User-level Differentially Private Stochastic Convex Optimization: Efficient Algorithms with Optimal Rates

Hilal Asi

Apple Inc.

Daogao Liu

University of Washington.

Abstract

We study differentially private stochastic convex optimization (DP-SCO) under user-level privacy, where each user may hold multiple data items. Existing work for user-level DP-SCO either requires super-polynomial runtime [Ghazi et al., 2023b] or requires the number of users to grow polynomially with the dimensionality of the problem with additional strict assumptions [Bassily and Sun, 2023]. We develop new algorithms for user-level DP-SCO that obtain optimal rates for both convex and strongly convex functions in polynomial time and require the number of users to grow only logarithmically in the dimension. Moreover, our algorithms are the first to obtain optimal rates for non-smooth functions in polynomial time. These algorithms are based on multiple-pass DP-SGD, combined with a novel private mean estimation procedure for concentrated data, which applies an outlier removal step before estimating the mean of the gradients.

1 Introduction

Differentially private stochastic convex optimization (DP-SCO) is a central problem in privacy-preserving machine learning, whose aim is to minimize a convex function

$$\begin{aligned} &\text{minimize } \; L_{\mathcal{P}}(\theta) := \mathop{\mathbb{E}}_{Z\sim\mathcal{P}}[\ell(\theta;Z)] \\ &\text{subject to } \; \theta \in \Theta \subset \mathbb{R}^d, \end{aligned} \tag{1}$$

under the constraint of differential privacy, given n users each holding a single sample $z_i \in \mathcal{Z}$ from the distribution $\mathcal{P}$. Numerous works have studied this problem, known as item-level DP-SCO, and it is by now relatively well understood [Bassily et al., 2014, Bassily et al., 2019, Feldman et al., 2020, Asi et al., 2021b, Asi et al., 2021a, Kulkarni et al., 2021].

A significant concern about item-level DP-SCO in practice is that each user may hold and contribute multiple items to the dataset, significantly degrading the actual privacy protection provided by the item-level differential privacy to users. This is the case in many machine learning applications in practice, such as training language and vision models on users' data in federated learning. To address this problem, prior work has studied user-level versions of differential privacy, where the algorithm preserves privacy concerning users that may contribute $m \geq 1$ data points [Liu et al., 2020, Badih et al., 2021, Levy et al., 2021]. This definition is stronger than item-level DP as it forces the algorithm not to be sensitive to changes of a single user or equivalently m data points.

Motivated by the realistic and strong privacy protections guaranteed by user-level privacy, many papers have studied DP-SCO under this notion of privacy. [Levy et al., 2021] has initiated the study of this problem and proposed new algorithms based on localized SGD. The main observation in [Levy et al., 2021] is that averaging the gradients

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

of users in SGD results in gradients that are concentrated in a ball of small radius of roughly $1/\sqrt{m}$, yielding a final excess risk of $1/\sqrt{nm} + d/n\sqrt{m}\varepsilon$. However, as the optimal rates for item-level DP-SCO ($m = 1$) are known to be $1/\sqrt{n} + \sqrt{d}/n\varepsilon$, it is evident that the rates of [Levy et al., 2021] are suboptimal. Moreover, their algorithms are applicable only to smooth functions.

Two recent works of [Bassily and Sun, 2023, Ghazi et al., 2023b] have resolved some of these issues. [Bassily and Sun, 2023] developed new algorithms based on DP-SGD with improved mean estimation procedures to obtain an optimal rate $1/\sqrt{nm} + \sqrt{d}/n\sqrt{m}\varepsilon$. However, their algorithms also require smoothness of the function and require an unnecessarily strong lower bound on the number of users $n \geq \sqrt{d}/\varepsilon$. Moreover, their algorithm can not work for large m and require $m \leq \max\{\sqrt{d}, n\varepsilon^2/\sqrt{d}\}$. On the other hand, [Ghazi et al., 2023b] observe that user-level DP-SCO has small local sensitivity to deletions, and used propose-test-release to desgin new algorithms. Their algorithm requires only $n \geq \log(d)/\varepsilon$ users and is also applicable to non-smooth functions. However, it runs in super-polynomial time and achieves sub-optimal error $1/\sqrt{nm} + \sqrt{d}/n\sqrt{m}\varepsilon^{2.5}$.

As a result, existing algorithms for user-level DP-SCO are not satisfactory: they either require smoothness and a large number of users that grows polynomially in the dimension [Bassily and Sun, 2023], or run in super-polynomial time [Ghazi et al., 2023b].

1.1 Contributions and Technical Overview

In this work, we develop new algorithms for user-level DP-SCO that resolve the abovementioned issues. In particular, our algorithms obtain optimal rates in polynomial time, are applicable for non-smooth functions, and requires a number of users that grows only logarithmically in the dimension $n \leq \frac{\log(d)}{\varepsilon}$. We summarize our results for the convex case and compare them to prior work in Table 1. Additionally, building on our algorithm for the convex case, we propose a new algorithm that obtains optimal rates for user-level DP-SCO in the strongly convex case.

Our algorithm follows a similar recipe to that of [Bassily and Sun, 2023]: as it is well known that DP-SGD is optimal in the item-level setting, we wish to extend it to user-level DP using new mean estimation procedures that add less noise to estimate the gradients at each iteration. To this end, note that if we average the gradients of each user using their m samples, this guarantees that the resulting averaged gradients of all users will lie in a ball of radius roughly $\tau = 1/\sqrt{m}$. This concentration allows to design algorithms for mean estimation with sensitivity τ/n (instead of $1/n$), hence obtaining error (e.g., [Bassily and Sun, 2023]) $\tau\sqrt{d}/n\varepsilon_i$ for estimating the gradients at iteration i, where ε_i is the privacy budget at iteration i. As we have T iterations, this requires $\varepsilon_i = \varepsilon/\sqrt{T}$. The key challenge here is that private mean estimation procedures for τ-concentrated data (e.g. [Levy et al., 2021, Bassily and Sun, 2023]) require $n \geq 1/\varepsilon_i = \sqrt{T}/\varepsilon$, which results in a strong restriction on the number of rounds T that we can run.

Our main challenge is then to design a private mean estimation procedure with T iterations. Each iteration we wish to estimate the mean of τ-concentrated data with privacy budget $\varepsilon_i = \varepsilon/\sqrt{T}$ such that the error at each iteration is $\tau\sqrt{d}/n\varepsilon_i$, and the algorithm uses only $n \leq \log(T)/\varepsilon$ samples. We develop a new private mean estimation algorithm for τ-concentrated data that satisfies these properties.

Our approach draws inspiration from the Friendly-Core framework [Tsfadia et al., 2022], which we use for removing outliers from the dataset. Our methodology has two distinct phases: in the initial stage, we employ an outlier-elimination process that yields a subset of data samples exhibiting τ-concentration. Subsequently, we privatize the mean of the concentrated sample by adding Gaussian noise proportional to τ.

Our outlier-detection phase is based on a score we give to each sample to measure how likely it is to be an outlier; the score measures how many samples in the dataset are in a ball of size τ around the sample. We then keep each sample in the dataset with probability proportional to its score, hence removing outliers which have low score. To guarantee that our final algorithm is private, we have to upper bound the sensitivity of the mean of the sub-sampled dataset is small. To this end, we apply an extra step via AboveThreshold [Dwork and Roth, 2014] to verify that the input dataset is nearly τ-concentrated, hence limiting the number of outliers that can be detected.

This improved mean estimation procedure is the building block of all of our results: it allows us to use DP-SGD with a small number of users and run it for large number of rounds to get the optimal rate. Moreover, the large number of rounds made possible

	Excess Risk	**Polynomial Runtime**	**Number of Users**
[Bassily and Sun, 2023]	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon}$	Yes	$n \geq \frac{\sqrt{d}}{\varepsilon}$
[Ghazi et al., 2023b]	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon^{2.5}}$	No	$n \geq \frac{1}{\varepsilon}$
This work	$\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}\varepsilon}$	Yes	$n \geq \frac{1}{\varepsilon}$

Table 1: Comparison of excess risk bounds for user-level DP-SCO with prior work, with logarithmic terms omitted. The work of [Bassily and Sun, 2023] additionally requires smoothness of the loss function and $m \leq \max\{\sqrt{d}, n\varepsilon^2/\sqrt{d}\}$.

by our mean estimation procedure allows us to use randomized smoothing in order to obtain optimal results in the non-smooth case as well, in contrast to prior work where randomized smoothing would not result in optimal rates in the non-smooth setting.

1.2 Related Work

User-level differential privacy (DP) is a relatively recent and less-explored area compared to the more established item-level DP setting. It has gained increased attention lately due to its significance in machine learning applications, particularly in the context of federated learning. Several works have studied user-level DP for several applications, including DP-SCO [Levy et al., 2021, Bassily and Sun, 2023], PAC learning [Badih et al., 2021], and discrete distribution estimation [Liu et al., 2020, Acharya et al., 2023]. In particular, the lower bound in [Levy et al., 2021] shows the tightness of the dependence on m in our results. In recent work, [Ghazi et al., 2023a] proposed a generic transformation of any item-level DP algorithm to a user-level DP algorithm. However, it is inefficient, and the dependence on ε may not be optimal.

DP-SCO has been studied in the item-level DP setting extensively [Bassily et al., 2019, Feldman et al., 2020, Asi et al., 2021b, Asi et al., 2021a, Kulkarni et al., 2021, Gopi et al., 2023]. The rates of DP-SCO in the item-level setting are well understood and [Bassily et al., 2019] obtained the optimal $1/\sqrt{n} + \sqrt{d\log(1/\delta)}/n\varepsilon$ rate using stability based analysis of DP-SGD with a large batch size. These algorithms are not efficient, leading [Feldman et al., 2020] to develop new optimal algorithms for the smooth case that run in linear time. However, the best runtime for the non-smooth setting is super-linear, and this is an ongoing research direction which is still open [Asi et al., 2021b, Kulkarni et al., 2021, Carmon et al., 2023]. Item-level DP-SCO has also been studied in various other settings, such as the stronger pure DP model [Asi et al., 2021c], heavy-tailed data distributions [Lowy and Razaviyayn, 2023], non-euclidean geometries [Asi et al., 2021b, Bassily et al., 2021], and non-convex loss functions [Ganesh et al., 2023, Arora et al., 2023].

