On User-Level Private Convex Optimization

Badih Ghazi [*1] **Pritish Kamath** [*1] **Ravi Kumar** [*1] **Raghu Meka** [*1 2] **Pasin Manurangsi** [*3] **Chiyuan Zhang** [*1]

Abstract

We introduce a new mechanism for stochastic convex optimization (SCO) with user-level differential privacy guarantees. The convergence rates of this mechanism are similar to those in the prior work of Levy et al. (2021); Narayanan et al. (2022), but with two important improvements. Our mechanism does not require any smoothness assumptions on the loss. Furthermore, our bounds are also the first where the minimum number of users needed for user-level privacy has no dependence on the dimension and only a logarithmic dependence on the desired excess error.

The main idea underlying the new mechanism is to show that the optimizers of strongly convex losses have low local deletion sensitivity, along with an output perturbation method for functions with low local deletion sensitivity, which could be of independent interest.

1. Introduction

Differential Privacy (DP) (Dwork et al., 2006b) is a formal notion that protects the privacy of each user contributing to a dataset when releasing statistics about the dataset. The settings considered in literature have typically involved each user contributing a single "item" to the dataset. Thus the most commonly used notion of DP protects the privacy of each item, and we refer to it as *item-level* DP. However, when a dataset contains multiple items contributed by each user, it is essential to simultaneously protect the privacy of all items contributed by any individual user; this notion has come to be known as *user-level* DP.

Convex optimization is one of the most basic and powerful computational tools in statistics and machine learning. In the most abstract setting, each item corresponds to a loss function. The goal is to return a value that achieves as small a loss as possible, either averaged over the data (empirical risk minimization) or the population distribution underlying the data (stochastic convex optimization).

Given its importance, a large body of work has tackled the convex optimization problem under item-level DP (e.g., Chaudhuri & Monteleoni (2008); Chaudhuri et al. (2011); Kifer et al. (2012); Bassily et al. (2014; 2019); Wang et al. (2017); Feldman et al. (2020); Asi et al. (2021); Gopi et al. (2022)) with the optimal risk bounds established in many standard settings, such as when the loss is Lipschitz or strongly convex. User-level DP has also been studied recently in various learning tasks (Liu et al., 2020; Ghazi et al., 2021); see also the survey by Kairouz et al. (2021, Section 4.3.2) for its relevance in federated learning, where the question of determining trade-offs between item-level and user-level DP is highlighted. Levy et al. (2021); Narayanan et al. (2022) have studied convex optimization with user-level DP; these results have two main drawbacks: they require the loss function to be smooth and they do not achieve good risk bounds in some regime of parameters. A question in Levy et al. (2021) was if the smoothness requirement can be removed. In this paper, we resolve this question in the affirmative by introducing novel mechanisms for convex optimization under user-level DP. En route, we also improve existing excess risk bounds for a large regime of parameters.

1.1. Background

We introduce some notation to state our results concretely. For $n, m \in \mathbb{N}$, suppose there are n users, and let the input to the ith user be $\boldsymbol{x}_i := (x_{i,1}, \ldots, x_{i,m})$. Two datasets $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and $\boldsymbol{x}' = (\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n)$ are said to be *user-level neighbors*, denoted $\boldsymbol{x} \sim \boldsymbol{x}'$, if there is an index $i_0 \in [n]$ such that $\boldsymbol{x}_i = \boldsymbol{x}'_i$ for all $i \in [n] \setminus \{i_0\}$.[1]

We recall the definition of DP, extended from Dwork et al. (2006a;b); see also Dwork & Roth (2014); Vadhan (2017):

Definition 1.1 ((User-Level) Differential Privacy (DP))**.** Let $\varepsilon > 0$ and $\delta \in [0, 1]$. A randomized algorithm $\mathcal{M} : \mathcal{X}^{n \times m} \to \mathcal{O}$ is (ε, δ)-*differentially private* ((ε, δ)-*DP*) if, for all $\boldsymbol{x} \sim \boldsymbol{x}'$ and all (measurable) outcomes $E \subseteq \mathcal{O}$, it

[1]Google Research, Mountain View, CA [2]UCLA, Los Angles, CA [3]Google Research, Thailand. Correspondence to: Pasin Manurangsi <pasin@google.com>, Pritish Kamath <pritish@alum.mit.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]We use *item-level* to refer to the case where $m = 1$.

holds that $\Pr[\mathcal{M}(\boldsymbol{x}) \in E] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(\boldsymbol{x}') \in E] + \delta$.

Throughout the paper, we assume that $\varepsilon \in (0, 1]$ and $\delta \in (0, 1/2]$, and we will not state this explicitly.

Convex Optimization. A convex optimization (CO) problem over a parameter space $\mathcal{K} \subseteq \mathbb{R}^d$ and domain $\mathcal{X}$, is specified by a *loss function* $\ell : \mathcal{K} \times \mathcal{X} \to \mathbb{R}$, that is convex in the first argument. Here, ℓ is said to be *G-Lipschitz* if all sub-gradients have norm at most[2] G, i.e., $\|\nabla_\theta \ell(\theta; x)\| \leq G$ for all θ, x. Moreover, ℓ is said to be *μ-strongly convex* if for all $x \in \mathcal{X}$, $\ell(\theta; x) - \frac{\mu}{2}\|\theta\|^2$ is convex. We consider the case where $\mathcal{K} \subseteq \mathbb{R}^d$ has ℓ_2-diameter at most R; we use $\mathcal{B}_d(\theta, r)$ to denote the ℓ_2 ball of radius r centered at θ.

The *empirical loss* on dataset $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is

$$\mathcal{L}(\theta; \boldsymbol{x}) \ := \ \tfrac{1}{nm} \textstyle\sum_{i\in[n]} \sum_{j\in[m]} \ell(\theta; x_{i,j}),$$

whereas the *population loss* over a distribution $\mathcal{D}$ on $\mathcal{X}$ is

$$\mathcal{L}(\theta; \mathcal{D}) := \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left[\ell(\theta; x)\right].$$

For a loss function ℓ and dataset $\boldsymbol{x}$, let $\theta^*_{\ell,\mathcal{K}}(\boldsymbol{x})$ denote an element of $\operatorname{argmin}_{\theta\in\mathcal{K}} \mathcal{L}(\theta; \boldsymbol{x})$ (ties broken arbitrarily), and let $\theta^*_{\ell,\mathcal{K}}(\mathcal{D})$ be defined similarly. When $\ell, \mathcal{K}$ are clear from context, we may drop them and simply write $\theta^*(\boldsymbol{x})$ or $\theta^*(\mathcal{D})$. When there is no ambiguity in $\boldsymbol{x}$ and $\mathcal{D}$, we may drop them and simply write θ^*. *Empirical risk minimization* (ERM) corresponds to the goal of minimizing $\mathcal{L}(\theta; \boldsymbol{x})$ and *stochastic convex optimization* (SCO) to the goal of minimizing $\mathcal{L}(\theta; \mathcal{D})$. If $\widehat{\theta}$ denotes the output of our algorithm, its *excess risk* is defined as $\mathbb{E}[\mathcal{L}(\widehat{\theta}; \boldsymbol{x}) - \mathcal{L}(\theta^*; \boldsymbol{x})]$ and $\mathbb{E}[\mathcal{L}(\widehat{\theta}; \mathcal{D}) - \mathcal{L}(\theta^*; \mathcal{D})]$ for ERM and SCO, respectively.

1.2. Our Results

We provide user-level DP algorithms for both the ERM as well as the SCO problems. For both problems, we consider the basic case of Lipschitz (including non-smooth) losses and the case of Lipschitz strongly convex losses.

DP-ERM. We give an algorithm for any Lipschitz and convex loss function with excess risk $O\left(\frac{\sqrt{d}}{n\sqrt{m}}\right)$ that works for any $n \geq \widetilde{\Omega}_\varepsilon(1)$. Previously, no user-level DP algorithm was known without a smoothness assumption on the loss function. Even with smoothness, the known algorithm of Narayanan et al. (2022) incurs an additional additive error of $\widetilde{O}_\varepsilon(1/\sqrt{n})$. In particular, the previous excess risk does *not* converge to zero if we fix the number of users (n) and let $m \to \infty$. Concretely, to achieve excess risk α, Narayanan et al. (2022) need $n \geq \widetilde{\Omega}_\varepsilon(1/\alpha^2)$. In contrast, we only need a logarithmic dependence of $n \geq \widetilde{\Omega}_\varepsilon(\log(1/\alpha))$. Additionally, for loss functions that are also strongly convex, we improve the excess risk bound to $\widetilde{O}_\varepsilon\left(\frac{d}{n^2m}\right)$. Again, no previous user-level DP algorithm was known in this setting (without the smoothness assumption).

