Journal of Machine Learning Research 21 (2020) 1-39 Submitted 4/19; Revised 3/20; Published 9/20

Empirical Risk Minimization in the Non-interactive Local Model of Differential Privacy

Di Wang DWANG45@BUFFALO.EDU
Department of Computer Science and Engineering
University at Buffalo, SUNY
Buffalo, NY 14260, USA

Marco Gaboardi GABOARDI@BU.EDU
Department of Computer Science
Boston University
Boston, MA 02215, USA

Adam Smith ADS22@BU.EDU
Department of Computer Science
Boston University
Boston, MA 02215, USA

Jinhui Xu JINHUI@BUFFALO.EDU
Department of Computer Science and Engineering
University at Buffalo, SUNY
Buffalo, NY 14260, USA

Editor: Mehryar Mohri

Abstract

In this paper, we study the Empirical Risk Minimization (ERM) problem in the non-interactive Local Differential Privacy (LDP) model. Previous research on this problem (Smith et al., 2017) indicates that the sample complexity, to achieve error α, needs to be exponentially depending on the dimensionality p for general loss functions. In this paper, we make two attempts to resolve this issue by investigating conditions on the loss functions that allow us to remove such a limit. In our first attempt, we show that if the loss function is (∞, T)-smooth, by using the Bernstein polynomial approximation we can avoid the exponential dependency in the term of α. We then propose player-efficient algorithms with 1-bit communication complexity and $O(1)$ computation cost for each player. The error bound of these algorithms is asymptotically the same as the original one. With some additional assumptions, we also give an algorithm which is more efficient for the server. In our second attempt, we show that for any 1-Lipschitz generalized linear convex loss function, there is an (ϵ, δ)-LDP algorithm whose sample complexity for achieving error α is only linear in the dimensionality p. Our results use a polynomial of inner product approximation technique. Finally, motivated by the idea of using polynomial approximation and based on different types of polynomial approximations, we propose (efficient) non-interactive locally differentially private algorithms for learning the set of k-way marginal queries and the set of smooth queries.

Keywords: Differential Privacy, Empirical Risk Minimization, Local Differential Privacy, Round Complexity, Convex Learning

1. Introduction

A tremendous amount of individuals' data is accumulated and shared every day. This data has the potential to bring improvements in scientific and medical research and to help improve several aspects of daily lives. However, due to the sensitive nature of such data, some care needs to be taken while analyzing them. Private data analysis seeks to combine the benefits of learning from data with the guarantee of privacy-preservation. Differential privacy (Dwork et al., 2006) has emerged as a rigorous notion for privacy-preserving accurate data analysis with a guaranteed bound on the increase in harm for each individual to contribute his/her data. Methods to guarantee differential privacy have been widely studied, and recently adopted in industry (Near, 2018; Erlingsson et al., 2014).

Two main user models have emerged for differential privacy: the central model and the local one. In the central model, data are managed by a trusted centralized entity which is responsible for collecting them and for deciding which differentially private data analysis to perform and to release. A classical use case for this model is the one of census data (Haney et al., 2017). In the local model instead, each individual manages his/her proper data and discloses them to a server through some differentially private mechanisms. The server collects the (now private) data of each individual and combines them into a resulting data analysis. A classical use case for this model is the one aiming at collecting statistics from user devices like in the case of Google's Chrome browser (Erlingsson et al., 2014), and Apple's iOS-10 (Near, 2018; Tang et al., 2017).

In the local model, there are two basic kinds of protocols: interactive and non-interactive. Bassily and Smith (2015) have recently investigated the power of non-interactive differentially private protocols. Because of its simplicity and its efficiency in term of network latency, this type of protocols seems to be more appealing for real world applications. Both Google and Apple use the non-interactive model in their projects (Near, 2018; Erlingsson et al., 2014).

Despite being used in industry, the local model has been much less studied than the central one. Part of the reason for this is that there are intrinsic limitations in what one can do in the local model. As a consequence, many basic questions, that are well studied in the central model, have not been completely understood in the local model, yet.

In this paper, we study differentially private Empirical Risk Minimization in the non-interactive local model. Before presenting our contributions and showing comparisons with previous works, we first introduce the problem and discuss our motivations.

Problem setting (Smith et al., 2017; Kasiviswanathan et al., 2011) Given a convex, closed and bounded constraint set $\mathcal{C} \subseteq \mathbb{R}^p$, a data universe $\mathcal{D}$, and a loss function $\ell : \mathcal{C} \times \mathcal{D} \mapsto \mathbb{R}$, a dataset $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \in \mathcal{D}^n$ with data records $\{x_i\}_{i=1}^n \subset \mathbb{R}^p$ and labels (responses) $\{y_i\}_{i=1}^n \subset \mathbb{R}$ defines an *empirical risk* function: $L(w; D) = \frac{1}{n}\sum_{i=1}^n \ell(w; x_i, y_i)$ (note that in some settings, such as mean estimation, there may not be separate labels). When the inputs are drawn i.i.d from an unknown underlying distribution $\mathcal{P}$ on $\mathcal{D}$, we can also define the *population risk* function: $L_{\mathcal{P}}(w) = \mathbb{E}_{D\sim\mathcal{P}^n}[\ell(w; D)]$.

Thus, we have the following two types of excess risk measured at a particular output w_{priv}: The empirical risk,

$$\text{Err}_D(w_{\text{priv}}) = L(w_{\text{priv}}; D) - \min_{w \in \mathcal{C}} L(w; D)\,,$$

and the population risk,

$$\text{Err}_{\mathcal{P}}(w_{\text{priv}}) = L_{\mathcal{P}}(w_{\text{priv}}) - \min_{w \in \mathcal{C}} L_{\mathcal{P}}(w).$$

The problem considered in this paper is to design non-interactive LDP protocols that find a private estimator w_{priv} to minimize the empirical and/or population excess risks. Alternatively, we can express our goal on this problem in terms of *sample complexity*: find the smallest n for which we can design protocols that achieve error at most α (in the worst case over data sets, or over generating distributions, depending on how we measure risk).

Duchi, Jordan, and Wainwright (2013) first considered the worst-case error bounds for LDP convex optimization. For 1-Lipchitz convex loss functions over a bounded constraint set, they gave a highly interactive SGD-based protocol with sample complexity $n = O(p/\epsilon^2\alpha^2)$; moreover, they showed that no LDP protocol which interacts with each player only once can achieve asymptotically better sample complexity, even for linear losses.

Smith, Thakurta, and Upadhyay (2017) considered the round complexity of LDP protocols for convex optimization. They observed that known methods perform poorly when constrained to be run non-interactively. They gave new protocols that improved on the state-of-the-art but nevertheless required sample complexity exponential in p. Specifically, they showed:

Theorem 1 ((Smith et al., 2017)) *Under some assumptions on the loss functions, there is a non-interactive ϵ-LDP algorithm such that for all distribution $\mathcal{P}$ on $\mathcal{D}$, with probability $1-\beta$, its population risk is upper bounded by*

$$Err_{\mathcal{P}}(w_{priv}) \leq \tilde{O}\big((\frac{\sqrt{p}\log^2(1/\beta)}{\epsilon^2 n})^{\frac{1}{p+1}}\big). \tag{1}$$

A similar result holds for empirical risk $Err_D(w_{priv})$. Equivalently, to ensure an error no more than α, the sample complexity needs to be $n = \tilde{O}(\sqrt{p}c^p\epsilon^{-2}\alpha^{-(p+1)})$, where c is some constant (approximately 2).

Furthermore, lower bounds on the parallel query complexity of stochastic optimization (e.g., Nemirovski (1994); Woodworth et al. (2018)) mean that, for natural classes of LDP optimization protocols (based on the measure of noisy gradients), the exponential dependence of the sample size on the dimensionality p (in the terms of $\alpha^{-(p+1)}$ and c^p) is, in general, unavoidable (Smith et al., 2017).

This situation is somehow undesirable: when the dimensionality p is high and the target error is low, the dependency on $\alpha^{-(p+1)}$ could make the sample size quite large. However, several results have already shown that for some specific loss functions, the exponential dependency on the dimensionality can be avoided. For example, Smith et al. (2017) show that, in the case of linear regression, there is a non-interactive (ϵ, δ)-LDP algorithm whose sample complexity for achieving error at most α in the empirical risk is $n = O(p \log(1/\delta)\epsilon^{-2}\alpha^{-2})$.[1] This indicates that there is a gap between the general case and some specific loss functions. This motivates us to consider the following basic question:

> *Are there natural conditions on the loss function which allow for non-interactive ϵ-LDP algorithms with sample complexity sub-exponentially (ideally, it should be polynomially or even linearly) depending on the dimensionality p in the terms of α or c?*

To answer this question, we make two attempts to approach the problem from different perspectives. In the first attempt, we show that the exponential dependency on p in the term of $\alpha^{-(p+1)}$ can be avoided if the loss function is sufficiently smooth. In the second attempt, we show that there exists a family of loss functions whose sample complexities is depending on p. Below is a summary of our main contributions.

Our Contributions:

1. In our first attempt, we investigate the conditions on the loss function guaranteeing a sample complexity which depends polynomially on p in the term of α. We first show that by using Bernstein polynomial approximation, it is possible to achieve a non-interactive ϵ-LDP algorithm in constant or low dimensions with the following properties. If the loss function is $(8, T)$-smooth (see Definition 7), then with a sample complexity of $n = \tilde{O}\big((c_0 p^{\frac{1}{4}})^p \alpha^{-(2+\frac{p}{2})}\epsilon^{-2}\big)$, the excess empirical risk is ensured to be $\text{Err}_D \leq \alpha$. If the loss function is (∞, T)-smooth, the sample complexity can be further improved to $n = \tilde{O}(4^{p(p+1)} D_p^2 p \epsilon^{-2}\alpha^{-4})$, where D_p depends only on p. Note that in the first case, the sample complexity is lower than the one in (Smith et al., 2017) when $\alpha \leq O(\frac{1}{p})$, and in the second case, the sample complexity depends only polynomially on α^{-1}, instead of the exponential dependence as in (Smith et al., 2017). Furthermore, our algorithm does not assume convexity for the loss function and thus can be applied to non-convex loss functions.

2. Then, we address the efficiency issue, which has only been partially studied in previous works (Smith et al., 2017). Following an approach similar to (Bassily and Smith, 2015), we propose an algorithm for our loss functions which has only 1-bit communication cost and $O(1)$ computation cost for each client, and achieves asymptotically the same error bound as the original one. Additionally, we present a novel analysis for the server showing that if the loss function is convex and Lipschitz and the convex set satisfies some natural conditions, then there is an algorithm which achieves the error bound of $O(p\alpha)$ and runs in polynomial time in $\frac{1}{\alpha}$ (instead of exponential time as in (Smith et al., 2017)) if the loss function is (∞, T)-smooth.

3. In our second attempt, we study the conditions on the loss function guaranteeing a sample complexity which depends polynomially on p (in both terms of α and c). We show that for any 1-Lipschitz generalized linear convex loss function, *i.e.,*

[1] Note that these two results are for non-interactive (ϵ, δ)-LDP, and we mainly focus on non-interactive ϵ-LDP algorithms. Thus, we omit terms related to $\log(1/\delta)$ in this paper.

$\ell(w; x, y) = f(y_i \langle w, x_i \rangle)$ for some 1-Lipschitz convex function f, there is a non-interactive (ϵ, δ)-LDP algorithm, whose sample complexity for achieving error α in empirical risk depends only linearly, instead of exponentially, on the dimensionality p. Our idea is based on results from Approximation Theory. We first consider the case of hinge loss functions. For this class of functions, we use Bernstein polynomials to approximate their derivative functions after smoothing, and then we apply the Stochastic Inexact Gradient Descent algorithm (Dvurechensky and Gasnikov, 2016). Next we extend the result to all convex general linear functions. The key idea is to show that any 1-Lipschitz convex function in $\mathbb{R}$ can be expressed as a linear combination of some linear functions and hinge loss functions, *i.e.,* plus functions of inner product $[\langle w, s \rangle]_+ = \max\{0, \langle w, s \rangle\}$. Based on this, we propose a general method which is called the polynomial of inner product approximation.

4. Finally, we show the generality of our technique by applying polynomial approximation to other problems. Specifically, we give a non-interactive LDP algorithm for answering the class of k-way marginals queries, by using Chebyshev polynomial approximation, and a non-interactive LDP algorithm for answering the class of smooth queries, by using trigonometric polynomial approximation.

Table 1 shows the detailed comparisons between our results and the results in (Smith et al., 2017; Zheng et al., 2017).

Preliminary results of this work have already appeared in the 2018 Thirty-second Conference on Neural Information Processing Systems (NeurIPS'18) (Wang et al., 2018) and in the 2019 Algorithmic Learning Theory (ALT'19) (Wang et al., 2019).

2. Related Work

Differentially private convex optimization, first formulated by Chaudhuri and Monteleoni (2009) and Chaudhuri, Monteleoni, and Sarwate (2011), has been the focus of an active line of work for the past decade, such as (Wang et al., 2017a; Bassily et al., 2014; Kifer et al., 2012; Chaudhuri et al., 2011; Talwar et al., 2015; Wang and Xu, 2019). We discuss here only those results which are related to the local model.

Kasiviswanathan et al. (2011) initiated the study of learning under local differential privacy. Specifically, they showed a general equivalence between learning in the local model and learning in the statistical query model. Beimel et al. (2008) gave the first lower bounds for the accuracy of LDP protocols, for the special case of counting queries (equivalently, binomial parameter estimation).

The general problem of LDP convex risk minimization was first studied by Duchi et al. (2013), which provided tight upper and lower bounds for a range of settings. Subsequent work considered a range of statistical problems in the LDP setting, providing upper and lower bounds—we omit a complete list here.

Smith et al. (2017) initiated the study of the round complexity of LDP convex optimization, connecting it to the parallel complexity of (non-private) stochastic optimization.

Convex risk minimization in the *non-interactive* LDP model received considerable recent attentions (Zheng et al., 2017; Smith et al., 2017; Wang et al., 2018) (see Table 1 for details). Smith et al. (2017) first studied the problem with general convex loss functions and showed

Methods	Sample Complexity	Assumption on the Loss Function
(Smith et al., 2017, Claim 4)	$\tilde{O}(4^p \alpha^{-(p+2)} \epsilon^{-2})$	1-Lipschitz
(Smith et al., 2017, Theorem 10)	$\tilde{O}(2^p \alpha^{-(p+1)} \epsilon^{-2})$	1-Lipschitz and Convex
Smith et al. (2017)	$\Theta(p\epsilon^{-2}\alpha^{-2})$	Linear Regression
Zheng et al. (2017)	$O\big(p(\frac{8}{\alpha})^{4\log\log(8/\alpha)}(\frac{4}{\epsilon})^{2c\log(8/\alpha)+2}(\frac{1}{\alpha^2\epsilon^2})\big)$	Smooth Generalized Linear
This Paper	$\tilde{O}\big((c_0 p^{\frac{1}{4}})^p \alpha^{-(2+\frac{p}{2})} \epsilon^{-2}\big)$	$(8, T)$-smooth
This Paper	$\tilde{O}(4^{p(p+1)} D_p^2 \epsilon^{-2} \alpha^{-4})$	(∞, T)-smooth
This Paper	$p \cdot \big(\frac{C}{\alpha^3}\big)^{O(1/\alpha^3)} / \epsilon^{O(\frac{1}{\alpha^3})}$	Hinge Loss
This Paper	$p \cdot \big(\frac{C}{\alpha^3}\big)^{O(1/\alpha^3)} / \epsilon^{O(\frac{1}{\alpha^3})}$	1-Lipschitz Convex Generalized Linear

Table 1: Comparisons on the sample complexities for achieving error α in the empirical risk, where c is a constant. We assume that $\|x_i\|_2, \|y_i\| \leq 1$ for every $i \in [n]$ and the constraint set $\|\mathcal{C}\|_2 \leq 1$. Asymptotic statements assume $\epsilon, \delta, \alpha \in (0, 1/2)$ and ignore dependencies on $\log(1/\delta)$.

that the exponential dependence on the dimensionality is unavoidable for a class of non-interactive algorithms. In this paper, we investigate the conditions on the loss function that allow us to avoid the issue of exponential dependence on p in the sample complexity.