2 Preliminaries

Let $[k] = \{1, \cdots, k\}$ be the set of positive integers no larger than k. Throughout the paper, we assume that the loss function $\ell(:, z) : \Theta \to \mathbb{R}$ is convex and G-Lipschitz for any $z \in \mathcal{Z}$, and $\Theta \subset \mathbb{R}^d$ is a closed convex domain of diameter R. There are n users, each holding m i.i.d. samples from the underlying distribution $\mathcal{P}$; we denote the samples of the i-th user by $Z_i = \{z_{i,j}\}_{j\in[m]}$. We use capital Z to denote one user and z to denote one item. The dataset $\mathcal{D} = \{Z_i\}_{i\in[n]}$ contains all the users along with all the items.

The objective is to design efficient algorithms for minimizing $L_{\mathcal{P}}(\theta) := \mathbb{E}_{z\sim\mathcal{P}}\,\ell(\theta, z)$, which is differentially private at the user level. For a user $Z_i = \{z_{i,j}\}_{j\in[m]}$, we let $\nabla L(\theta; Z_i) := \frac{1}{m}\sum_{j\in[m]} \nabla\ell(\theta; Z_{i,j})$ denote the average of the gradients for the user's samples. We denote the empirical function $L_{\mathcal{D}}(\theta) := \frac{1}{nm}\sum_{z\in Z_i}\sum_{Z_i\in\mathcal{D}} \ell(\theta, z)$. For a distribution X, we let $\operatorname{supp}(X)$ be the support of the distribution X.

2.1 Differential Privacy

In this work, we use the notion of user-level differential privacy where each user has a sample $z \in \mathcal{Z}^m$.

Definition 2.1 (User-Level Differential Privacy). A mechanism $\mathcal{M} : (\mathcal{Z}^m)^n \to \mathbb{R}^d$ is (ε, δ) user-level differentially private, if for any neighboring datasets $\mathcal{D}, \mathcal{D}' \in (\mathcal{Z}^m)^n$ that differ in one user, and for any event O in the range of $\mathcal{M}$, we have

$$\Pr[M(\mathcal{D}) \in O] \le e^{\varepsilon} \Pr[M(\mathcal{D}') \in O] + \delta.$$

Note that item-level differential privacy is a specific case of this definition where $m = 1$.

Additionally, our analysis requires the notion of indistinguishability between two random variables.

Definition 2.2 (Indistinguishablity). Two random variables X and Y are (ε, δ)-Indistinguishable if for any event $\mathcal{O}$, we have

$$\Pr[X \in \mathcal{O}] \le e^{\varepsilon} \Pr[Y \in \mathcal{O}] + \delta,$$
$$\text{and } \Pr[Y \in \mathcal{O}] \le e^{\varepsilon} \Pr[X \in \mathcal{O}] + \delta.$$

Moreover, for two distributions X and Y, we use the notation $X \sim_{\gamma} Y$ to denote that the total variation distance between X and Y is bounded by γ. We also define the following divergence.

Definition 2.3. Given two distributions X and Y, the δ-approximate max divergence between X and Y is defined as

$$D_{\infty}^{\delta}(X \| Y) = \sup_{Z \in \operatorname{supp}(X) : \Pr[X \in Z] \ge \delta} \log \frac{\Pr[X \in Z] - \delta}{\Pr[Y \in Z]}$$

2.1.1 AboveThreshold

Our algorithms use the AboveThreshold algorithm [Dwork and Roth, 2014] which is a key tool in differential privacy to identify whether there is a query $q_i : \mathcal{Z} \to \mathbb{R}$ in a stream of queries $q_1, \ldots, q_T$ that is above a certain threshold Δ. The AboveThreshold algorithm (presented in appendix) has the following guarantees.

Lemma 2.4 ([Dwork and Roth, 2014], Theorem 3.24). *AboveThreshold is $(\varepsilon, 0)$-DP. Moreover, let $\alpha = \frac{8 \log(2T/\gamma)}{\varepsilon}$ and $\mathcal{D} \in \mathcal{Z}^n$. For any sequence of T queries $q_1, \cdots, q_T : \mathcal{Z}^n \to \mathbb{R}$ each of sensitivity 1, AboveThreshold halts at time $k \in [T+1]$ such that with probability at least $1 - \gamma$,*

- *For all $t < k$, $a_t = \top$ and $q_t(\mathcal{D}) \ge \Delta - \alpha$;*
- *$a_k = \perp$ and $q_k(\mathcal{D}) \le \Delta + \alpha$ or $k = T + 1$.*

2.2 Randomized Smoothing

To develop optimal algorithms in the non-smooth setting, our algorithm use randomized smoothing [Yousefian et al., 2012, Duchi et al., 2012] to make the functions smooth. To this end, for a convex function $\ell(:; Z)$, we denote the convolution function $\widehat{\ell}(:; Z) := \ell(:; Z) * n_r.$, where n_r is the uniform density in the ℓ_2 ball of radius r centered at the origin in $\mathbb{R}^d$. Specifically, $n_r(y) = \frac{\Gamma(\frac{d}{2}+1)}{\pi^{\frac{d}{2}} r^d}$ for $\|y\| \le r$, and $n_r(y) = 0$ otherwise. For simplicity, we may omit the dependence on z, and write the function as $\widehat{\ell}$ and ℓ. Denote $\widehat{L}_{\mathcal{P}}(\theta) := \mathbb{E}_{z \sim P, y \sim n_r} \ell(\theta + y; z)$ and $\widehat{L}_{\mathcal{D}}(\theta) := \frac{1}{|\mathcal{D}|} \sum_{z \in \mathcal{D}} \mathbb{E}_{y \sim n_r} \ell(\theta + y; z)$.

Lemma 2.5 (Randomized Smoothing, [Yousefian et al., 2012, Duchi et al., 2012]). *The convolution function has the following properties:*

- *$\widehat{\ell}(\theta) \le \ell(\theta) \le \widehat{\ell}(\theta) + Gr$.*
- *$\widehat{\ell}$ is G-Lipschitz and convex.*
- *$\widehat{\ell}$ is $\frac{G\sqrt{d}}{r}$-smooth.*
- *For random variables $y \sim n_r$, and $z \in \mathcal{D}$, we have $\mathbb{E}_{y,z}[\nabla \ell(\theta + y; z)] = \nabla \widehat{L}_{\mathcal{D}}(\theta)$.*

2.3 Norm-Subgaussian Concentration

Our analysis also uses a notion named concentration properties for norm-Subgaussian random variables.

Definition 2.6 (norm-Subgaussian). A random vector $X \in \mathbb{R}^d$ is norm-SubGaussian with parameter σ, denoted $\mathrm{nSG}(\sigma)$, if for all $t \in \mathbb{R}$

$$\Pr[\|X - \mathbb{E} X\| \ge t] \le 2 \exp(-\frac{t^2}{2\sigma^2}).$$

The following concentration result holds for norm-Subgaussian random variables.

Lemma 2.7 ([Jin et al., 2019], concentration of NormSubgaussian). *There exists a constant $c > 0$, such that for zero-mean independent random vectors $X_1, \cdots, X_n \in \mathbb{R}^d$ where X_i is $\mathrm{nSG}(\sigma_i)$ for all $i \in [n]$, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\|\sum_{i \in [n]} X_i\| \le c \sqrt{\sum_{i \in [n]} \sigma_i^2 \log \frac{2d}{\delta}}.$$

3 Adaptive Mean Estimation for Concentrated Samples

The main component in our algorithms is a novel mean estimation procedure for adaptive queries for τ-concentrated samples where the samples lie in a ball of radius τ (see Definition 3.1). This algorithm will be used to estimate the gradients in our optimization procedure, as the user-level setting will guarantee that $\tau \approx 1/\sqrt{m}$ for an i.i.d. input. We add Gaussian noise scales with τ; hence, the final loss bound benefits from small τ.

Definition 3.1. A random samples $\{X_i\}_{i\in[n]}$ is (τ, γ)-concentrated if there exists a point $x \in \mathbb{R}^d$ such that with probability at least $1 - \gamma$,

$$\max_{i\in[n]} \|X_i - x\| \le \tau.$$

Given T adaptive mean estimation queries $q_1, \dots, q_T : (\mathcal{Z}^m)^n \to \mathbb{R}^d$ such that the n users are τ-concentrated with respect to these queries, our goal is to get a nearly unbiased estimate of the mean of each query with variance $\frac{\tau^2 Td}{n^2\varepsilon^2}$ under (ε, δ)-DP. The standard approach for solving this task, as done in [Bassily and Sun, 2023], is to assign a privacy budget $\varepsilon_i = \varepsilon/\sqrt{T}$ for each query, hence resulting in variance $\frac{\tau^2 T^2 d}{n^2\varepsilon^2}$. However, this procedure requires $n \ge \frac{1}{\varepsilon_i} = \sqrt{T}/\varepsilon$ to guarantee the desired utility bounds, which is too prohibitive for our purposes.

In this section, we design a new algorithm for adaptive mean estimation that achieves the desired variance with only $n \ge 1/\varepsilon$. Our algorithm is inspired by the FriendlyCore framework [Tsfadia et al., 2022], where we use the basic filter to identify outliers in the dataset. Our procedure consists of two stages: first, we apply an outlier-removal procedure, which returns a subset of the samples that is τ-concentrated. Then, we add Gaussian noise proportional to τ to privatize the mean of the concentrated sample.

To identify outliers, we give a score to each sample, which measures how many samples in the dataset are in a ball of size τ around the sample. As outlier samples will have a low score, we then keep each sample in the dataset with probability proportional to its score. This will preserve privacy for samples that are nearly τ-concentrated, whereas we aim to preserve privacy for all input datasets. Therefore, we add an initial check to the algorithm which verifies that the algorithm is nearly τ-concentrated. To this end, we define a τ-concentration score of the dataset for a query q_i to be

$$s_i^{\mathsf{conc}}(\mathcal{D}, \tau) := \frac{1}{n} \sum_{z\in\mathcal{D}} \sum_{z'\in\mathcal{D}} \mathbf{1}(\|q_i(z) - q_i(z')\| \le \tau). \tag{2}$$

and check via $\mathsf{AboveThreshold}$ that this score is above the desired threshold for all queries. The following procedure will be processed only if the dataset and the queries pass the check, which means our samples are nearly concentrated and ensures the privacy guarantee of the following procedure. We describe the full details of our algorithm in Algorithm 1.