DP-SCO. Here, we give algorithms with similar excess risk bounds except with additive terms of $\widetilde{O}\left(\frac{1}{\sqrt{nm}}\right)$ and $\widetilde{O}\left(\frac{1}{nm}\right)$ for the convex and strongly convex cases, respectively. These additive terms are known to be tight, even without privacy. Again, previous results (Levy et al., 2021; Narayanan et al., 2022) are only known under the smoothness assumption and the excess risk bounds do not converge to zero when n is fixed.

A summary of the previous and new bounds is in Table 1.

Tightness of our Risk Bounds. Our excess risk bounds are nearly tight for a large regime of parameters. In particular, Levy et al. (2021) proved a lower bound of $\Omega\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{\varepsilon n\sqrt{m}}\right)$ for DP-SCO. It is not hard to extend this to prove a lower bound of $\Omega\left(\frac{1}{nm} + \frac{d}{\varepsilon^2 n^2 m}\right)$ for the strongly convex DP-SCO case. These two lower bounds hold for any $n \geq \Theta(\sqrt{d}/\varepsilon)$. For DP-ERM, it is possible to extend these lower bounds to get $\Omega\left(\frac{\sqrt{d}}{\varepsilon n\sqrt{m}}\right)$ and $\Omega\left(\frac{d}{\varepsilon n^2 m}\right)$ lower bounds for the convex and strongly convex cases, respectively; however, these DP-ERM lower bounds require an additional assumption that $n = O(d/\varepsilon^2)$. We discuss these lower bounds in more detail in Appendix D.

2. Technical Overview

Our main technical contribution is an *improved output perturbation* algorithm for user-level DP compared to item-level DP. Recall that in item-level DP, the output perturbation algorithm (Chaudhuri et al., 2011) computes the empirical risk minimizer θ^* and outputs $\theta^* + Z$ where $Z \sim \mathcal{N}(\sigma^2 \cdot I)$ for a suitable σ; naturally, the smaller the σ for which DP guarantees hold, the better the accuracy. It is known that for *strongly convex* loss functions, this algorithm is DP for $\sigma = \widetilde{O}_\varepsilon\left(\frac{1}{n}\right)$. As discussed more below, we give a similar algorithm that only requires $\sigma = \widetilde{O}_\varepsilon(\frac{1}{n\sqrt{m}})$. This improvement is critical in our results.

Deletion Sensitivity. We exploit a refined notion of sensitivity to facilitate our improved output perturbation algorithm. Bounding the sensitivity of the quantity to be computed is one of the most used methods for achieving DP guarantees. Indeed, the DP guarantee of the output perturbation algorithm in (Chaudhuri et al., 2011) for item-level privacy follows from the fact that the (ℓ_2-)sensitivity of the empirical risk minimizer is $O\left(\frac{1}{n}\right)$ (Shalev-Shwartz et al., 2009). Formally,

$$\|\theta^*(\boldsymbol{x}) - \theta^*(\boldsymbol{x}')\| \leq O(1/n), \tag{1}$$

[2] We use $\|\cdot\|$ to denote the Euclidean, i.e., ℓ_2-norm.

	Additional Assumptions on ℓ	Item-Level DP (Previous Work)	User-Level DP (Previous Work)	User-Level DP (Our Results)
ERM	(no additional assumption)	$\widetilde{O}_\varepsilon\left(\frac{\sqrt{d}}{n}\right)$ (Bassily et al., 2014)	—	$\widetilde{O}_\varepsilon\left(\frac{\sqrt{d}}{n\sqrt{m}}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Theorem 4.1)
	Smooth		$\widetilde{O}_\varepsilon\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\sqrt{m}}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Narayanan et al., 2022)	
	Strongly Convex	$\widetilde{O}_\varepsilon\left(\frac{d}{n^2}\right)$ (Bassily et al., 2014)	—	$\widetilde{O}_\varepsilon\left(\frac{d}{n^2m}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Theorem 4.3)
	Strongly Convex & Smooth		$O_\varepsilon\left(\frac{d}{n^2m}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Narayanan et al., 2022)	
SCO	(no additional assumption)	$\widetilde{O}_\varepsilon\left(\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n}\right)$ (Bassily et al., 2019)	—	$\widetilde{O}_\varepsilon\left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Theorem 5.1)
	Smooth		$\widetilde{O}_\varepsilon\left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{n\sqrt{m}}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(\min\{\sqrt[3]{m}, \sqrt{m}/d\})$ (Narayanan et al., 2022)	
	Strongly Convex	$\widetilde{O}_\varepsilon\left(\frac{1}{n} + \frac{d}{n^2}\right)$ (Feldman et al., 2020)	—	$\widetilde{O}_\varepsilon\left(\frac{1}{nm} + \frac{d}{n^2m}\right)$ for $n \geq \widetilde{\Omega}_\varepsilon(1)$ (Theorem 5.3)

Table 1. Summary of our results and previous results. In all rows, the loss function is assumed to be convex and Lipschitz. The $\widetilde{O}_\varepsilon$ hides polynomial dependency on the convexity, Lipschitzness, strong convexity and smoothness parameters, ε, and polylogarithmic dependency on $1/\delta, n, m$. We remark that, while it seems plausible to derive bounds using their techniques, Levy et al. (2021); Narayanan et al. (2022) did not explicitly consider the strongly convex (and smooth) case for DP-SCO.

for any two neighboring datasets $\boldsymbol{x}, \boldsymbol{x}'$.

Ideally, we would like the "sensitivity" of θ^* to become $\widetilde{O}\left(\frac{1}{n\sqrt{m}}\right)$ for some notion of "sensitivity". However, the standard notion of sensitivity as above (or even local sensitivity (Nissim et al., 2007)) does not work: even for mean estimation[3], we can change a user to have all their input vectors far from the mean, resulting in the same $O(1/n)$ sensitivity as before. Instead, we use the notion of *deletion sensitivity*. Here, instead of considering $\boldsymbol{x}'$ that results from *changing* a user's data in $\boldsymbol{x}$, we only consider $\boldsymbol{x}'$ that results from *removing* a user's data completely.

Bounding Deletion Sensitivity of Empirical Risk Minimizer. We show that the (local) deletion sensitivity of $\theta^*(\boldsymbol{x})$ is at most $\widetilde{O}\left(\frac{1}{n\sqrt{m}}\right)$. To describe our approach, let us briefly recall the proof of (1) (item level setting, i.e., $m = 1$) from Shalev-Shwartz et al. (2009). The proof proceeds by bounding the norm of the gradient at $\theta^* := \theta^*(\boldsymbol{x})$ with respect to $\boldsymbol{x}'$ (i.e., $\|\nabla\mathcal{L}(\theta^*(\boldsymbol{x}); \boldsymbol{x}')\|$); strong convexity then implies that $\theta^*(\boldsymbol{x}')$ is close to $\theta^*(\boldsymbol{x})$. The gradient norm bound is based on the observation that $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) = 0$ due to optimality, and that $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) - \nabla\mathcal{L}(\theta^*; \boldsymbol{x}')$ is only $1/n$ times a difference of the gradients of *two input points* (that got changed from $\boldsymbol{x}$ to $\boldsymbol{x}'$). These two claims yield the desired $O(1/n)$ bound.

For the user-level setting, the situation is similar except that $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) - \nabla\mathcal{L}(\theta^*; \boldsymbol{x}')$ now becomes $O\left(\frac{1}{nm}\right)$ times the gradient of *all input points of a single user* (that got removed from $\boldsymbol{x}$ to $\boldsymbol{x}'$). An observation we use here is that in SCO—where all nm input points are drawn i.i.d.—we may view the input generation as a two-step process: (i) draw the nm input points, and (ii) randomly allocate these nm input points to n users. With this view in mind, $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) = 0$ means that the sum of the nm gradients is zero. The randomness in (ii) means that $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) - \nabla\mathcal{L}(\theta^*; \boldsymbol{x}')$ is now $O\left(\frac{1}{nm}\right)$ times the sum of m vectors *randomly chosen* from these nm vectors that sum to zero. Applying concentration inequalities (and a union bound), we can show that w.h.p. $\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) - \nabla\mathcal{L}(\theta^*; \boldsymbol{x}')\| \leq \widetilde{O}\left(\frac{1}{n\sqrt{m}}\right)$ as desired.