The work most related to ours (*i.e.,* the second attempt) is that of (Zheng et al., 2017), which also considered some specific loss functions in high dimensions, such as sparse linear regression and kernel ridge regression. The major differences with our results are the following. Firstly, although they studied a similar class of loss functions (*i.e.,* Smooth Generalized Linear Loss functions) and used the polynomial approximation approach, their approach needs quite a few assumptions on the loss function in addition to the smoothness condition, such as Lipschitz smoothness and boundedness on the higher order derivative functions, which are clearly not satisfied by the hinge loss functions. Contrarily, our results only assume the 1-Lipschitz convex condition on the loss function. Secondly, even though the idea in our algorithm for the hinge loss functions is similar to theirs, we also consider generalized linear loss function by using techniques from approximation theory.

Kulkarni et al. (2017); Zhang et al. (2018) recently studied the problem of releasing k-way marginal queries in LDP. They compared different LDP methods to release marginal statistics, but did not consider methods based on polynomial approximation.

For other problems under LDP model, (Bun et al., 2018; Bassily and Smith, 2015; Bassily et al., 2017; Hsu et al., 2012) studied heavy hitter problem, (Ye and Barg, 2017; Kairouz et al., 2016; Wang et al., 2017b; Acharya et al., 2018) considered local private distribution estimation. The polynomial approximation approach has been used under the central model in (Aldà and Rubinstein, 2017; Wang et al., 2016; Thaler et al., 2012; Zheng et al., 2017).

3. Preliminaries

Differential privacy in the local model. In LDP, we have a data universe $\mathcal{D}$, n players with each holding a private data record $x_i \in \mathcal{D}$, and a server coordinating the protocol. An LDP protocol executes a total of T rounds. In each round, the server sends a message, which is also called a query, to a subset of the players requesting them to run a particular algorithm. Based on the query, each player i in the subset selects an algorithm Q_i, runs it on her own data, and sends the output back to the server.

Definition 2 *(Evfimievski et al., 2003; Dwork et al., 2006) An algorithm Q is (ϵ, δ)-locally differentially private (LDP) if for all pairs $x, x' \in \mathcal{D}$, and for all events E in the output space of Q, we have*

$$Pr[Q(x) \in E] \leq e^{\epsilon} Pr[Q(x') \in E] + \delta.$$

A multi-player protocol is (ϵ, δ)-LDP if for all possible inputs and runs of the protocol, the transcript of player i's interaction with the server is (ϵ, ϵ)-LDP. If $T = 1$, we say that the protocol is ϵ non-interactive LDP. When $\delta = 0$, we call it is ϵ-LDP.

Algorithm 1 1-dim LDP-AVG

```
Input: Player $i \in [n]$ holding data $v_i \in [0, b]$, privacy parameter $\epsilon$.
for Each Player $i$ do
    Send $z_i = v_i + \text{Lap}(\frac{b}{\epsilon})$
end for
for The Server do
    Output $a = \frac{1}{n}\sum_{i=1}^{n} z_i$.
end for
```

Since we only consider non-interactive LDP through the paper, we will use LDP as non-interactive LDP below.

As an example that will be useful throughout the paper, the next lemma shows a property of an ϵ-LDP algorithm for computing 1-dimensional average.

Lemma 3 *For any $\epsilon > 0$, Algorithm 1 is ϵ-LDP. Moreover, if player $i \in [n]$ holds value $v_i \in [0, b]$ and $n > \log \frac{2}{\beta}$ with $0 < \beta < 1$, then, with probability at least $1 - \beta$, the output $a \in \mathbb{R}$ satisfies:*

$$|a - \frac{1}{n}\sum_{i=1}^{n} v_i| \leq \frac{2b\sqrt{\log \frac{2}{\beta}}}{\sqrt{n}\epsilon}.$$

Bernstein polynomials and approximation We give here some basic definitions that will be used in the sequel; more details can be found in (Aldà and Rubinstein, 2017; Lorentz, 1986; Micchelli, 1973).

Definition 4 *Let k be a positive integer. The Bernstein basis polynomials of degree k are defined as $b_{v,k}(x) = \binom{k}{v} x^v (1-x)^{k-v}$ for $v = 0, \cdots, k$.*

Definition 5 *Let $f : [0,1] \mapsto \mathbb{R}$ and k be a positive integer. Then, the Bernstein polynomial of f of degree k is defined as $B_k(f;x) = \sum_{v=0}^{k} f(v/k) b_{v,k}(x)$. We denote by B_k the Bernstein operator $B_k(f)(x) = B_k(f,x)$.*

Bernstein polynomials can be used to approximate some smooth functions over $[0,1]$.

Definition 6 ((Micchelli, 1973)) *Let h be a positive integer. The iterated Bernstein operator of order h is defined as the sequence of linear operators $B_k^{(h)} = I - (I - B_k)^h = \sum_{i=1}^{h} \binom{h}{i} (-1)^{i-1} B_k^i$, where $I = B_k^0$ denotes the identity operator and B_k^i is defined as $B_k^i = B_k \circ B_k^{k-1}$. The iterated Bernstein polynomial of order h can be computed as $B_k^{(h)}(f;x) = \sum_{v=0}^{k} f(\frac{v}{k}) b_{v,k}^{(h)}(x)$, where $b_{v,k}^{(h)}(x) = \sum_{i=1}^{h} \binom{h}{i} (-1)^{i-1} B_k^{i-1}(b_{v,k};x)$.*

Iterated Bernstein operator can well-approximate multivariate (h,T)-smooth functions.

Definition 7 ((Micchelli, 1973)) *Let h be a positive integer and $T > 0$ be a constant. A function $f : [0,1]^p \mapsto \mathbb{R}$ is (h,T)-smooth if it is in class $\mathcal{C}^h([0,1]^p)$ and its partial derivatives up to order h are all bounded by T. We say it is (∞,T)-smooth, if for every $h \in \mathbb{N}$ it is (h,T)-smooth.*[2]

Note that (h,T)-smoothness is incomparable with the Lipschitz smoothness. In (h,T)-smoothness, we assume it is smooth up to the h-th order while Lipschitz smooth is only for the first order, from this view, (h,T)-smoothness is stronger than the Lipschitz smoothness. However, in Lipschitz smoothness we assume the gradient norm of the function will be bounded by some constant while (h,T)-smoothness assumes that each partial derivative (or each coordinate of the gradient) is bounded by some constant, so from this view Lipschitz smoothness is stronger than (h,T)-smoothness.

Lemma 8 ((Micchelli, 1973)) *If $f : [0,1] \mapsto \mathbb{R}$ is a $(2h,T)$-smooth function, then for all positive integers k and $y \in [0,1]$, we have $|f(y) - B_k^{(h)}(f;y)| \leq T D_h k^{-h}$, where D_h is a constant independent of k, f and y.*

The above lemma is for univariate functions, which has been extended to multivariate functions in Aldà and Rubinstein (2017).

Definition 9 *Assume $f : [0,1]^p \mapsto \mathbb{R}$ and let $k_1, \cdots, k_p, h$ be positive integers. The multivariate iterated Bernstein polynomial of order h at $y = (y_1, \ldots, y_p)$ is defined as:*

$$B_{k_1,\ldots,k_p}^{(h)}(f;y) = \sum_{j=1}^{p} \sum_{v_j=0}^{k_j} f(\frac{v_1}{k_1}, \ldots, \frac{v_p}{k_p}) \prod_{i=1}^{p} b_{v_i,k_i}^{(h)}(y_i). \tag{2}$$

We denote $B_k^{(h)} = B_{k_1,\ldots,k_p}^{(h)}(f;y)$ if $k = k_1 = \cdots = k_p$.

Lemma 10 ((Aldà and Rubinstein, 2017)) *If $f : [0,1]^p \mapsto \mathbb{R}$ is a $(2h,T)$-smooth function, then for all positive integers k and $y \in [0,1]^p$, we have*

$$|f(y) - B_k^{(h)}(f;y)| \leq O(pTD_h k^{-h}).$$

Where D_h is a universal constant only related to h.

[2] $\mathcal{C}^h([0,1]^p)$ means the class of functions that is h-th order smooth in the interval $[0,1]^p$.

In the following, we will rephrase some basic definitions and lemmas on Chebyshev polynomial approximation.

Definition 11 *The Chebyshev polynomials $\{\mathcal{T}(x)_n\}_{n\geq 0}$ are recursively defined as follows*

$$\mathcal{T}_0(x) \equiv 1, \mathcal{T}_1(x) \equiv x \text{ and } \mathcal{T}_{n+1}(x) = 2x\mathcal{T}_n(x) - \mathcal{T}_{n-1}(x).$$

It satisfies that for any $n \geq 0$

$$\mathcal{T}_n(x) = \begin{cases} \cos(n \arccos(x)), \text{ if } |x| \leq 1 \\ \cosh(n arccosh(x)), \text{ if } x \geq 1 \\ (-1)^n \cosh(n arccosh(-x)), \text{ if } x \leq -1 \end{cases}$$

Definition 12 *For every $\rho > 0$, let Γ_ρ be the ellipse Γ of foci ± 1 with major radius $1 + \rho$.*

Definition 13 *For a function f with a domain containing in $[-1, 1]$, its degree-n Chebyshev truncated series is denoted by $P_n(x) = \sum_{k=0}^{n} a_k \mathcal{T}_k(x)$, where the coefficient $a_k = \frac{2-1[k=0]}{\pi} \int_{-1}^{1} \frac{f(x)\mathcal{T}_k(x)}{\sqrt{1-x^2}} dx$.*

Lemma 14 (Cheybeshev Approximation Theorem (Trefethen, 2013)) *Let $f(z)$ be a function that is analytic on Γ_ρ and has $|f(z)| \leq M$ on Γ_ρ. Let $P_n(x)$ be the degree-n Chebyshev truncated series of $f(x)$ on $[-1, 1]$. Then, we have*

$$\max_{x\in[-1,1]} |f(x) - P_n(x)| \leq \frac{2M}{\rho + \sqrt{2\rho + \rho^2}} (1 + \rho + \sqrt{2\rho + \rho^2})^{-n},$$

$|a_0| \leq M$, and $|a_k| \leq 2M(1 + \rho + \sqrt{2\rho + \rho^2})^{-k}$.

The following theorem shows the convergence rate of the Stochastic Inexact Gradient Method (Dvurechensky and Gasnikov, 2016), which will be used in our algorithm. We first give the definition of inexact oracle (see Appendix C for the algorithm of SIGM).

Definition 15 *For an objective function f, a (γ, β, σ) stochastic oracle returns a tuple $(F_{\gamma,\beta,\sigma}(w; \xi), G_{\gamma,\beta,\sigma}(w; \xi))$ (ξ means the randomness in the algorithm) such that*

$$\begin{aligned}
&\mathbb{E}_\xi[F_{\gamma,\beta,\sigma}(w; \xi)] = f_{\gamma,\beta,\sigma}(w), \\
&\mathbb{E}_\xi[G_{\gamma,\beta,\sigma}(w; \xi)] = g_{\gamma,\beta,\sigma}(w), \\
&\mathbb{E}_\xi[\|G_{\gamma,\beta,\sigma}(w; \xi) - g_{\gamma,\beta,\sigma}(w)\|_2^2] \leq \sigma^2, \\
&0 \leq f(v) - f_{\gamma,\beta,\sigma}(w) - \langle g_{\gamma,\beta,\sigma}(w), v - w\rangle \leq \frac{\beta}{2}\|v - w\|^2 + \gamma, \forall v, w \in \mathcal{C}.
\end{aligned}$$

Lemma 16 ((Dvurechensky and Gasnikov, 2016)) *Assume that $f(w)$ is endowed with a (γ, β, σ) stochastic oracle with $\beta \geq O(1)$. Then, the sequence w_k generated by SIGM algorithm satisfies the following inequality*

$$\mathbb{E}[f(w_k)] - \min_{w\in\mathcal{C}} f(w) \leq \Theta(\frac{\beta\sigma\|\mathcal{C}\|_2^2}{\sqrt{k}} + \gamma).$$

4. LDP-ERM with Smooth Loss Functions

In this section, we will mainly focus on reducing the sample complexity of $\frac{1}{\alpha}$. We first show that if the loss function is ∞-smooth (with some additional assumptions), then its sample complexity can be reduced to only polynomial in $\frac{1}{\alpha}$ instead of exponential dependency in the previous paper. Then we talk about how to reduce the communication and computation cost for each user and also provide an algorithm which can let the server solve the problem more efficient.

In this section, we impose the following assumptions on the loss function.

Assumption 1: We let x denote (x, y) for simplicity unless specified otherwise. We assume that there is a constraint set $\mathcal{C} \subseteq [0,1]^p$ and for every $x \in \mathcal{D}$ and $w \in \mathcal{C}$, $\ell(\cdot; x)$ is well defined on $[0,1]^p$ and $\ell(w; x) \in [0,1]$. These closed intervals can be extended to arbitrarily bounded closed intervals.

Note that our assumptions are similar to the 'Typical Settings' in (Smith et al., 2017), where $\mathcal{C} \subseteq [0,1]^p$ appears in their Theorem 10, and $\ell(w; x) \in [0,1]$ from their 1-Lipschitz requirement and $\|\mathcal{C}\|_2 \leq 1$. We note that the above assumptions on x_i, y_i and $\mathcal{C}$ are quite common for the studies of LDP-ERM (Smith et al., 2017; Zheng et al., 2017).

4.1 Basic Idea

Definition 9 and Lemma 10 tell us that if the value of the empirical risk function, *i.e.* the average of the sum of loss functions, is known at each of the grid points $(\frac{v_1}{k}, \frac{v_2}{k} \cdots \frac{v_p}{k})$, where $(v_1, \cdots, v_p) \in \mathcal{T} = \{0, 1, \cdots, k\}^p$ for some large k, then the function can be well approximated. Our main observation is that this can be done in the local model by estimating the average of the sum of loss functions at each of the grid points using Algorithm 1. This is the idea of Algorithm 2.

Algorithm 2 Local Bernstein Mechanism

```
Input: Player $i \in [n]$ holds a data record $x_i \in \mathcal{D}$, public loss function $\ell : [0,1]^p \times \mathcal{D} \mapsto [0,1]$, privacy parameter $\epsilon > 0$, and parameter $k$.
Construct the grid $\mathcal{T} = \{\frac{v_1}{k}, \dots, \frac{v_p}{k}\}_{\{v_1,\dots,v_p\}}$, where $\{v_1, \dots, v_p\} \in \{0, 1, \cdots, k\}^p$.
for Each grid point $v = (\frac{v_1}{k}, \dots, \frac{v_p}{k}) \in \mathcal{T}$ do
    for Each Player $i \in [n]$ do
        Calculate $\ell(v; x_i)$.
    end for
    Run Algorithm 1 with $\epsilon = \frac{\epsilon}{(k+1)^p}$ and $b = 1$ and denote the output as $\tilde{L}(v; D)$.
end for
for The Server do
    Construct Bernstein polynomial, as in (2), based on the perturbed empirical loss function values $\{\tilde{L}(v; D)\}_{v \in \mathcal{T}}$. Denote $\tilde{L}(\cdot; D)$ the corresponding function.
    Compute $w_{\text{priv}} = \arg\min_{w \in \mathcal{C}} \tilde{L}(w; D)$.
end for
```

Theorem 17 *For any $\epsilon > 0$ and $0 < \beta < 1$, Algorithm 2 is ϵ-LDP.*[3] *Assume that the loss function $\ell(\cdot; x)$ is $(2h, T)$-smooth for all $x \in \mathcal{D}$, some positive integer h and constant $T = O(1)$. If the sample complexity n satisfies the condition of $n = O\Big(\frac{\log\frac{1}{\beta}4^{p(h+1)}}{\epsilon^2 D_h^2}\Big)$, then by setting $k = O\Big((\frac{D_h\sqrt{pn}\epsilon}{2^{(h+1)p}\sqrt{\log\frac{1}{\beta}}})^{\frac{1}{h+p}}\Big)$, with probability at least $1-\beta$ we have:*

$$Err_D(w_{priv}) \leq \tilde{O}\Big(\frac{\log^{\frac{h}{2(h+p)}}(\frac{1}{\beta})D_h^{\frac{p}{p+h}}p^{\frac{p}{2(h+p)}}2^{(h+1)p\frac{h}{h+p}}}{n^{\frac{h}{2(h+p)}}\epsilon^{\frac{h}{h+p}}}\Big), \tag{3}$$

where $\tilde{O}$ hides the log *and T terms.*

Proof The proof of the ϵ-LDP comes from Lemma 3 and the basic composition theorem of differential privacy. Without loss of generality, we assume that T=1.