Algorithm 1: Outlier-Removal Based Mean Estimation for Concentrated Data

```
Input: Dataset $\mathcal{D} = (Z_1, \dots, Z_n)$, privacy parameters $(\varepsilon, \delta)$, parameters $\tau$ ;
for $i = 1$ to $T$ do
    Receive a new mean estimation query $q_i : \mathcal{Z} \to \mathbb{R}^d$ ;
    Define concentration score
        $s_i^{\mathsf{conc}}(\mathcal{D}, \tau) := \frac{1}{n} \sum_{Z\in\mathcal{D}} \sum_{Z'\in\mathcal{D}} \mathbf{1}(\|q_i(Z) - q_i(Z')\| \le \tau)$
    if $\mathsf{AboveThreshold}(s_i^{\mathsf{conc}}, \varepsilon/2, 4n/5) = \top$ then
        Set $S_i = \emptyset$;
        for Each User $Z_j \in \mathcal{D}$ do
            Set $f_{i,j} = \sum_{Z\in\mathcal{D}} \mathbf{1}(\|q_i(Z_j) - q_i(Z)\| \le 2\tau)$;
            Add $Z_j$ to $S_i$ with probability $p_{i,j}$ for
                $p_{i,j} = \begin{cases} 0 & f_{i,j} < n/2 \\ 1 & f_{i,j} \ge 2n/3 \\ \frac{f_{i,j} - n/2}{n/6} & o.w. \end{cases}$
        end
        Let $g_i = \frac{1}{|S_i|} \sum_{Z\in S_i} q_i(Z)$ if $S_i$ is not empty, and $0$ otherwise ;
        Output: $\widehat{g}_i \leftarrow g_i + \nu_i$, where
        $\nu_i \sim \mathcal{N}(0, \frac{8\tau^2 T \log(e^{\varepsilon/2} T/\delta) \log(e^{\varepsilon/2}/\delta)}{n^2\varepsilon^2} I_d)$
    end
    else
        Output: $g_i = \mathbf{0}$;
        Halt;
    end
end
```

The following theorem summarizes the main guarantees of our algorithm.

Theorem 3.2. *For $0 < \varepsilon < 10, 0 < \delta < 1$. Let $\mathcal{D} = (Z_1, \dots, Z_n) \in (\mathcal{Z}^m)^n$ be a dataset with $n \geq \frac{8\log(T/\gamma)+8\log(T/\delta)}{\varepsilon}$ users. Algorithm 1 is (ε,δ)-DP. Moreover, if $(q_i(Z_1), \dots, q_i(Z_n))$ is (τ,γ)-concentrated for all $i \in [T]$, then there exists random variables $\widehat{g}'_1, \dots, \widehat{g}'_T$ such that the outputs $\widehat{g}_1, \dots, \widehat{g}_T$ of Algorithm 1 satisfy $\widehat{g}_i \sim_{2\gamma} \widehat{g}'_i$ for all $i \in [T]$. Moreover, $\widehat{g}'_i$ has*

$$\mathbb{E}\,\widehat{g}'_i = \frac{1}{n}\sum_{j=1}^{n} q_i(Z_j),$$

$$\mathbb{E}\left\|\widehat{g}'_i - \frac{1}{n}\sum_{j=1}^{n} q_i(Z_j)\right\|^2 \leq \frac{\tau^2 T\log(T/\delta)\log(1/\delta)}{n^2\varepsilon^2}.$$

To prove Theorem 3.2, we consider the privacy and utility guarantees separately. We argue about privacy first. The following are two technical lemmas.

Lemma 3.3. *For any neighboring dataset $\mathcal{D}, \mathcal{D}'$ that differs in one user, let $p_i = (p_{i,1}, \cdots, p_{i,n})$ be the probability for users to be selected into S_i for $\mathcal{D}$, and let p'_i be the corresponding probability for $\mathcal{D}'$. Then*

$$\|p_i - p'_i\|_1 \leq 2.$$

Lemma 3.4. *Let $p, p' \in [0,1]^n$ such that $\|p-p'\|_1 \leq 10$, and let V and V' be drawn from* $\mathrm{Ber}(p)$ *and* $\mathrm{Ber}(p')$ *respectively. For any $\zeta \in (0,1)$, there exists a coupling Γ over V and V' such that for (x,y) drawn from Γ, with probability at least $1-\zeta$,*

$$\|x-y\|_1 \leq O(\log(1/\zeta)).$$

Now, we analyze the privacy guarantee. We already know the AboveThreshold is private, and hence, it suffices to consider the following procedure when AboveThreshold always outputs "$\top$", which means the dataset is well concentrated with respect to the queries. We can show the sensitivity of the estimate g_i of the algorithm is bounded with the concentration, and hence, the privacy guarantee of the outputs $\{\widehat{g}_i\}$ can be proved via the property of Gaussian Mechanism.

Let $a_i \in \{\top, \bot\}$ be the output of AboveThreshold for i-th query. Let E be the event that for all $a_i = \top$, we have $q_i \geq \frac{4n}{5} - \alpha$ and for all $a_i = \bot$ we have $q_i \leq \frac{4n}{5} + \alpha$. Note that $\frac{4n}{5} - \alpha \geq \frac{2n}{3}$ by the value of α and the precondition that $n \geq \frac{40\log(2T/\zeta)}{\varepsilon}$. The guarantees of AboveThreshold (Lemma 2.4) also imply that the measure of E is at least $1-\zeta$. Define E' to be the event w.r.t. input $\mathcal{D}'$.

The following lemma upper bounds the sensitivity of the mean of the sub-sampled datasets.

Lemma 3.5. *For any i-th iteration and any neighboring datasets $\mathcal{D}, \mathcal{D}'$, conditional on E and E' and conditional on $a_i = a'_i$, there exists a coupling Γ_i over g_i and g'_i, such that for (x,y) drawn from Γ_i, with probability at least $1-\zeta$,*

$$\|x-y\|_2 \lesssim \frac{\tau\log(1/\zeta)}{n}.$$

Given the sensitivity bound of Lemma 3.5, we can argue for indistinguishability of the outputs using standard guarantees of the Gaussian mechanism.

Lemma 3.6. *For any dataset $\mathcal{D}$, if $n \geq \frac{40\log(4T/\delta)}{\varepsilon}$, then for any neighboring dataset $\mathcal{D}'$, the outputs of Algorithm 1 with $\mathcal{D}$ and $\mathcal{D}'$ as inputs are (ε,δ)-indistinguishable.*

Having estabilished the privacy guarantee of Algorithm 1, we now prove its utility. The following lemma shows that if the dataset is well concentrated with respect to the query, then no user will be removed in the outlier-removal stage with high probability, hence the estimate is nearly unbiased.

Lemma 3.7. *For all $i \in [T]$, if $(q_i(Z_1), \dots, q_i(Z_n))$ is (τ,γ)-concentrated and $n \geq \frac{8\log(T/\gamma)}{\varepsilon}$, then with probability at least $1-(T+1)\gamma$, it holds that $S_i = \mathcal{D}$ for all $i \in [T]$.*

Theorem 3.2 follows from Lemma 3.6 and Lemma 3.7.

4 Optimal Rates for User-Level DP-SCO

In this section, we present our main algorithm for user-level DP-SCO based on the gradient estimation procedure constructed above. Our algorithm leverages the Stochastic Gradient Descent (SGD) over a smoothed version of the loss function using randomized smoothing by applying the gradient estimation procedure to get (nearly) unbiased stochastic gradients. We present the full details of the algorithm in Algorithm 2.

Three key techniques are crucial for our algorithm and its analysis: first, for a fixed $\theta \in \Theta$, a simple concentration argument shows that the average gradient of each user will lie with high probability in a ball of small radius around the population gradient (see Lemma 4.4)

$$\|\nabla L(\theta; Z_i) - \nabla L_{\mathcal{P}}(\theta)\| \leq \frac{G\log(nd/\gamma)}{\sqrt{m}}.$$

This is not sufficient for our algorithms as we need this property to hold for data-dependent θ_t. To this end, similarly to [Bassily and Sun, 2023], we use the generalization properties of differential privacy to show in Lemma 4.6 that a similar concentration holds for $\nabla L(\theta_t; Z_i)$. Given this concentration, our mean estimation procedure (Algorithm 1) adds lower noise to estimate of the gradients.

Our second technique is based on the observation that smoothness is necessary to obtain the full potential of DP-SGD in user-level DP-SCO (similarly to existing work that used SGD-based algorithms for user-level DP-SCO [Levy et al., 2021, Bassily and Sun, 2023]). Convergence rates of SGD cause the limitation for non-smooth functions, which depend on the second moment of the gradients, whereas it depends on the variance for smooth functions (Proposition 4.10). As averaging the gradients of m samples reduces the variance while keeping the second moment the same, this yields better performance for smooth functions. To address this, we adopt randomized smoothing to smooth the loss functions and apply SGD over the smoothed functions. This is made possible due to our mean estimation procedure, which only requires $n \geq \log(mnd/\delta)/\varepsilon$, in contrast to prior work, which required $n \geq \sqrt{T}/\varepsilon$; this strict bound on the number of rounds is not sufficient to obtain optimal rates with randomized smoothing.

Finally, as we are using multi-pass SGD, an additional argument is needed to guarantee a low risk for population error. To this end, we analyze the stability of our algorithm for non-smooth functions using [Bassily et al., 2020], which implies that our algorithm has low generalization error.

Let $\Theta_r = \{\theta + y : \theta \in \Theta, \|y\| \leq r\}$. The following theorem summarizes our main result.