Noise Addition Algorithm for Deletion Sensitivity. Adding noise is still not trivial, even after bounding the (local) deletion sensitivity. As stated earlier, since we do not have the bound for the (standard) sensitivity, adding Gaussian noise directly to $\theta^*(\boldsymbol{x})$ will not yield the desired DP guarantee. To overcome this, we give an algorithm that adds noise w.r.t. the (local) deletion sensitivity. At a high-level, our algorithm has to perform a test to ensure that $\boldsymbol{x}$ is "sufficiently stable" (akin to

[3] This corresponds to $\ell(\theta; x) = \|\theta - x\|^2$ (here $x \in \mathbb{R}^d$) for which the empirical risk minimizer $\theta^*(\boldsymbol{x})$ is the average over all the input points.

propose-test-release (Dwork & Lei, 2009)) before adding Gaussian noise. Our algorithm is an adaptation of that of Kohli & Laskowsk (2021), which focuses on the real-valued case and adds Laplace noise.

From Output Perturbation to DP-SCO/DP-ERM. Finally, once we have the improved output perturbation algorithm, we use the localization-based algorithms (called Phased-SCO/Phased-ERM) of Feldman et al. (2020) with the enhanced output perturbation algorithm as subroutines to arrive at our results for DP-SCO/DP-ERM in the convex case. The strongly convex case follows from a known black-box reduction from Bassily et al. (2014).

Remark. Our algorithm for ERM guarantees an $\widetilde{O}_\varepsilon\left(\frac{\sqrt{d}}{n\sqrt{m}}\right)$ excess risk w.h.p. over the input being a random permutation of any given dataset $\boldsymbol{x}$. We emphasize that this is a mild assumption on the distribution of the dataset, and the same guarantees immediately follow for stronger assumptions such as the dataset $\boldsymbol{x}$ being drawn from any exchangeable distribution e.g. drawn i.i.d. from $\mathcal{D}$. Furthermore, we stress that it is impossible to have an excess risk bound for ERM that is better than $\widetilde{O}_\varepsilon(\sqrt{d}/n)$ for worst-case datasets since $x_{i,j}$ could be all the same for each i, which becomes essentially identical to the item-level setting with $m = 1$.

Comparison to Previous Work. Previous work (Levy et al., 2021; Narayanan et al., 2022) on user-level DP-SCO and DP-ERM tackle the problem using privatized first order methods (i.e., variants of gradient descent), sometimes with regularization. The main tool in these works is a user-level DP algorithm for mean estimation of vectors, which is used to aggregate the gradients with errors smaller than in the item-level setting. Such a result needs to rely on the fact that the average of the gradients of each user is well-concentrated; this can be interpreted as the average gradient having low deletion sensitivity. As discussed earlier, our result significantly generalizes this by showing that this also holds for the minimizer of any strongly convex function. Our algorithms also provide a novel paradigm of output perturbation for user-level DP learning—deviating from the first order methods explored in previous works.

In addition to the aforementioned work of Kohli & Laskowsk (2021), a notion similar to local deletion sensitivity has been studied in the context of DP graph analysis under the names of "empirical sensitivity" (Chen & Zhou, 2013) and "down sensitivity" (Raskhodnikova & Smith, 2016). Several mechanisms were developed using this notion, including an algorithm for monotonic real-valued functions (Chen & Zhou, 2013) and for many graph parameters. However, we are not aware of a generic algorithm for the high-dimensional case similar to our Algorithm 1.

3. Output Perturbation for Strongly Convex Losses

At the heart of our results is a new DP output perturbation algorithm (Algorithm 2) for strongly convex losses. The guarantee of this algorithm does not hold for any worst-case dataset, but instead holds for a *random permutation* of any given dataset. In particular, for any permutation π over $[n] \times [m]$, let $\boldsymbol{x}^\pi$ be the permutation of $\boldsymbol{x}$ by π, i.e., $x^\pi_{i,j} := x_{\pi(i,j)}$. As discussed in the previous section, this is a mild assumption but is required for our results.

Theorem 3.1. *Fix a G-Lipschitz and μ-strongly convex loss ℓ and a sufficiently large constant C. For all $\varepsilon, \delta, \beta > 0$ and $n \geq C\log(1/\delta)/\varepsilon$, there exists an (ε, δ)-DP algorithm* SCOutputPert*, such that for all $\boldsymbol{x}$, with probability $\geq 1 - \beta$ over a random permutation π over $[n] \times [m]$,* SCOutputPert$(\boldsymbol{x}^\pi)$ *outputs $\theta^*(\boldsymbol{x}^\pi) + \mathcal{N}(0, \sigma^2 \cdot I)$ where $\sigma = O\left(\frac{G\sqrt{\log n \log(1/\delta)/\varepsilon + \log(1/\beta)}}{\mu n\sqrt{m}} \cdot \frac{(\log(1/\delta))^{1.5}}{\varepsilon^2}\right)$.*

The expected ℓ_2-distance between the output estimate and the true minimizer thus scales as $\widetilde{O}_\varepsilon\left(\frac{\sqrt{d}}{n\sqrt{m}}\right)$. This should be compared with the item-level (i.e., $m = 1$) setting where the bound is $\widetilde{O}_\varepsilon(\sqrt{d}/n)$ (Chaudhuri & Monteleoni, 2008; Chaudhuri et al., 2011).

3.1. Deletion Sensitivity & A Generic Output Perturbation Algorithm

Before we can prove Theorem 3.1, we need to introduce the notion of local deletion sensitivity and present a generic output perturbation algorithm for low local deletion sensitivity functions and datasets. We stress that the algorithm in this section works for any such function and can be applied beyond the context of convex optimization.

Local Deletion Sensitivity. For any $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, let $\boldsymbol{x}_{-i}$, denote the dataset obtained by deleting the ith user's data $\boldsymbol{x}_i$ from $\boldsymbol{x}$. For any subset $S \subseteq [n]$, let $\boldsymbol{x}_{-S}$ denote the dataset obtained by deleting $\boldsymbol{x}_i$ from $\boldsymbol{x}$ for all $i \in S$.

Definition 3.2. The *local (ℓ_2-)deletion sensitivity* of function f at dataset $\boldsymbol{x}$ with n users is defined as $\overline{\Delta} f(\boldsymbol{x}) := \max_{i\in[n]} \|f(\boldsymbol{x}) - f(\boldsymbol{x}_{-i})\|$. Moreover, for $r \in \mathbb{N}$, let $\overline{\Delta}_r f(\boldsymbol{x}) := \max_{S\subseteq[n],|S|\leq r} \overline{\Delta} f(\boldsymbol{x}_{-S})$.

The difference between the usual definition of local sensitivity (Nissim et al., 2007) and that of local deletion sensitivity is that the latter definition only applies to *removal* of a user's data. This means that standard frameworks such as propose-test-release (Dwork & Lei, 2009) cannot be directly used here. We however show that this sensitivity notion still allows us to design an algorithm with small error on any dataset for which $\overline{\Delta}_r f(\boldsymbol{x})$ is small for any sufficiently large $r = \Theta(\log(1/\delta)/\varepsilon)$. The guarantee is given below.

Theorem 3.3. *Let $f : \mathcal{X}^{*\times m} \to \mathbb{R}^d$, and $\Delta > 0$ be a predefined parameter. There exists an (ε, δ)-DP algorithm that either outputs $\perp$ or a vector in $\mathbb{R}^d$. Furthermore, there exists $r = O(\log(1/\delta)/\varepsilon)$ such that, on input dataset $\boldsymbol{x}$ that satisfies $\overline{\Delta}_r f(\boldsymbol{x}) \le \Delta$, it never returns $\perp$ and simply returns $f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \cdot I)$ where $\sigma = O\left(\Delta \cdot \frac{(\log(1/\delta))^{1.5}}{\varepsilon^2}\right)$.*

The general idea is to find a "sufficiently stable" dataset $\hat{\boldsymbol{x}}$ and add noise to $f(\hat{\boldsymbol{x}})$. Although we may wish to just set $\hat{\boldsymbol{x}} = \boldsymbol{x}$ directly and check that the local sensitivity is small, we cannot do this, as changing a single datapoint can change whether we pass the test. Therefore, similar to the propose-test-release framework, we check for $\boldsymbol{x}_{-S}$ for all subsets S with $|S| \le R_1$ where R_1 is a shifted truncated discrete Laplace random variable, as defined below. This allows us to maintain the closeness of acceptance probability across neighboring input datasets. The full description is given in Algorithm 1. As stated earlier, our algorithm is a modification of that of Kohli & Laskowsk (2021), which uses Laplace noise and a different distribution of R_1.

