracle for the noisy gradients. Specifically, the algorithm is allowed N queries to an oracle $\mathscr{O}$ at each agent. Each query to the oracle $\mathscr{O}$ reveals a noisy gradient at the queried point x that follows the same distribution as in (17). Thus, querying an oracle is equivalent to computing the gradient at a new data point.

The benefit of the oracle framework is that it allows us to consider a more powerful oracle for the subgradient. Specifically, we consider an oracle $\mathscr{O}'$ which when queried at a point x reveals the tuple

$$\mathscr{O}'(x) = (\alpha \cdot a_{i(x)} + \mathcal{N}(0, (\sigma^2/d) \cdot I_d), s_{i(x)}(x)),$$

i.e., it separately provides the gradient and the sign information. Clearly, $\mathscr{O}'$ is a more informative oracle. Consequently, if $\mathcal{A}$ and $\mathcal{A}'$ denote the class of algorithms that can optimize functions in $\mathcal{F}'$ using observations from oracles $\mathscr{O}$ and $\mathscr{O}'$ respectively, then $\mathcal{A} \subseteq \mathcal{A}'$. For the remainder of the analysis, we focus on the algorithms in the class $\mathcal{A}'$.

A.2 Reduction to mean estimation

In this part, we show that any algorithm $\mathscr{A} \in \mathcal{A}'$ that achieves a small optimization error, needs to solve at least $\Omega(d)$ mean estimation problems. We establish this reduction in four steps.

Step 1: Low optimization error is equivalent to learning b. For any $A = [a_1, a_2, \ldots, a_d]$ and $\mathbf{b}$, a given algorithm $\mathscr{A}$ and all $j \in \{1, 2, \ldots, d\}$, define

$$p_j(\mathscr{A}; A, \mathbf{b}) = \Pr\left(\operatorname{sgn}(a_j^\top \widehat{x}_{\mathscr{A}}) = b_j\right), \tag{20}$$

where $\widehat{x}_{\mathscr{A}}$ denotes the output of $\mathscr{A}$ when run on the function corresponding to $(A, \mathbf{b})$ and $\operatorname{sgn}(\cdot)$ is the sign function. Here the probability is taken over the randomness in $\mathscr{A}$ and noise distribution. If $(A', \mathbf{b}')$ is an instance such that for some index i, $p_i(\mathscr{A}; A', \mathbf{b}') < 5/6$, then for the function f corresponding to $(A', \mathbf{b}')$,

$$\begin{aligned}\mathbb{E}[\mathsf{ER}(\mathscr{A}; f)] = \mathbb{E}[f(\widehat{x}_{\mathscr{A}})] &\geq \mathbb{E}[\alpha f_i(\widehat{x}_{\mathscr{A}})] \\ &\geq \mathbb{E}\left[\alpha \left| a_i^\top \widehat{x}_{\mathscr{A}} - \frac{b_i}{\sqrt{d}} \right| \;\middle|\; \operatorname{sgn}(a_i^\top \widehat{x}_{\mathscr{A}}) \neq b_i \right] \cdot \Pr(\operatorname{sgn}(a_i^\top \widehat{x}_{\mathscr{A}}) \neq b_i) > \frac{\alpha}{6\sqrt{d}}.\end{aligned}$$

For the first equality we use the fact that the minimum value of f is 0. Thus, for any algorithm $\mathscr{A}$

$$\sup_{f \in \mathcal{F}'} \mathbb{E}[\mathsf{ER}(\mathscr{A}, f)] \leq \frac{\alpha}{6\sqrt{d}} \implies \max_j \sup_{(A, \mathbf{b})} p_j(\mathscr{A}; A, \mathbf{b}) \geq \frac{5}{6}. \tag{21}$$

In other words, if $\mathscr{A}$ achieves a small excess risk for all functions in $\mathcal{F}'$, then it must correctly learn all the b_i's with probability at least $5/6$.

Step 2: Learning b_j's is equivalent to finding a point in $\mathcal{X}_j$. An algorithm can estimate b_j's only through issuing appropriate queries to the oracle $\mathscr{O}'$. For all $i \in \{1, 2, \ldots, d\}$, we define $\mathcal{X}_i$ as

$$\mathcal{X}_i := \left\{ x \in \mathcal{X} \;\middle|\; \left(\bigcap_{j<i} \{f_i(x) > f_j(x)\} \right) \cap \left(\bigcap_{j \geq i} \{f_i(x) \geq f_j(x)\} \right) \cap \left\{ |a_i^\top x| \leq \frac{1}{\sqrt{d}} \right\} \right\}. \tag{22}$$

Note that whenever an algorithm queries a point $x \in \mathcal{X}_i$, the oracle returns $s_i(x) = -b_i$. Moreover, if the queried point $x \notin \mathcal{X}_i$, the value returned by the oracle is either $-b_j$ for $j \neq i$ (when x does not satisfy one of the first two conditions of being in $\mathcal{X}_i$) or a fixed value in $\{-1, +1\}$ given by $\operatorname{sgn}(a_i^\top x)$, which is independent of the value of b_i (when x does not satisfy the third condition of being in $\mathcal{X}_i$). Thus, in both cases, the output of the oracle is uncorrelated with b_i. Hence, querying a point $x \notin \mathcal{X}_i$ yields no information about b_i.

Consequently, an algorithm can learn b_i *only if* it can determine a point $x \in \mathcal{X}_i$. Moreover, since $s_i(x)$ is noiseless, it is also sufficient to determine such an $x \in \mathcal{X}_i$.

Hence, in order for an algorithm to estimate b_i, it needs to build an estimator for a point $x \in \mathcal{X}_i$. Let φ be an estimator for determining a point $x \in \mathcal{X}_i$ such that $\Pr(\varphi \in \mathcal{X}_i) < 2/3$. Here, we slightly abuse notation and use φ to also denote the output of estimator φ. If $\widehat{b}_i(\varphi)$ is an estimator of b_i that uses φ, then $\sup_{\mathbf{b}} \Pr(\widehat{b}_i(\varphi) \neq b_i) \geq \sup_{\mathbf{b}} \Pr(\widehat{b}_i(\varphi) \neq b_i | \varphi \notin \mathcal{X}_i) \Pr(\varphi \notin \mathcal{X}_i) > \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6}$. Here, we used the observation that the output of the oracle at a point $x \notin \mathcal{X}_i$ is uncorrelated with b_i and hence cannot be better than a random guess. Thus, to correctly determine all b_i's with probability 5/6, an algorithm $\mathscr{A}$ needs to determine a set of points $(\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d)$ such that

$$\Pr\left(\bigcap_{i=1}^{d} \{\widetilde{x}_i \in \mathcal{X}_i\}\right) \geq \frac{2}{3}. \tag{23}$$

Step 3: Hardness of finding a point in $\mathcal{X}_i$. In this step, we characterize the hardness of finding a point $w \in \mathcal{X}_i$ for some fixed $i \in \{1, 2, \ldots, d\}$. To characterize the hardness, we make use of the reduction outlined in the following lemma, whose proof is deferred to Appendix A.4.

Lemma 1. *Let*

$$\mathcal{X}_i' := \left\{ x \in \mathcal{X} \,\middle|\, \left(\bigcap_{j<i} \{\langle a_i b_i,\, x\rangle < \langle a_j b_j,\, x\rangle\}\right) \cap \left(\bigcap_{j \geq i} \{\langle a_i b_i,\, x\rangle \leq \langle a_j b_j,\, x\rangle\}\right) \cap \left\{ |\, \langle a_i b_i,\, x\rangle \,| \leq \frac{1}{\sqrt{d}} \right\} \right\} \tag{24}$$

for all $i \in [d]$. It follows $\{w \in \mathcal{X}_i\} \implies \{w \in \mathcal{X}_i'\}$.