To prove the theorem, it is sufficient to estimate $\sup_{w\in\mathcal{C}}|\tilde{L}(w; D) - L(w; D)| \leq \alpha$ for some α. Since if it is true, denoting $w^* = \arg\min_{w\in\mathcal{C}} L(w; D)$, we have $L(w_{\text{priv}}; D) - L(w^*; D) \leq L(w_{\text{priv}}; D) - \tilde{L}(w_{\text{priv}}; D) + \tilde{L}(w_{\text{priv}}; D) - \tilde{L}(w^*; D) + \tilde{L}(w^*; D) - L(w^*; D) \leq L(w_{\text{priv}}; D) - \tilde{L}(w_{\text{priv}}; D) + \tilde{L}(w^*; D) - L(w^*; D) \leq 2\alpha$.

Since we have

$$\sup_{w\in\mathcal{C}}|\tilde{L}(w; D) - L(w; D)| \leq \sup_{w\in\mathcal{C}}|\tilde{L}(w; D) - B_k^{(h)}(\hat{L}, w)| + \sup_{w\in\mathcal{C}}|B_k^{(h)}(\hat{L}, w) - L(w; D)|.$$

The second term is bounded by $O(D_h p\frac{1}{k^h})$ by Lemma 10.

For the first term, by (2) and Algorithm 2, we have

$$\sup_{w\in\mathcal{C}}|\tilde{L}(w; D) - B_k^{(h)}(\hat{L}, w)| \leq \max_{v\in\mathcal{T}}|\tilde{L}(v; D) - \hat{L}(v; D)| \sup_{w\in\mathcal{C}}\sum_{j=1}^{p}\sum_{v_j=0}^{k}|\prod_{i=1}^{p}b_{v_i,k}^{(h)}(w_i)|. \tag{4}$$

By Proposition 4 in (Aldà and Rubinstein, 2017), we have

$$\sum_{j=1}^{p}\sum_{v_j=0}^{k}|\prod_{i=1}^{p}b_{v_i,k}^{(h)}(w_i)| \leq (2^h - 1)^p.$$

The following lemma bounds the term of $\max_{v\in\mathcal{T}}|\tilde{L}(v; D) - L(v; D)|$, which is obtained by Lemma 3.

Lemma 18 *If $0 < \beta < 1, k$ and n satisfy the condition of $n \geq p\log(2/\beta)\log(k+1)$, then with probability at least $1-\beta$, for each $v \in \mathcal{T}$, the following holds*

$$|\tilde{L}(v; D) - L(v; D)| \leq O(\frac{\sqrt{\log\frac{1}{\beta}}\sqrt{p}\sqrt{\log(k)}(k+1)^p}{\sqrt{n}\epsilon}).$$

[3] Note that we can use Advanced Composition Theorem in (Dwork et al., 2014) to reduce the noise. For simplicity, we omit it here; the following algorithms are also the same.

Proof [Proof of Lemma 18] By Lemma 3, for a fixed $v \in \mathcal{T}$, if $n \geq \log \frac{2}{\beta}$, we have, with probability $1-\beta$, $|\tilde{L}(v;D) - L(v;D)| \leq \frac{2\sqrt{\log \frac{2}{\beta}}}{\sqrt{n}\epsilon}$. Taking the union of all $v \in \mathcal{T}$ and then taking $\beta = \frac{\beta}{(k+1)^p}$ (since there are $(k+1)^p$ elements in $\mathcal{T}$) and $\epsilon = \frac{\epsilon}{(k+1)^p}$, we get the proof. ■

By the fact that $(k+1) < 2k$, we have in total

$$\sup_{w\in\mathcal{C}} |\tilde{L}(w;D) - L(w;D)| \leq O(\frac{D_h p}{k^h} + \frac{2^{(h+1)p}\sqrt{\log\frac{1}{\beta}}\sqrt{p\log k}k^p}{\sqrt{n}\epsilon}). \tag{5}$$

Now, we take $k = O(\frac{D_h\sqrt{pn}\epsilon}{2^{(h+1)p}\sqrt{\log\frac{1}{\beta}}})^{\frac{1}{h+p}}$. Since $n = O(\frac{4^{p(h+1)}}{\epsilon^2 p D_h^2})$, we have $\log k > 1$. Plugging it into (5), we get

$$\begin{aligned}\sup_{w\in\mathcal{C}} |\tilde{L}(w;D) - L(w;D)| &\leq \tilde{O}(\frac{\log^{\frac{h}{2(h+p)}}(\frac{1}{\beta}) D_h^{\frac{p}{p+h}} p^{\frac{1}{2}+\frac{p}{2(h+p)}} 2^{(h+1)p\frac{h}{h+p}}}{\sqrt{h+p}n^{\frac{h}{2(h+p)}}\epsilon^{\frac{h}{h+p}}}) \\ &= \tilde{O}(\frac{\log^{\frac{h}{2(h+p)}}(\frac{1}{\beta}) D_h^{\frac{p}{p+h}} p^{\frac{p}{2(h+p)}} 2^{(h+1)p}}{n^{\frac{h}{2(h+p)}}\epsilon^{\frac{h}{h+p}}}).\end{aligned}$$

Also, we can see that $n \geq p\log(2/\beta)\log(k+1)$ is true for $n = O(\frac{4^{p(h+1)}}{\epsilon^2 p D_h^2})$. Thus, the theorem follows. ■

From (3) we can see that in order to achieve error α, the sample complexity needs to be

$$n = \tilde{O}(\log\frac{1}{\beta} D_h^{\frac{2p}{h}} p^{\frac{p}{h}} 4^{(h+1)p}\epsilon^{-2}\alpha^{-(2+\frac{2p}{h})}). \tag{6}$$

This implies the following special cases.

Corollary 19 *If the loss function $\ell(\cdot;x)$ is $(8,T)$-smooth for all $x \in \mathcal{D}$ and some constant T, and n, ϵ, β, k satisfy the condition in Theorem 17 with $h = 4$, then with probability at least $1-\beta$, the sample complexity to achieve α error is*

$$n = \tilde{O}\big(\alpha^{-(2+\frac{p}{2})}\epsilon^{-2}(4^5\sqrt{D_4}p^{\frac{1}{4}})^p\big).$$

Note that the sample complexity for general convex loss functions in (Smith et al., 2017) is $n = \tilde{O}\big(\alpha^{-(p+1)}\epsilon^{-2}2^p\big)$, which is considerably worse than ours when $\alpha \leq O(\frac{1}{p})$, that is either in the low dimensional case or with high accuracy.

Corollary 20 *If the loss function $\ell(\cdot;x)$ is (∞,T)-smooth for all $x \in \mathcal{D}$ and some constant T, and n, ϵ, β, k satisfy the condition in Theorem 17 with $h = p$, then with probability at least $1-\beta$, the output w_{priv} of Algorithm 2 satisfies:*

$$Err_D(w_{priv}) \leq \tilde{O}\Big(\frac{\log\frac{1}{\beta}^{\frac{1}{4}} D_p^{\frac{1}{2}} p^{\frac{1}{4}} \sqrt{2}^{(p+1)p}}{n^{\frac{1}{4}}\epsilon^{\frac{1}{2}}}\Big),$$

where $\tilde{O}$ hides the $\log$ *and T terms. Thus, to achieve error α, with probability at least $1-\beta$, the sample complexity needs to be*

$$n = \tilde{O}\Big(\max\{4^{p(p+1)}\log(\frac{1}{\beta})D_p^2 p\epsilon^{-2}\alpha^{-4}, \frac{\log\frac{1}{\beta}4^{p(p+1)}}{\epsilon^2 D_p^2}\}\Big). \tag{7}$$

It is worth noticing that from (6) we can see that when the term $\frac{h}{p}$ grows, the term α decreases. Thus, for loss functions that are (∞, T)-smooth, we can get a smaller dependency than the term α^{-4} in (7). For example, if we take $h = 2p$, then the sample complexity is $n = O(\max\{c_2^{p^2}\log\frac{1}{\beta}D_{2p}\sqrt{p}\epsilon^{-2}\alpha^{-3}, \frac{\log\frac{1}{\beta}c^{p^2}}{\epsilon^2 D_{2p}^2}\})$ for some constants c, c_2. When $h \to \infty$, the dependency on the error becomes α^{-2}, which is the optimal bound, even for convex functions.

Our analysis on the empirical excess risk does not use the convexity assumption. While this gives a bound which is not optimal, even for $p = 1$, it also says that our result holds for non-convex loss functions and constrained domain set, as long as they are smooth enough.

From (7), we can see that our sample complexity is lower than the one in (Smith et al., 2017) when $\alpha \leq O(\frac{1}{16^p})$. It is notable that this bound is less reasonable since in practice could be very large. However, there are still many cases where the condition still holds. For example, in low dimensional space to achieve the best performance for ERM, quite often the error is set to be extremely small, *e.g.,* $\alpha = 10^{-10} \sim 10^{-14}$(Johnson and Zhang, 2013).

Using the convexity assumption of the loss function, we can also give a bound on the population excess risk. Here we will show only the case of (∞, T), as the general case is basically the same.

Theorem 21 *Under the conditions in Corollary 20, if we further assume that the loss function $\ell(\cdot; x)$ is convex and* 1*-Lipschitz for all $x \in \mathcal{D}$, then with probability at least $1-2\beta$, we have:*

$$Err_{\mathcal{P}}(w_{priv}) \leq \tilde{O}\Big(\frac{(\sqrt{\log 1/\beta})^{\frac{1}{4}} D_p^{\frac{1}{4}} p^{\frac{1}{8}} \sqrt[4]{2}^{p(p+1)}}{\beta n^{\frac{1}{12}}\epsilon^{\frac{1}{4}}}\Big).$$

That is, if we have sample complexity

$$n = \tilde{O}\big(\max\{\frac{\log\frac{1}{\beta}4^{p(p+1)}}{\epsilon^2 D_p^2}, (\sqrt{\log 1/\beta})^3 D_p^3 p^{\frac{3}{2}} 8^{p(p+1)}\epsilon^{-3}\alpha^{-12}\beta^{-12}\big),$$

then $Err_{\mathcal{P}}(w_{priv}) \leq \alpha$.

Corollary 20 provides a partial answer to our motivational questions. That is, for loss functions which are (∞, T)-smooth, there is an ϵ-LDP algorithm for the empirical and population excess risks achieving error α with sample complexity which is independent of the dimensionality p in the term of α. This result does not contradict the results in (Smith et al., 2017). Indeed, the example used to show the unavoidable dependency between the sample complexity and $\alpha^{-\Omega(p)}$, to achieve an α error, is actually non-smooth.

4.2 More Efficient Algorithms

Algorithm 2 has computational time and communication complexity for each player which are exponential in the dimensionality. This is clearly problematic for every realistic practical application. For this reason, in this section, we investigate more efficient algorithms. For convenience, in this section we focus only on the case of (∞, T)-smooth loss functions, but our results can easily be extended to more general cases.

We first consider the computational issue on the users side. The following lemma, shows an ϵ-LDP algorithm (which is different from Algorithm 1) for efficiently computing p-dimensional average (notice the extra conditions on n and p compared with Lemma 3).

Lemma 22 ((Nissim and Stemmer, 2018)) *Consider player $i \in [n]$ holding data $v_i \in \mathbb{R}^p$ with coordinate between 0 and b. Then for $0 < \beta < 1$, $0 < \epsilon$ such that $n \geq 8p\log(\frac{8p}{\beta})$ and $\sqrt{n} \geq \frac{12}{\epsilon}\sqrt{\log\frac{32}{\beta}}$, there is an ϵ-LDP algorithm, LDP-AVG, with probability at least $1-\beta$, the output $a \in \mathbb{R}^p$ satisfying[4]:*

$$\max_{j\in[d]} |a_j - \frac{1}{n}\sum_{i=1}^{n}[v_i]_j| \leq O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}}).$$

Moreover, the computational cost for each user is $O(1)$.

By using Lemma 22 and by discretizing the grid with some interval steps, we can design an algorithm which requires $O(1)$ computation time and $O(\log n)$-bits communication per player (see (Nissim and Stemmer, 2018) for details; in Appendix B we have an algorithm with $O(\log\log n)$-bits communication per player). However, we would like to do even better and obtain constant communication complexity.

Instead of discretizing the grid, we apply a technique, proposed first by Bassily and Smith (2015), which permits us to transform any 'sampling resilient' ϵ-LDP protocol into a protocol with 1-bit communication complexity (at the expense of increasing the shared randomness in the protocol). Roughly speaking, a protocol is sampling resilient if its output on any dataset S can be approximated well by its output on a random subset of half of the players.

Since our algorithm only uses the LDP-AVG protocol, we can show that it is indeed sampling resilient. Inspired by this result and the algorithm behind Lemma 22, we propose Algorithm 3 and obtain the following theorem.

Theorem 23 *For any $0 < \epsilon \leq \ln 2$ and $0 < \beta < 1$, Algorithm 3 is ϵ-LDP. If the loss function $\ell(\cdot; x)$ is (∞, T)-smooth for all $x \in \mathcal{D}$ and $n = \tilde{O}(\max\{\frac{\log\frac{1}{\beta}4^{p(p+1)}}{\epsilon^2 D_p^2}, p(k+1)^p\log(k+1), \frac{1}{\epsilon^2}\log\frac{1}{\beta}\})$, then by setting $k = O\big((\frac{D_p\sqrt{pn}\epsilon}{2^{(p+1)p}\sqrt{\log\frac{1}{\beta}}})^{\frac{1}{2p}}\big)$, the results in Corollary 20 hold with probability at least $1-4\beta$. Moreover, for each player the time complexity is $O(1)$, and the communication complexity is 1-bit.*

[4] Note that here we use an weak version of their result, one can get a finer analysis. For simplicity, we will omit it in the paper.

Algorithm 3 Player-Efficient Local Bernstein Mechanism with 1-bit communication per player

```
Input: Player $i \in [n]$ holds a data record $x_i \in \mathcal{D}$, public loss function $\ell : [0,1]^p \times \mathcal{D} \mapsto [0,1]$, privacy parameter $\epsilon \leq \ln 2$, and parameter $k$.
Preprocessing:
Generate $n$ independent public strings
$y_1 = \text{Lap}(\frac{1}{\epsilon}), \cdots, y_n = \text{Lap}(\frac{1}{\epsilon})$.
Construct the grid $\mathcal{T} = \{\frac{v_1}{k}, \ldots, \frac{v_p}{k}\}_{\{v_1,\ldots,v_p\}}$, where $\{v_1, \ldots, v_p\} \in \{0, 1, \cdots, k\}^p$.
Partition randomly $[n]$ into $d = (k+1)^p$ subsets $I_1, I_2, \cdots, I_d$, and associate each $I_j$ to a grid point $\mathcal{T}(j) \in \mathcal{T}$.
for Each Player $i \in [n]$ do
    Find $I_l$ such that $i \in I_l$. Calculate $v_i = \ell(\mathcal{T}(l); x_i)$.
    Compute $p_i = \frac{1}{2}\frac{\Pr[v_i+\text{Lap}(\frac{1}{\epsilon})=y_i]}{\Pr[\text{Lap}(\frac{1}{\epsilon})=y_i]}$
    Sample a bit $b_i$ from Bernoulli$(p_i)$ and send it to the server.
end for
for The Server do
    for $i = 1 \cdots n$ do
        Check if $b_i = 1$, set $\tilde{z}_i = y_i$, otherwise $\tilde{z}_i = 0$.
    end for
    for each $l \in [d]$ do
        Compute $v_\ell = \frac{n}{|I_l|}\sum_{i \in I_\ell} \tilde{z}_i$
        Denote the corresponding grid point $(\frac{v_1}{k}, \ldots, \frac{v_p}{k}) \in \mathcal{T}$ of $I_l$, then denote $\hat{L}((\frac{v_1}{k}, \cdots, \frac{v_p}{k}); D) = v_l$.
    end for
    Construct Bernstein polynomial for the perturbed empirical loss $\{\hat{L}(v; D)\}_{v \in \mathcal{T}}$ as in Algorithm 2. Denote $\tilde{L}(\cdot; D)$ the corresponding function.
    Compute $w_{\text{priv}} = \arg\min_{w \in \mathcal{C}} \tilde{L}(w; D)$.
end for
```

Now we study the algorithm from the server's computational complexity perspective. The polynomial construction time complexity is $O(n)$, where the most inefficient part is finding $w_{\text{priv}} = \arg\min_{w\in\mathcal{C}} \tilde{L}(w; D)$. In fact, this function may be non-convex; but unlike general non-convex functions, it can be α-uniformly approximated by the empirical loss function $L(\cdot; D)$ if the loss function is convex (by the proof of Theorem 17), although we do not have access to the empirical risk function. Thus, we can see this problem as an instance of Approximately-Convex Optimization, which has been studied recently by (Risteski and Li, 2016). Before doing that, we first give the definition of the condition on the constraint set.