Definition 3.4 (Shifted Truncated Discrete Laplace Distribution)**.** For any $\varepsilon, \delta > 0$, let $\kappa = \kappa(\varepsilon, \delta) := 1 + \lceil \ln(1/\delta)/\varepsilon \rceil$ and let $\mathsf{TDLap}(\varepsilon, \delta)$ be the distribution supported on $\{0, \dots, 2\kappa\}$ with probability mass function at x being proportional to $\exp(-\varepsilon \cdot |x - \kappa|)$.

Algorithm 1 $\mathsf{DelOutputPert}_{\varepsilon,\delta,\Delta}(f; \boldsymbol{x})$

1: **Input:** Dataset $\boldsymbol{x}$, function $f : \mathcal{X}^{*\times m} \to \mathbb{R}^d$
2: **Parameters:** Privacy parameters ε, δ; Target deletion sensitivity parameter Δ
3: $\overline{\varepsilon} \leftarrow \frac{\varepsilon}{2}, \overline{\delta} \leftarrow \frac{\delta}{e^{\overline{\varepsilon}}+2}, \kappa \leftarrow \kappa(\overline{\varepsilon}, \overline{\delta}), \sigma \leftarrow \frac{2\sqrt{\ln(2/\overline{\delta})}(8\kappa\Delta)}{\overline{\varepsilon}}$
4: Sample $R_1 \sim \mathsf{TDLap}(\overline{\varepsilon}, \overline{\delta})$ ▷ See Definition 3.4
5: $\mathcal{X}^{R_1}_{\text{stable}} \leftarrow \left\{\boldsymbol{x}_{-S} : |S| \le R_1, \overline{\Delta}_{4\kappa-|S|} f(\boldsymbol{x}_{-S}) \le \Delta\right\}$
6: **if** $|\mathcal{X}^{R_1}_{\text{stable}}| = \emptyset$ **then**
7: **return** $\perp$
8: **end if**
9: Choose $\boldsymbol{x}_{-S} \in \mathcal{X}^{R_1}_{\text{stable}}$ with smallest $|S|$ ▷ Ties broken arbitrarily
10: **return** $f(\boldsymbol{x}_{-S}) + \mathcal{N}(0, \sigma^2 \cdot I)$

To prove Theorem 3.3, the following observation is useful.

Observation 3.5. *For neighboring datasets $\boldsymbol{x}, \boldsymbol{x}'$, and all $r_1 \in \mathbb{Z}_{\ge 0}$, if $\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}') \neq \emptyset$, then $\mathcal{X}^{r_1+1}_{\text{stable}}(\boldsymbol{x}) \neq \emptyset$.*

Proof. Suppose $\boldsymbol{x}' = (\boldsymbol{x}'_1, \dots, \boldsymbol{x}'_n)$. Let $\boldsymbol{x}'_{-S'}$ be an element of $\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}')$. That is, we have $|S'| \le r_1$ and $\overline{\Delta}_{r_2} f(\boldsymbol{x}'_{-S'}) \le \Delta$ for $r_2 = 4\kappa - r_1$. Let $i \in [n]$ denote the user on which $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ. We consider two cases: If $i \in S'$, then we simply have $\boldsymbol{x}_{-S'} = \boldsymbol{x}'_{-S'}$ and therefore $\boldsymbol{x}_{-S'}$ also belongs to $\mathcal{X}^{r_1+1}_{\text{stable}}(\boldsymbol{x})$. If $i \notin S'$, let $S = S' \cup \{i\}$. We have $|S| \le r_1 + 1$ and $\overline{\Delta}_{r_2-1}(\boldsymbol{x}_{-S}) \le \overline{\Delta}_{r_2}(\boldsymbol{x}'_{-S'})$. This means that $\boldsymbol{x}_{-S} \in \mathcal{X}^{r_1+1}_{\text{stable}}(\boldsymbol{x})$ as well. □

Proof of Theorem 3.3. Let $\mathcal{A}$ be the DelOutputPert algorithm (Algorithm 1).

Privacy Analysis. Let $\boldsymbol{x}, \boldsymbol{x}'$ be neighboring datasets. First,

$$\begin{aligned}
&\Pr[\mathcal{A}(\boldsymbol{x}) = \perp] \\
&= \textstyle\sum_{r_1=0}^{2\kappa} \mathbf{1}[\mathcal{A}(\boldsymbol{x}) = \perp \mid R_1 = r_1] \cdot \Pr[R_1 = r_1] \\
&= \textstyle\sum_{r_1=0}^{2\kappa} \mathbf{1}[\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}) = \emptyset] \cdot \Pr[R_1 = r_1] \\
&\le \Pr[R_1 = 0] + \textstyle\sum_{r_1=1}^{2\kappa} \mathbf{1}[\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}) = \emptyset] \cdot \Pr[R_1 = r_1] \\
&\le \overline{\delta} + \textstyle\sum_{r_1=1}^{2\kappa} \mathbf{1}[\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}) = \emptyset] \cdot e^{\overline{\varepsilon}} \cdot \Pr[R_1 = r_1 - 1] \\
&\le \overline{\delta} + \textstyle\sum_{r_1=1}^{2\kappa} \mathbf{1}[\mathcal{X}^{r_1-1}_{\text{stable}}(\boldsymbol{x}') = \emptyset] \cdot e^{\overline{\varepsilon}} \cdot \Pr[R_1 = r_1 - 1] \\
&\le \overline{\delta} + e^{\overline{\varepsilon}} \cdot \textstyle\sum_{r_1=0}^{2\kappa} \mathbf{1}[\mathcal{A}(\boldsymbol{x}') = \perp \mid R_1 = r_1] \cdot \Pr[R_1 = r_1] \\
&= \overline{\delta} + e^{\overline{\varepsilon}} \cdot \Pr[\mathcal{A}(\boldsymbol{x}') = \perp]. \qquad (2)
\end{aligned}$$

Next, consider any set $S_0 \subseteq \mathbb{R}^d$. We have

$$\begin{aligned}
&\Pr[\mathcal{A}(\boldsymbol{x}) \in S_0] \\
&\le \textstyle\sum_{r_1=0}^{2\kappa} \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1] \cdot \Pr[R_1 = r_1] \\
&= \textstyle\sum_{r_1=0}^{2\kappa-1} \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1] \cdot \Pr[R_1 = r_1] \\
&\qquad + \Pr[R_1 = 2\kappa] \\
&\le \overline{\delta} + e^{\overline{\varepsilon}} \cdot \textstyle\sum_{r_1=0}^{2\kappa-1} \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1] \\
&\qquad\qquad \cdot \Pr[R_1 = r_1 + 1], \qquad (3)
\end{aligned}$$

where the last inequality follows since $R_1 \sim \mathsf{TDLap}(\overline{\varepsilon}, \overline{\delta})$. To bound the term $\Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1]$ for $r_1 < 2\kappa$, observe that if it is non-zero, then it must be that $\mathcal{A}(\boldsymbol{x}) \neq \perp$ or equivalently that $\mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x}) \neq \emptyset$; Observation 3.5 then implies that $\mathcal{X}^{r_1+1}_{\text{stable}}(\boldsymbol{x}) \neq \emptyset$, or equivalently, $\mathcal{A}(\boldsymbol{x}') \neq \perp$. Let $\hat{\boldsymbol{x}}_{r_1}$ and $\hat{\boldsymbol{x}}'_{r_1+1}$ be the sets chosen on Line 9 when we run the algorithm on input $\hat{\boldsymbol{x}}, R_1 = r_1$ and $\hat{\boldsymbol{x}}', R_1 = r_1 + 1$ respectively. Let $\boldsymbol{x}^* := \hat{\boldsymbol{x}}_{r_1} \cap \hat{\boldsymbol{x}}'_{r_1+1}$. We have $|\boldsymbol{x}^*| \ge |\boldsymbol{x}| - r_1 - (r_1+1) \ge |\boldsymbol{x}| - 4\kappa$. Therefore, since $\hat{\boldsymbol{x}}_{r_1} \in \mathcal{X}^{r_1}_{\text{stable}}(\boldsymbol{x})$ and $\hat{\boldsymbol{x}}'_{r_1+1} \in \mathcal{X}^{r_1+1}_{\text{stable}}(\boldsymbol{x}')$, we must have

$$\|f(\hat{\boldsymbol{x}}_{r_1}) - f(\boldsymbol{x}^*)\| \le \Delta \cdot |\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}'| \le 4\kappa \cdot \Delta,$$

and similarly,

$$\|f(\hat{\boldsymbol{x}}'_{r_1+1}) - f(\boldsymbol{x}^*)\| \le \Delta \cdot |\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}'| \le 4\kappa \cdot \Delta.$$