In other words, finding a point $w \in \mathcal{X}_i$ is at least as hard as finding a point $w \in \mathcal{X}_i'$. Consequently, any set of points $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d\}$ that satisfy (23) must also satisfy

$$\Pr\left(\bigcap_{i=1}^{d} \{\widetilde{x}_i \in \mathcal{X}_i'\}\right) \geq \frac{2}{3}. \tag{25}$$

Moreover, note that by definition, the sets $\{\mathcal{X}_i'\}_{i=1}^d$ are disjoint and thus the points $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d\}$ need to be distinct.

For the rest of the proof, we focus on characterizing the hardness of finding a set of points $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d\}$ satisfying Eqn. (25). We claim that any routine $\mathscr{M}$ that determines a set of points satisfying Eqn. (25) can also determine a set of vectors $(y_1, y_2, \ldots, y_{d'})$ such that $\langle v_j,\, y_j\rangle > 0$ holds for all $j \leq d'$ with probability 2/3 for some $(d-1)/2 \leq d' \leq d$ and $\{v_1, v_2, \ldots, v_{d'}\} \subseteq \{a_1 b_1, a_2 b_2, \ldots, a_d b_d\}$. To prove this claim, we define

$$\mathcal{X}_{\text{sol}} := \left\{(w_1, w_2, \ldots, w_d) \in \mathcal{X}^d \mid w_j \in \mathcal{X}_j' \quad \forall\, j \in \{1, 2, \ldots, d\}\right\} \tag{26}$$

to be the set of all possible solutions. For all $(w_1, w_2, \ldots, w_d) \in \mathcal{X}_{\text{sol}}$ define,

$$n_+(w_1, w_2, \ldots, w_d) := |\{j \mid \langle a_j b_j,\, w_j\rangle \geq 0\}|; \quad n_-(w_1, w_2, \ldots, w_d) := |\{j \mid \langle a_j b_j,\, w_j\rangle < 0\}|. \tag{27}$$

Lastly, let

$$\mathcal{X}_{\text{sol}}^+ := \{(w_1, w_2, \ldots, w_d) \in \mathcal{X}_{\text{sol}} \mid n_+(w_1, w_2, \ldots, w_d) > d/2\}, \tag{28}$$

$$\mathcal{X}_{\text{sol}}^- := \{(w_1, w_2, \ldots, w_d) \in \mathcal{X}_{\text{sol}} \mid n_-(w_1, w_2, \ldots, w_d) \geq d/2\}. \tag{29}$$

It is straightforward to note that $\mathcal{X}_{\text{sol}}^+$ and $\mathcal{X}_{\text{sol}}^-$ form a partition of $\mathcal{X}_{\text{sol}}$. Thus, $\mathscr{M}$ can determine an element of $\mathcal{X}_{\text{sol}}$ with probability 2/3 *only if* it can either find an element in $\mathcal{X}_{\text{sol}}^+$ with probability 2/3 or find an element in $\mathcal{X}_{\text{sol}}^-$ with probability 2/3. Let us consider the two possible cases.

- **Case (i): $\mathscr{M}$ finds a point in $\mathcal{X}_{\text{sol}}^-$.** By definition of $\mathcal{X}_{\text{sol}}^-$, we know that there exists a set of indices $\{j_1, j_2, \ldots, j_{d'}\}$ with $d' \geq d/2$ for which $\langle a_{j_r} b_{j_r},\, \widetilde{x}_{j_r}\rangle < 0$ holds for all $r \in [d']$. If we choose $\{v_1, \ldots, v_d\}$ such that $v_r = a_{j_r} b_{j_r}$, then $(y_1, y_2, \ldots, y_{d'}) = (-\widetilde{x}_{j_1}, -\widetilde{x}_{j_2}, \ldots, -\widetilde{x}_{j_{d'}})$ is the required set of vectors. Note that finding a point $\widetilde{x}$ is statistically equivalent to finding a point $-\widetilde{x}$.

- **Case (ii): $\mathscr{M}$ finds a point in $\mathcal{X}^+_{\text{sol}}$.** By definition of $\mathcal{X}^+_{\text{sol}}$, we know that there exists a set of indices $\{j_1, j_2, \ldots, j_{d''}\}$ with $d'' > d/2$ for which $\langle a_{j_r} b_{j_r}, \widetilde{x}_{j_r} \rangle \geq 0$ holds for all $r \in [d'']$. WLOG, we assume that $j_1 < j_2 < \cdots < j_{d''}$. Note that from the definition of $\mathcal{X}'_i$, we can conclude that $\left\langle a_{j_r} b_{j_r}, \widetilde{x}_{j_{r-1}} \right\rangle > 0$ holds for all $r = 2, 3, \ldots, d''$. Thus, $\mathscr{M}$ determines the required collection $(y_1, y_2, \ldots, y_{d'}) = (\widetilde{x}_{j_1}, \widetilde{x}_{j_2}, \ldots, \widetilde{x}_{j_{d''-1}})$ corresponding to the subset $\{v_1, v_2, \ldots, v_{d'}\} = \{a_{j_2} b_{j_2}, a_{j_3} b_{j_3}, \ldots, a_{j_{d''}} b_{j_{d''}}\}$ with $d' = d'' - 1 > d/2 - 1$.

On combining the cases, we arrive at the statement. Consequently, we can conclude that determining a solution $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d\}$ satisfying (25) is at least as hard as finding a set of points $(y_1, y_2, \ldots, y_{d'})$ such that $\langle v_j, y_j \rangle > 0$ holds for all $j \leq d'$ with probability 2/3 for some $(d-1)/2 \leq d' \leq d$ and $\{v_1, v_2, \ldots, v_{d'}\} \subseteq \{a_1 b_1, a_2 b_2, \ldots, a_d b_d\}$.

Step 4: Establishing the equivalence to mean estimation. Consider the problem of estimating a vector y such that $\langle \mu, y \rangle > 0$ w.p. 2/3, using samples of the form $\mathcal{N}(\mu, (\sigma^2/d) I_d)$, where μ is an unknown vector. We claim that this problem is at least as hard as solving the Gaussian mean estimation problem to within an error of $C\|\mu\|_2^2$ for some numerical constant $C > 0$.

To establish this claim, note that under the aforementioned model, the problem is at least as hard as finding $\widehat{\mu}$, an estimate of μ based on the samples, satisfying $\langle \mu, \widehat{\mu} \rangle > 0$ w.p. 2/3. Let $Z = \mu - \widehat{\mu}$ denote the estimation error and $u = \mu/\|\mu\|_2$ denote a unit vector in the direction of μ. Note that under this model, Z is independent of μ. We have,

$$\langle \mu, \widehat{\mu} \rangle > 0 \implies \langle \mu, \widehat{\mu} - \mu \rangle > -\|\mu\|_2^2 \implies \langle u, Z \rangle < \|\mu\|.$$

Consequently, any estimator $\widehat{\mu}$ that ensures $\langle \mu, \widehat{\mu} \rangle > 0$ w.p. 2/3 for all choices of u must ensure $\sup_u \langle u, Z \rangle < \|\mu\|$ holds w.p. 2/3, or equivalently, $\|Z\| \leq \|\mu\|$ with probability at least 2/3. This is equivalent to solving the Gaussian mean estimation problem such that $\|\widehat{\mu} - \mu\|^2 \leq \|\mu\|^2$ with probability at least 2/3.