Definition 24 ((Risteski and Li, 2016)) *We say that a convex set $\mathcal{C}$ is μ-well conditioned for $\mu \geq 1$, if there exists a function $F : \mathbb{R}^p \mapsto \mathbb{R}$ such that $\mathcal{C} = \{x | F(x) \leq 0\}$ and for every $x \in \partial K$: $\frac{\|\nabla^2 F(x)\|_2}{\|\nabla F(x)\|_2} \leq \mu$.*

Lemma 25 (Theorem 3.2 in (Risteski and Li, 2016)) *Let ϵ, Δ be two real numbers such that $\Delta \leq \max\{\frac{\epsilon^2}{\mu\sqrt{p}}, \frac{\epsilon}{p}\} \times \frac{1}{16348}$. Then, there exists an algorithm $\mathcal{A}$ such that for any given Δ-approximate convex function $\tilde{f}$ over a μ-well-conditioned convex set $\mathcal{C} \subseteq \mathbb{R}^p$ of diameter 1 (that is, there exists a 1-Lipschitz convex function $f : \mathcal{C} \mapsto \mathbb{R}$ such that for every $x \in \mathcal{C}, |f(x) - \tilde{f}(x)| \leq \Delta$), $\mathcal{A}$ returns a point $\tilde{x} \in \mathcal{C}$ with probability at least $1 - \delta$ in time Poly($p, \frac{1}{\epsilon}, \log\frac{1}{\delta}$) and with the following guarantee: $\tilde{f}(\tilde{x}) \leq \min_{x\in\mathcal{C}} \tilde{f}(x) + \epsilon$.*

Based on Lemma 25 (for $\tilde{L}(w; D)$) and Corollary 20, and taking $\epsilon = O(p\alpha)$, we have the following.

Theorem 26 *Under the conditions in Corollary 20, and assuming that n satisfies $n = \tilde{O}(4^{p(p+1)} \log(1/\beta) D_p^2 p\epsilon^{-2}\alpha^{-4})$, that the loss function $\ell(\cdot; x)$ is 1-Lipschitz and convex for every $x \in \mathcal{D}$, that the constraint set $\mathcal{C}$ is convex and $\|\mathcal{C}\|_2 \leq 1$, and satisfies μ-well-condition property (see Definition 24), if the error α satisfies $\alpha \leq C\frac{\mu}{p\sqrt{p}}$ for some universal constant C, then there is an algorithm $\mathcal{A}$ which runs in Poly($n, \frac{1}{\alpha}, \log\frac{1}{\beta}$) time for the server,[5] and with probability $1-2\beta$ the output $\tilde{w}_{priv}$ of $\mathcal{A}$ satisfies $\tilde{L}(\tilde{w}_{priv}; D) \leq \min_{w\in\mathcal{C}} \tilde{L}(w; D) + O(p\alpha)$, which means that Err$_D(\tilde{w}_{priv}) \leq O(p\alpha)$.*

Combining Theorem 26 with Corollary 20, and taking $\alpha = \frac{\alpha}{p}$, we have our final result:

Theorem 27 *Under the conditions of Corollary 20, Theorem 23 and 26, for any $C\frac{\mu}{\sqrt{p}} > \alpha > 0$, if we further set*

$$n = \tilde{O}(4^{p(p+1)} \log(1/\beta) D_p^2 p^5 \epsilon^{-2}\alpha^{-4}),$$

then there is an ϵ-LDP algorithm, with $O(1)$ running time and 1-bit communication per player, and Poly($\frac{1}{\alpha}, \log\frac{1}{\beta}$) running time for the server. Furthermore, with probability at least $1 - 5\beta$, the output $\tilde{w}_{priv}$ satisfies Err$_D(\tilde{w}_{priv}) \leq O(\alpha)$.

Note that comparing with the sample complexity in Theorem 27 and Corollary 20, we have an additional factor of $O(p^4)$; however, the α terms are the same.

[5] Note that since here we assume n is at least exponential in p, thus the algorithm is not fully polynomial.

5. LDP-ERM with Convex Generalized Linear Loss Functions

In Section 4, we have seen that under the condition of (∞, T)-smoothness for the loss function, the sample complexity can actually have polynomial dependence on p and α. However, as shown in (7), there is still another exponential term c^{p^2} in the sample complexity that needs to be removed.

In this section, we show that if the loss function is generalized linear, the sample complexity for achieving error α is only linear in the dimensionality p. We first give the assumptions that will be used throughout this section.

Assumption 2: We assume that $\|x_i\|_2 \leq 1$ and $|y_i| \leq 1$ for each $i \in [n]$ and the constraint set $\|\mathcal{C}\|_2 \leq 1$. Unless specified otherwise, the loss function is assumed to be generalized linear, that is, the loss function $\ell(w; x_i, y_i) \equiv f(y_i\langle x_i, w\rangle)$ for some 1-Lipschitz convex function f.

The generalized linear assumption holds for a large class of functions such as Generalized Linear Model and SVM. We also note that there is another definition for general linear functions, $\ell(w; x, y) = f(\langle w, x\rangle, y)$, which is more general than our definition. This class of functions has been studied in (Kasiviswanathan and Jin, 2016); we leave as future research to extend our work to this class of loss functions.

5.1 Sample Complexity for Hinge Loss Function

We first consider LDP-ERM with hinge loss function and then extend the obtained result to general convex linear functions.

The hinge loss function is defined as $\ell(w; x_i, y_i) = f(y_i\langle x_i, w\rangle) = [\frac{1}{2} - y_i\langle w, x_i\rangle]_+$, where the plus function $[x]_+ = \max\{0, x\}$, *i.e.,* $f(x) = \max\{0, \frac{1}{2} - x\}$ for $x \in [-1, 1]$.[6] Note that to avoid the scenario that $1 - y_i\langle w, x_i\rangle$ is always greater than or equal to 0, we use $\frac{1}{2}$, instead of 1 as in the classical setting.

Before showing our idea, we first smooth the function $f(x)$. The following lemma shows one of the smooth functions that is close to f in the domain of $[-1, 1]$ (note that there are other ways to smooth f; see (Chen and Mangasarian, 1996) for details).

Lemma 28 *Let* $f_\beta(x) = \frac{\frac{1}{2} - x + \sqrt{(\frac{1}{2} - x)^2 + \beta^2}}{2}$ *be a function with parameter* $\beta > 0$. *Then, we have*

1. $|f_\beta(x) - f(x)|_\infty \leq \frac{\beta}{2}$, $\forall x \in \mathbb{R}$.

2. $f_\beta(x)$ *is 1-Lipschitz, that is,* $f'(x)$ *is bounded by* 1 *for* $x \in \mathbb{R}$.

3. f_β *is* $\frac{1}{\beta}$*-smooth and convex.*

4. $f'_\beta(x)$ *is* $(2, O(\frac{1}{\beta^2}))$*-smooth if* $\beta \leq 1$.

The above lemma indicates that $f_\beta(x)$ is a smooth and convex function which well approximates $f(x)$. This suggests that we can focus on $f_\beta(y_i\langle w, x_i\rangle)$, instead of f. Our idea is to construct a locally private (γ, β, σ) stochastic oracle for some γ, β, σ to approximate

[6] The reader should think about about particular function f, not just a general f.

$f'_\beta(y_i\langle w, x_i\rangle)$ in each iteration, and then run the SIGM step of (Dvurechensky and Gasnikov, 2016). By Lemma 28, we know that f'_β is $(2, O(\frac{1}{\beta^2}))$-smooth; thus, we can use Lemma 8 to approximate $f'_\beta(x)$ via Bernstein polynomials.

Let $P_d(x) = \sum_{i=0}^d c_i \binom{d}{i} x^i(1-x)^{d-i}$ be the d-th order Bernstein polynomial ($c_i = f'_\beta(\frac{i}{d})$, where $\max_{x\in[-1,1]} |P_d(x) - f'_\beta(x)| \le \frac{\alpha}{4}$ (*i.e.*, $d = c\frac{1}{\beta^2\alpha}$ for some constant $c > 0$). Then, we have $\nabla_w \ell(w; x, y) = f'(y\langle w, x\rangle)yx^T$, which can be approximated by $[\sum_{i=0}^d c_i \binom{d}{i}(y\langle w, x\rangle)^i(1-y\langle w, x\rangle)^{d-i}]yx^T$. The idea is that if $(y\langle w, x\rangle)^i$, $(1 - y\langle w, x\rangle)^{d-i}$ and yx^T can be approximated locally differentially privately by directly adding $d + 1$ numbers of independent Gaussian noises, which means it is possible to form an unbiased estimator of the term $[\sum_{i=0}^d c_i \binom{d}{i}(y\langle w, x\rangle)^i(1 - y\langle w, x\rangle)^{d-i}]yx^T$. The error of this procedure can be estimated by Lemma 16. Details of the algorithm are given in Algorithm 4.

Algorithm 4 Hinge Loss-LDP

```
Input: Player $i \in [n]$ holds data $(x_i, y_i) \in \mathcal{D}$, where $\|x_i\|_2 \le 1, \|y_i\|_2 \le 1$; privacy
  parameters $\epsilon, \delta$; $P_d(x) = \sum_{j=0}^d c_i \binom{d}{j} x^j(1-x)^{d-j}$ be the $d$-th order Bernstein polynomial
  for the function of $f'_\beta$, where $c_i = f'_\beta(\frac{i}{d})$ and $f_\beta(x)$ is the function in Lemma 28.
for Each Player $i \in [n]$ do
    Calculate $x_{i,0} = x_i + \sigma_{i,0}$ and $y_{i,0} = y_i + z_{i,0}$, where $\sigma_{i,0} \sim \mathcal{N}(0, \frac{32\log(1.25/\delta)}{\epsilon^2} I_p)$ and
    $z_{i,0} \sim \mathcal{N}(0, \frac{32\log(1.25/\delta)}{\epsilon^2})$.
    for $j = 1, \cdots, d(d+1)$ do
        $x_{i,j} = x_i + \sigma_{i,j}$, where $\sigma_{i,j} \sim \mathcal{N}(0, \frac{8\log(1.25/\delta)d^2(d+1)^2}{\epsilon^2} I_p)$
        $y_{i,j} = y_i + z_{i,j}$, where $z_{i,j} \sim \mathcal{N}(0, \frac{8\log(1.25/\delta)d^2(d+1)^2}{\epsilon^2})$
    end for
    Send $\{x_{i,j}\}_{j=0}^{d(d+1)}$ and $\{y_{i,j}\}_{j=0}^{d(d+1)}$ to the server.
end for
for the Server side do
    for $t = 1, 2, \cdots, n$ do
        Randomly sample $i \in [n]$ uniformly.
        Set $t_{i,0} = 1$
        for $j = 0, \cdots, d$ do
            $t_{i,j} = \Pi_{k=jd+1}^{jd+j} y_{i,k}\langle w_t, x_{i,k}\rangle$ and $t_{i,0} = 1$
            $s_{i,j} = \Pi_{k=jd+j+1}^{jd+d}(1 - y_{i,k}\langle w_t, x_{i,k}\rangle)$ and $s_{i,d} = 1$
        end for
        Denote $G(w_t, i) = (\sum_{j=0}^d c_j \binom{d}{j} t_{i,j}s_{i,j})y_{i,0}x_{i,0}^T$.
        Update SIGM in (Dvurechensky and Gasnikov, 2016) by $G(w_t, i)$
    end for
end for
return $w_n$
```

Theorem 29 *For each* $i \in [n]$, *the term* $G(w_t, i)$ *generated by Algorithm 4 will be an* $\left(\frac{\alpha}{2}, \frac{1}{\beta}, O(\frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}} + \alpha + 1)\right)$ *stochastic oracle (see Definition 15) for function* $L_\beta(w; D) = \frac{1}{n}\sum_{i=1}^n f_\beta(y_i\langle x_i, w\rangle)$, *where* f_β *is the function in Lemma 28, where* C_4 *is some constant.*

From Lemmas 28, 16 and Theorem 29, we have the following sample complexity bound for the hinge loss function under the non-interactive local model.

Theorem 30 *For any $\epsilon > 0$ and $0 < \delta < 1$, Algorithm 4 is (ϵ, δ) non-interactively locally differentially private.*[7] *Furthermore, for the target error α, if we take $\beta = \frac{\alpha}{4}$ and $d = \frac{2}{\beta^2 \alpha} = O(\frac{1}{\alpha^3})$. Then with the sample size $n = \tilde{O}(\frac{d^{6d} C^d p}{\epsilon^{4d+4} \alpha^2})$, the output w_n satisfies the following inequality*

$$\mathbb{E}L(w_n, D) - \min_{w \in \mathcal{C}} L(w, D) \leq \alpha,$$

where C is some constant.

Remark 31 *Note that the sample complexity bound in Theorem 30 is quite loose for parameters other than p. This is mainly due to the fact that we use only the basic composition theorem to ensure local differential privacy.*[8] *It is possible to obtain a tighter bound by using Advanced Composition Theorem (Dwork et al., 2010) (this is the same for other algorithms in this section). Details of the improvement are omitted from this version. We can also extend to the population risk by the same algorithm, the main difference is that now $G(w, i)$ is a $\big(\frac{\alpha}{2}, \frac{1}{\beta}, O(\frac{d^{3d} C_4^d \sqrt{p}}{\epsilon^{2d+2}} + \alpha + 1)\big)$ stochastic oracle, where $\sigma^2 = \mathbb{E}_{(x,y)\sim\mathcal{P}} \|\ell(w; x, y) - \mathbb{E}_{(x,y)\sim\mathcal{P}} \ell(w; x, y)\|_2^2$. For simplicity of presentation, we omitt the details here.*

5.2 Extension to Generalized Linear Convex Loss Functions

In this section, we extend our results for the hinge loss function to generalized linear convex loss functions $L(w, D) = \frac{1}{n} \sum_{i=1}^{n} f(y_i \langle x_i, w \rangle)$ for any 1-Lipschitz convex function f.

One possible way (for the extension) is to follow the same approach used in previous section. That is, we first smooth the function f by some function f_β. Then, we use Bernstein polynomials to approximate the derivative function f'_β, and apply an algorithm similar to Algorithm 4. One of the main issues of this approach is that we do not know whether Bernstein polynomials can be directly used for every smooth convex function. Instead, we will use some ideas in approximation theory, which says that every 1-Lipschitz convex function can be expressed by a linear combination of the absolute value functions and linear functions.

To implement this approach, we first note that for the plus function $f(x) \equiv \max\{0, x\}$, by using Algorithm 4 we can get the same result as in Theorem 30. Since the absolute value function $|x| = 2\max\{0, x\} - x$, Theorem 30 clearly also holds for the absolute function. The following key lemma shows that every 1-dimensional 1-Lipschitz convex function $f : [-1, 1] \mapsto [-1, 1]$ is contained in the convex hull of the set of absolute value and identity functions. We need to point out that Smith et al. (2017) gave a similar lemma. Their proof is, however, somewhat incomplete and thus we give a complete one in this paper.

[7] Note that in the non-interactive local model, (ϵ, δ)-LDP is equivalent to ϵ-LDP by using the protocol given in Bun et al. (2018); this allows us to omit the term of δ.