Theorem 4.1 (User-level DP-SCO). *Let $0 < \varepsilon < 10$ and $0 < \delta < 1$. Algorithm 2 is user-level (ε, δ)-DP. Setting $\widehat{R} = R, r = \frac{d^{1/4}\widehat{R}}{\sqrt{T}}, \eta = \frac{\widehat{R}}{G} \cdot \min\{\frac{\sqrt{mn\varepsilon}}{T\sqrt{d\log^2(mnd/\delta)}}, \frac{1}{T^{3/4}}, \frac{\sqrt{nm}}{T}\}$, $\tau = \frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $T = O(m^2n^2 + mn\sqrt{d})$, if $\Theta \subset \mathbb{R}^d$ is a convex set of diameter R, $\{\ell(:, z)\}_{z\in\mathcal{Z}}$ is a family of G-Lipschitz convex function over Θ_r, each item in $\mathcal{D}$ is drawn i.i.d. from the underlying distribution P, and $n \gtrsim \frac{\log(mdn/\delta)}{\varepsilon}$, then the output $\widehat{\theta}$ of Algorithm 2 satisfies*

$$\begin{aligned}&\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star\in\Theta} L_{\mathcal{P}}(\theta^\star)\right] \\ &\leq O\left(GR \cdot \left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d\log^2(ndm/\delta)}}{n\sqrt{m}\varepsilon}\right)\right).\end{aligned}$$

Algorithm 2: DP-SGD for user-level DP

```
Input: Dataset $\mathcal{D} = (Z_1, \dots, Z_n) \in (\mathcal{Z}^m)^n$, private parameters $(\varepsilon, \delta)$, initial point $\theta_0$, convolution parameter $r$, number of rounds $T$, stepsize $\eta$, concentration parameter $\tau$, initial distance $\widehat{R}$;
for $t = 1, \cdots, T$ do
    Define a query $q_t(Z) = \frac{1}{m}\sum_{j=1}^{m} \nabla\widehat{\ell}(\theta_t; z_{i,j})$ for $Z \in \mathcal{Z}^m$, See Equation (3) for the definition ;
    Run Algorithm 1 with query $q_t$ and parameters $\mathcal{D}, \varepsilon, \frac{\delta}{2Tmnd}, \tau$ ;
    Let $\overline{g}_t$ be the output of Algorithm 1 ;
    if $\overline{g}_t \neq \perp$ then
        Update $\theta_{t+1} \leftarrow \Pi(\theta_t - \eta\overline{g}_t)$;
    end
    else
        Output: Initial point $\theta_0$;
        Halt
    end
end
Return: $\widehat{\theta} = \frac{1}{T}\sum_{t\in[T]} \theta_t$
```

Remark 4.2. If we have a random initial point θ_0 such that $\mathbb{E}[\|\theta_0 - \theta^*\|^2] \leq R'^2$ for $\theta^* = \arg\min L_{\mathcal{D}}(\theta)$ and some $R' < R$, then we can replace the parameter setting $\widehat{R} = R$ by $\widehat{R} = R'$ in the population loss bound and the dependence on R can be reduced to R' in the loss bound.

Remark 4.3. We define the functions on Θ_r rather than Θ to make use of the randomized smoothing technique. As r is much smaller than R, this impact can be minimal. One can eliminate this domain extension by applying other smoothing techniques, such as the Moreau envelope smoothing method, but this method will increase the gradient computation cost.

We begin by showing that the gradients are concentrated. For any user Z_i who holds m items denoted by $\{z_{i,j}\}_{j\in[m]}$ and any point $\theta \in \Theta$, we denote

$$\nabla\widehat{\ell}(\theta; Z_i) := \frac{1}{m}\sum_{j\in[m]} \nabla\ell(\theta + y_j; z_{i,j}), \qquad (3)$$

the average stochastic gradients of all items owned by Z_i, where $y_j \sim n_r$ is drawn independently of θ and $z_{i,j}$ for the randomized smoothing.

Our goal is to eventually prove that $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$ are concentrated. To this end, we start with proving concentration for $\{\nabla\widehat{\ell}(\theta; Z_i)\}_{Z_i\in\mathcal{D}}$ for a fixed $\theta \in \Theta$.

Lemma 4.4. *For any fixed θ and for each Z_i, if each item in Z_i is drawn i.i.d. from $\mathcal{P}$, with probability at least $1-\gamma/n$, we have*

$$\|\nabla\widehat{\ell}(\theta;Z_i)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\leq\frac{G\log(nd/\gamma)}{\sqrt{m}},$$

One issue with applying Lemma 4.4 to demonstrate the concentration property of the stochastic gradients is that the dataset $\mathcal{D}$ and the points $\{\theta_i\}_{i\in[T]}$ are not independent. To tackle this, similarly to [Bassily and Sun, 2023], we make use of the generalization properties of private mechanisms. We need the following lemma.

Lemma 4.5 (Lemma 3.7 in [Feldman et al., 2022])**.** *Let* ALG *be an (ε,δ)-DP algorithm with respect the input $\mathcal{D}$. Then there exists an $(2\varepsilon,0)$-DP algorithm* ALG′*, such that*

$$d_{TV}(\mathrm{ALG}(\mathcal{D}),\mathrm{ALG}'(\mathcal{D}))\leq\delta.$$

Lemma 4.6 (Similar to Theorem 3.4 in [Bassily and Sun, 2023])**.** *Suppose $\mathcal{D}=\{z_{i,j}\}_{i\in[n],j\in[m]}$ are drawn i.i.d. from the distribution $\mathcal{P}$. In Algorithm 2, for all $t\in[T]$, $\{\nabla\widehat{\ell}(\theta_t;Z_i)\}_{Z_i\in\mathcal{D}}$ is (τ,γ')-concentrated for*

$$\tau=\frac{G\log(nd/\gamma)}{\sqrt{m}},\gamma'=T(e^{2\varepsilon}\gamma+\frac{\delta}{2Tmnd}).$$

Having established the concentration property of $\{\nabla\widehat{\ell}(\theta_t;Z_i)\}_{Z_i\in\mathcal{D}}$, we can bound the utility of our procedure for the empirical function $\widehat{L}_{\mathcal{D}}$. Now, we turn to prove the upper bounds for the generalization error, which needs the following well-known Lemma.

Lemma 4.7 ([Bousquet and Elisseeff, 2002])**.** *For an algorithm* ALG*, a dataset $\mathcal{D}=\{z_{i,j}\}_{i\in[n],j\in[m]}$ drawn i.i.d. from the distribution $\mathcal{P}$. If we replace one random data $z_{i,j}$ in $\mathcal{D}$ by a fresh new sample $z'_{i,j}$ from $\mathcal{P}$ and get the dataset $\mathcal{D}'$ and let* ALG($\mathcal{D}$) *be the (random) output of the algorithm, one has*

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathcal{D},\mathrm{ALG}}\left[L_{\mathcal{P}}(\mathrm{ALG}(\mathcal{D}))-L_{\mathcal{D}}(\mathrm{ALG}(\mathcal{D}))\right]\\&=\mathop{\mathbb{E}}_{\mathcal{D},z'_{i,j},\mathrm{ALG}}\left[\ell(\mathrm{ALG}(\mathcal{D});z'_{i,j})-\ell(\mathrm{ALG}(\mathcal{D}');z'_{i,j})\right].\end{aligned}$$

As we are considering Lipschitz functions, if we can bound the total variation distance between ALG($\mathcal{D}$) and ALG($\mathcal{D}'$) where $\mathcal{D}$ and $\mathcal{D}'$ differs from one single item, named by algorithmic stability, then we can bound the generalization error. Formally, we define the algorithmic stability of ALG as follows:

$$\Lambda(\mathrm{ALG}):=d_{TV}(\mathrm{ALG}(\mathcal{D}),\mathrm{ALG}(\mathcal{D}')),$$

where $d_{TV}(\mathrm{ALG}(\mathcal{D}),\mathrm{ALG}(\mathcal{D}'))$ denotes the total variation distance between ALG($\mathcal{D}$) and ALG($\mathcal{D}'$). Notably, the user-level differential privacy concerns replace m data of one user, while the algorithmic stability only concerns replacing one single item of a user. We have the following Lemma.

Lemma 4.8 (Lemma 3.1 in [Bassily et al., 2020])**.** *Let $(x^t)_{t\in[T]}$ and $(y^t)_{t\in[T]}$ be two trajectories of running SGD for G-Lipschitz convex function f, that is $x^t=\Pi(x^{t-1}-\eta\nabla f(x^{t-1}))$ and $y^t=\Pi(y^{t-1}-\eta\nabla f'(y^{t-1}))$. Suppose $\|\nabla f(x^t)-\nabla f'(x^t)\|\leq a_t\leq 2G$ for all $t\in[T]$, then*

$$\|x^T-y^T\|\leq 2G\sqrt{\sum_{t\in[T-1]}\eta_t^2}+2\sum_{t\in[T-1]}\eta_t a_t.$$

We use ALG to represent Algorithm 2. Then, we can bound the algorithmic stability of ALG based on the unbiased property of our mean estimate procedure (Lemma 3.7) constructed in the previous section.

Lemma 4.9 (Algorithmic stability bound)**.** *Suppose $\{Z_i\}$ are drawn i.i.d. from the underlying distribution $\mathcal{P}$. Suppose $\tau\geq\frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $n\gtrsim\frac{\log(mdn/\delta)}{\varepsilon}$, with probability at least $1-\frac{\delta}{mnd}$, the stability of Algorithm 2 is bounded as follows:*

$$\Lambda(\mathrm{ALG})\leq G\eta\sqrt{T}+\frac{G\eta T}{nm}.$$

Finally, to prove our main result, we need the following convergence rates for SGD.

Proposition 4.10 (SGD, [Bubeck, 2015])**.** *Consider a convex function f over a convex domain X. Suppose the random initial point x_0 satisfies $\mathbb{E}[\|x_0-x^*\|]\leq R^2$ where $x^*=\arg\min_{x\in X}f(x)$. Assume the unbiased stochastic oracle is such that $\mathbb{E}[\|\tilde{g}(x)\|^2]\leq\sigma^2$. Running gradient descent with step size η satisfies*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}f(x_{t+1})-\min_{x^*}f(x^*)\right]\leq\frac{R^2}{\eta T}+\eta\sigma^2.$$

Moreover, if the function f is β-smooth and the unbiased stochastic oracle is such that $\mathbb{E}[\|\tilde{g}(x)-\nabla f(x)\|^2]\leq\sigma^2$, then running SGD for T steps with step size η satisfies that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}f(x_{t+1})-\min_{x^*}f(x^*)\right]\leq(\beta+\frac{1}{\eta})\frac{R^2}{T}+\frac{\eta\sigma^2}{2}.$$

Combining these lemmas, we prove Theorem 4.1 in the Appendix.

4.1 Implication for Strongly convex functions

Building on our optimal algorithm for the convex setting, in the section, we proceed to obtain optimal rates for the strongly convex case using the localization framework [Feldman et al., 2020]. The idea is to iteratively run Algorithm 2 for $\log\log(mn)$ rounds, where at each round, we run it with improved parameters. We present the full details in the Appendix, and defer the full proof to the supplement with detailed parameter settings therein.