Note that the problem faced by the learner, i.e., of identifying the set of vectors $(y_1, y_2, \ldots, y_{d'})$ corresponding to $\{v_1, v_2, \ldots, v_{d'}\} \subseteq \{a_1 b_1, a_2 b_2, \ldots, a_d b_d\}$, is identical to the one outlined above. In particular, for y_j, $\mu = \alpha v_j$, and the samples correspond to the queries to the oracle. As a result, the problem of identifying the set of vectors $(y_1, y_2, \ldots, y_{d'})$ is equivalent to solving $d' = \Theta(d)$ mean estimation problems to within an error proportional to α, the norm of the mean vector.

A.3 Establishing the final bound

We can restate the problem of interest as follows. Let $\{\theta_1, \ldots, \theta_{d'}\}$ be a collection of distinct vectors with (unknown) norm α. Let $\mathscr{A}'_{ME}$ be any distributed mean estimation algorithm with M clients such that whose communication cost matches that of our optimization algorithm $\mathscr{A}$, i.e., $\mathsf{CC}(\mathscr{A}'_{ME}) = \mathsf{CC}(\mathscr{A})$. For any vector θ_j, each client can query the oracle to obtain an independent sample from $\mathcal{N}(\theta_j, (\sigma^2/d) I_d)$. Using a total of N such samples at each agent, the algorithms need to determine a set of estimates $\{\widehat{\theta}_1, \ldots, \widehat{\theta}_{d'}\}$ such that with probability at least 2/3,

$$\max_{j \leq d'} \|\widehat{\theta}_j - \theta_j\|^2 \leq \|\theta_1\|_2^2 = \alpha^2. \tag{30}$$

For all j, let N_j and B_j denote the number of samples used and the number of bits transmitted by each client respectively to estimate θ_j.[6] Using the results for distributed mean estimation [Barnes et al., 2020b, Braverman et al., 2016, Duchi et al., 2014], we can conclude that

$$\|\widehat{\theta}_j - \theta_j\|^2 \geq c_0 \cdot \min\left\{ \frac{\sigma^2 d}{d N_j}, \max\left\{ \frac{\sigma^2 d^2}{d M N_j B_j}, \frac{\sigma^2 d}{d M N_j} \right\} \right\} \geq c_0 \cdot \min\left\{ \frac{\sigma^2}{N_j}, \max\left\{ \frac{\sigma^2 d}{M N_j B_j}, \frac{\sigma^2}{M N_j} \right\} \right\}, \tag{31}$$

holds for each $j \leq d'$ w.p. at least 1/3 where $c_0 > 0$ is a numerical constant. Define $\mathcal{J}_1 := \{j : N_j > 3N/d'\}$ and $\mathcal{J}_2 := \{j : B_j > 3\mathsf{CC}(\mathscr{A}'_{ME})/d'\}$. It is straightforward to note that $|\mathcal{J}_1| \leq d'/3$ and $|\mathcal{J}_2| \leq d'/3$. Consequently, $|\mathcal{J}_1^c \cap \mathcal{J}_2^c| = d' - |\mathcal{J}_1 \cup \mathcal{J}_2| \geq d' - |\mathcal{J}_1| - |\mathcal{J}_2| \geq d'/3 > 0$. This implies that there exists an

[6] For simplicity of exposition, we assume that the values N_j and B_j are same across all clients. This idea can be extended to the general case with different values at different clients using the sequence of arguments outlined in Duchi et al. [2014].

index j' such that $N_{j'} \leq 3N/d'$ and $B_{j'} \leq 3\mathsf{CC}(\mathscr{A}'_{ME})/d'\}$. Using this choice of j' along with (31) and the relations $d' \geq (d-1)/2$ and $\mathsf{CC}(\mathscr{A}'_{ME}) = \mathsf{CC}(\mathscr{A})$, we can conclude that

$$\max_{j \leq d'} \|\widehat{\theta}_j - \theta_j\|^2 \geq c_1 \cdot \min\left\{\frac{\sigma^2 d}{N}, \max\left\{\frac{\sigma^2 d^3}{MN\mathsf{CC}(\mathscr{A})}, \frac{\sigma^2 d}{MN}\right\}\right\} \tag{32}$$

holds with probability at least $1/3$ for some numerical constant $c_1 > 0$.

Let us consider the scenario where

$$\alpha^2 := \min\left\{\frac{c_1}{2} \cdot \min\left\{\frac{\sigma^2 d}{N}, \max\left\{\frac{\sigma^2 d^3}{MN\mathsf{CC}(\mathscr{A})}, \frac{\sigma^2 d}{MN}\right\}\right\}, 1\right\}. \tag{33}$$

For this choice of α, based on Eqn. (32) we can conclude that $\mathscr{A}$ cannot solve the mean estimation problems to the required level of precision. As elaborated in the previous step, this implies that $\mathscr{A}$ cannot identify the set of points $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_d\}$ with the required confidence and hence

$$\begin{aligned}\mathsf{ER}(\mathscr{A}) = \sup_{f \in \mathcal{F}} \mathsf{ER}(\mathscr{A}; f) &\geq \sup_{f \in \mathcal{F}'} \mathsf{ER}(\mathscr{A}; f) \\ &\geq \frac{\alpha}{6\sqrt{d}} \gtrsim \min\left\{\min\left\{\sqrt{\frac{\sigma^2}{N}}, \max\left\{\sqrt{\frac{\sigma^2 d^2}{MN\mathsf{CC}(\mathscr{A})}}, \sqrt{\frac{\sigma^2}{MN}}\right\}\right\}, \frac{1}{\sqrt{d}}\right\}. \end{aligned} \tag{34}$$

As mentioned at the beginning of the proof, the above analysis can be easily extended to a domain with diameter R by replacing b_i with $Rb_i/2$ in the definition of the functions f_i. In a such a case, the corresponding relation for Eqn. (21) would read as

$$\sup_{f \in \mathcal{F}'} \mathbb{E}[\mathsf{ER}(\mathscr{A}, f)] \leq \frac{R\alpha}{12\sqrt{d}} \implies \max_j \sup_{(A, \mathbf{b})} p_j(\mathscr{A}; A, \mathbf{b}) \geq \frac{5}{6}. \tag{35}$$

Consequently, Eqn. (36) would be updated as

$$\mathsf{ER}(\mathscr{A}) = \geq \frac{R\alpha}{12\sqrt{d}} \gtrsim R \min\left\{\min\left\{\sqrt{\frac{\sigma^2}{N}}, \max\left\{\sqrt{\frac{\sigma^2 d^2}{MN\mathsf{CC}(\mathscr{A})}}, \sqrt{\frac{\sigma^2}{MN}}\right\}\right\}, \frac{1}{\sqrt{d}}\right\}. \tag{36}$$

The statistical term and the privacy term in the statement of Theorem 1 follow from the standard lower bounds established in the literature [Agarwal et al., 2009, Bassily et al., 2014, Levy et al., 2021]. Specifically, Theorem 1 from Agarwal et al. [2009] states that the error rate of any convex optimization algorithm $\mathscr{A}$ with a total of MN queries to a (sub)gradient oracle corresponding to a 1-Lipschitz function satisfies

$$\mathsf{ER}(\mathscr{A}) \geq c_2 \min\left\{R \cdot \sqrt{\frac{\sigma^2}{MN}}, \frac{R}{\sqrt{d}}\right\}. \tag{37}$$

To obtain the lower bound corresponding to the private estimation, note that the problem considered in this work is at least as hard as stochastic convex optimization in a centralized setting with MN samples out of which M samples can change in neighboring datasets (akin to user-level privacy). Thus, using the corresponding lower bounds from Bassily et al. [2014], Levy et al. [2021], we can conclude that

$$\mathsf{ER}(\mathscr{A}) \geq c_3 \cdot \frac{R\sqrt{d}}{\sqrt{M}N\varepsilon_{\mathsf{DP}}}. \tag{38}$$

On combining the results in Eqn. (36), (37) and (38), we arrive at the final result.