[8] There could be some improvement on the term of $\frac{1}{\alpha}$ if we use advanced composition theorem. However, since the dependency of $\frac{1}{\alpha}$ is already exponential, and it will be still exponential after the improvement. So here the improvement will be very incremental.

Lemma 32 *Let $f : [-1,1] \mapsto [-1,1]$ be a 1-Lipschitz convex function. If we define the distribution $\mathcal{Q}$ which is supported on $[-1,1]$ as the output of the following algorithm:*

1. *first sample $u \in [f'(-1), f'(1)]$ uniformly,*
2. *then output s such that $u \in \partial f(s)$ (note that such an s always exists due to the fact that f is convex and thus f' is non-decreasing); if multiple number of such as s exist, return the maximal one,*

then, there exists a constant c such that

$$\forall \theta \in [-1,1], f(\theta) = \frac{f'(1) - f'(-1)}{2}\mathbb{E}_{s\sim\mathcal{Q}}|\theta - s| + \frac{f'(1) + f'(-1)}{2}\theta + c.$$

Proof Let $g(\theta) = \mathbb{E}_{s\sim\mathcal{Q}}|s - \theta|$. Then, we have the following for every θ, where $f'(\theta)$ is well defined,

$$\begin{aligned} g'(\theta) &= \mathbb{E}_{s\sim\mathcal{Q}}[1_{s\leq\theta}] - \mathbb{E}_{s\sim\mathcal{Q}}[1_{s>\theta}] \\ &= \frac{[f'(\theta) - f'(-1)] - [f'(1) - f'(\theta)]}{f'(1) - f(-1)} \\ &= \frac{2f'(\theta) - (f'(1) + f'(-1))}{f'(1) - f'(-1)}. \end{aligned}$$

Thus, we get

$$F'(\theta) = \frac{f'(1) - f'(-1)}{2}g'(\theta) + \frac{f'(1) + f'(-1)}{2} = f'(\theta).$$

Next, we show that if $F'(\theta) = f'(\theta)$ for every $\theta \in [0,1]$, where $f'(\theta)$ is well defined, there is a constant c which satisfies the condition of $F(\theta) = f(\theta) + c$ for all $\theta \in [0,1]$.

Lemma 33 *If f is convex and 1-Lipschitz, then f is differentiable at all but countably many points. That is, f' has only countable many discontinuous points.*

Proof [Proof of Lemma 33] Since f is convex, we have the following for $0 \leq s < u \leq v < t \leq 1$

$$\frac{f(u) - f(s)}{u - s} \leq \frac{f(t) - f(v)}{t - v},$$

This is due to the property of 3-point convexity, where

$$\frac{f(u) - f(s)}{u - s} \leq \frac{f(t) - f(u)}{t - u} \leq \frac{f(t) - f(v)}{t - v}.$$

Thus, we can obtain the following inequality of one-sided derivation, that is,

$$f'_-(x) \leq f'_+(x) \leq f'_-(y) \leq f'_+(y)$$

for every $x < y$. For each point where $f'_-(x) < f'_+(x)$, we pick a rational number $q(x)$ which satisfies the condition of $f'_-(x) < q(x) < f'_+(x)$. From the above discussion, we can see that all these $q(x)$ are different. Thus, there are at most countable many points where f is non-differentiable. ■

From the above lemma, we can see that the Lebesgue measure of these dis-continuous points is 0. Thus, f' is Riemann Integrable on $[-1,1]$. By Newton-Leibniz formula, we have the following for any $\theta \in [0,1]$,

$$\int_{-1}^{\theta} f'(x)dx = f(\theta) - f(-1) = \int_{-1}^{\theta} F'(x)dx = F(x) - F(-1).$$

Therefore, we get $F(\theta) = f(\theta) + c$ and complete the proof. ∎

Algorithm 5 General Linear-LDP

```
Input: Player $i \in [n]$ holds raw data record $(x_i, y_i) \in \mathcal{D}$, where $\|x_i\|_2 \leq 1$ and $\|y_i\|_2 \leq 1$; privacy parameters $\epsilon, \delta$; $h_\beta(x) = \frac{x+\sqrt{x^2+\beta^2}}{2}$ and $P_d(x) = \sum_{j=0}^{d} c_j \binom{d}{j} x^j (1-x)^j$ is the $d$-th order Bernstein polynomial approximation of $h'_\beta(x)$. Loss function $\ell$ can be represented by $\ell(w; x, y) = f(y\langle w, x\rangle)$.
for Each Player $i \in [n]$ do
    Calculate $x_{i,0} = x_i + \sigma_{i,0}$ and $y_{i,0} = y_i + z_{i,0}$, where $\sigma_{i,0} \sim \mathcal{N}(0, \frac{32\log(1.25/\delta)}{\epsilon^2} I_p)$ and $z_{i,0} \sim \mathcal{N}(0, \frac{32\log(1.25/\delta)}{\epsilon^2})$
    for $j = 1, \cdots, d(d+1)$ do
        $x_{i,j} = x_i + \sigma_{i,j}$, where $\sigma_{i,j} \sim \mathcal{N}(0, \frac{8\log(1.25/\delta) d^2 (d+1)^2}{\epsilon^2} I_p)$
        $y_{i,j} = y_i + z_{i,j}$, where $z_{i,j} \sim \mathcal{N}(0, \frac{8\log(1.25/\delta) d^2 (d+1)^2}{\epsilon^2})$
    end for
    Send $\{x_{i,j}\}_{j=0}^{d(d+1)}$ and $\{y_{i,j}\}_{j=0}^{d(d+1)}$ to the server.
end for
for the Server side do
    for $t = 1, 2, \cdots, n$ do
        Randomly sample $i \in [n]$ uniformly.
        Randomly sample $d(d+1)$ numbers of i.i.d $s = \{s_k\}_{k=1}^{d(d+1)} \in [-1,1]$ based on the distribution $\mathcal{Q}$ in Lemma 32.
        Set $t_{i,0} = 1$
        for $j = 0, \cdots, d$ do
            $t_{i,j} = \Pi_{k=jd+1}^{jd+i} (\frac{y_{i,k}\langle w_t, x_{i,k}\rangle - s_k}{2})$ and $t_{i,0} = 1$
            $r_{i,j} = \Pi_{k=jd+i+1}^{jd+d} (1 - \frac{y_{i,k}\langle w_t, x_{i,k}\rangle - s_k}{2})$ and $r_{i,d} = 1$
        end for
        Denote $G(w_t, i, s) = (f'(1) - f'(-1))(\sum_{j=0}^{d} c_j \binom{d}{j} t_{i,j} r_{i,j}) y_{i,0} x_{i,0}^T + f'(-1)$.
        Update SIGM in (Dvurechensky and Gasnikov, 2016) by $G(w_t, i, s)$
    end for
end for
return $w_n$
```

Using Lemma 32 and the ideas discussed in the previous section, we can now show that the sample complexity in Theorem 30 also holds for any general linear convex function. See Algorithm 5 for the details.

Theorem 34 *Under Assumption 2, where the loss function ℓ is $\ell(w; x, y) = f(y\langle w, x\rangle)$ for any 1-Lipschitz convex function f, for any $\epsilon, \delta \in (0, 1]$, Algorithm 5 is (ϵ, δ) non-interactively differentialy private. Moreover, given the target error α, if we take $\beta = \frac{\alpha}{4}$ and $d = \frac{2}{\beta^2\alpha} = O(\frac{1}{\alpha^3})$. Then with the sample size $n = \tilde{O}(\frac{d^{6d}C^d p}{\epsilon^{4d+4}\alpha^2})$, the output w_n satisfies the following inequality*

$$\mathbb{E}L(w_n, D) - \min_{w\in\mathcal{C}} L(w, D) \leq \alpha,$$

where C is some universal constant independent of f.

Remark 35 *The above theorem suggests that the sample complexity for any generalized linear loss function depends only linearly on p. However, there are still some not so desirable issues. Firstly, the dependence on α is exponential, while we have already shown in the Section 4 that it is only polynomial (*i.e.*, α^{-4}) for sufficiently smooth loss functions. Secondly, the term of ϵ is not optimal in the sample complexity, since it is $\epsilon^{-\Omega(\frac{1}{\alpha^3})}$, while the optimal one is ϵ^{-2} (Smith et al., 2017). We leave it as an open problem to remove the exponential dependency. Thirdly, the assumption on the loss function is that $\ell(w; x, y) = f(y\langle w, x\rangle)$, which includes the generalized linear models and SVM. However, as mentioned earlier, there is another slightly more general function class $\ell(w; x, y) = f(\langle w, x\rangle, y)$ which does not always satisfy our assumption,* e.g.*, linear regression and ℓ_1 regression. For linear regression, we have already known its optimal bound $\Theta(p\alpha^{-2}\epsilon^{-2})$; for ℓ_1 regression, we can use a method similar to Algorithm 4 to achieve a sample complexity which is linear in p. Thus, a natural question is whether the sample complexity is still linear in p for all loss functions $\ell(w; x, y)$ that can be written as $f(\langle w, x\rangle, y)$.*

We can see from Algorithm 4 and 5 that, both of the computation and communication cost of each user will be $O(d^2) = O(\frac{1}{\alpha^6})$. So, our question is, can we reduce these costs just as in the Section 4? We will leave it as future research.

Additional to the aforementioned improvements, another advantage of our method is that it can be extended to other LDP problems. Below we show how it can be used to answer the class of k-way marginals and smooth queries under LDP.

6. LDP Algorithms for Learning k-way Marginals Queries and Smooth Queries

In this section, we show further applications of our idea by giving LDP algorithms for answering sets of queries. All the queries considered in this section are linear, that is, of the form $q_f(D) = \frac{1}{|D|}\sum_{x\in D} f(x)$ for some function f. It will be convenient to have a notion of accuracy for the algorithm to be presented with respect to a set of queries. This is defined as follow:

Definition 36 *Let $\mathcal{Q}$ denote a set of queries. An algorithm $\mathcal{A}$ is said to have (α, β)-accuracy for size n databases with respect to $\mathcal{Q}$, if for every n-size dataset D, the following holds: $Pr[\exists q \in \mathcal{Q}, |\mathcal{A}(D, q) - q(D)| \geq \alpha] \leq \beta$.*

6.1 k-way Marginals Queries

Now we consider a database $D = (\{0,1\}^p)^n$, where each row corresponds to an individuals record. A marginal query is specified by a set $S \subseteq [p]$ and a pattern $t \in \{0,1\}^{|S|}$. Each such query asks: 'What fraction of the individuals in D has each of the attributes set to t_j?'. We will consider here k-way marginals which are the subset of marginal queries specified by a set $S \subseteq [p]$ with $|S| \leq k$. K-way marginals could represent several statistics over datasets, including contingency tables, and the problem is to release them under differential privacy has been studied extensively in the literature (Hardt et al., 2012; Gupta et al., 2013; Thaler et al., 2012; Gaboardi et al., 2014). All these previous works have considered the central model of differential privacy, and only the recent work (Kulkarni et al., 2017) studies this problem in the local model, while their methods are based on Fourier Transform. We now use the LDP version of Chebyshev polynomial approximation to give an efficient way of constructing a sanitizer for releasing k-way marginals.

Since learning the class of k-way marginals is equivalent to learning the class of monotone k-way disjunctions (Hardt et al., 2012), we will only focus on the latter. The reason of why we can locally privately learning them is that they form a $\mathcal{Q}$-Function Family.

Definition 37 ($\mathcal{Q}$-Function Family) *Let $\mathcal{Q} = \{q_y\}_{y \in Y_{\mathcal{Q}} \subseteq \{0,1\}^m}$ be a set of counting queries on a data universe $\mathcal{D}$, where each query is indexed by an m-bit string. We define the index set of $\mathcal{Q}$ to be the set $Y_{\mathcal{Q}} = \{y \in \{0,1\}^m | q_y \in \mathcal{Q}\}$. We define a $\mathcal{Q}$-Function Family $\mathcal{F}_{\mathcal{Q}} = \{f_{\mathcal{Q},x} : \{0,1\}^m \mapsto \{0,1\}\}_{x \in \mathcal{D}}$ as follows: for every data record $x \in D$, the function $f_{\mathcal{Q},x} : \{0,1\}^m \mapsto \{0,1\}$ is defined as $f_{\mathcal{Q},x}(y) = q_y(x)$. Given a database $D \in \mathcal{D}^n$, we define $f_{\mathcal{Q},D}(y) = \frac{1}{n}\sum_{i=1}^n f_{\mathcal{Q},x^i}(y) = \frac{1}{n}\sum_{i=1}^n q_y(x^i) = q_y(D)$, where x^i is the i-th row of D.*

This definition guarantees that $\mathcal{Q}$-function queries can be computed from their values on the individual's data x^i. We can now formally define the class of monotone k-way disjunctions.

Definition 38 *Let $\mathcal{D} = \{0,1\}^p$. The query set $\mathcal{Q}_{disj,k} = \{q_y\}_{y \in Y_k \subseteq \{0,1\}^p}$ of monotone k-way disjunctions over $\{0,1\}^p$ contains a query q_y for every $y \in Y_k = \{y \in \{0,1\}^p || y| \leq k\}$. Each query is defined as $q_y(x) = \vee_{j=1}^p y_j x_j$. The $\mathcal{Q}_{disj,k}$-function family $\mathcal{F}_{\mathcal{Q}_{disj,k}} = \{f_x\}_{x \in \{0,1\}^p}$ contains a function $f_x(y_1, y_2, \cdots, y_p) = \vee_{j=1}^p y_j x_j$ for each $x \in \{0,1\}^p$.*

Definition 38 guarantees that if we can uniformly approximate the function $f_{\mathcal{Q},x}$ by polynomials p_x, then we can also have an approximation of $f_{\mathcal{Q},D}$, *i.e.,* we can approximate $q_y(D)$ for every y or all the queries in the class $\mathcal{Q}$. Thus, if we can locally privately estimate the sum of coefficients of the monomials for the m-multivariate functions $\{p_x\}_{x \in D}$, then we can uniformly approximate $f_{\mathcal{Q},D}$. Clearly, this can be done by Lemma 22, if the coefficients of the approximated polynomial are bounded.

In order to uniformly approximate the $\mathcal{Q}_{disj,k}$-function, we use Chebyshev polynomials.

Definition 39 (Chebyshev Polynomials) *For every $k \in \mathbb{N}$ and $\gamma > 0$, there exists a univariate real polynomial $p_k(x) = \sum_{j=0}^{t_k} c_i x^i$ of degree t_k such that $t_k = O(\sqrt{k}\log(\frac{1}{\gamma}))$; for every $i \in [t_k], |c_i| \leq 2^{O(\sqrt{k}\log(\frac{1}{\gamma}))}$; and $p(0) = 0, |p_k(x) - 1| \leq \gamma, \forall x \in [k]$.*

Algorithm 6 Local Chebyshev Mechanism for $\mathcal{Q}_{\text{disj,k}}$

```
Input: Player $i \in [n]$ holds a data record $x_i \in \{0,1\}^p$, privacy parameter $\epsilon > 0$, error bound $\alpha$, and $k \in \mathbb{N}$.
for Each Player $i \in [n]$ do
    Consider the $p$-multivariate polynomial $q_{x_i}(y_1, \ldots, y_p) = p_k(\sum_{j=1}^{p} y_j [x_i]_j)$, where $p_k$ is defined as in Lemma 40 with $\gamma = \frac{\alpha}{2}$.
    Denote the coefficients of $q_{x_i}$ as a vector $\tilde{q}_i \in \mathbb{R}^{\binom{p+t_k}{t_k}}$(since there are $\binom{p+t_k}{t_k}$ coefficients in a $p$-variate polynomial with degree $t_k$), note that each $\tilde{q}_i$ can bee seen as a $p$-multivariate polynomial $q_{x_i}(y)$.
end for
for The Server do
    Run LDP-AVG from Lemma 3 on $\{\tilde{q}_i\}_{i=1}^{n} \in \mathbb{R}^{\binom{p+t_k}{t_k}}$ with parameter $\epsilon$, $b = p^{O(\sqrt{k}\log(\frac{1}{\gamma}))}$, denote the output as $\tilde{q}_D \in \mathbb{R}^{\binom{p+t_k}{t_k}}$, note that $\tilde{q}_D$ also corresponds to a $p$-multivariate polynomial.
    For each query $y$ in $\mathcal{Q}_{\text{disj,k}}$ (seen as a $d$ dimension vector), compute the $p$-multivariate polynomial $\tilde{q}_D(y_1, \ldots, y_p)$.
end for
```

Lemma 40 ((Thaler et al., 2012)) *For every $k, p \in \mathbb{N}$, such that $k \leq p$, and every $\gamma > 0$, there is a family of p-multivariate polynomials of degree $t = O(\sqrt{k}\log(\frac{1}{\gamma}))$with coefficients bounded by $T = p^{O(\sqrt{k}\log(\frac{1}{\gamma}))}$, which uniformly approximate the family $\mathcal{F}_{\mathcal{Q}_{disj,k}}$ over the set Y_k (Definition 38) with error bound γ. That is, there is a family of polynomials $\mathcal{P}$ such that for every $f_x \in \mathcal{F}_{\mathcal{Q}_{disj,k}}$, there is $p_x \in \mathcal{P}$ which satisfies $\sup_{y \in Y_k} |p_x(y) - f_x(y)| \leq \gamma$.*

By combining the ideas discussed above and Lemma 40, we have Algorithm 6 and the following theorem.