Combining the two, we can conclude that

$$\|f(\hat{\boldsymbol{x}}_{r_1}) - f(\hat{\boldsymbol{x}}'_{r_1+1})\| \le 8\kappa \cdot \Delta.$$

From the privacy guarantee of the Gaussian mechanism (e.g., Dwork & Roth (2014, Appendix A)), we have

$$\begin{aligned}
&\Pr[f(\hat{\boldsymbol{x}}_{r_1}) + \mathcal{N}(\sigma^2 \cdot I) \in S_0] \\
&\le e^{\overline{\varepsilon}} \cdot \Pr[f(\hat{\boldsymbol{x}}'_{r_1+1}) + \mathcal{N}(\sigma^2 \cdot I) \in S_0] + \overline{\delta}.
\end{aligned}$$

Note that $\Pr[f(\hat{\boldsymbol{x}}_{r_1}) + \mathcal{N}(\sigma^2 \cdot I) \in S_0] = \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1]$, while $\Pr[f(\hat{\boldsymbol{x}}'_{r_1+1}) + \mathcal{N}(\sigma^2 \cdot I) \in S_0] = \Pr[\mathcal{A}(\boldsymbol{x}') \in S_0 \mid R_1 = r_1 + 1]$.

Plugging this back to (3), we get

$$\begin{aligned}
&\Pr[\mathcal{A}(\boldsymbol{x}) \in S_0] \\
&\le \overline{\delta} + e^{\overline{\varepsilon}} \cdot \textstyle\sum_{r_1=0}^{2\kappa-1} \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0 \mid R_1 = r_1] \\
&\qquad\qquad \cdot \Pr[R_1 = r_1 + 1] \\
&= \overline{\delta} + e^{\overline{\varepsilon}} \cdot \textstyle\sum_{r_1=0}^{2\kappa-1} \left(e^{\overline{\varepsilon}} \cdot \Pr[\mathcal{A}(\boldsymbol{x}') \in S_0 \mid R_1 = r_1 + 1] + \overline{\delta}\right) \\
&\qquad\qquad \cdot \Pr[R_1 = r_1 + 1] \\
&\le (e^{\overline{\varepsilon}} + 1)\overline{\delta} + e^{2\overline{\varepsilon}} \cdot \textstyle\sum_{r_1=0}^{2\kappa} \Pr[\mathcal{A}(\boldsymbol{x}') \in S_0 \mid R_1 = r_1] \\
&\qquad\qquad \cdot \Pr[R_1 = r_1] \\
&= (e^{\overline{\varepsilon}} + 1)\overline{\delta} + e^{2\overline{\varepsilon}} \cdot \Pr[\boldsymbol{A}(\boldsymbol{x}') \in S_0]. \qquad (4)
\end{aligned}$$

Now, consider any set S of outcomes. Let $S_0 = S \cap \mathbb{R}^d$ and $S_\perp = S \cap \{\perp\}$. Then, we have

$$\begin{aligned}
&\Pr[\mathcal{A}(\boldsymbol{x}) \in S] = \Pr[\mathcal{A}(\boldsymbol{x}) \in S_0] + \Pr[\mathcal{A}(\boldsymbol{x}) \in S_\perp] \\
&\overset{(4),(2)}{\le} \left((e^{\overline{\varepsilon}} + 1)\overline{\delta} + e^{2\overline{\varepsilon}} \cdot \Pr[\boldsymbol{A}(\boldsymbol{x}') \in S_0]\right) \\
&\qquad + \left(\overline{\delta} + e^{\overline{\varepsilon}} \cdot \Pr[\mathcal{A}(\boldsymbol{x}') = S_\perp]\right) \\
&\le (e^{\overline{\varepsilon}} + 2)\overline{\delta} + e^{2\overline{\varepsilon}} \Pr[\mathcal{A}(\boldsymbol{x}') \in S] \\
&\le \delta + e^{\varepsilon} \cdot \Pr[\mathcal{A}(\boldsymbol{x}') \in S].
\end{aligned}$$

Therefore, the algorithm is (ε, δ)-DP as desired.

Accuracy Analysis. Let $\boldsymbol{x}$ be any dataset such that $\overline{\Delta}_{4\kappa}\, \boldsymbol{x} \le \Delta$. This means that, for any $0 \le R_1 \le 4\kappa$, $\overline{\Delta}_{4\kappa - R_1}\, \boldsymbol{x} \le \Delta$. In other words, $\boldsymbol{x}$ belongs to $\mathcal{X}^{R_1}_{\text{stable}}$. Thus, we always have $\hat{\boldsymbol{x}} = \boldsymbol{x}$ and the output is simply drawn from $f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \cdot I)$ as claimed. □

3.2. Deletion Sensitivity of Optimizers of Strongly Convex Losses

Having provided a generic noising algorithm for functions with low local deletion sensitivity, the next step is to show that the function that we care about for convex optimization—the empirical risk minimizer—has low deletion sensitivity (with high probability), as formalized below.

Theorem 3.6. *Let ℓ be any μ-strongly convex loss function such that $\|\nabla \ell(\theta; x)\| \le G$ for all $\theta \in \mathcal{K}, x \in \mathcal{X}$. For all $\boldsymbol{x} \in \mathcal{X}^{n\times m}$ and $\beta < 1/e$, with probability $1 - \beta$ over the choice of a random permutation π over $[n] \times [m]$, we have*

$$\|\theta^*(\boldsymbol{x}^\pi) - \theta^*(\boldsymbol{x}^\pi_{-n})\| \;\le\; \tfrac{5G\sqrt{\log(1/\beta)}}{\mu(n-1)\sqrt{m}}.$$

Before proving this, we note that by applying a union bound over all the n users and all subsets S of size at most r, we arrive at Corollary 3.7. Theorem 3.1 now follows by defining SCOutputPert (Algorithm 2) that invokes DelOutputPert on the function f being the empirical loss, and combining Corollary 3.7 with Theorem 3.3 (setting $r = 4\kappa$).

Algorithm 2 $\mathsf{SCOutputPert}_{\varepsilon,\delta,\beta,G,\mu,\mathcal{K}}(\ell; \boldsymbol{x})$

1: **Input:** Dataset $\boldsymbol{x}$, loss function $\ell : \mathcal{K} \times \mathcal{X} \to \mathbb{R}$
2: **Parameters:** Privacy parameters ε, δ; Target failure probability β; Lipschitz parameter G; Strong convexity μ
3: $\Delta \leftarrow \frac{10G\sqrt{\log(1/\beta)}}{\mu n\sqrt{m}}$
4: **return** $\mathsf{DelOutputPert}_{\varepsilon,\delta,\Delta}(f; \boldsymbol{x})$, where $f(\cdot) := \operatorname{argmin}_\theta \mathcal{L}(\theta; \cdot)$

Corollary 3.7. *Let ℓ be any G-Lipschitz, μ-strongly convex loss. For all $\boldsymbol{x} \in \mathcal{X}^{n\times m}$ and $r \le n/2$, with probability $1 - \beta$ over the choice of a random permutation π over $[n] \times [m]$, we have*

$$\overline{\Delta}_r\, \theta^*(\boldsymbol{x}^\pi) \;\le\; O\left(\tfrac{G\sqrt{r\log n + \log(1/\beta)}}{\mu n\sqrt{m}}\right).$$

In order to prove Theorem 3.6, we use the following lemma, proved in Appendix A.