Extension to L-Lipschitz functions. The above analysis can be easily extended to accommodate L-Lipschitz functions. In particular, for the hard instance, we replace $f(x)$ with $Lf(x)$ and carry out the same series of steps. As a result, Eqn. (21) gets modified to

$$\sup_{f \in \mathcal{F}'} \mathbb{E}[\mathsf{ER}(\mathscr{A}, f)] \leq \frac{RL\alpha}{12\sqrt{d}} \implies \max_j \sup_{(A, \mathbf{b})} p_j(\mathscr{A}; A, \mathbf{b}) \geq \frac{5}{6}. \tag{39}$$

Moreover, since the gradients scale by a factor of L, the condition in Eqn. (30) changes to

$$\max_{j \le d'} \|\widehat{\theta}_j - \theta_j\|^2 \le \|\theta_1\|_2^2 = L^2\alpha^2 \tag{40}$$

and the corresponding choice of α needs to be updated to

$$\alpha^2 := \frac{1}{L^2} \min\left\{ \frac{c_1}{2} \cdot \min\left\{ \frac{\sigma^2 d}{N}, \max\left\{ \frac{\sigma^2 d^3}{MN\mathsf{CC}(\mathscr{A})}, \frac{\sigma^2 d}{MN} \right\} \right\}, 1 \right\}. \tag{41}$$

In the light of Eqn. (39), the above choice of α results in a lower bound identical to Eqn. (36). Due to a similar flavour of analysis, the relation in Eqn. (37) also does not get affected by the choice of L. However, the change in lipschitz constant results in a change of sensitivity in the gradient estimation procedures. As a result, Eqn. (38) exhibits a linear dependence with L for L-Lipschitz functions.

A.4 Proof of Lemma 1

Throughout the proof, we fix a vector $w \in \mathcal{X}_i$. For simplicity of presentation, we use the shorthand

$$\alpha_j := \langle a_j b_j, w \rangle$$

for all $j \in [d]$. Using this shorthand, we can rewrite the condition $w \in \mathcal{X}_i$ as

$$\{w \in \mathcal{X}_i\} = \left(\bigcap_{j<i} \left\{ \left| \frac{1}{\sqrt{d}} - \alpha_i \right| > \left| \frac{1}{\sqrt{d}} - \alpha_j \right| \right\} \right) \cap \left(\bigcap_{j \ge i} \left\{ \left| \frac{1}{\sqrt{d}} - \alpha_i \right| \ge \left| \frac{1}{\sqrt{d}} - \alpha_j \right| \right\} \right) \cap \left\{ |\alpha_i| \le \frac{1}{\sqrt{d}} \right\}.$$

To establish the statement of the lemma, we fix a value of j and define the following events:

$$\mathcal{E}_0 := \mathcal{E}_1 \cap \mathcal{E}_2, \tag{42a}$$
$$\mathcal{E}_1 := \left\{ \left| \frac{1}{\sqrt{d}} - \alpha_i \right| > \left| \frac{1}{\sqrt{d}} - \alpha_j \right| \right\}, \tag{42b}$$
$$\mathcal{E}_2 := \left\{ |\alpha_i| \le \frac{1}{\sqrt{d}} \right\}, \tag{42c}$$
$$\mathcal{E}_3 := \{|\alpha_j| < |\alpha_i|\}, \tag{42d}$$
$$\mathcal{E}_4 := \{\alpha_i < 0\}, \tag{42e}$$
$$\mathcal{E}_5 := \{\alpha_j > 0\}. \tag{42f}$$

Let us first analyze the condition for $j < i$. Given $\mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4^c$, we have,

$$\left| \frac{1}{\sqrt{d}} - \alpha_i \right| \overset{(a)}{=} \frac{1}{\sqrt{d}} - \alpha_i \overset{(b)}{=} \frac{1}{\sqrt{d}} - |\alpha_i| \overset{(c)}{\le} \frac{1}{\sqrt{d}} - |\alpha_j| \le \left| \frac{1}{\sqrt{d}} - |\alpha_j| \right| \overset{(d)}{\le} \left| \frac{1}{\sqrt{d}} - \alpha_j \right|, \tag{43}$$

where $(a), (b)$ and (c) are a consequence of $\mathcal{E}_2, \mathcal{E}_4^c$ and $\mathcal{E}_3$ respectively, and (d) follows from triangle inequality. As a result, $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4^c = \emptyset$ which implies $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4$. Similarly, given $\mathcal{E}_2 \cap \mathcal{E}_3^c \cap \mathcal{E}_5^c$, we have,

$$\left| \frac{1}{\sqrt{d}} - \alpha_i \right| \overset{(a)}{=} \frac{1}{\sqrt{d}} - \alpha_i \le \frac{1}{\sqrt{d}} + |\alpha_i| \overset{(b)}{\le} \frac{1}{\sqrt{d}} + |\alpha_j| \overset{(c)}{\le} \left| \frac{1}{\sqrt{d}} - \alpha_j \right|, \tag{44}$$

where $(a), (b)$ and (c) are a consequence of $\mathcal{E}_2, \mathcal{E}_3^c$ and $\mathcal{E}_5^c$ respectively. As a result, $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3^c \cap \mathcal{E}_5^c = \emptyset$ and hence, $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3^c = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3^c \cap \mathcal{E}_5$. Using these two relations, we can conclude that

$$\begin{aligned}
\mathcal{E}_0 &= \mathcal{E}_1 \cap \mathcal{E}_2 \\
&= (\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) \cup (\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3^c) \\
&= (\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3 \cap \mathcal{E}_4) \cup (\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3^c \cap \mathcal{E}_5)
\end{aligned}$$

$$\subseteq \mathcal{E}_2 \cap ((\mathcal{E}_3 \cap \mathcal{E}_4) \cup (\mathcal{E}_3^c \cap \mathcal{E}_5)) \cap \{\alpha_i \neq \alpha_j\}$$
$$\subseteq \mathcal{E}_2 \cap \{\alpha_i < \alpha_j\},$$

where in the fourth step the condition $\{\alpha_i \neq \alpha_j\}$ is a consequence of $\mathcal{E}_1$ and the last step follows by noting $((\mathcal{E}_3 \cap \mathcal{E}_4) \cup (\mathcal{E}_3^c \cap \mathcal{E}_5)) \cap \{\alpha_i \neq \alpha_j\} = \{\alpha_i < \alpha_j\}$. By a very similar sequence of arguments, we can also show that for all $j > i$

$$\left\{\left|\frac{1}{\sqrt{d}} - \alpha_i\right| \geq \left|\frac{1}{\sqrt{d}} - \alpha_j\right|\right\} \cap \left\{|\alpha_i| \leq \frac{1}{\sqrt{d}}\right\} \subseteq \left\{|\alpha_i| \leq \frac{1}{\sqrt{d}}\right\} \cap \{\alpha_i \leq \alpha_j\}.$$

Note that the only difference in this case is we allow $\alpha_i = \alpha_j$. On combining the two cases, we can conclude that $w \in \mathcal{X}_i \implies w \in \mathcal{X}_i'$, where $\mathcal{X}_i'$ is defined in Eqn. (24).

B Proof of Theorem 2

We separately establish the accuracy, privacy and communication complexity guarantees of CHARTER.