Theorem 41 *For $\epsilon > 0$ Algorithm 6 is ϵ-LDP. Also, for $0 < \beta < 1$, there are constants C, C_1 such that for every $k, p, n \in \mathbb{N}$ with $k \leq p$, if*

$$n = O(\max\{\frac{p^{C\sqrt{k}\log\frac{1}{\alpha}}\log\frac{1}{\beta}}{\epsilon^2\alpha^2}, \frac{\log\frac{1}{\beta}}{\epsilon^2}, p^{C_1\sqrt{k}\log\frac{1}{\alpha}}\log\frac{1}{\beta}\}),$$

this algorithm is (α, β)-accurate with respect to $\mathcal{Q}_{disj,k}$. The running time for each player is Poly($p^{O(\sqrt{k}\log\frac{1}{\alpha})}$), and the running time for the server is at most $O(n)$ and the time for answering a query is $O(p^{C_2\sqrt{k}\log\frac{1}{\alpha}})$ for some constant C_2. Moreover, as in Section 4.2, the communication complexity can be improved to 1-bit per player.

6.2 Smooth Queries

We now consider the case where each player $i \in [n]$ holds a data record in the continuous interval $x_i \in [-1, 1]^p$ and we want to estimate the kernel density for a given point $x_0 \in \mathbb{R}^p$. A natural question is: If we want to estimate Gaussian kernel density of a given point x_0 with many different bandwidths, can we do it simultaneously under ϵ local differential privacy?

Algorithm 7 Local Trigonometry Mechanism for $\mathcal{Q}_{C_T^h}$

```
Input: Player $i \in [n]$ holds a data record $x_i \in [-1,1]^p$, privacy parameter $\epsilon > 0$, error bound $\alpha$, and $t \in \mathbb{N}$. $\mathcal{T}_t^p = \{0, 1, \cdots, t-1\}^p$. For a vector $x = (x_1, \dots, x_p) \in [-1,1]^p$, denote operators $\theta_i(x) = \arccos(x_i), i \in [p]$.
for Each Player $i \in [n]$ do
    for Each $v = (v_1, v_2, \cdots, v_p) \in \mathcal{T}_t^p$ do
        Compute $p_{i;v} = \cos(v_1\theta_1(x_i)) \cdots \cos(v_p\theta_p(x_i))$
    end for
    Let $p_i = (p_{i;v})_{v \in \mathcal{T}_t^p}$.
end for
for The Server do
    Run LDP-AVG from Lemma 3 on $\{p_i\}_{i=1}^n \in \mathbb{R}^{t^p}$ with parameter $\epsilon$, $b = 1$, denote the output as $\tilde{p}_D$.
    For each query $q_f \in \mathcal{Q}_{C_T^h}$. Let $g_f(\theta) = f(\cos(\theta_1), \cos(\theta_2), \cdots, \cos(\theta_p))$.
    Compute the trigonometric polynomial approximation $p_t(\theta)$ of $g_f(\theta)$, where $p_t(\theta) = \sum_{r=(r_1, r_2 \cdots r_p), \|r\|_\infty \leq t-1} c_r \cos(r_1\theta_1) \cdots \cos(r_p\theta_p)$ as in (8). Denote the vector of the coefficients $c \in \mathbb{R}^{t^p}$.
    Compute $\tilde{p}_D \cdot c$.
end for
```

We can view this kind of queries as a subclass of the smooth queries. So, like in the case of k-way marginals queries, we will give an ϵ-LDP sanitizer for smooth queries. Now we consider the data universe $\mathcal{D} = [-1,1]^p$, and dataset $D \in \mathcal{D}^n$. For a positive integer h and constant $T > 0$, we denote the set of all p-dimensional (h, T)-smooth function (Definition 7) as C_T^h, and $\mathcal{Q}_{C_T^h} = \{q_f(D) = \frac{1}{n}\sum_{x \in D} f(D), f \in C_T^h\}$ the corresponding set of queries. The idea of the algorithm is similar to the one used for the k-way marginals; but instead of using Chebyshev polynomials, we will use trigonometric polynomials. We now assume that the dimensionality p, h and T are constants so all the result in big O notation will be omitted. The idea of Algorithm 7 is based on the following Lemma.

Lemma 42 ((Wang et al., 2016)) *Assume $\gamma > 0$. For every $f \in C_T^h$, defined on $[-1,1]^p$, let $g_f(\theta_1, \dots, \theta_p) = f(\cos(\theta_1), \dots, \cos(\theta_p))$, for $\theta_i \in [-\pi, \pi]$. Then there is an even trigonometric polynomial p whose degree for each variable is $t(\gamma) = (\frac{1}{\gamma})^{\frac{1}{h}}$:*

$$p(\theta_1, \dots, \theta_p) = \sum_{0 \leq r_1, \dots, r_p < t(\gamma)} c_{r_1, \dots, r_p} \prod_{i=1}^{p} \cos(r_i\theta_i), \tag{8}$$

such that 1) p γ-uniformly approximates g_f, i.e. $\sup_{x \in [-\pi,\pi]^p} |p(x) - g_f(x)| \leq \gamma$, 2) the coefficients are uniformly bounded by a constant M which only depends on h, T and p, 3) moreover, the entire set of the coefficients can be computed in time $O\big((\frac{1}{\gamma})^{\frac{p+2}{h} + \frac{2p}{h^2}} poly \log \frac{1}{\gamma}\big)$.

By (8), we can see that all the $p(x)$ which corresponds to $g_f(x)$, representing functions $f \in C_T^h$, have the same basis $\prod_{i=1}^p \cos(r_i\theta_i)$. So we can use Lemma 3 and 22 to estimate the average of the basis. Then, for each query f the server can only compute the corresponding coefficients $\{c_{r_1,r_2,\cdots,r_p}\}$. This idea is implemented in Algorithm 7 for which we have the following result.

Theorem 43 *For any $\epsilon > 0$, Algorithm 7 is ϵ-LDP. Also for $\alpha > 0$, $0 < \beta < 1$, if*

$$n = O(\max\{\log^{\frac{5p+2h}{2h}}(\frac{1}{\beta})\epsilon^{-2}\alpha^{-\frac{5p+2h}{h}}, \frac{1}{\epsilon^2}\log(\frac{1}{\beta})\})$$

and $t = O((\sqrt{n}\epsilon)^{\frac{2}{5p+2h}})$, then Algorithm 7 is (α, β)-accurate with respect to $\mathcal{Q}_{C^h_T}$. Moreover, the time for answering each query is $\tilde{O}((\sqrt{n}\epsilon)^{\frac{4p+4}{5p+2h}+\frac{4p}{5ph+2h^2}})$, where O omits h, T, p and some log *terms. For each player, the computation and communication cost could be improved to $O(1)$ and 1 bit, respectively, as in Section 4.2.*

7. Conclusions and Discussions

In this paper, we studied ERM under the non-interactive local differential privacy model and made two attempts to resolve the issue of exponential dependency in the dimentionality. In our first attempt, we showed that if the loss function is smooth enough, then the sample complexity to achieve α error is α^{-c} for some positive constant c, which improves significantly on the previous result of $\alpha^{-(p+1)}$.

Moreover, we proposed efficient algorithms for both player and server views. In our second attempt, we show that the sample complexity for any 1-Lipschtiz generalized linear convex function is only linear in p and exponential on other terms by using polynomial of inner product approximation. Moreover, our techniques can also be extended some other related problems such as answering k-way-marginals and smooth queries in the local model.

There are still many open problems left. Firstly, as we showed in this paper, the α term can be polynomial in the sample complexity when the loss function is smooth enough while the p term can be polynomial when the loss function is generalized linear. Thus, a natural question is to determine whether it is possible to get an algorithm whose sample complexity is fully polynomial in all the terms when the loss function is generalized linear and smooth enough, such as logistic regression. Secondly, although we have shown the advantages of these two methods, we do not know the practical performance of these methods.

Acknowledgments

D.W. and J.X. were supported in part by NSF through grants CCF-1422324 and CCF-1716400. M.G was supported by NSF awards CNS-1565365 and CCF-1718220. A.S. was supported by NSF awards IIS-1447700 and AF-1763786 and a Sloan Foundation Research Award. Part of this research was done while D.W. was visiting Boston University and Harvard University's Privacy Tools Project.

References

Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Communication efficient, sample optimal, linear time locally private discrete distribution estimation. *arXiv preprint arXiv:1802.04705*, 2018.

Francesco Aldà and Benjamin IP Rubinstein. The bernstein mechanism: Function release under differential privacy. In *AAAI*, pages 1705–1711, 2017.

Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 127–135. ACM, 2015.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 464–473. IEEE, 2014.

Raef Bassily, Uri Stemmer, Abhradeep Guha Thakurta, et al. Practical locally private heavy hitters. In *Advances in Neural Information Processing Systems*, pages 2288–2296, 2017.

Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *CRYPTO*, volume 5157, pages 451–468. Springer, 2008.

Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 435–447. ACM, 2018.

Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in neural information processing systems*, pages 289–296, 2009.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Chunhui Chen and Olvi L Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2): 97–138, 1996.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

Pavel Dvurechensky and Alexander Gasnikov. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

Alexandre V. Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Principles of Database Systems (PODS)*, pages 211–222, 2003.

Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1170–1178, 2014. URL http://jmlr.org/proceedings/papers/v32/gaboardi14.html.

Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM Journal on Computing*, 42(4):1494–1520, 2013.

Samuel Haney, Ashwin Machanavajjhala, John M. Abowd, Matthew Graham, Mark Kutzbach, and Lars Vilhuber. Utility cost of formal privacy for releasing national employer-employee statistics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1339–1354, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4197-4. doi: 10.1145/3035918.3035940. URL http://doi.acm.org/10.1145/3035918.3035940.

Moritz Hardt, Guy N Rothblum, and Rocco A Servedio. Private data release via learning thresholds. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 168–187. Society for Industrial and Applied Mathematics, 2012.

Justin Hsu, Sanjeev Khanna, and Aaron Roth. Distributed private heavy hitters. *Automata, Languages, and Programming*, pages 461–472, 2012.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2436–2444. JMLR.org, 2016.

Shiva Prasad Kasiviswanathan and Hongxia Jin. Efficient private empirical risk minimization for high-dimensional learning. In *International Conference on Machine Learning*, pages 488–497, 2016.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1(41): 3–1, 2012.

Tejas Kulkarni, Graham Cormode, and Divesh Srivastava. Marginal release under local differential privacy. *CoRR*, abs/1711.02952, November 2017. URL `http://arxiv.org/abs/1711.02952`.

G.G. Lorentz. *Bernstein Polynomials*. AMS Chelsea Publishing Series. Chelsea Publishing Company, 1986. ISBN 9780828403238.

Charles Micchelli. The saturation class and iterates of the bernstein polynomials. *Journal of Approximation Theory*, 8(1):1–18, 1973.

Joe Near. Differential privacy at scale: Uber and berkeley collaboration. In *Enigma 2018 (Enigma 2018)*, Santa Clara, CA, 2018. USENIX Association.

Arkadi Nemirovski. On parallel complexity of nonsmooth convex optimization. *J. Complexity*, 10(4):451–463, 1994. doi: 10.1006/jcom.1994.1025.

Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. In *Algorithmic Learning Theory*, pages 619–653, 2018.

Andrej Risteski and Yuanzhi Li. Algorithms and matching lower bounds for approximately-convex optimization. In *Advances in Neural Information Processing Systems*, pages 4745–4753, 2016.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

Adam Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *IEEE Symposium on Security and Privacy*, 2017.

Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3025–3033, 2015.

Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *CoRR*, abs/1709.02753, 2017.

Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

Lloyd N Trefethen. *Approximation theory and approximation practice*, volume 128. Siam, 2013.

Di Wang and Jinhui Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. *Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA, January 27-February 1, 2019*, 2019.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2719–2728, 2017a.

Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, 3-8 December 2018, Montreal, QC, Canada*, 2018.

Di Wang, Adam Smith, and Jinhui Xu. Noninteractive locally private learning of linear models via polynomial approximations. In *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, IL, USA*, 2019.

Shaowei Wang, Yiwen Nie, Pengzhan Wang, Hongli Xu, Wei Yang, and Liusheng Huang. Local private ordinal data distribution estimation. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017b.

Ziteng Wang, Chi Jin, Kai Fan, Jiaqi Zhang, Junliang Huang, Yiqiao Zhong, and Liwei Wang. Differentially private data releasing for smooth queries. *The Journal of Machine Learning Research*, 17(1):1779–1820, 2016.

Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 8505–8515, 2018.

M. Ye and A. Barg. Optimal Schemes for Discrete Distribution Estimation under Locally Differential Privacy. *ArXiv e-prints*, February 2017.

Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. Calm: Consistent adaptive local marginal for marginal release under local differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 212–229. ACM, 2018.

Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at once, use effectively: Making non-interactive locally private learning possible. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 4130–4139, 2017.

Appendix A. Details of Omitted Proofs

In this section, we provide the details of the omitted proofs for the theorems, lemmas, and corollaries stated in previous sections.

A.1 Proofs in Section 3

Lemma 44 ((Nissim and Stemmer, 2018)) *Suppose that $x_1, \cdots, x_n$ are i.i.d sampled from Lap($\frac{1}{\epsilon}$). Then for every $0 \leq t < \frac{2n}{\epsilon}$, we have*

$$Pr(|\sum_{i=1}^{n} x_i| \geq t) \leq 2\exp(-\frac{\epsilon^2 t^2}{4n}).$$

Proof [**Proof of Lemma 3**] Consider Algorithm 1. We have $|a - \frac{1}{n}\sum_{i=1}^{n} v_i| = |\frac{\sum_{i=1}^{n} x_i}{n}|$, where $x_i \sim \text{Lap}(\frac{b}{\epsilon})$. Taking $t = \frac{2\sqrt{n}\sqrt{\log\frac{2}{\beta}}}{\epsilon}$ and applying Lemma 44, we prove the lemma. ■

A.2 Proofs in Section 4.1

Proof [**Proof of Corollaries 19 and 20**] Since the loss function is (∞, T)-smooth, it is $(2p, T)$-smooth for all p. Thus, taking $h = p$ in Theorem 17, we get the proof. ■

Lemma 45 *[**(Shalev-Shwartz et al., 2009)**] If the loss function ℓ is L-Lipschitz and μ-strongly convex, then with probability at least $1 - \beta$ over the randomness of sampling the data set $\mathcal{D}$, the following is true,*

$$Err_{\mathcal{P}}(\theta) \leq \sqrt{\frac{2L^2}{\mu}}\sqrt{Err_{\mathcal{D}}(\theta)} + \frac{4L^2}{\beta\mu n}.$$

Proof [**Proof of Theorem 21**] For the general convex loss function ℓ, we let $\hat{\ell}(\theta; x) = \ell(\theta; x) + \frac{\mu}{2}\|\theta\|^2$ for some $\mu > 0$. Note that in this case the new empirical risk becomes $\bar{L}(\theta; D) = \hat{L}(\theta; D) + \frac{\mu}{2}\|\theta\|^2$. Since $\frac{\mu}{2}\|\theta\|^2$ does not depend on the dataset, we can still use the Bernstein polynomial approximation for the original empirical risk $\hat{L}(\theta; D)$ as in Algorithm 2, and the error bound for $\bar{L}(\theta; D)$ is the same. Thus, we can get the population excess risk of the loss function $\hat{\ell}$, $\text{Err}_{\mathcal{P},\hat{\ell}}(\theta_{\text{priv}})$ by Corollary 20 and have the following relation,

$$\text{Err}_{\mathcal{P},\ell}(\theta_{\text{priv}}) \leq \text{Err}_{\mathcal{P},\hat{\ell}}(\theta_{\text{priv}}) + \frac{\mu}{2}.$$

By Lemma 45 for $\text{Err}_{\mathcal{P},\hat{\ell}}(\theta_{\text{priv}})$, where $\hat{\ell}(\theta; x)$ is $1 + \|\mathcal{C}\|_2 = O(1)$-Lipschitz, we have the following,

$$\text{Err}_{\mathcal{P},\ell}(\theta_{\text{priv}}) \leq \tilde{O}(\sqrt{\frac{2}{\mu}\frac{\log^{\frac{1}{8}}\frac{1}{\beta}D_p^{\frac{1}{4}}p^{\frac{1}{8}}\sqrt[4]{2}^{(p+1)p}}{n^{\frac{1}{8}}\epsilon^{\frac{1}{4}}}} + \frac{4}{\beta\mu n} + \frac{\mu}{2}).$$

Taking $\mu = O(\frac{1}{\sqrt[12]{n}})$, we get

$$\text{Err}_{\mathcal{P},\ell}(\theta_{\text{priv}}) \leq \tilde{O}(\frac{\log^{\frac{1}{8}}\frac{1}{\beta}D_p^{\frac{1}{4}}p^{\frac{1}{8}}\sqrt[4]{2}^{(p+1)p}}{\beta n^{\frac{1}{12}}\epsilon^{\frac{1}{4}}}).$$

Thus, we have the theorem. ■

A.3 Proofs in Section 4.2

Proof [**Proof of Theorem 23**] By (Bassily and Smith, 2015) it is ϵ-LDP. The time complexity and communication complexity is obvious. As in (Bassily and Smith, 2015), it is sufficient to show that the LDP-AVG is sampling resilient.