Lemma 3.8. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N \in \mathbb{R}^d$ be any set of vectors satisfying $\sum_i \boldsymbol{v}_i = \boldsymbol{0}$ and $\|\boldsymbol{v}_i\| \le G$ for all i. For all $\beta < 1/e$, over choice of a random permutation π over $[N]$, it holds that*

$$\Pr\left[\left\|\textstyle\sum_{j\in[m]} \boldsymbol{v}_{\pi(j)}\right\| \;>\; 5G\sqrt{m\log(1/\beta)}\right] \;\le\; \beta.$$

Proof of Theorem 3.6. Let $\theta^* := \theta^*(\boldsymbol{x})$; note that due to the symmetric nature of $\mathcal{L}(\theta; \cdot)$, it holds that $\theta^*(\boldsymbol{x}) = \theta^*(\boldsymbol{x}^\pi)$ for all permutations π. Let $\theta^{*,\pi}_{-n} := \theta^*(\boldsymbol{x}^\pi_{-n})$. Since $\mathcal{L}(\cdot; \boldsymbol{x}^\pi_{-n})$ is μ-strongly convex[4], we have that

$$\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_{-n}) - \nabla\mathcal{L}(\theta^{*,\pi}_{-n}; \boldsymbol{x}^\pi_{-n})\| \;\ge\; \mu\|\theta^* - \theta^{*,\pi}_{-n}\|. \qquad (5)$$

Next, we upper bound $\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_{-n})\|$.

$$0 = \nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi) = \tfrac{n-1}{n} \cdot \nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_{-n}) + \tfrac{1}{n} \cdot \nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_n),$$

and hence

$$\begin{aligned}
\left\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_{-n})\right\| &= \tfrac{1}{n-1}\left\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_n)\right\| \\
&= \left\|\tfrac{1}{(n-1)m}\textstyle\sum_{j\in[m]} \nabla\ell(\theta; x_{\pi(n,j)})\right\|. \qquad (6)
\end{aligned}$$

Since $\sum_{i\in[n], j\in[m]} \nabla\ell(\theta; x_{i,j}) = 0$ and $\|\nabla\ell(\theta; x_{i,j})\| \le G$, we have from Lemma 3.8 that

$$\Pr\left[\left\|\textstyle\sum_{j\in[m]} \nabla\ell(\theta; x_{\pi(n,j)})\right\| \le 5G\sqrt{m\log(1/\beta)}\right] \ge 1 - \beta. \qquad (7)$$

Putting (6) and (7) together, we have that,

$$\Pr\left[\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}^\pi_{-n})\| \le \tfrac{5G\sqrt{\log\frac{1}{\beta}}}{(n-1)\sqrt{m}}\right] \ge 1 - \beta.$$

[4] f is *μ-strongly convex* iff $\|\nabla f(\theta) - \nabla f(\theta')\| \ge \mu\|\theta - \theta'\|$.

Combining this with (5), and noting that $\nabla \mathcal{L}(\theta^{*,\pi}_{-n}; \boldsymbol{x}^{\pi}_{-n}) = 0$ since $\theta^{*,\pi}_{-n}$ is the minimizer of $\mathcal{L}(\cdot; \boldsymbol{x}^{\pi}_{-n})$, completes the proof. □

4. User-Level DP-ERM

In this section, we describe our algorithms for DP-ERM and prove their excess risk bounds. As in Section 3, our guarantee holds for a random permutation of any dataset—a mild but necessary assumption.

4.1. Convex Losses

The formal guarantee when the loss is only assumed to be convex (and Lipschitz) is given below.

Theorem 4.1. *For any G-Lipschitz loss ℓ, there exists an (ε, δ)-DP mechanism that, for all $n \geq \widetilde{\Omega}\left(\frac{\log(1/\delta)\log(m)}{\varepsilon}\right)$, outputs $\widehat{\theta}$ such that*

$$\mathbb{E}_{\pi, \widehat{\theta} \leftarrow \mathcal{M}(\boldsymbol{x}^{\pi})}[\mathcal{L}(\widehat{\theta}; \boldsymbol{x}^{\pi})] - \mathcal{L}(\theta^*; \boldsymbol{x}^{\pi}) \;\leq\; \widetilde{O}_{\varepsilon}\left(\frac{RG\sqrt{d}}{n\sqrt{m}}\right),$$

where $\widetilde{O}_{\varepsilon}$ hides a multiplicative factor of $\mathrm{poly}(\log(1/\delta), \log(nm), 1/\varepsilon)$.

We use Phased-ERM algorithm similar to Feldman et al. (2020), which requires solving a regularized ERM in each step. Our proof below closely follows their proofs, although we change the algorithm slightly because their proof is for SCO whereas the analysis below is for ERM. In particular, for ERM, we need to use the full dataset in each step. We also change some parameters accordingly. The full description is in Algorithm 3; note that on line 12, we only optimize over the set $\mathcal{K}_i$, and use Lipschitz constant $2G$ and strong convexity parameter λ_i.

Algorithm 3 Phased-ERM.

1: **Input:** Dataset $\boldsymbol{x}$, loss function $\ell : \mathcal{K} \times \mathcal{X} \to \mathbb{R}$ that is convex and G-Lipschitz
2: **Parameters:** Privacy parameters ε, δ; Regularizer coefficient λ; Target failure probability β
3: $T \leftarrow \lceil \log(nm) \rceil$ ▷ Number of iterations
4: $\varepsilon' \leftarrow \varepsilon/T, \delta' \leftarrow \delta/T$ ▷ Per-iteration privacy budgets
5: $\beta' \leftarrow \beta/T$ ▷ Per-iteration failure probability
6: $\widehat{\theta}_0 \leftarrow$ arbitrary element of $\mathcal{K}$ ▷ Initial parameter
7: **for** $i = 1, \ldots, T$ **do**
8: $\quad \lambda_i \leftarrow \lambda \cdot 4^i$
9: $\quad R_i \leftarrow G/\lambda_i$
10: $\quad$ Let $\ell_i(\theta; x) := \ell(\theta; x) + \frac{\lambda_i}{2} \cdot \|\theta - \widehat{\theta}_{i-1}\|^2$
11: $\quad \mathcal{K}_i \leftarrow \mathcal{K} \cap \mathcal{B}_d(\theta_{i-1}, R_i)$
12: $\quad \widehat{\theta}_i \leftarrow \mathsf{SCOutputPert}_{\varepsilon', \delta', \beta', 2G, \lambda_i, \mathcal{K}_i}(\ell_i; \boldsymbol{x})$
13: **end for**
14: **return** $\widehat{\theta}_T$

To analyze the accuracy, let $\theta^*_i := \theta^*_{\ell_i, \mathcal{K}_i}(\boldsymbol{x})$ for all $i \in [T]$. It should be noted that θ^*_i is also equal to $\theta^*_{\ell_i, \mathcal{K}}(\boldsymbol{x})$ (where the optimization is over $\mathcal{K}$ instead of $\mathcal{K}_i$). Furthermore, within $\mathcal{K}_i$, the loss $\mathcal{L}_i$ is $2G$-Lipschitz. We start with the following lemma, which is an analogue of Feldman et al. (2020, Lemma 4.7).

Lemma 4.2. *For any $\theta \in \mathcal{K}$ and $i \in [T]$, we have*

$$\mathcal{L}(\theta^*_i; \boldsymbol{x}) - \mathcal{L}(\theta; \boldsymbol{x}) \leq \tfrac{\lambda_i}{2} \cdot \|\widehat{\theta}_{i-1} - \theta\|^2.$$

Proof. This is simply because

$$\begin{aligned}\mathcal{L}(\theta^*_i; \boldsymbol{x}) - \mathcal{L}(\theta; \boldsymbol{x}) &\leq \mathcal{L}_i(\theta^*_i; \boldsymbol{x}) - \mathcal{L}(\theta; \boldsymbol{x}) \\ &\leq \mathcal{L}_i(\theta; \boldsymbol{x}) - \mathcal{L}(\theta; \boldsymbol{x}) = \tfrac{\lambda_i}{2} \cdot \|\theta - \widehat{\theta}_{i-1}\|^2.\end{aligned}$$

□

We are now ready to prove Theorem 4.1. The usual analysis of the "standard" Phased-ERM algorithm in Feldman et al. (2020)—where SCOutputPert is replaced by an algorithm that just adds Gaussian noise to the true optimizer—shows that it has small excess risk. We then relate Algorithm 3 back to this "standard" version by using Theorem 3.1 to show that the output distribution of our algorithm (over random π) is very close in total variation distance to this standard version. This idea is formalized below.