Communication Cost. The bound on communication cost is straightforward. Note that in the learning stage, CHARTER quantizes each gradient such that it can be expressed in $d \cdot J_0$ bits (J_0 bits for each coordinate). Since each agent transmits K such gradients, one for each of the K iterations, the communication cost during the learning phase is KdJ_0 bits. During the verification phase, each client transmits $K+1$ scalars, where each scalar is expressed using J_1 bits. Thus, the communication cost during the verification stage is $(K+1)J_1$. On combining the two and plugging in the values from Section 4.3, we obtain,

$$\begin{aligned}\mathsf{CC}(\text{CHARTER}) &= KdJ_0 + (K+1)J_1 \\ &\leq C_1 \cdot \left(d^2 \log(dMN) \cdot \log\left(\frac{2D_0 N \varepsilon_{\mathsf{DP}}}{\sqrt{d} + \varepsilon_{\mathsf{DP}}\sqrt{N}}\right) + d\log(dMN) \cdot \log\left(\frac{2D_1 N \varepsilon_{\mathsf{DP}}}{\sqrt{d} + \varepsilon_{\mathsf{DP}}\sqrt{N}}\right)\right) = \widetilde{\mathcal{O}}(d^2),\end{aligned}$$

as required. Here $C_1 > 0$ is a numerical constant.

Privacy. To establish the privacy guarantees, note that it is sufficient to establish that both stages of the algorithm are $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differentially private as they use distinct subsets of $\mathcal{D}$. Since the analysis is identical for all the clients, we drop the subscript m for notational simplicity. We begin with stating some useful lemmas followed by the proof.

Definition 4. *Let $f : \mathcal{Z}^N \to \mathbb{R}^k$. The ℓ_2 sensitivity of f is defined as*

$$\Delta_{2,f} := \sup_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2,$$

where $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^N$ are neighboring datasets.

Lemma 2 (Gaussian Mechanism [Dwork et al., 2006])**.** *Let $f : \mathcal{Z}^N \to \mathbb{R}^k$ obeying Definition 4 and $Y \in \mathbb{R}^k$ be a random vector each of whose entries is an i.i.d. random variable drawn according to zero mean Gaussian with variance $\frac{2\log(5/(4\delta))\Delta_{2,f}^2}{\varepsilon^2}$. The algorithm*

$$\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + Y$$

is (ε, δ)-differentially private.

Lemma 3 (Amplification by subsampling [Balle et al., 2018])**.** *For $\varepsilon, \delta \in (0,1)$, let $\mathcal{A} : \mathcal{Z}^k \to \Theta$ be an (ε, δ)-differentially private algorithm. For $N > k$ and a dataset $S \subset \mathcal{Z}^N$, let S^{WOR} be a dataset of size k obtained by randomly sampling points from S without replacement. Then $\mathcal{A}'$ obtained via $\mathcal{A}'(S) = \mathcal{A}(S^{\text{WOR}})$ is a $\left((e-1)\frac{k\varepsilon}{N}, \frac{k\delta}{N}\right)$-differentially private algorithm.*

Lemma 4 (Advanced Composition Theorem [Dwork et al., 2015, Kairouz et al., 2015]). *For any $\varepsilon > 0, \delta \in [0,1]$ and $\widetilde{\delta} \in [0,1]$, the class of (ε, δ)-differentially private mechanisms satisfies $(\widetilde{\varepsilon}_{\widetilde{\delta}}, 1-(1-\delta)^k(1-\widetilde{\delta}))$-differentially privacy under k-fold adaptive composition for*

$$\widetilde{\varepsilon}_{\widetilde{\delta}} := \min\left\{k\varepsilon, \frac{k(e^\varepsilon - 1)\varepsilon}{(e^\varepsilon+1)} + \varepsilon\sqrt{2k\log\left(\min\left\{e + \frac{\sqrt{k\varepsilon^2}}{\widetilde{\delta}}, \frac{1}{\widetilde{\delta}}\right\}\right)}\right\}.$$

We begin with the main proof starting with the privacy guarantees of the learning stage. Note that the ℓ_2 sensitivity of $\widehat{\partial\mathcal{L}}^{(\mathsf{NonPriv,b})}$ is $6KG_0/N$. Thus, using the privacy guarantees of Gaussian Mechanism (Lemma 2), we can conclude that for all iterations k, $\widehat{\partial\mathcal{L}}^{(\mathsf{Priv,b})}(x_k)$ is $(\varepsilon_0, \delta_0)$ private with respect to $\mathcal{D}^{(1,k)}$, where

$$\varepsilon_0 := \varepsilon_{\mathsf{DP}} \cdot \sqrt{\frac{K}{15\log(2.5/\delta_{\mathsf{DP}})}}; \quad \delta_0 := \frac{\delta_{\mathsf{DP}}}{2}. \tag{45}$$

Using Lemma 3 and the condition $\varepsilon_{\mathsf{DP}} \leq \frac{1}{\sqrt{K}}$, we can conclude that $\widehat{\partial\mathcal{L}}^{(\mathsf{Priv,b})}(x_k)$ $(\varepsilon_1, \delta_1)$ private with respect to $\mathcal{D}^{(1)}$, where

$$\varepsilon_1 := \frac{(e-1)\varepsilon_{\mathsf{DP}}}{2} \cdot \sqrt{\frac{1}{15K\log(2.5/\delta_{\mathsf{DP}})}}; \quad \delta_1 := \frac{\delta_{\mathsf{DP}}}{2K}. \tag{46}$$

Lastly, using Lemma 4 with $\widetilde{\delta} = \delta_{\mathsf{DP}}/2$, we can conclude that CHARTER is $(\varepsilon_{\mathsf{DP}}, \delta_{\mathsf{DP}})$ differentially private during the learning stage.

For the verification stage, note that for all k, $\widehat{\mathcal{L}}(x_k)$ is $(\varepsilon_2, \delta_2)$ private, where

$$\varepsilon_2 := \varepsilon_{\mathsf{DP}} \cdot \sqrt{\frac{9}{20K\log(2.5/\delta_{\mathsf{DP}})}}; \quad \delta_2 := \frac{\delta_{\mathsf{DP}}}{2K}. \tag{47}$$

The final privacy guarantee of the verification stage then follows by again invoking Lemma 4 with $\widetilde{\delta} = \delta_{\mathsf{DP}}/2$.

Accuracy. We establish the utility guarantees of CHARTER in four steps. In the first step, we establish that the loss estimates obtained at the end of the verification stage are close to the true values by bounding the estimation error during the verification stage. In the second step, we use these bounds to relate the excess risk of the algorithm to that of the minimum among the iterates. In the third step, we show that the iterates generated by the algorithm are such that there exists at least one iterate with sufficiently small excess risk. In the last step, we combine the results to obtain the final bound.

Step 1: Bounding the estimation error. In the verification stage, we have the following relation for all $k \in \{0, 1, 2, \ldots, K\}$

$$\begin{aligned}&\widehat{\mathcal{L}}(x_k) - \mathcal{L}(x_k)\\ &= \underbrace{\frac{1}{M}\sum_{m=1}^{M}(\widehat{\mathcal{L}}_m(x_k) - \widehat{\mathcal{L}}_m^{\mathsf{Priv}}(x_k))}_{:=L_1} + \underbrace{\frac{1}{M}\sum_{m=1}^{M}(\widehat{\mathcal{L}}_m^{\mathsf{Priv}}(x_k) - \widehat{\mathcal{L}}_m^{\mathsf{NonPriv}}(x_k))}_{:=L_2} + \underbrace{\frac{1}{M}\sum_{m=1}^{M}(\widehat{\mathcal{L}}_m^{\mathsf{NonPriv}}(x_k) - \mathcal{L}(x_k))}_{:=L_3}.\end{aligned} \tag{48}$$

We separately bound each of the three terms on the RHS.