The STAT in (Bassily and Smith, 2015) corresponds to the average in our problem, and $\phi(x,y)$ corresponds to $\max_{j\in[p]}|[x]_j-[y]_j|$. By Lemma 22, we can see that with probability at least $1-\beta$,

$$\phi(\text{Avg}(v_1,v_2,\cdots,v_n);a)=O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}}).$$

Now let $\mathcal{S}$ be the set obtained by sampling each point $v_i, i\in[n]$ independently with probability $\frac{1}{2}$. Note that by Lemma 22, we have the subset $\mathcal{S}$. If $|S|\geq\Omega(\max\{p\log(\frac{p}{\beta}),\frac{1}{\epsilon^2}\log\frac{1}{\beta}\})$ with probability $1-\beta$,

$$\phi(\text{Avg}(\mathcal{S});\text{LDP-AVG}(\mathcal{S}))=O(\frac{b\sqrt{p}}{\sqrt{|\mathcal{S}|}\epsilon}\sqrt{\log\frac{p}{\beta}}).$$

Now by Hoeffdings inequality, we can get $|n/2-|\mathcal{S}||\leq\sqrt{n\log\frac{4}{\beta}}$ with probability $1-\beta$. Also since $n=\Omega(\log\frac{1}{\beta})$, we know that $|\mathcal{S}|\geq O(n)\geq\Omega(p\log(\frac{p}{\beta}))$ is true. Thus, with probability at least $1-2\beta$, $\phi(\text{Avg}(\mathcal{S});\text{LDP-AVG}(\mathcal{S}))=O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}})$.

Actually, we can also get $\phi(\text{Avg}(\mathcal{S});\text{Avg}(v_1,v_2,\cdots,v_n))\leq O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}})$. We now assume that $v_i\in\mathbb{R}$. Note that $\text{Avg}(\mathcal{S})=\frac{v_1x_1+\cdots+v_nx_n}{x_1+\cdots+x_n}$, where each $x_i\sim\text{Bernoulli}(\frac{1}{2})$. Denote $M=x_1+x_2+\cdots+x_n$. By Hoeffdings Inequality, we have with probability at least $1-\frac{\beta}{2}$, $|M-\frac{n}{2}|\leq\sqrt{n\log\frac{4}{\beta}}$. We further denote $N=v_1x_1+\cdots+v_nx_n$. Also, by Hoeffdings inequality, with probability at least $1-\beta$, we get $|N-\frac{v_1+\cdots+v_n}{2}|\leq b\sqrt{n\log\frac{2}{\beta}}$. Thus, with probability at least $1-\beta$, we have:

$$\begin{aligned}|\frac{N}{M}-\frac{v_1+\cdots+v_n}{n}|&\leq\frac{|N-\sum_{i=1}^n v_i/2|}{M}+|\sum_{i=1}^n v_i/2||\frac{1}{M}-\frac{2}{n}|\\&\leq\frac{|N-\sum_{i=1}^n v_i/2|}{M}+\frac{nb}{2}|\frac{1}{M}-\frac{2}{n}|.\end{aligned}\tag{9}$$

For the second term of (9), $|\frac{1}{M}-\frac{2}{n}|=\frac{|n/2-M|}{M\frac{n}{2}}$. We know from the above $|n/2-M|\leq\sqrt{n\log\frac{4}{\beta}}$. Also since $n=\Omega(\log\frac{1}{\beta})$, we get $M\geq O(n)$. Thus, $|\frac{1}{M}-\frac{2}{n}|\leq O(\frac{\sqrt{\log\frac{1}{\beta}}}{\sqrt{n}n})$. The upper bound of the second term is $O(\frac{b\sqrt{\log\frac{1}{\beta}}}{\sqrt{n}})$, and the same for the first term. For p dimensions, we just choose $\beta=\frac{\beta}{p}$ and take the union. Thus in total we have $\phi(\text{Avg}(\mathcal{S});\text{Avg}(v_1,v_2,\cdots,v_n))\leq O(\frac{b}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}})\leq O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}})$.

In summary, we have shown that

$$\phi(\text{AVG-LDP}(\mathcal{S});\text{Avg}(v_1,v_2,\cdots,v_n))\leq O(\frac{bp}{\sqrt{n}\epsilon}\sqrt{\log\frac{p}{\beta}})$$

with probability at least $1 - 4\beta$. ■

Proof [**Proof of Theorem 26**] Let $\theta^* = \arg\min_{\theta \in \mathcal{C}} L(\theta; D)$, $\theta_{\text{priv}} = \arg\min_{\theta \in \mathcal{C}} \tilde{L}(\theta; D)$. Under the assumptions of $\alpha, n, k, \epsilon, \beta$, we know from the proof of Theorem 17 and Corollary 20 that $\sup_{\theta \in \mathcal{C}} |\tilde{L}(\theta; D) - L(\theta; D)| \leq \alpha$. Also by setting $\epsilon = 16348 p\alpha$ and $\alpha \leq \frac{1}{16348} \frac{\mu}{p\sqrt{p}}$, we can see that the condition in Lemma 25 holds for $\Delta = \alpha$. So there is an algorithm whose output $\tilde{\theta}_{\text{priv}}$ satisfies

$$\tilde{L}(\tilde{\theta}_{\text{priv}}; D) \leq \min_{\theta \in \mathcal{C}} \tilde{L}(\theta; D) + O(p\alpha).$$

Thus, we have

$$L(\tilde{\theta}_{\text{priv}}; D) - L(\theta^*; D) \leq L(\tilde{\theta}_{\text{priv}}; D) - \tilde{L}(\theta_{\text{priv}}; D) + \tilde{L}(\theta_{\text{priv}}; D) - L(\theta^*; D),$$

where

$$\begin{aligned} L(\tilde{\theta}_{\text{priv}}; D) - \tilde{L}(\theta_{\text{priv}}; D) &\leq L(\tilde{\theta}_{\text{priv}}; D) - \tilde{L}(\tilde{\theta}_{\text{priv}}; D) + \tilde{L}(\tilde{\theta}_{\text{priv}}; D) - \tilde{L}(\theta_{\text{priv}}; D) \\ &\leq O(p\alpha). \end{aligned}$$

Also $\tilde{L}(\theta_{\text{priv}}; D) - \hat{L}(\theta^*; D) \leq \tilde{L}(\theta^*; D) - \hat{L}(\theta^*; D) \leq \alpha$. Thus, the theorem follows. The running time is determined by n. This is because when we use the algorithm in Lemma 25, we have to use the first order optimization. That is, we have to evaluate some points at $\tilde{L}(\theta; D)$, which will cost at most $O(\text{Poly}(n, \frac{1}{\alpha}))$ time (note that $\tilde{L}$ is a polynomial with $(k+1)^p \leq n$ coefficients). ■

A.4 Proofs in Section 5

Proof [**Proof of Lemma 28**] It is easy to see that items 1 is true. Item 2 is due to the following $|f'_\beta(x)| = |\frac{-1+\frac{x-\frac{1}{2}}{\sqrt{(x-\frac{1}{2})^2+\beta^2}}}{2}| \leq 1$. Item 3 is because of the following $0 \leq f''_\beta(x) = \frac{\beta^2}{((x-\frac{1}{2})^2+\beta^2)^{\frac{3}{2}}} \leq \frac{1}{\beta}$. For item 4 we have $|f^{(3)}_\beta(x)| = \frac{3\beta^2 x}{(x^2+\beta^2)^{\frac{5}{2}}} \leq \frac{3}{\beta^2}$.

■

Proof [**Proof of Theorem 29**] For simplicity, we omit the term of δ, which will not affect the linear dependency. Let

$$\hat{G}(w, i) = [\sum_{j=0}^{d} c_j \binom{d}{j} (y_i \langle w, x_i \rangle)^j (1 - y_i \langle w, x_i \rangle)^{d-j}] y_i x_i^T,$$

where $c_j = f'_\beta(\frac{j}{d})$ and

$$\mathbb{E}_i \hat{G}(w, i) = \frac{1}{n} \sum_{i=1}^{n} \hat{G}(w, i) = \hat{G}(w).$$

For the term of $G(w, i)$, the randomness comes from sampling the index i and the Gaussian noises added for preserving local privacy.

Note that in total $\mathbb{E}_{\sigma,z,i}G(w,i) = \hat{G}(w)$, where $\sigma = \{\sigma_{i,j}\}_{j=0}^{\frac{d(d+1)}{2}}$ and $z = \{z_{i,j}\}_{j=0}^{\frac{d(d+1)}{2}}$.

It is easy to see that $\mathbb{E}_{\sigma,z}G(w,i) = \mathbb{E}[(\sum_{j=0}^{d} c_j \binom{d}{j} t_{i,j}s_{i,j})y_{i,0}x_{i,0}^T \mid i] = \hat{G}(w,i)$, which is due to the fact that $\mathbb{E}t_{i,j} = (y_i\langle w, x_i\rangle)^j$, $\mathbb{E}s_{i,j} = (1 - y_i\langle w, x_i\rangle)^{d-j}$ and each $t_{i,j}, s_{i,j}$ is independent. We now calculate the variance for this term with fixed i. Firstly, we have $\mathrm{Var}(y_{i,0}x_{i,0}^T) = O(\frac{p}{\epsilon^4})$. For each $t_{i,j}$, we get

$$\mathrm{Var}(t_{i,j}) \leq \Pi_{k=jd+1}^{jd+j}\mathrm{Var}(y_{i,k})(\mathrm{Var}(< w_i, x_{i,k} >) + (\mathbb{E}(w_i^T x_{i,k}))^2) \leq \tilde{O}\big((C_1\frac{d(d+1)}{\epsilon^2})^{2j}\big).$$

and similarly we have

$$\mathrm{Var}(s_{i,j}) \leq \tilde{O}\big((C_2\frac{d(d+1)}{\epsilon^2})^{2(d-j)}\big).$$

Thus we have

$$\mathrm{Var}(t_{i,j}s_{i,j}) \leq \tilde{O}\big((C_3\frac{d(d+1)}{\epsilon^2})^{2d}\big).$$

Since function f'_β is bounded by 1 and $\binom{d}{j} \leq d^d$ for each j. In total, we have

$$\mathrm{Var}(G(w_t,i)|i) \leq O(d \cdot d^d \cdot (C_3\frac{d(d+1)}{\epsilon^2})^{2d} \cdot \frac{p}{\epsilon^4}) = \tilde{O}\big(\frac{d^{6d}C^d p}{\epsilon^{4d+4}}\big).$$

Next we consider $\mathrm{Var}(\hat{G}(w,i))$. Since

$$\begin{aligned}\|\hat{G}(w,i) - f'_\beta(y_i x_i^T w)y_i x_i^T\|_2^2 &= \|[\sum_{j=0}^{d} c_j \binom{d}{j}(y_i\langle w, x_i\rangle)^j(1 - y_i\langle w, x_i\rangle)^{d-j} - f'_\beta(w)]y_i x_i^T\|_2^2 \\ &\leq (\frac{1}{\beta^2 d})^2 \leq \frac{\alpha^2}{4},\end{aligned}$$

we get

$$\begin{aligned}\mathrm{Var}(\hat{G}(w,i)) \leq O\big(&\mathbb{E}[\|\hat{G}(w,i) - f'_\beta(y_i x_i^T w)y_i x_i^T\|_2^2] + \mathbb{E}[\hat{G}(w) - \nabla L_\beta(w;D)\|_2^2] \\ &+ \mathbb{E}[\|f'_\beta(y_i x_i^T w)y_i x_i^T - \nabla L_\beta(w;D)\|_2^2]\big) \leq O((\alpha+1)^2).\end{aligned}$$

In total, we have $\mathbb{E}[\|G(w,i) - \hat{G}(w)\|_2^2] \leq \mathbb{E}[\|G(w,i) - \hat{G}(w,i)\|_2^2] + \mathbb{E}[\|\hat{G}(w,i) - \hat{G}(w)\|_2^2] \leq \tilde{O}\big((\frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}} + \alpha + 1)^2\big)$.

Also, we know that

$$\begin{aligned}&L_\beta(v;D) - L_\beta(w;D) - \langle\hat{G}(w), v - w\rangle = \\ &L_\beta(v;D) - L_\beta(w;D) - \langle\nabla L_\beta(w;D), v - w\rangle + \langle\nabla L_\beta(w;D) - G(w), v - w\rangle \\ &\leq \frac{1}{2\beta}\|v - w\|_2^2 + \frac{\alpha}{2},\end{aligned}$$

since L_β is $\frac{1}{\beta}$-smooth and $|\langle\nabla L_\beta(w) - G(w), v - w\rangle| \leq \frac{\alpha}{2}$.

Thus, $G(w,i)$ is an $\big(\frac{\alpha}{2}, \frac{1}{\beta}, O(\frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}} + \alpha + 1)\big)$ stochastic oracle of L_β. ■

Proof [**Proof of Theorem 30**]

The guarantee of differential privacy is by Gaussian mechanism and composition theorem.