Proof of Theorem 4.1. We run the Phased-ERM algorithm (Algorithm 3) where we set $\lambda = \frac{G\sqrt{d}}{Rn\sqrt{m}}$ and $\beta = \frac{1}{nm}$; throughout the proof, we will use $\mathcal{M}$ as a shorthand for this algorithm. The privacy guarantee follows immediately from the fact that each call to SCOutputPert is (ε', δ')-DP and the basic composition of DP.

To understand its accuracy guarantee, let us start by considering another algorithm $\mathcal{M}'$ that is the same as $\mathcal{M}$ except that on line 12 we do not call SCOutputPert but instead directly let $\widehat{\theta}_i \leftarrow \theta^*_i(\boldsymbol{x}) + \mathcal{N}(0, \sigma_i^2 \cdot I)$ where $\sigma_i := \sigma(\varepsilon', \delta', \beta', 2G, \lambda_i)$ is as in Theorem 3.1.

For convenience, we let $\theta_0 = \theta^*(\boldsymbol{x})$. We have

$$\begin{aligned}&\mathcal{L}(\widehat{\theta}_T; \boldsymbol{x}) - \mathcal{L}(\theta^*_0; \boldsymbol{x}) \\ &= (\mathcal{L}(\widehat{\theta}_T; \boldsymbol{x}) - \mathcal{L}(\theta^*_T; \boldsymbol{x})) \\ &\qquad + \textstyle\sum_{i=1}^{T} \left(\mathcal{L}(\theta^*_i; \boldsymbol{x}) - \mathcal{L}(\theta^*_{i-1}; \boldsymbol{x})\right) \\ &\leq G \cdot \|\widehat{\theta}_T - \theta^*_T\| + \textstyle\sum_{i=1}^{T} \frac{\lambda_i}{2} \cdot \|\widehat{\theta}_{i-1} - \theta^*_{i-1}\|^2 \\ &\leq O\left(\lambda R^2\right) + G \cdot \|\widehat{\theta}_T - \theta^*_T\| + \textstyle\sum_{i=1}^{T-1} \frac{\lambda_{i+1}}{2} \cdot \|\widehat{\theta}_i - \theta^*_i\|^2,\end{aligned}$$

where we used Lemma 4.2 for the first inequality.

Thus, we have (where the expectation is over the randomness of $\mathcal{M}'$, i.e., noise drawn from $\mathcal{N}(0, \sigma_i^2 \cdot I)$ for each $i \in [T]$)

$$\mathbb{E}_{\widehat{\theta} \leftarrow \mathcal{M}'(\boldsymbol{x})}[\mathcal{L}(\widehat{\theta}; \boldsymbol{x})] - \mathcal{L}(\theta^*; \boldsymbol{x})$$

$$\le O\left(\lambda R^2 + G\sqrt{d}\cdot\sigma_T + \textstyle\sum_{i=1}^{T-1}\lambda_{i+1}\cdot d\cdot\sigma_i^2\right)$$
$$\le \widetilde{O}_\varepsilon\left(\lambda R^2 + \tfrac{G^2\sqrt{d}}{\lambda_T\cdot n\sqrt{m}} + \textstyle\sum_{i=1}^{T-1}\tfrac{\lambda_{i+1}\cdot dG^2}{\lambda_i^2\cdot n^2 m}\right)$$
$$\le \widetilde{O}_\varepsilon\left(\lambda R^2 + \tfrac{dG^2}{\lambda\cdot n^2 m}\right),$$

where the last inequality comes from our setting $\lambda_i = \lambda\cdot 4^i$. Finally, from our setting of $\lambda = \frac{G\sqrt{d}}{Rn\sqrt{m}}$, we can conclude

$$\mathbb{E}_{\widehat{\theta}\leftarrow\mathcal{M}'(\boldsymbol{x})}[\mathcal{L}(\widehat{\theta};\boldsymbol{x})] - \mathcal{L}(\theta^*;\boldsymbol{x}) \le \widetilde{O}_\varepsilon\left(\tfrac{RG\sqrt{d}}{n\sqrt{m}}\right). \quad (8)$$

Let P' denote the distribution of the output of running $\mathcal{M}'$ on $\boldsymbol{x}$, and let P denote the distribution of the output of running $\mathcal{M}$ on $\boldsymbol{x}^\pi$ where π is a uniformly random permutation. Next, we will show that

$$d_{\mathrm{tv}}(P', P) \le \beta. \quad (9)$$

Before proving this, note that (8) and (9) together imply the bound in the theorem because we have

$$\mathbb{E}_{\pi,\widehat{\theta}\leftarrow\mathcal{M}(\boldsymbol{x}^\pi)}[\mathcal{L}(\widehat{\theta};\boldsymbol{x})] - \mathcal{L}(\theta^*;\boldsymbol{x})$$
$$\overset{(9)}{\le} \mathbb{E}_{\widehat{\theta}\leftarrow\mathcal{M}'(\boldsymbol{x})}[\mathcal{L}(\widehat{\theta};\boldsymbol{x})] - \mathcal{L}(\theta^*;\boldsymbol{x}) + \beta\cdot RG$$
$$\overset{(8)}{\le} \widetilde{O}_\varepsilon\left(\tfrac{RG\sqrt{d}}{n\sqrt{m}}\right).$$

We are left with proving (9). To do this, for every $i \in \{0,\dots,T\}$, consider a hybrid algorithm $\mathcal{M}_i$ where, in the first i iterations, we follow $\mathcal{M}'$ and, in the remaining iterations, we follow $\mathcal{M}$. Let P_i denote the probability distribution of the output of $\mathcal{M}_i$ on input $\boldsymbol{x}^\pi$ where π is a uniformly random permutation. Notice that $P_0 = P$ and $P_T = P'$.

For every $i \in [T]$, consider P_i and P_{i-1}. They differ only in the ith iteration. Thus, $d_{\mathrm{tv}}(P_i, P_{i-1})$ is at most the probability that SCOutputPert does not output a sample from $\theta_i^* + \mathcal{N}(0,\sigma_i^2\cdot I)$. By Theorem 3.1, the probability (over π) that this happens is at most[5] β'. Therefore, we have that $d_{\mathrm{tv}}(P_i, P_{i-1}) \le \beta'$.

Thus, $d_{\mathrm{tv}}(P,P') \le \sum_{i\in[T]} d_{\mathrm{tv}}(P_{i-1},P_i) \le T\cdot\beta' = \beta$, concluding our proof. □

4.2. Strongly Convex Losses

For strongly convex losses, we can get an improved bound:

Theorem 4.3. *For any G-Lipschitz, μ-strongly convex loss ℓ, there exists an (ε,δ)-DP mechanism that, for all $n \ge \widetilde{\Omega}\left(\frac{\log(1/\delta)\log(m)}{\varepsilon}\right)$, outputs $\widehat{\theta}$ such that*

$$\mathbb{E}_{\pi,\widehat{\theta}\leftarrow\mathcal{M}(\boldsymbol{x}^\pi)}[\mathcal{L}(\widehat{\theta};\boldsymbol{x}^\pi)] - \mathcal{L}(\theta^*;\boldsymbol{x}^\pi) \le \widetilde{O}_\varepsilon\left(\tfrac{G^2 d}{\mu n^2 m}\right),$$

where $\widetilde{O}_\varepsilon$ hides a multiplicative factor of $\mathrm{poly}(\log(1/\delta),\log(nm),1/\varepsilon)$.

We obtain the above result by reducing back to the convex case. This reduction essentially dates back to Bassily et al. (2014) and works as follows: first we apply the output perturbation algorithm (Theorem 3.1). With high probability, the output is within a ball of radius $R = \widetilde{O}_\varepsilon\left(\frac{G\sqrt{d}}{\mu n\sqrt{m}}\right)$. We then run Theorem 4.1 using this R, which yields the final excess risk of $\widetilde{O}_\varepsilon\left(\frac{G\sqrt{d}}{\mu n\sqrt{m}}\cdot\frac{G\sqrt{d}}{n\sqrt{m}}\right) = \widetilde{O}_\varepsilon\left(\frac{G^2 d}{\mu n^2 m}\right)$ as desired. The full proof is presented in Appendix B.1.

5. User-Level DP-SCO

We next describe our algorithms for DP-SCO and their excess risk guarantees.