- **Bounding L_1:** We use the following lemma that provides concentration guarantees for clipped sub-Gaussian random variables to obtain a bound on L_1.

Lemma 5. *(Lemma B.1 from Salgia and Zhao [2023]) Let $X_1, X_2, \ldots, X_n$ be a collection of i.i.d. σ^2-sub-Gaussian random variables with mean μ. For all i, define $Y_i = X_i \mathbb{1}\{|x| \leq B\}$, where $B \geq |\mu| + \sigma\sqrt{2\log(4n)}$. Then, with probability $1-\delta$,*

$$\left|\frac{1}{n}\sum_{i=1}^{n} Y_i - \mu\right| \leq \sigma\sqrt{\frac{2}{n}\log\left(\frac{4}{\delta}\right)}.$$

Note that the prescribed choice of G_1 satisfies the condition in the above lemma. On invoking the above lemma along with the choice of G_1, we can conclude that

$$|L_1| \leq \sigma_f\sqrt{\frac{6}{MN}\log\left(\frac{32(K+1)}{\delta_{\mathsf{Err}}}\right)} \tag{49}$$

holds for x_k with probability $1-\delta_{\mathsf{Err}}/(8(K+1))$. Moreover, on taking a union bound over all k, we can conclude that the above relation holds for all k with probability $1-\delta_{\mathsf{Err}}/8$.

- **Bounding L_2:** The term L_2 corresponds to the error induced by the privatization noise. Since privatization just involves the addition of Gaussian noise, we can use the concentration of Gaussian random variables to bound L_2. Thus,

$$|L_2| \leq \sigma_1\sqrt{\frac{2}{M}\log\left(\frac{16(K+1)}{\delta_{\mathsf{Err}}}\right)}. \tag{50}$$

holds with probability $1-\delta_{\mathsf{Err}}/8(K+1)$. Upon again invoking the union bound argument, we can conclude that the above relation holds for all k with probability $1-\delta_{\mathsf{Err}}/8$.

- **Bounding L_3:** On using the prescribed choice of D_1 along with the concentration of Gaussian random variables, we obtain that $|\widehat{\mathcal{L}}_m^{\mathsf{Priv}}(x_k))| \leq D_1$ holds for all m, k with probability $1-\delta_{\mathsf{Err}}/8$. Moreover, in the stochastic quantization routine, the quantization noise is bounded and hence sub-Gaussian with parameter $4D_1^2 \cdot 4^{-J_1}$. Consequently, the following relation holds for all k with probability $1-2\delta_{\mathsf{Err}}/8$:

$$|L_3| \leq 2D_1 \cdot 2^{-J_1}\sqrt{\frac{2}{M}\log\left(\frac{16(K+1)}{\delta_{\mathsf{Err}}}\right)}. \tag{51}$$

On combining the relations in (49), and (50), (51) and plugging them into (48), we obtain that

$$\begin{aligned}&|\widehat{\mathcal{L}}(x_k) - \mathcal{L}(x_k)| \\ &\leq \sigma_f\sqrt{\frac{6}{MN}\log\left(\frac{32(K+1)}{\delta_{\mathsf{Err}}}\right)} + \sigma_1\sqrt{\frac{2}{M}\log\left(\frac{16(K+1)}{\delta_{\mathsf{Err}}}\right)} + 2D_1 \cdot 2^{-J_1}\sqrt{\frac{2}{M}\log\left(\frac{16(K+1)}{\delta_{\mathsf{Err}}}\right)}\end{aligned} \tag{52}$$

holds with probability $1-\delta_{\mathsf{Err}}/2$.

Step 2: Relating the excess risks. Let $k^\star$ be as defined in (14) and $k^\dagger$ be

$$k^\dagger := \arg\min_k \mathcal{L}(x_k). \tag{53}$$

Then,

$$\mathcal{L}(x_{k^\star}) \leq \widehat{\mathcal{L}}(x_{k^\star}) + \zeta \leq \widehat{\mathcal{L}}(x_{k^\dagger}) + \zeta \leq \mathcal{L}(x_{k^\dagger}) + 2\zeta, \tag{54}$$

where ζ corresponds to the expression on the RHS in (52).

This implies that the excess risk of the point output by the algorithm is at most an additive factor larger than that of the point with the smallest excess risk in $\{x_0, x_1, \ldots, x_{K+1}\}$. Thus, it is sufficient for CHARTER

to ensure that at least one iterate obtained during the learning stage has a small excess risk. We analyze the performance of the learning stage in the step to establish the existence of such a point.

Step 3: Existence of an iterate with small excess risk. Our analysis in this step builds upon the analysis of Vaidya's method [Anstreicher, 1997, Vaidya, 1996]. Let x_c be the center of $\mathcal{X}$ and $\mathcal{X}_0$ be the set given by

$$\mathcal{X}_0 = \left\{ x_c \pm \frac{R}{2\sqrt{d}} \cdot e_1, x_c \pm \frac{R}{2\sqrt{d}} \cdot e_2, \ldots, x_c \pm \frac{R}{2\sqrt{d}} \cdot e_d \right\}, \tag{55}$$

where $\{e_1, e_2, \ldots, e_d\}$ denote the canonical basis of $\mathbb{R}^d$. In other words, $\mathcal{X}_0$ denote the vertices of an ℓ_1-ball of radius $\frac{R}{2\sqrt{d}}$, centered at x_c. Note that $\mathcal{X}_0 \subset \mathcal{X}$. Let $x^\star$ be any fixed minimizer of the function f in $\mathcal{X}$ and $\mathcal{X}_1$ be the set given by

$$\mathcal{X}_1 := (1-\omega)x^\star + \omega \mathcal{X}_0, \tag{56}$$

where $\omega = \frac{\sigma_{\max}}{\sqrt{MN}}$. Thus, $\mathrm{conv}(\mathcal{X}_1)$ is an ℓ_1-ball of radius $\frac{R\sigma_{\max}}{2\sqrt{dMN}}$, centered at $x^\star + \omega(x_c - x^\star)$. Here $\mathrm{conv}(\mathcal{Y})$ denotes the convex hull of the set $\mathcal{Y}$. Using convexity of $\mathcal{X}$ and the relation $\mathcal{X}_0 \subset \mathcal{X}$, we can conclude that $\mathcal{X}_1 \subset \mathcal{X}$. Let $\overline{x}_1 \in \mathcal{X}_1$ and $\overline{x}_0$ be the corresponding point in $\mathcal{X}_0$. Thus,

$$\|\overline{x}_1 - x^\star\|_2 = \omega \|\overline{x}_0 - x^\star\|_2 \leq \omega R = \frac{R\sigma_{\max}}{\sqrt{MN}}. \tag{57}$$

We claim that there exists an iteration $k' \in \{0, 2, \ldots, K\}$ such that $\mathcal{X}_1 \subset P_{k'}$ and $\mathcal{X}_1 \not\subset P_{k'+1}$, where P_k denotes the polyhedron (A_k, b_k) constructed during Vaidya's method. In other words, during the iteration k', one of the points in $\mathcal{X}_1$ is eliminated. We defer the proof of the claim to the end of the section.