Algorithm 3: User-level DP-SCO for strongly convex functions

```
Input: Dataset $\mathcal{D} = (Z_1, \dots, Z_n) \in (\mathcal{Z}^m)^n$, privacy parameters $(\varepsilon, \delta)$, initial point $\theta_0$;
Set $k = \lceil \log\log mn \rceil$;
Divide $\mathcal{D}$ into $k$ disjoint datasets $\{\mathcal{D}_i\}_{i \in [k]}$, where $\mathcal{D}_i$ is of size $n_i := n/2^{k+1-i}$;
for $i = 1, \cdots, k$ do
    Run Algorithm 2 with $\mathcal{D}_i, \varepsilon, \delta, \theta_{i-1}, r_i, T_i, \eta_i, \tau_i, \widehat{R}_i$ as inputs, and get its output $\theta_i$;
end
Output: $\widehat{\theta} = \theta_k$;
```

Theorem 4.11 (Strongly convex case). *For $0 < \varepsilon < 10, 0 < \delta < 1$, Algorithm 3 is user-level (ε, δ)-DP. Under the same assumptions as in Theorem 4.1, additionally assuming that $n > \frac{\log(mdn)\log(mdn/\delta)}{\varepsilon}$ and the functions are μ-strongly convex, then with proper parameter settings, Algorithm 3 outputs $\widehat{\theta}$ such that*

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star \in \Theta} L_{\mathcal{P}}(\theta^\star)\right] \le O\left(\frac{G^2}{\mu}\left(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2 m \varepsilon^2}\right)\right).$$

5 Conclusion

In this work, we have studied user-level DP-SCO and proposed new efficient algorithms that obtain near-optimal rates even in the non-smooth setting. There remain open questions in this domain. First, our rates are optimal up to logarithmic factors and we leave it for future work to improve these factors. Moreover, our algorithms requires number of rounds $T \ge n^2m^2 \cdot \min(1, n^2/d)$, and it remains open whether there is a more efficient algorithm. In particular, are there linear time algorithms for user-level DP-SCO in the smooth setting, similar to the item-level setting where such results are known [Feldman et al., 2020]?

References

[Acharya et al., 2023] Acharya, J., Liu, Y., and Sun, Z. (2023). Discrete distribution estimation under user-level local differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 8561–8585. PMLR.

[Arora et al., 2023] Arora, R., Bassily, R., González, T., Guzmán, C. A., Menart, M., and Ullah, E. (2023). Faster rates of convergence to stationary points in differentially private optimization. In *International Conference on Machine Learning*, pages 1060–1092. PMLR.

[Asi et al., 2021a] Asi, H., Duchi, J., Fallah, A., Javidbakht, O., and Talwar, K. (2021a). Private adaptive gradient methods for convex optimization. In *Proceedings of the 38th International Conference on Machine Learning*, pages 383–392.

[Asi et al., 2021b] Asi, H., Feldman, V., Koren, T., and Talwar, K. (2021b). Private stochastic convex optimization: Optimal rates in ℓ_1 geometry. In *Proceedings of the 38th International Conference on Machine Learning*.

[Asi et al., 2021c] Asi, H., Levy, D., and Duchi, J. (2021c). Adapting to function difficulty and growth conditions in private optimization. In *Advances in Neural Information Processing Systems 34*, volume 34, pages 19069–19081.

[Badih et al., 2021] Badih, G., Ravi, K., and Pasin, M. (2021). User-level private learning via correlated sampling. *Advances in Neural Information Processing Systems*.

[Bassily et al., 2020] Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. (2020). Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33:4381–4391.

[Bassily et al., 2019] Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. (2019). Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, volume 32, pages 11282–11291.

[Bassily et al., 2021] Bassily, R., Guzman, C., and Nandi, A. (2021). Non-euclidean differentially private stochastic convex optimization. *arXiv:2103.01278 [cs.LG]*.

[Bassily et al., 2014] Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th Annual Symposium on Foundations of Computer Science*, pages 464–473.

[Bassily and Sun, 2023] Bassily, R. and Sun, Z. (2023). User-level private stochastic convex optimization with optimal rates.

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

[Bubeck, 2015] Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357.

[Bun and Steinke, 2016] Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

[Carmon et al., 2023] Carmon, Y., Jambulapati, A., Jin, Y., Lee, Y. T., Liu, D., Sidford, A., and Tian, K. (2023). Resqueing parallel and private stochastic convex optimization. *arXiv preprint arXiv:2301.00457*.

[Duchi et al., 2012] Duchi, J. C., Bartlett, P. L., and Wainwright, M. J. (2012). Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701.

[Dwork and Roth, 2014] Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

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

[Feldman et al., 2022] Feldman, V., McMillan, A., and Talwar, K. (2022). Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 954–964. IEEE.

[Ganesh et al., 2023] Ganesh, A., Liu, D., Oh, S., and Thakurta, A. (2023). Private (stochastic) non-convex optimization revisited: Second-order stationary points and excess risks. *arXiv:2302.09699 [cs.LG]*.

[Ghazi et al., 2023a] Ghazi, B., Kamath, P., Kumar, R., Manurangsi, P., Meka, R., and Zhang, C. (2023a). User-level differential privacy with few examples per user. *arXiv preprint arXiv:2309.12500*.

[Ghazi et al., 2023b] Ghazi, B., Kamath, P., Kumar, R., Meka, R., Manurangsi, P., and Zhang, C. (2023b). On user-level private convex optimization. *arXiv preprint arXiv:2305.04912*.

[Gopi et al., 2023] Gopi, S., Lee, Y. T., Liu, D., Shen, R., and Tian, K. (2023). Private convex optimization in general norms. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5068–5089. SIAM.

[Jin et al., 2019] Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*.

[Kulkarni et al., 2021] Kulkarni, J., Lee, Y. T., and Liu, D. (2021). Private non-smooth empirical risk minimization and stochastic convex optimization in subquadratic steps. *arXiv preprint arXiv:2103.15352*.

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

[Liu et al., 2020] Liu, Y., Suresh, A. T., Yu, F. X. X., Kumar, S., and Riley, M. (2020). Learning discrete distributions: user vs item-level privacy. *Advances in Neural Information Processing Systems*, 33:20965–20976.

[Lowy and Razaviyayn, 2023] Lowy, A. and Razaviyayn, M. (2023). Private stochastic optimization with large worst-case lipschitz parameter: Optimal rates for (non-smooth) convex losses and extension to non-convex losses. In *International Conference on Algorithmic Learning Theory*, pages 986–1054. PMLR.

[Tsfadia et al., 2022] Tsfadia, E., Cohen, E., Kaplan, H., Mansour, Y., and Stemmer, U. (2022). Friendlycore: Practical differentially private aggregation. In *International Conference on Machine Learning*, pages 21828–21863. PMLR.

[Yousefian et al., 2012] Yousefian, F., Nedić, A., and Shanbhag, U. V. (2012). On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica*, 48(1):56–67.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Preliminaries

Lemma A.1 (Chernoff-Hoeffding Bound)**.** *Let $X_1, \cdots, X_n$ be independent Bernoulli random variables such that $\mathbb{E}[X_i] = p_i$. Let $X = \sum_{i\in[n]} X_i$ and $\mu = \mathbb{E}[X]$. Then we know for any $\lambda > 0$, we have*

$$\Pr[X \geq (1+\lambda)\mu] \leq \exp(-\frac{\lambda^2\mu}{2+\lambda}).$$

A.1 AboveThreshold

The pseudocode of AboveThreshold can be found below for completeness.

Algorithm 4: AboveThreshold

```
Input: Dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$, threshold $\Delta \in \mathbb{R}$, privacy parameter $\varepsilon$;
Let $\widehat{\Delta} := \Delta - \mathrm{Lap}(\frac{2}{\varepsilon})$;
for $t = 1$ to $T$ do
    Receive a new query $q_t : \mathcal{Z}^n \to \mathbb{R}$ ;
    Sample $\nu_i \sim \mathrm{Lap}(\frac{4}{\varepsilon})$;
    if $q_t(\mathcal{D}) + \nu_i < \widehat{\Delta}$ then
        Output: $a_i = \perp$;
        Halt;
        else
            Output: $a_i = \top$;
        end
    end
end
```

B Missing Proofs in Section 3

B.1 Proof of Lemma 3.3

Lemma 3.3. *For any neighboring dataset $\mathcal{D}, \mathcal{D}'$ that differs in one user, let $p_i = (p_{i,1}, \cdots, p_{i,n})$ be the probability for users to be selected into S_i for $\mathcal{D}$, and let p_i' be the corresponding probability for $\mathcal{D}'$. Then*

$$\|p_i - p_i'\|_1 \leq 2.$$

Proof. Without loss of generality, let $\mathcal{D} = (Z_1, Z_2, \ldots, Z_n)$ and $\mathcal{D}' = (Z_1', Z_2, \ldots, Z_n)$ differ in the first user. Note that $f_{i,j}$ has sensitivity 1 for $j \neq 1$, hence $|p_{i,j} - p_{j,j}'| \leq 1/n$ for all $j \neq 1$. Moreover, $|p_{i,1} - p_{i,1}'| \leq 1$. Therefore, $\|p_i - p_i'\|_1 \leq 2$. □

B.2 Proof of Lemma 3.4

Lemma 3.4. *Let $p, p' \in [0,1]^n$ such that $\|p - p'\|_1 \leq 10$, and let V and V' be drawn from* $\mathrm{Ber}(p)$ *and* $\mathrm{Ber}(p')$ *respectively. For any $\zeta \in (0,1)$, there exists a coupling Γ over V and V' such that for (x,y) drawn from Γ, with probability at least $1-\zeta$,*

$$\|x - y\|_1 \leq O(\log(1/\zeta)).$$

Proof. We construct the coupling by considering each coordinate separately. Let p_i and p_i' be the i-th coordinate of p and p' respectively. Consider i-th coordinate, without losing generality, let $p_i \geq p_i'$. Then, we set

$$(x_i, y_i) = \begin{cases} (1,1), & \text{w.p. } p_i' \\ (1,0), & \text{w.p. } p_i - p_i' \\ (0,0), & \text{w.p. } 1-p_i \end{cases}$$

And coordinates are independent of each other. We draw (x,y) from the coupling Γ, and set $X_i = 1$ if $x_i = y_i$ and, $X_i = 0$ otherwise. Hence we know $\{X_i\}$ are independent Bernoulli random variables such that $\mathbb{E}[X_i] = |p_i - p_i'|$. By Lemma A.1, we know

$$\Pr[\|x-y\|_1 \geq O(\log(1/\zeta))] = \Pr[\sum_i X_i \geq O(\log(1/\zeta))] \leq \zeta.$$

This completes the proof. □

B.3 Proof of Lemma 3.5

Recall that the event E corresponds to the success of AboveThreshold in all time-steps for the dataset $\mathcal{D}$. Similarly, E' is defined for $\mathcal{D}'$. More precisely, let E be the event that for all $a_i = \top$, we have $q_i \geq \frac{4n}{5} - \alpha$ and for all $a_i = \bot$ we have $q_i \leq \frac{4n}{5} + \alpha$. Define E' to be the event w.r.t. input $\mathcal{D}'$.