By Theorem 29, Lemma 28 and 16, we have

$$\mathbb{E}L_\beta(w_n, D) - \min_{w\in\mathcal{C}} L_\beta(w, D) \leq O(\frac{(\frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}} + \alpha + 1)}{\beta\sqrt{n}} + \frac{1}{\beta^2 d}) = O(\frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}\beta\sqrt{n}} + \frac{\alpha}{2}).$$

By Lemma 28, we know that

$$\mathbb{E}L(w_n, D) - \min_{w\in\mathcal{C}} L(w, D) \leq O(\beta + \frac{d^{3d}C_4^d\sqrt{p}}{\epsilon^{2d+2}\beta\sqrt{n}} + \frac{\alpha}{2}).$$

Thus, if we take $\beta = \frac{\alpha}{4}$, $d = \frac{2}{\beta^2\alpha} = O(\frac{1}{\alpha^3})$ and $n = \Omega(\frac{d^{6d}C_5^d p}{\epsilon^{4d+4}\alpha^2})$, we have

$$\mathbb{E}L(w_n, D) - \min_{w\in\mathcal{C}} L(w, D) \leq \alpha.$$

■

Proof [**Proof of Theorem 34**]

Let h_β denote the function $h_\beta(x) = \frac{x+\sqrt{x^2+\beta^2}}{2}$. By Lemma 32 we have

$$f(\theta) = (f'(1) - f'(-1))\mathbb{E}_{s\sim\mathcal{Q}}\frac{|s-\theta|}{2} + \frac{f'(1)+f'(-1)}{2}\theta + c.$$

Now, we consider function $F_\beta(\theta)$, which is

$$F_\beta(\theta) = (f'(1) - f'(-1))\mathbb{E}_{s\sim\mathcal{Q}}[2h_\beta(\frac{\theta-s}{2}) - \frac{\theta-s}{2}] + \frac{f'(1)+f'(-1)}{2}\theta + c.$$

From this, we have

$$\nabla F_\beta(\theta) = (f'(1) - f'(-1))\mathbb{E}_{s\sim\mathcal{Q}}[\nabla h_\beta(\frac{\theta-s}{2})] + \frac{f'(1)+f'(-1)}{2} - \frac{f'(1)-f'(-1)}{2}.$$

Note that since $|x| = 2\max\{x, 0\} - x$, we can get 1) $|F_\beta(\theta) - f(\theta)| \leq O(\beta)$ for any $\theta \in \mathbb{R}$, 2) $F_\beta(x)$ is $O(\frac{1}{\beta})$-smooth and convex since $h_\beta(\theta - s)$ is $\frac{1}{\beta}$-smooth and convex, and 3) $F_\beta(\theta)$ is $O(1)$-Lipschitz. Now, we optimize the following problem in the non-interactive local model:

$$F_\beta(w; D) = \frac{1}{n}\sum_{i=1}^n F_\beta(y_i\langle x_i, w\rangle).$$

For each fixed i and s, we let

$$\hat{G}(w, i, s) = (f'(1) - f'(-1))[\sum_{j=1}^d c_j \binom{d}{j} t_{i,j} r_{i,j}] y_i x_i^T + f'(-1).$$

Then, we have $\mathbb{E}_{\sigma,z}G(w,i,s) = \hat{G}(w,i,s)$. By using a similar argument given in the proof of Theorem 29, we get

$$\mathrm{Var}(\hat{G}(w,i,s)|i,s) \leq \tilde{O}(\frac{d^{6d}C^{d}p}{\epsilon^{4d+4}}).$$

Thus, for each fixed i we have

$$\mathbb{E}_s\hat{G}(w,i,s) = \bar{G}(w,i) = (f'(1)-f'(-1))[\mathbb{E}_{s\sim\mathcal{Q}}\sum_{j=1}^{d}c_j\binom{d}{j}(\frac{y_i\langle w,x_i\rangle - s}{2})^j \\ (1-\frac{y_i\langle w,x_i\rangle - s}{2})^{d-j}]y_ix_i^T + f'(-1).$$

Next, we bound the term of $\mathrm{Var}(\hat{G}(w,i,s)|i) \leq O(d^{2d+2})$.

Let $t_{i,j} = \Pi_{k=jd+1}^{jd+j}(\frac{y_i\langle w_t,x_i\rangle - s_k}{2})$. Then, we have

$$\mathrm{Var}(t_{i,j}) \leq \Pi_{k=jd+1}^{jd+j}|y_i|^2\mathrm{Var}(\langle w_t,x_i\rangle - s_k) \leq O(1).$$

And similarly for $\mathrm{Var}(r_{i,j})$. Thus, we get

$$\mathrm{Var}(\hat{G}(w,i,s)|i) \leq O(\sum_{j=1}^{d}c_j^2\binom{d}{j}^2\mathrm{Var}(t_{i,j}r_{i,j})) = O(d^{2d+2}).$$

Since $\mathbb{E}_i\bar{G}(w,i) = \hat{G} = \frac{1}{n}\sum_{i=1}^{n}\bar{G}(w,i)$, we have $\mathrm{Var}(\bar{G}(w,i)) \leq O((\alpha+1)^2)$ by a similar argument given in the proof of Theorem 29. Thus, in total we have

$$\mathbb{E}\|G(w,i,s)-\hat{G}\|_2^2 \leq \tilde{O}(\frac{d^{6d}C^{d}p}{\epsilon^{4d+4}})$$

The other part of the proof is the same as that of Theorem 29. ■

A.5 Proofs in Section 6

Proof [Proof of Theorem 41] It is sufficient to prove that

$$\sup_{y\in Y_k}|\tilde{q}_D(y) - q_y(D)| \leq \gamma + \frac{T\binom{p+t_k}{t_k}^2\sqrt{\log\frac{\binom{p+t_k}{t_k}}{\beta}}}{\sqrt{n}\epsilon},$$

where $T = p^{O(\sqrt{k}\log(\frac{1}{\gamma}))}$. Now we denote $p_D \in \mathbb{R}^{\binom{p+t_k}{t_k}}$ as the average of $\tilde{q}_i$. That is, it is the unperturbed version of $\tilde{p}_D$. By Lemma 40, we have $\sup_{y\in Y_k}|p_D(y) - q_y(D)| \leq \gamma$. Thus it is sufficient to prove that

$$\sup_{y\in Y_k}|\tilde{q}_D(y) - p_D(y)| \leq \frac{T\binom{p+t_k}{t_k}^2\sqrt{\log\frac{\binom{p+t_k}{t_k}}{\beta}}}{\sqrt{n}\epsilon}.$$

Since both $\tilde{q}_D$ and p_D can be viewed as $\binom{p+t_k}{t_k}$-dimensional vectors, we then have

$$\sup_{y \in Y_k} |\tilde{p}_D(y) - p_D(y)| \leq \|\tilde{p}_D - p_D\|_1.$$

Also, since each coordinate of $p_D(y)$ is bounded by T by Lemma 40, we can see that if $n = \Omega(\max\{\frac{1}{\epsilon^2} \log \frac{1}{\beta}, \binom{p+t_k}{t_k} \log \binom{p+t_k}{t_k} \log 1/\beta\})$, then by Lemma 3, with probability at least $1-\beta$, the following is true

$$\|\tilde{p}_D - p_D\|_1 \leq \frac{T\binom{p+t_k}{t_k}^2 \sqrt{\log \frac{\binom{p+t_k}{t_k}}{\beta}}}{\sqrt{n}\epsilon}.$$

Thus, if taking $\gamma = \frac{\alpha}{2}$ and by the fact that $\binom{p+t_k}{t_k} = p^{O(t_k)}$, we get the proof. ■

Proof [Proof of Theorem 43] Let $t = (\frac{1}{\gamma})^{\frac{1}{h}}$. It is sufficient to prove that $\sup_{q_f \in \mathcal{Q}_{C_T^h}} |\tilde{p}_D \cdot c_f - q_f(D)| \leq \alpha$. Let p_D denote the average of $\{p_i\}_{i=1}^n$, *i.e.* the unperturbed version of $\tilde{p}_D$. Then by Lemma 42, we have $\sup_{q_f \in \mathcal{Q}_{C_T^h}} |p_D \cdot c_f - q_f(D)| \leq \gamma$. Also since $\|c_f\|_\infty \leq M$, we have $\sup_{q_f \in \mathcal{Q}_{C_T^h}} |\tilde{p}_D \cdot c_f - p_D \cdot c_f| \leq O(\|\tilde{p}_D - p_D\|_1)$. By Lemma 3, we know that if $n = \Omega(\max\{\frac{1}{\epsilon^2} \log \frac{1}{\beta}, t^{2p} \log \frac{1}{\beta}\})$, then $\|\tilde{p}_D - p_D\|_1 \leq O(\frac{t^{\frac{5p}{2}} \sqrt{\log(\frac{1}{\beta})}}{\sqrt{n}\epsilon})$ with probability at least $1-\beta$. Thus, we have $\sup_{q_f \in \mathcal{Q}_{C_T^h}} |\tilde{p}_D \cdot c_f - q_f(D)| \leq O(\gamma + \frac{(\frac{1}{\gamma})^{\frac{5p}{2h}} \sqrt{\log(\frac{1}{\beta})}}{\sqrt{n}\epsilon})$. Taking $\gamma = O((1/\sqrt{n}\epsilon)^{\frac{2h}{5p+2h}})$, we get $\sup_{q_f \in \mathcal{Q}_{C_T^h}} |\tilde{p}_D \cdot c_f - q_f(D)| \leq O(\sqrt{\log(\frac{1}{\beta})}(\frac{1}{\sqrt{n}\epsilon})^{\frac{2h}{5p+2h}}) \leq \alpha$. The computational cost for answering a query follows from Lemma 42 and $b \cdot c = O(t^p)$. ■

Appendix B. Omitted Details in Section 4.2

Recently, Bun et al. (2018) proposed a generic transformation, GenProt, which could transform any (ϵ, δ) (so as for ϵ) non-interactive LDP protocol to an $O(\epsilon)$-LDP protocol with the communication complexity for each player being $O(\log \log n)$ (at the expense of increasing the shared randomness in the protocol), which removes the condition of 'sample resilient' in (Bassily and Smith, 2015). The detail is in Algorithm 8. The transformation uses $O(n \log \frac{n}{\beta})$ independent public string. The reader is referred to (Bun et al., 2018) for details. Actually, by Algorithm 8, we can easily get an $O(\epsilon)$-LDP algorithm with the same error bound.

Theorem 46 *For any given $\epsilon \leq \frac{1}{4}$, under the condition of Corollary 20, Algorithm 8 is 10ϵ-LDP. If $T = O(\log \frac{n}{\beta})$, then with probability at least $1 - 2\beta$, Corollary 20 holds. Moreover, the communication complexity of each layer is $O(\log \log n)$ bits, and the computational complexity for each player is $O(\log \frac{n}{\beta})$.*

Algorithm 8 Player-Efficient Local Bernstein Mechanism with $O(\log\log n)$ bits communication complexity.

```
Input: Each user $i \in [n]$ has data $x_i \in \mathcal{D}$, privacy parameter $\epsilon$, public loss function $\ell : [0,1]^p \times \mathcal{D} \mapsto [0,1]$, and parameter $k, T$.
Preprocessing:
For every $(i,T) \in [n] \times [T]$, generate independent public string $y_{i,t} = \text{Lap}(\perp)$.
Construct the grid $\mathcal{T} = \{\frac{v_1}{k}, \frac{v_2}{k}, \cdots, \frac{v_p}{k}\}_{v_1,v_2,\cdots,v_p}$, where $\{v_1, v_2, \cdots, v_p\} = \{0, 1, \cdots, k\}^p$.
Randomly partition $[n]$ in to $d = (k+1)^p$ subsets $I_1, I_2, \cdots, I_d$, with each subset $I_j$ corresponding to an grid in $\mathcal{T}$ denoted as $\mathcal{T}(j)$.
for Each Player $i \in [n]$ do
    Find the subset $I_\ell$ such that $i \in I_\ell$. Calculate $v_i = \ell(\mathcal{T}(l); x_i)$.
    For each $t \in [T]$, compute $p_{i,t} = \frac{1}{2}\frac{\Pr[v_i + Lap(\frac{1}{\epsilon}) = y_{i,t}]}{\Pr[\text{Lap}(\perp) = y_{i,t}]}$
    For every $t \in [T]$, if $p_{i,t} \notin [\frac{e^{-2\epsilon}}{2}, \frac{e^{2\epsilon}}{2}]$, then set $p_{i,t} = \frac{1}{2}$.
    For every $t \in [T]$, sample a bit $b_{i,t}$ from Bernoulli($p_{i,t}$).
    Denote $H_i = \{t \in [T] : b_{i,t} = 1\}$
    If $H_i = \emptyset$, set $H_i = [T]$
    Sample $g_i \in H_i$ uniformly, and send $g_i$ to the server.
end for
for The Server do
    for Each $l \in [d]$ do
        Compute $v_\ell = \frac{n}{|I_\ell|}\sum_{i \in I_\ell} g_i$.
        Denote the corresponding grid point $(\frac{v_1}{k}, \frac{v_2}{k}, \cdots, \frac{v_p}{k}) \in \mathcal{T}$ as $\ell$; then let $\hat{L}((\frac{v_1}{k}, \frac{v_2}{k}, \cdots, \frac{v_p}{k}); D) = v_\ell$.
    end for
    Construct perturbed Bernstein polynomial of the empirical loss $\tilde{L}$ as in Algorithm 2. Denote the function as $\tilde{L}(\cdot, D)$.
    Compute $w_{\text{priv}} = \arg\min_{w \in \mathcal{C}} \tilde{L}(w; D)$.
end for
```

Appendix C. Detailed Algorithm of SIGM in Lemma 16

Let $a \geq 1, b \geq 0, p \geq 1$ be some parameters. Let us assume that we know a number R such that $\|w^*\|_2 \leq R$. We choose

$$\alpha_i = \frac{1}{a}(\frac{i+p}{p})^{p-1} \tag{10}$$

$$\beta_i = \beta + \frac{b\sigma}{R}(i+p+1)^{\frac{2p-1}{2}} \tag{11}$$

$$B_i = a\alpha_i^2 = \frac{1}{a}(\frac{i+p}{p})^{2p-2}. \tag{12}$$

We also define $A_k = \sum_{i=0}^n \alpha_i$ and $\eta_i = \frac{\alpha_{i+1}}{B_{i+1}}$ and $\alpha_0 = A_0 = B_0$

Lemma 47 (Theorem 3.4 in (Dvurechensky and Gasnikov, 2016)) *Assume that $f(w)$ is endowed with a (γ, β, σ) stochastic oracle $(F_{\gamma,\beta,\sigma}(w;\xi), G_{\gamma,\beta,\sigma}(w;\xi))$ with $\beta \geq O(1)$. By* *choosing the parameters above with* $a = 2^{\frac{p-1}{2}}$ *and* $b = 2^{\frac{5-2p}{4}} p^{\frac{1-2p}{2}}$*, then the sequence* y_k *generated by Algorithm 9*

$$\mathbb{E}_{x_0, x_1, \cdots, x_k}[f(y_k)] - \min_{y \in \mathcal{C}} f(y) \leq \Theta(\frac{\beta R^2}{k^p} + \frac{\sigma R}{\sqrt{k}} + k^{p-1}\gamma).$$

Taking $p = 1$*, this is just Lemma 16.*

Algorithm 9 Stochastic Intermediate Gradient Method

```
**Input:** The sequences $\{\alpha_i\}_{i \geq 0}, \{\beta_i\}_{i \geq 0}, \{B_i\}_{i \geq 0}$, functions $d(x) = \frac{1}{2}\|x\|^2$, Bregman distance $V(x, z) = d(X) - d(Z) - \langle \nabla d(z), x - z \rangle$.
Compute $x_0 = \arg\min_{x \in \mathcal{C}}\{d(x)\}$.
Let $\xi_0$ be a realization of the random variable $\xi$.
Computer $G_{\gamma,\beta,\sigma}(x_0; \xi_0)$.
Compute
    $$y_0 = \arg\min_{x \in \mathcal{C}}\{\beta_0 d(x) + \alpha_0 \langle G_{\gamma,\beta,\sigma}(x_0; \xi_0), x - x_0 \rangle. \tag{13}$$
**for** $k = 0, \cdots, T-1$ **do**
    Compute
        $$z_k = \arg\min_{x \in \mathcal{C}} \beta_k d(x) + \sum_{i=0}^{k} \alpha_i \langle G_{\gamma,\beta,\sigma}(x_i; \xi_i), x - x_i \rangle \tag{14}$$
    Let $x_{k+1} = \eta_k z_k + (1 - \eta_k) y_k$.
    Let $\xi_{k+1}$ be a realization of the random variable $\xi$.
    Compute $G_{\gamma,\beta,\sigma}(x_{k+1}; \xi_{k+1})$
    Compute
        $$\hat{x}_{k+1} = \arg\min_{x \in \mathcal{C}} \beta_k V(x, z_k) + \alpha_{k+1} \langle G_{\gamma,\beta,\sigma}(x_{k+1}; \xi_{k+1}), x - z_k \rangle. \tag{15}$$
    Let $w_{k+1} = \eta \hat{x}_{k+1} + (1 - \eta_k) y_k$.
    Let $y_{k+1} = \frac{A_{k+1} - B_{k+1}}{A_{k+1}} y_k + \frac{B_{k+1}}{A_{k+1}} w_{k+1}$.
**end for**
**return** $y_T$.
```