We show that $x_{k'}$ is the required point that has a small excess risk. Firstly, note that a point is eliminated in Vaidya's method only when a constraint is added. Secondly, recall that in the k^{th} iteration, we add the constraint $c_k^\top x \geq \beta_k$, where $c_k = -\widehat{\partial \mathcal{L}}(x_k)$ and $\beta_k \leq c_k^\top x_k$. This implies all points eliminated during the k^{th} iteration satisfy $c_k^\top x < \beta_k \leq c_k^\top x_k$. Thus, if a point $\overline{x} \in \mathcal{X}_1$ is eliminated in iteration k', then $\overline{x}$ satisfies

$$\left\langle -\widehat{\partial \mathcal{L}}(x_{k'}), \overline{x} - x_{k'} \right\rangle < 0 \implies \left\langle \widehat{\partial \mathcal{L}}(x_{k'}), \overline{x} - x_{k'} \right\rangle > 0. \tag{58}$$

Consequently,

$$\begin{aligned}
\mathcal{L}(x_{k'}) &< \mathcal{L}(x_{k'}) + \left\langle \widehat{\partial \mathcal{L}}(x_{k'}), \overline{x} - x_{k'} \right\rangle \\
&< \mathcal{L}(x_{k'}) + \langle \partial \mathcal{L}(x_{k'}), \overline{x} - x_{k'} \rangle + \left\langle \widehat{\partial \mathcal{L}}(x_{k'}) - \partial \mathcal{L}(x_{k'}), \overline{x} - x_{k'} \right\rangle \\
&< \mathcal{L}(\overline{x}) + \left\langle \widehat{\partial \mathcal{L}}(x_{k'}) - \partial \mathcal{L}(x_{k'}), \overline{x} - x_{k'} \right\rangle \\
&< \mathcal{L}(x^\star) + \frac{R\sigma_{\max}}{\sqrt{MN}} + \left\langle \widehat{\partial \mathcal{L}}(x_{k'}) - \partial \mathcal{L}(x_{k'}), \overline{x} - x_{k'} \right\rangle,
\end{aligned} \tag{59}$$

where the first line follows from (58), the third line from the convexity of $\mathcal{L}$ and the fourth line from (57) and 1-Lipschitzness of $\mathcal{L}$ (Assumption 1). Thus, if the error $\left\langle \widehat{\partial \mathcal{L}}(x_{k'}) - \partial \mathcal{L}(x_{k'}), \overline{x} - x_{k'} \right\rangle$ is small, the excess risk at k' is also small.

To establish this result, we first state a relation that will be useful for the analysis. We claim that

$$\frac{1}{4} \leq T_{k,m} \cdot \frac{3K}{N} \leq 1 \tag{60}$$

holds for all clients m and iterations k with probability $1 - \delta_{\mathsf{Err}}/10$ as long as $N = \Omega(d \log(MK))$. We defer the proof of the claim to the end of the section. Moreover, we carry out the remainder of the analysis conditioned on this event.

We establish that $\left\langle \widehat{\partial\mathcal{L}}(x_k) - \partial\mathcal{L}(x_k), \overline{x} - x_k \right\rangle$ is small for all iterations k which immediately yields the bound for iteration k'. We use a similar modus operandi as used in Step 1. Consider the k^{th} iteration and any fixed $\overline{x} \in \mathcal{X}_1$. We have,

$$\left\langle \widehat{\partial\mathcal{L}}(x_k) - \partial\mathcal{L}(x_k), \overline{x} - x_k \right\rangle = \underbrace{\frac{1}{M}\sum_{m=1}^{M}\left\langle \frac{N}{3KT_{k,m}}\widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k) - \partial\mathcal{L}(x_k), \overline{x} - x_k \right\rangle}_{:=W_1}$$
$$+ \underbrace{\frac{1}{M}\sum_{m=1}^{M}\frac{N}{3KT_{k,m}}\cdot\left\langle \widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,b}}(x_k) - \widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k), \overline{x} - x_k \right\rangle}_{:=W_2}$$
$$+ \underbrace{\frac{1}{M}\sum_{m=1}^{M}\left\langle \widehat{\partial\mathcal{L}}_m(x_k) - \widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,u}}(x_k), \overline{x} - x_k \right\rangle}_{:=W_3}. \tag{61}$$

Similar to Step 1, we separately bound each of the three terms on the RHS of Eq. (61).

- **Bounding W_1:** To bound W_1, note that

$$\frac{N}{3KT_{k,m}}\widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k) = \frac{1}{T_{k,m}}\sum_{z\in\mathcal{D}_m^{(1,k)}}\mathsf{clip}(\partial\ell(x_k;z);G_0)\cdot\mathbb{1}\{z\notin\cup_{j=1}^{k-1}\mathcal{D}_m^{(1,j)}\},$$

  is an estimate of $\partial\mathcal{L}(x_k)$ using $T_{k,m}$ independent (clipped) samples. If $\overline{v}$ denotes the unit vector along $\overline{x} - x_k$, then using the sub-Gaussianity of the samples (Assumption 2), we know that $\langle\partial\ell(x;z),\overline{v}\rangle$ is a sub-Gaussian random variable with parameter σ_g^2/d. Moreover, the choice of G_0 satisfies the condition in Lemma 5 for the random variable $\langle\ell(x;z),\overline{v}\rangle$. Thus, using Lemma 5, we can conclude that

$$|W_1| = \|\overline{x} - x_k\|_2\cdot\left|\frac{1}{M}\sum_{m=1}^{M}\left\langle\frac{N}{3KT_{k,m}}\widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k) - \partial\mathcal{L}(x_k),\overline{v}\right\rangle\right|$$
$$\le R\sigma_g\sqrt{\log\left(\frac{80d(K+1)}{\delta_{\mathsf{Err}}}\right)\cdot\frac{2}{M^2}\sum_{m=1}^{M}\frac{1}{T_{k,m}}}$$
$$\le R\sigma_g\sqrt{\frac{24K}{dMN}\cdot\log\left(\frac{80d(K+1)}{\delta_{\mathsf{Err}}}\right)}, \tag{62}$$

  holds with probability $1-\delta_{\mathsf{Err}}/(20d(K+1))$. Here, the last line follows using (60). Using a union bound we obtain that the above relation holds for all k with probability $1-\delta_{\mathsf{Err}}/(20d)$.

- **Bounding W_2:** Note that $\frac{N}{3KT_{k,m}}\cdot\left\langle\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,b}}(x_k) - \widehat{\partial\mathcal{L}}_m^{\mathsf{NonPriv,b}}(x_k),\overline{x}-x_k\right\rangle$ is a Gaussian random variable with variance $\left(\frac{N}{3KT_{k,m}}\right)^2 R^2\sigma_0^2 \le 16R^2\sigma_0^2$ where the inequality follows from the bound in Eqn. (60). Consequently, for all k,

$$|W_2| \le 4R\sigma_0\cdot\sqrt{\frac{2}{M}\log\left(\frac{40d(K+1)}{\delta_{\mathsf{Err}}}\right)} \tag{63}$$

  holds with probability $1-\delta_{\mathsf{Err}}/(20d)$.

- **Bounding W_3:** Lastly, to bound W_3, we use the same approach as used for L_3. For the choice D_0 and in light of Eqn. (60), we can conclude that the event $\mathcal{E} = \{\|\widehat{\partial\mathcal{L}}_m^{\mathsf{Priv,u}}(x_k)\|_\infty \le D_0 \;\; \forall m,k\}$ holds with probability $1-\delta_{\mathsf{Err}}/10$. Since each coordinate of the quantized vector is an independent sub-Gaussian

random variable with parameter $4D_0^2 \cdot 4^{-J_0}$, $\left\langle \widehat{\partial \mathcal{L}}_m(x_k) - \widehat{\partial \mathcal{L}}_m^{\mathsf{Priv,u}}(x_k), \overline{x} - x_k \right\rangle$ is a sub-Gaussian random variable with parameter $4D_0^2 \cdot 4^{-J_0} \cdot \|\overline{x} - x_k\|^2$. Using the concentration of sub-Gaussian random variables and a union bound argument, we can conclude that, conditioned on $\mathcal{E}$,

$$|W_3| \leq 2D_0 \cdot 2^{-J_0} \cdot R \cdot \sqrt{\frac{2}{M} \log\left(\frac{40d(K+1)}{\delta_{\mathsf{Err}}}\right)} \tag{64}$$

holds for all k with probability $1 - \delta_{\mathsf{Err}}/(20d)$. Here, we used the relation $\|\overline{x} - x_k\| \leq R$.