Lemma 3.5. *For any i-th iteration and any neighboring datasets $\mathcal{D}, \mathcal{D}'$, conditional on E and E' and conditional on $a_i = a_i'$, there exists a coupling Γ_i over g_i and g_i', such that for (x,y) drawn from Γ_i, with probability at least $1-\zeta$,*

$$\|x-y\|_2 \lesssim \frac{\tau \log(1/\zeta)}{n}.$$

Proof. If $a_i = a_i' = \bot$, then both g_i and g_i' will be $\mathbf{0}$.

Consider the non-trivial case when $a_i = a_i' = \top$. As $s_i^{\mathsf{conc}}(\mathcal{D}, \tau) > \frac{2n}{3}$, we know there exists $Z^* \in \mathcal{D}$ such that $\sum_{Z \in \mathcal{D}} \mathbf{1}(\|q_i(Z^*) - q_i(Z)\| \leq \tau) \geq \frac{2n}{3}$. Let $H_i = \{Z \in \mathcal{D} : \|q_i(Z) - q_i(Z^*)\| \leq \tau\}$ be the set of users whose queried values are close to Z^*. We know $H_i \subset S_i$. Moreover, we can argue for any $Z \in S_i$, $\|q_i(Z) - q_i(Z^*)\| \leq 4\tau$. The same argument holds for $\mathcal{D}'$, that is there exists $Z'^* \in \mathcal{D}'$, such that $H_i' \subset S_i'$ and for any $Z \in S_i'$, $\|q_i(Z) - q_i(Z'^*)\| \leq 4\tau$.

We know $\|q_i(Z^*) - q_i(Z'^*)\| \leq 2\tau$, as there exists Z in $\mathcal{D} \cap \mathcal{D}'$ such that $\|q_i(Z^*) - q_i(Z)\| \leq \tau$ and $\|q_i(Z'^*) - q_i(Z)\| \leq \tau$. Hence for any point $Z_1, Z_2 \in S_i \cup S_i'$, $\|q_i(Z_1) - q_i(Z_2)\| \leq 10\tau$.

Note that $g_i = \frac{1}{|S_i|}\sum_{Z \in S_i} q_i(Z)$ and $g_i' = \frac{1}{|S_i'|}\sum_{Z \in S_i'} q_i(Z)$. By Lemma 3.3 and Lemma 3.4, we know there exists a Coupling Γ_i over S_i and S_i' such that if we draw (S, S') from Γ_i, with probability at least $1-\zeta$, we have

$$\|S - S'\|_0 \lesssim \log(1/\zeta).$$

Assume $|S'| \geq |S|$ without loss of generality and let $Z_0 \in S$. Note that we have

$$\begin{aligned}
\|g_i - g_i'\|_2 &= \left\| \frac{1}{|S|}\sum_{Z \in S} q_i(Z) - \frac{1}{|S'|}\sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|}\left\| \frac{|S'|}{|S|}\sum_{Z \in S} q_i(Z) - \sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|}\left\| \frac{|S'| - |S|}{|S|}\sum_{Z \in S} q_i(Z) + \sum_{Z \in S} q_i(Z) - \sum_{Z \in S'} q_i(Z) \right\|_2 \\
&= \frac{1}{|S'|}\left\| \frac{|S'| - |S|}{|S|}\sum_{Z \in S} q_i(Z) + \sum_{Z \in S \setminus S'} q_i(Z) - \sum_{Z \in S' \setminus S} q_i(Z) \right\|_2
\end{aligned}$$

$$
\begin{aligned}
&\le \frac{1}{|S'|}\left\| \frac{|S'|-|S|}{|S|}\sum_{Z\in S} q_i(Z) + \sum_{Z\in S\setminus S'} q_i(Z) - |S'\setminus S|\cdot q_i(Z_0)\right\|_2 + \frac{1}{|S'|}\left\| |S'\setminus S|\cdot q_i(Z_0) - \sum_{Z\in S'\setminus S} q_i(Z)\right\|_2 \\
&\overset{(i)}{=} \frac{1}{|S'|}\left\| \frac{|S'|-|S|}{|S|}\sum_{Z\in S} (q_i(Z)-q_i(Z_0)) + \sum_{Z\in S\setminus S'} (q_i(Z)-q_i(Z_0))\right\|_2 + \frac{1}{|S'|}\left\| \sum_{Z\in S'\setminus S} (q_i(Z_0)-q_i(Z))\right\|_2 \\
&\overset{(ii)}{\le} \frac{10\tau}{|S'|}\cdot ((|S'|-|S|) + |S\setminus S'| + |S'\setminus S|) \\
&\overset{(iii)}{\lesssim} \frac{\tau\log(1/\zeta)}{n}.
\end{aligned}
$$

where (i) follows since $|S'|-|S|+|S\setminus S'| = |S'\setminus S|$, and (ii) follows since $\max_{Z_1,Z_2\in S\cup S'}\|q_i(Z_1)-q_i(Z_2)\|_2 \le 10\tau$, and (iii) follows since $\|S-S'\|_0 \lesssim \log(1/\zeta)$ and hence $|S'|-|S|+|S'\setminus S|+|S\setminus S'| \lesssim \log(1/\zeta)$.

This completes the proof.

□

B.4 Proof of Lemma 3.6

Lemma 3.6. *For any dataset $\mathcal{D}$, if $n \ge \frac{40\log(4T/\delta)}{\varepsilon}$, then for any neighboring dataset $\mathcal{D}'$, the outputs of Algorithm 1 with $\mathcal{D}$ and $\mathcal{D}'$ as inputs are (ε,δ)-indistinguishable.*

Proof. Let $\{a_i\}_{i\in T} = \{\top,\bot\}^T$ be the outputs of Algorithm 4 with input $\mathcal{D}$, where if $a_i=\bot$ we set $a_j=\bot$ for all $j\ge i$. Define the $\{a_i'\}$ correspondingly with input $\mathcal{D}'$. Then by Theorem 2.4, we know $\{a_i\}$ and $\{a_i'\}$ are $(\varepsilon/2,0)$-indistinguishable.

Now conditional on that E and E' hold. If $a_i=a_i'=\bot$, then the algorithm halts and outputs the initial point, hence no privacy leakage.

Our proof proceeds by assuming the Gaussian noise v_i we add is drawn from $\mathcal{N}(0,\frac{4\tau^2 T\log(1/\zeta')\log(1/\delta')}{n^2\varepsilon^2})$. Then the statement follows from setting ζ' and δ'.

Under the assumption on $n\ge\frac{40\log(2T/\zeta')}{\varepsilon}$, for any $b\in\{\top,\bot\}^T$, by Lemma 3.5, the Union Bound, we know there exists a coupling over $\{g_i\}_{i\in T}$ and $\{g_i'\}_{i\in T}$, such that for $(\{x_i\},\{y_i\})$ drawn from Γ, with probability at least $1-T\zeta'$,

$$\text{for all } i\in[T], \|x_i-y_i\| \lesssim \frac{\tau\log(1/\zeta')}{n}.$$

By the guarantee of the Gaussian Mechanism and the composition [Bun and Steinke, 2016], we know

$$\Pr[\{g_i+\nu_i\}\in\mathcal{O}\mid E,\{a_i\}=b] \le e^{\varepsilon/2}\Pr[\{g_i'+\nu_i'\}\in\mathcal{O}\mid E',\{a_i'\}=b]+\delta'+T\zeta',$$

where we note that the Gaussian noise of $\{\nu_i\}$ and $\{\nu_i'\}$ are independent of the Laplacian noise we add in Algorithm 4.

To conclude, letting $\{g_i+\nu_i\}$ be the sequence of output, we have for any event $\mathcal{O}$,

$$
\begin{aligned}
\Pr[\{g_i+\nu_i\}\in\mathcal{O}] &= \Pr[\{g_i+\nu_i\}\in\mathcal{O}\mid E]\Pr[E] + \Pr[\{g_i+\nu_i\}\in\mathcal{O}\mid\neg E]\Pr[\neg E] \\
&\le \Pr[\{g_i+\nu_i\}\in\mathcal{O}\mid E]\Pr[E]+\zeta' \\
&= \sum_{b\in\{\top,\bot\}^T}\Pr[\{g_i+\nu_i\}\in\mathcal{O}\mid E,\{a_i\}=b]\Pr[E,\{a_i\}=b]+\zeta' \\
&\le \sum_{b\in\{\top,\bot\}^T} e^{\varepsilon/2}(\Pr[\{g_i'+\nu_i'\}\in\mathcal{O}\mid E',\{a_i'\}=b]+\delta')\Pr[E,\{a_i\}=b]+(T+1)\zeta'
\end{aligned}
$$

$$\leq \sum_{b\in\{\top,\bot\}^T} e^{\varepsilon/2}\Pr[\{g_i'+\nu_i'\}\in\mathcal{O}\mid E',\{a_i'\}=b]\Pr[E,\{a_i\}=b]+(T+1)\zeta'+e^{\varepsilon/2}\delta'.$$

Note that the randomness of $\{a_i\}$ and whether E holds comes from the Laplacian variables we draw. By the privacy guarantee of $\mathsf{AboveThreshold}$, for any $b\in\{\top,\bot\}^T$, we have

$$\Pr[\{a_i\}=b]\leq e^{\varepsilon/2}\Pr[\{a_i'\}=b].$$

It is not hard to observe that

$$\Pr[\{a_i\}=b,E]\leq e^{\varepsilon/2}\Pr[\{a_i'\}=b,E']+e^{\varepsilon/2}\zeta'.$$

Hence

$$\begin{aligned}&\Pr[\{g_i+\nu_i\}\in\mathcal{O}]\\ \leq&\sum_{b\in\{\top,\bot\}^T} e^{\varepsilon/2}\Pr[\{g_i'+\nu_i'\}\in\mathcal{O}\mid E',\{a_i'\}=b]\Pr[E,\{a_i\}=b]+(T+1)\zeta'+e^{\varepsilon/2}\delta'\\ \leq&\sum_{b\in\{\top,\bot\}^T} e^{\varepsilon}\Pr[\{g_i'+\nu_i'\}\in\mathcal{O}\mid E',\{a_i'\}=b]\Pr[E',\{a_i'\}=b]+(T+1+e^{\varepsilon})\zeta'+e^{\varepsilon/2}\delta'\end{aligned}$$

Setting $\zeta'=\frac{\delta}{2(e^{\varepsilon}+1+T)}$ and $\delta'=\frac{\delta}{2e^{\varepsilon/2}}$, we get the Noise scale as stated in the pseudo-code of Algorithm 1 and complete the proof. □

B.5 Proof of Lemma 3.7

Lemma 3.7. *For all $i\in[T]$, if $(q_i(Z_1),\ldots,q_i(Z_n))$ is (τ,γ)-concentrated and $n\geq\frac{8\log(T/\gamma)}{\varepsilon}$, then with probability at least $1-(T+1)\gamma$, it holds that $S_i=\mathcal{D}$ for all $i\in[T]$.*

Proof. To prove the lemma, we have to show that $\mathsf{AboveThreshold}$ will succeed (output $\top$) for each $i\in[T]$, and that the outlier-removal stage will not remove any item from the set.

To this end, fix any $i\in[T]$. Under the precondition that $(q_i(Z_1),\ldots,q_i(Z_n))$ is (τ,γ)-concentrated, we know that $s_i^{\mathsf{conc}}(\mathcal{D},\tau)=n$ with probability $1-\gamma$ for each $i\in[T]$. Moreover, the guarantees of $\mathsf{AboveThreshold}$ (Lemma 2.4) imply that it will output "$\top$" with probability at least $1-\gamma/T$ for each $i\in[T]$ when $s_i^{\mathsf{conc}}(\mathcal{D},\tau)=n$. Finally, under the event that $(q_i(Z_1),\ldots,q_i(Z_n))$ is τ-concentrated, we have that $f_{i,j}=n$ for each user $Z_j\in\mathcal{D}$, and hence Z_j will be added into S_i. The statement follows by applying a union bound. □

C Missing Proof in Section 4

C.1 Proof of Lemma 4.4

Lemma 4.4. *For any fixed θ and for each Z_i, if each item in Z_i is drawn i.i.d. from $\mathcal{P}$, with probability at least $1-\gamma/n$, we have*

$$\|\nabla\widehat{\ell}(\theta;Z_i)-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\leq\frac{G\log(nd/\gamma)}{\sqrt{m}},$$

Proof. The lemma follows from the concentration of Norm Subgaussian random variables (Lemma 2.7). Specifically, we know for each $z_{i,j}\in Z_i$, $\mathbb{E}\,\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)=0$, and $\|\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)\|\leq 2G$, which implies $\nabla\widehat{\ell}(\theta+y_j;z_{i,j})-\nabla\widehat{L}_{\mathcal{P}}(\theta)$ is zero-mean and nSG$(2G)$. The statement follows. □

C.2 Proof of Lemma 4.6

Lemma 4.6 (Similar to Theorem 3.4 in [Bassily and Sun, 2023])**.** *Suppose $\mathcal{D} = \{z_{i,j}\}_{i\in[n],j\in[m]}$ are drawn i.i.d. from the distribution $\mathcal{P}$. In Algorithm 2, for all $t \in [T]$, $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$ is (τ, γ')-concentrated for*

$$\tau = \frac{G\log(nd/\gamma)}{\sqrt{m}}, \gamma' = T(e^{2\varepsilon}\gamma + \frac{\delta}{2Tmnd}).$$

Proof. It suffices to prove that for each $t \in [T]$, $\{\nabla\widehat{\ell}(\theta_t; Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau, e^{2\varepsilon}\gamma + \frac{\delta}{2Tmnd})$-concentrated. Note that by Theorem 3.2 and the parameter settings in the precondition, Algorithm 2 is user-level $(\varepsilon, \frac{\delta}{2Tnmd})$-DP. Then there exists an $(2\varepsilon, 0)$-DP ALG′ such that $d_{TV}(\mathrm{ALG}(\mathcal{D}), \mathrm{ALG}'(\mathcal{D})) \le \delta/2Tmnd$ by Lemma 4.5. Let $\{\theta'_t\}_{t\in[T]}$ be the output of $\mathrm{ALG}'(\mathcal{D})$. It suffices to show for any $t \in [T], \{\nabla\widehat{\ell}(\theta'_t; Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau, e^{2\varepsilon}\gamma)$-concentrated.

Let $f_{Z_i}(Z)$ be the density of $Z_i = Z$ and and $f_{Z_i}(Z \mid \theta'_t = \theta)$ be the density conditional on $\theta'_t = \theta$. Similarly, we let $f_{\theta'_t}(\theta)$ and $f_{\theta'_t}(\theta \mid Z_i = Z)$ be the (conditional) density of θ'_t. For any θ, Z, we have

$$\frac{f_{Z_i}(Z \mid \theta'_t = \theta)}{f_{Z_i}(Z)} = \frac{f_{\theta'_t}(\theta \mid Z_i = Z)}{f_{\theta'_t}(\theta)} \le e^{2\varepsilon},$$

where the last inequality comes from the privacy guarantee of ALG′.

One has

$$\begin{aligned}&\Pr_{Z_i,\theta'_t}\left[\|\nabla\widehat{\ell}(\theta'_t; Z_i) - \nabla\widehat{L}_{\mathcal{P}}(\theta'_t)\| \ge \tau\right]\\ =&\int\int f_{\theta'_t}(\theta)f_{Z_i}(Z \mid \theta'_t = \theta)\mathbf{1}(\|\nabla\widehat{\ell}(\theta; Z) - \nabla\widehat{L}_{\mathcal{P}}(\theta)\| \ge \tau)\mathrm{d}Z\mathrm{d}\theta\\ \le& e^{2\varepsilon}\int\int f_{\theta'_t}(\theta)f_{Z_i}(Z)\mathbf{1}(\|\nabla\widehat{\ell}(\theta; Z) - \nabla\widehat{L}_{\mathcal{P}}(\theta)\| \ge \tau)\mathrm{d}Z\mathrm{d}\theta.\end{aligned}$$

Note that for any θ, we have

$$\int f_{Z_i}(Z)\mathbf{1}(\|\nabla\widehat{\ell}(\theta; Z) - \nabla\widehat{L}_{\mathcal{P}}(\theta)\| \ge \tau)\mathrm{d}Z \le \gamma/n.$$

Then by union bound, we know $\{\nabla\widehat{\ell}(\theta'_t; Z_i)\}_{Z_i\in\mathcal{D}}$ is $(\tau, e^{2\varepsilon}\gamma)$-concentrated which completes the proof as $d_{TV}(\mathrm{ALG}(\mathcal{D}), \mathrm{ALG}'(\mathcal{D})) \le \delta/2Tmnd.$ □

C.3 Proof of Lemma 4.9

Lemma 4.9 (Algorithmic stability bound)**.** *Suppose $\{Z_i\}$ are drawn i.i.d. from the underlying distribution $\mathcal{P}$. Suppose $\tau \ge \frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $n \gtrsim \frac{\log(mdn/\delta)}{\varepsilon}$, with probability at least $1 - \frac{\delta}{mnd}$, the stability of Algorithm 2 is bounded as follows:*

$$\Lambda(\mathrm{ALG}) \le G\eta\sqrt{T} + \frac{G\eta T}{nm}.$$

Proof. We use Lemma 4.8 to upper bound the stability of our algorithm. As we are using fixed step sizes $\eta_t = \eta$, Lemma 4.8 implies that

$$\begin{aligned}\Lambda(\mathrm{ALG}) &\le 2G\sqrt{\sum_{t\in[T-1]}\eta_t^2} + 2\sum_{t\in[T-1]}\eta_t a_t\\ &\le 2G\eta\sqrt{T} + 2\eta\sum_{t\in[T-1]}a_t\end{aligned}$$

Thus it suffices to upper bound a_t for all $t \in [T]$.

By Lemma 4.6, we know for all $t \in [T]$, $\{\nabla \widehat{\ell}(\theta_t; Z_i)\}_{Z_i \in \mathcal{D}}$ is (τ, γ')-concentrated for

$$\tau = \frac{G \log(nd/\gamma)}{\sqrt{m}}, \gamma' = T(e^{2\varepsilon}\gamma + \frac{\delta}{2Tmnd}).$$

Then by Theorem 3.2 and Lemma 3.7, we know

$$\overline{g}_t \sim_{2\gamma'} \frac{1}{nm} \sum_{Z_i \in \mathcal{D}} \sum_{z_{i,j} \in Z_i} \nabla \widehat{\ell}(\theta_t + y_j; z_{i,j}) + \nu,$$

where ν is Gaussian noise independent of the data. Thus we have $a_t \leq \frac{G}{nm}$. Setting $\gamma = \frac{\delta}{2Te^{2\varepsilon}}$ completes the proof. □

C.4 Proof of Theorem 4.1

Theorem 4.1 (User-level DP-SCO). *Let $0 < \varepsilon < 10$ and $0 < \delta < 1$. Algorithm 2 is user-level (ε, δ)-DP. Setting $\widehat{R} = R, r = \frac{d^{1/4}\widehat{R}}{\sqrt{T}}, \eta = \frac{\widehat{R}}{G} \cdot \min\{\frac{\sqrt{mn\varepsilon}}{T\sqrt{d\log^2(mnd/\delta)}}, \frac{1}{T^{3/4}}, \frac{\sqrt{nm}}{T}\}$, $\tau = \frac{G\log(ndme^{\varepsilon}T/\delta)}{\sqrt{m}}$ and $T = O(m^2n^2 + mn\sqrt{d})$, if $\Theta \subset \mathbb{R}^d$ is a convex set of diameter R, $\{\ell(:, z)\}_{z \in \mathcal{Z}}$ is a family of G-Lipschitz convex function over Θ_r, each item in $\mathcal{D}$ is drawn i.i.d. from the underlying distribution P, and $n \gtrsim \frac{\log(mdn/\delta)}{\varepsilon}$, then the output $\widehat{\theta}$ of Algorithm 2 satisfies*

$$\begin{aligned} &\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star \in \Theta} L_{\mathcal{P}}(\theta^\star)\right] \\ &\leq O\left(GR \cdot \left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d\log^2(ndm/\delta)}}{n\sqrt{m}\varepsilon}\right)\right). \end{aligned}$$

Proof. The privacy guarantee of Algorithm 2 follows from the privacy guarantee of our mean estimation procedure (Algorithm 1), as Algorithm 2 is post processing of the outputs of Algorithm 1.