Thus, the relations (61), (64), (63), and (62) taken together along with a union bound over $\overline{x} \in \mathcal{X}_1$ imply that

$$\begin{aligned} |\left\langle \widehat{\partial \mathcal{L}}(x_k) - \partial \mathcal{L}(x_k), \overline{x} - x_k \right\rangle| \leq 2D_0 \cdot 2^{-J_0} \cdot R \cdot \sqrt{\frac{2}{M} \log\left(\frac{40d(K+1)}{\delta_{\mathsf{Err}}}\right)} + 4R\sigma_0 \cdot \sqrt{\frac{2}{M} \log\left(\frac{40d(K+1)}{\delta_{\mathsf{Err}}}\right)} \\ + R\sigma_g \sqrt{\frac{24K}{dMN} \cdot \log\left(\frac{80d(K+1)}{\delta_{\mathsf{Err}}}\right)} \end{aligned} \tag{65}$$

holds for all $k \in \{0, 1, \ldots, K\}$, $m \in \{1, 2, \ldots, M\}$ and $\overline{x} \in \mathcal{X}_1$ with probability $1 - 3\delta_{\mathsf{Err}}/10$.

Step 4: Putting it together. On combining (52), (54), (59), and (65), plugging in the prescribed parameter values from Section 4.3 and accounting for the conditioning on the $\mathcal{E}$ and Eqn. (60) , we obtain that

$$\begin{aligned} &\mathcal{L}(x_{k^\star}) - \mathcal{L}(x^\star) \\ &\leq C_1(R\sigma_g + \sigma_f)\sqrt{\frac{\log N}{MN} \cdot \log\left(\frac{d^2 \log(MN)}{\delta_{\mathsf{Err}}}\right)} + C_2 R' \frac{\sqrt{d}\log(MN)}{N\varepsilon_{\mathsf{DP}}} \log\left(\frac{d\log(MN)}{\delta_{\mathsf{DP}}}\right) \sqrt{\log\left(\frac{d^2\log(MN)}{\delta_{\mathsf{Err}}}\right)} \end{aligned} \tag{66}$$

holds with probability $1 - \delta_{\mathsf{Err}}$ for some constants C_1, C_2 that are independent of all problem parameters and $R' = R(1 + \sigma_g) + \sigma_f$.

Proving the claim (60). To establish this result, firstly note that we sample (uniformly at random) $N/3K$ points for K rounds from a dataset of size $2N/3$. Thus, for all rounds, the number of previously seen data points are at most $N/3$, which is half the dataset. To lower bound the value of $T_{k,m}$, we obtain an upper bound on $\widetilde{T}_{k,m} = \frac{N}{3K} - T_{k,m}$, i.e., the number of points in the set that have been seen previously by the algorithm. Using Hoeffding's inequality, which also holds for sampling without replacement [Bardenet and Maillard, 2015, Hoeffding, 1994], we can conclude that with probability $1 - \delta_{\mathsf{Err}}/(10M(K+1))$,

$$\widetilde{T}_{k,m} \leq \frac{N}{3K} \cdot \frac{3N_{k,m}}{2N} + \sqrt{\frac{N}{6K} \log\left(\frac{10M(K+1)}{\delta_{\mathsf{Err}}}\right)}, \tag{67}$$

where $N_{k,m}$ denotes the number of samples that have been seen before iteration k at client m. As shown above, $N_{k,m} \leq N/3$ for all k, m with probability 1. Thus, if $N \geq 24K \log\left(\frac{10M(K+1)}{\delta_{\mathsf{Err}}}\right)$, then,

$$\widetilde{T}_{k,m} \leq \frac{N}{3K} \cdot \left(\frac{1}{2} + \sqrt{\frac{3K}{2N} \log\left(\frac{10M(K+1)}{\delta_{\mathsf{Err}}}\right)}\right) \leq \frac{N}{3K} \cdot \left(\frac{1}{2} + \sqrt{\frac{1}{16}}\right) \leq \frac{3}{4} \cdot \frac{N}{3K}. \tag{68}$$

On taking union bound over k and m, we can conclude that

$$T_{k,m} \geq \frac{1}{4} \cdot \frac{N}{3K} \tag{69}$$

holds for all $k \in \{0, 1, \ldots, K\}$ and $m \in \{1, 2, \ldots, M\}$ with probability $1 - \delta_{\mathsf{Err}}/10$. The upper bound on $T_{k,m}$ follows directly by definition.

Proving that $\overline{x} \in \mathcal{X}_1$ is eliminated. We establish this claim using contradiction. Specifically, if we assume $\mathcal{X}_1 \subset P_k$ for all $k \leq K$, then P_k contains an ℓ_1 ball of radius $\frac{\omega R}{2\sqrt{d}}$ for all $k \leq K$. This is because P_k is a convex set and if $\mathcal{X}_1 \subset P_k$, then the convex hull of $\mathcal{X}_1$, which is an ℓ_1 ball, also lies in P_k. If for all $k \leq K$, P_k contains an ℓ_1 ball of radius $\frac{\omega R}{2\sqrt{d}}$, then for all $k \leq K$

$$\log(\text{vol}(P_k)) \geq d \log\left(\frac{\omega R}{\sqrt{d}}\right) - \log(d!) \geq d \log\left(\frac{R\sigma_g}{d\sqrt{dMN}}\right). \tag{70}$$

The RHS is the logarithm of the volume of an ℓ_1 ball of radius $\frac{\omega R}{2\sqrt{d}}$, where $d!$ denotes the factorial of d. On the other hand, Vaidya [1996] shows that the volume of the polyhedron after k^{th} iteration of Vaidya's method is given by

$$\log(\text{vol}(P_k)) \leq d \log\left(\frac{2d}{\gamma}\right) - V^0 - \frac{\gamma k}{2}, \tag{71}$$

where γ is the parameter of Vaidya's algorithm and V^0 is the initial volumetric barrier. Since we start with a hypercube, its volumetric center is the same as the geometric center of the hypercube. Consequently, the volumetric barrier of a cube of side b is given by

$$V_{\text{cube}} = \frac{d}{2} \log\left(\frac{8}{b^2}\right). \tag{72}$$

Since the diameter of $\mathcal{X}$ is R, the initial volumetric barrier is given by

$$V^0 = \frac{d}{2} \log\left(\frac{8d}{R^2}\right). \tag{73}$$

On plugging the above relation into (71) along with the value of K, we obtain,

$$\begin{aligned}
\log(\text{vol}(P_K)) &\leq d \log\left(\frac{2d}{\gamma}\right) - \frac{d}{2} \log\left(\frac{8d}{R^2}\right) - \frac{\gamma}{2} \cdot \frac{4d}{\gamma} \log\left(\frac{d\sqrt{MN}}{\gamma\sigma_g}\right) \\
&\leq d \log\left(\frac{2d}{\gamma} \cdot \frac{R}{\sqrt{8d}} \cdot \frac{\gamma\sigma_g}{d^2 MN}\right) \\
&\leq d \log\left(\frac{R\sigma_g}{dMN\sqrt{2d}}\right).
\end{aligned}$$

This results in a contradiction with the lower bound on the volume of P_K from (70). This implies that $\mathcal{X}_1 \not\subset P_K$ and hence some $\overline{x} \in \mathcal{X}_1$ was eliminated during the algorithm.