Now, we prove utility. Let $\widehat{\theta} = \frac{1}{T}\sum_{t \in [T]} \theta_t$ denote the output of the algorithm. We upper bound the error by splitting it to two terms: one for generalization error and empirical error,

$$\begin{aligned} \mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)\right] &= \mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})\right] + \mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta \in \Theta} L_{\mathcal{D}}(\theta)\right] + \mathbb{E}\left[\min_{\theta \in \Theta} L_{\mathcal{D}}(\theta) - \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)\right] \\ &\leq \mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})\right] + \mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta \in \Theta} L_{\mathcal{D}}(\theta)\right]. \end{aligned} \tag{4}$$

where the second inequality holds since $\mathbb{E}[\min_{\theta \in \Theta} L_{\mathcal{D}}(\theta)] \leq \min_{\theta^* \in \Theta} L_{\mathcal{P}}(\theta^*)$.

For the empirical quantity (the second quantity in Equation (4)), first note that the error caused by randomized smoothing is Gr (Lemma 2.5), hence

$$\mathbb{E}\left[L_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta \in \Theta} L_{\mathcal{D}}(\theta)\right] \leq \mathbb{E}\left[\widehat{L}_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta \in \Theta} \widehat{L}_{\mathcal{D}}(\theta)\right] + 2Gr.$$

As our algorithm basically applies noisy SGD over $\widehat{L}_{\mathcal{D}}$, we now use Proposition 4.10 to bound the empirical error. By Lemma 3.7 and Theorem 3.2, we have

$$\overline{g}_t \sim_{\delta/Tnmd} \nabla \widehat{L}_{\mathcal{D}}(\theta_{t-1}) + \zeta,$$

where $\zeta \sim \mathcal{N}(0, \frac{G^2 T \log^2(Tmnd/\delta)}{mn^2\varepsilon^2})$. Hence we know the variance of the stochastic (sub)gradients we get is bounded by $O(\frac{G^2 Td\log^2(Tmnd/\delta)}{mn^2\varepsilon^2})$. Moreover, we know that $\widehat{\ell}$ is $\frac{G\sqrt{d}}{r}$-smooth by Lemma 2.5. Thus, Proposition 4.10 now implies that

$$\mathbb{E}[\widehat{L}_{\mathcal{D}}(\widehat{\theta}) - \min_{\theta} \widehat{L}_{\mathcal{D}}(\theta)] \lesssim \left(\frac{G\sqrt{d}}{r} + \frac{1}{\eta}\right)\frac{R^2}{T} + \frac{\eta G^2 T d \log^2(Tmnd/\delta)}{mn^2\varepsilon^2} + \frac{GR\delta}{mnd},$$

where the term $\frac{GR\delta}{mnd}$ comes from the failure probability.

Now we proceed to upper bound the generalization error (first quantity in Equation (4)). Combining Lemma 4.7 and Lemma 4.9, and the assumption that the functions are G-Lipschitz, we get

$$\mathbb{E}[L_{\mathcal{P}}(\widehat{\theta}) - L_{\mathcal{D}}(\widehat{\theta})] \le G^2\eta\sqrt{T} + \frac{G^2\eta T}{nm} + \frac{GR\delta}{mnd}.$$

Overall, combining these together and putting them back into Equation (4), we get

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim \frac{G\sqrt{d}R^2}{rT} + \frac{R^2}{\eta T} + \frac{\eta G^2 T d \log^2(Tmnd/\delta)}{mn^2\varepsilon^2} + Gr + G^2\eta\sqrt{T} + \frac{G^2\eta T}{nm} + \frac{GR\delta}{mnd}.$$

Optimizing the above parameters by setting $r = \frac{d^{1/4}R}{\sqrt{T}}$, $\eta = \frac{R}{G}\cdot\min\{\frac{\sqrt{m}n\varepsilon}{T\sqrt{d\log^2(Tmnd/\delta)}}, \frac{1}{T^{3/4}}, \sqrt{nm}/T\}$, we get

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim GR\cdot\left(\frac{d^{1/4}}{\sqrt{T}} + \frac{1}{T^{1/4}} + \frac{\sqrt{d\log^2(Tmnd/\delta)}}{\sqrt{m}n\varepsilon} + \frac{1}{\sqrt{nm}}\right).$$

By setting $T = O(m^2n^2 + mn\sqrt{d})$, we have

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)\right] \lesssim GR\cdot\left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d\log^2(ndm/\delta)}}{n\varepsilon\sqrt{m}}\right),$$

which completes the proof. □

C.5 Proof of Theorem 4.11

Theorem 4.11 (Strongly convex case)**.** *For $0 < \varepsilon < 10, 0 < \delta < 1$, Algorithm 3 is user-level (ε,δ)-DP. Under the same assumptions as in Theorem 4.1, additionally assuming that $n > \frac{\log(mdn)\log(mdn/\delta)}{\varepsilon}$ and the functions are μ-strongly convex, then with proper parameter settings, Algorithm 3 outputs $\widehat{\theta}$ such that*

$$\mathbb{E}\left[L_{\mathcal{P}}(\widehat{\theta}) - \min_{\theta^\star\in\Theta} L_{\mathcal{P}}(\theta^\star)\right] \\ \le O\left(\frac{G^2}{\mu}\left(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2m\varepsilon^2}\right)\right).$$

Proof. Let $L_{\mathcal{P}}^* = \min_{\theta^*\in\Theta} L_{\mathcal{P}}(\theta^*)$, $\Delta_i := \mathbb{E}[L_{\mathcal{P}}(\theta_i) - L_{\mathcal{P}}^*]$ and $R_i^2 := \mathbb{E}[\|\theta_i - \theta^*\|^2]$. Due to the strong convexity, we know $\frac{1}{2}\mu R_i^2 \le \Delta_i$.

Let $C > 2$ be the constant hidden in the population loss bound in Theorem 4.1. For $i \ge 0$, define $E_i := \frac{4C^2G^2}{\mu}(\frac{1}{n_i m} + \frac{d\log^2(n_i dm/\delta)}{n_i^2\varepsilon^2 m})$ and we know $E_i/E_{i+1} \le 4$. Define $D_i = 16E_i\sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{16E_0}}$. By the definition, we know

$$\frac{D_{i+1}}{16E_{i+1}} = \sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{16E_0}} \le \sqrt{\frac{D_i}{16E_i}},$$

$$\sqrt{D_iE_{i+1}} = 4\sqrt{E_iE_{i+1}}\sqrt[2^i]{\frac{2G^2}{\mu}\cdot\frac{1}{4E_1}} \le 16E_{i+1}\sqrt[2^{i+1}]{\frac{2G^2}{\mu}\cdot\frac{1}{4E_1}} = D_{i+1}.$$

Hence by setting $k \geq \log\log(D_1/(16E_1))$, then $\frac{D_k}{16E_k} \leq 2$. Note that $E_0 \geq \frac{4C^2G^2}{\mu nm}$, and $D_0 = \frac{2G^2}{\mu}$. We get $\frac{D_0}{16E_0} \leq mn$ and setting $k = \log\log(mn)$ is large enough to get $D_k \leq 32E_k$. Note that $\Delta_0 \leq \frac{2G^2}{\mu}$ and $R_0 \leq \frac{2G}{\mu}$ by the strong convexity and assumption on being Lipschitz.

For $j \geq 1$, set $\widehat{R}_j = \sqrt{2D_{j-1}/\mu}$, $r_j = \frac{d^{1/4}\widehat{R}_j}{\sqrt{T_j}}, \eta_j = \frac{\widehat{R}_j}{G} \cdot \min\{\frac{\sqrt{m}n_j\varepsilon}{T_j\sqrt{d\log^2(mn_jd/\delta)}}, \frac{1}{T_j^{3/4}}, \frac{\sqrt{n_jm}}{T_j}\}$, $\tau = \frac{G\log(n_jdme^{\varepsilon}T_j/\delta)}{\sqrt{m}}$ and $T_j = O(m^2n_j^2 + mn_j\sqrt{d})$. As $n_j \geq n/\log(nm) \geq \frac{\log(mdn/\delta)}{\varepsilon}$ by the precondition, $R_0 \leq \widehat{R}_1 = \frac{2G}{\mu}$, by Theorem 4.1 and our parameter setting, recursively we know

$$\begin{aligned}
\Delta_j \leq & CG\widehat{R}_j \cdot (\frac{1}{\sqrt{n_jm}} + \frac{\sqrt{d\log^2(n_jdm/\delta)}}{n_j\varepsilon\sqrt{m}}) \\
\leq & CG\sqrt{2D_{j-1}/\mu} \cdot (\frac{1}{\sqrt{n_jm}} + \frac{\sqrt{d\log^2(n_jdm/\delta)}}{n_j\varepsilon\sqrt{m}}) \\
\leq & CG\sqrt{2D_{j-1}/\mu} \cdot \sqrt{\frac{\mu E_j}{2C^2G^2}} \\
\leq & \sqrt{D_{j-1}E_j} \leq D_j,
\end{aligned}$$

where we used $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a+b)}$ for $a, b > 0$. We know $R_i \leq \sqrt{\frac{2\Delta_i}{\mu}} \leq \sqrt{\frac{2D_i}{\mu}} = \widehat{R}_{i+1}$ recursively as well. After k-iteration, we have

$$\mathbb{E}[L_{\mathcal{P}}(\theta_k) - L_{\mathcal{P}}^*] = \Delta_k \leq D_k \leq 32E_k = O(\frac{G^2}{\mu}(\frac{1}{nm} + \frac{d\log^2(ndm/\delta)}{n^2\varepsilon^2m})).$$

The statement follows. □