5.1. Convex Losses

For the convex (and Lipschitz) loss case, the risk bound is similar to that of Theorem 4.1 except with an additional additive term $O(1/\sqrt{nm})$:

Theorem 5.1. *For any G-Lipschitz convex loss ℓ, there exists an (ε,δ)-DP mechanism that, for all $n \ge \widetilde{O}\left(\frac{\log(1/\delta)\log(m)}{\varepsilon}\right)$, outputs $\widehat{\theta}$ such that*

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}^{n\times m},\widehat{\theta}\leftarrow\mathcal{M}(\boldsymbol{x})}[\mathcal{L}(\widehat{\theta};\mathcal{D})] - \mathcal{L}(\theta^*;\mathcal{D})$$
$$\le \widetilde{O}_\varepsilon\left(RG\left(\tfrac{\sqrt{d}}{n\sqrt{m}} + \tfrac{1}{\sqrt{nm}}\right)\right),$$

where $\widetilde{O}_\varepsilon$ hides a multiplicative factor of $\mathrm{poly}(\log(1/\delta),\log(nm),1/\varepsilon)$.

The arguments in the previous section also extend to SCO. The idea as before is to replace the output perturbation step in Algorithm 3 of Feldman et al. (2020) with SCOutputPert. The full algorithm is presented in Algorithm 4; note that in the ith iteration, we only use the input from users $2^{-i}n,\dots,2^{-i+1}n$ to perform SCOutputPert.

Similar to before, let $\theta_i^* := \theta^*_{\ell_i,\mathcal{K}_i}(\mathcal{D})$ for all $i\in[T]$. Again, note that $\theta_i^* = \theta^*_{\ell_i,\mathcal{K}}(\mathcal{D})$ (where the optimization is over $\mathcal{K}$ instead of $\mathcal{K}_i$). Furthermore, within $\mathcal{K}_i$, the loss $\mathcal{L}_i$ is $2G$-Lipschitz.

We use the following lemma, analogous to Lemma 4.2.

Lemma 5.2. *For any $\theta\in\mathcal{K}$ and $i\in[T]$, we have*

$$\mathcal{L}(\theta_i^*;\mathcal{D}) - \mathcal{L}(\theta;\mathcal{D}) \le \tfrac{\lambda_i}{2}\cdot\|\widehat{\theta}_{i-1}-\theta\|^2 + \tfrac{16G^2}{\lambda_i n_i}.$$

Proof. The objective $\ell_i(\theta;\boldsymbol{x}^{(i)})$ is $(2G)$-Lipschitz and is λ_i-strongly convex. Therefore, by Shalev-Shwartz et al. (2009, Theorem 7), we get that

$$\mathcal{L}(\theta_i^*;\mathcal{D}) - \mathcal{L}(\theta;\mathcal{D}) \le \tfrac{\lambda_i}{2}\cdot\|\theta-\widehat{\theta}_{i-1}\|^2 + \tfrac{4(2G)^2}{\lambda_i n_i}.$$ □

[5] Note that the distribution of θ_i^* is independent of π since we are running $\mathcal{M}'$ for the first $i-1$ steps. Thereby, we can still apply Theorem 3.1, which only relies on the randomness in π.

Algorithm 4 Phased-SCO

1: **Input:** Dataset $\boldsymbol{x}$, loss function $\ell : \Theta \times \mathcal{X} \to \mathbb{R}$ that is convex and G-Lipschitz
2: **Parameters:** Privacy parameters ε, δ; Regularizer coefficient λ; Target failure probability β
3: $N_0 = C \log(1/\delta)/\varepsilon$ for some sufficiently large constant C
4: $T \leftarrow \lceil \log(n/N_0) \rceil$ ▷ Number of iterations
5: $\beta' = \beta/T$ ▷ Per-iteration failure probability
6: $\widehat{\theta}_0 \leftarrow$ arbitrary element of $\mathcal{K}$ ▷ Initial parameter
7: **for** $i = 1, \ldots, T$ **do**
8: $\quad \lambda_i = \lambda \cdot 4^i$
9: $\quad R_i = G/\lambda_i$
10: $\quad$ Let $\ell_i(\theta; x) := \ell(\theta; x) + \frac{\lambda_i}{2} \cdot \|\theta - \widehat{\theta}_{i-1}\|^2$
11: $\quad \mathcal{K}_i \leftarrow \mathcal{K} \cap \mathcal{B}_d(\theta_{i-1}, R_i)$
12: $\quad \boldsymbol{x}^{(i)} = (\boldsymbol{x}_\ell : \ell \in [2^{-i}n, 2^{-i+1}n])$
13: $\quad \widehat{\theta}_i \leftarrow \mathsf{SCOutputPert}_{\varepsilon,\delta,\beta',2G,\lambda_i,\mathcal{K}_i}(\ell_i; \boldsymbol{x}^{(i)})$
14: **end for**
15: **return** $\widehat{\theta}_T$

Proof of Theorem 5.1. We run the Phased-SCO algorithm (Algorithm 4) where we set $\lambda = \frac{G\sqrt{d}}{Rn\sqrt{m}}$ and $\beta = \frac{1}{nm}$; we will use $\mathcal{M}$ as a shorthand for this algorithm. The main difference from Algorithm 3 is that we use different batch sizes (and do not reuse sample points across phases). The analysis is similar to the proof of Theorem 4.1 with corresponding changes to Lemma 4.2. The privacy guarantee follows immediately from the fact that each call to SCOutputPert is (ε, δ)-DP and the parallel composition of DP (McSherry, 2010). Note that we maintain a minimum batch size as required for SCOutputPert so that we maintain DP.

Further, as in the analysis of Theorem 4.1, we can consider another algorithm $\mathcal{M}'$ which is the same as $\mathcal{M}$ except that on line 13 it does not call SCOutputPert but instead directly lets $\widehat{\theta}_i \leftarrow \theta_i^*(\boldsymbol{x}) + \mathcal{N}(0, \sigma_i^2 \cdot I)$ where $\sigma_i := \sigma(\varepsilon', \delta', \beta', 2G, \lambda_i)$ is as in Theorem 3.1. Proof of Theorem 5.1 is completed by following the same analysis as in the proof of Theorem 4.1 with Lemma 4.2 replaced with Lemma 5.2. □

5.2. Strongly Convex Losses

We obtain better excess risk bounds for the case of strongly convex losses, as stated below. The proof is similar to that of DP-ERM for strongly convex loss, i.e., we use output perturbation and then run DP-SCO for convex losses (Theorem 5.1) using a smaller radius. An additional step here is to show that an empirical minimizer is $\widetilde{O}\left(\frac{G}{\mu\sqrt{nm}}\right)$-close to the population minimizer w.h.p. (which might be of independent interest; see Proposition B.1). The full proof is presented in Appendix B.2.

Theorem 5.3. *For any G-Lipschitz, μ-strongly convex loss ℓ, there exists an (ε, δ)-DP mechanism that, for all $n \geq \widetilde{O}\left(\frac{\log(1/\delta)\log(m)}{\varepsilon}\right)$, outputs $\widehat{\theta}$ such that*

$$\mathbb{E}_{\pi, \widehat{\theta} \leftarrow \mathcal{M}(\boldsymbol{x}^\pi)}[\mathcal{L}(\widehat{\theta}; \mathcal{D})] - \mathcal{L}(\theta^*; \mathcal{D}) \leq \widetilde{O}_\varepsilon\left(\frac{G^2}{\mu}\left(\frac{d}{n^2m} + \frac{1}{nm}\right)\right),$$

where $\widetilde{O}_\varepsilon$ hides a multiplicative factor of $\mathrm{poly}(\log(1/\delta), \log(nm), 1/\varepsilon)$.

6. Discussion & Open Problems

Although we do not discuss the running time of our algorithm, it can be seen that they run in $n^{O(\log(1/\delta)/\varepsilon)}(md)^{O(1)}$ time; the bottleneck comes from the step to compute $\mathcal{X}^{R_1}_{\text{stable}}$ in DelOutputPert, which requires enumerating all subsets S of size $R_1 = O\left(\frac{\log(1/\delta)}{\varepsilon}\right)$. In Appendix C, we describe a speed up for all our DP-SCO/ERM results that makes the algorithm run in polynomial (in n, m, d) time *with high probability*. However, with the remaining $o(1)$ probability, the algorithm may still take $n^{O(\log(1/\delta)/\varepsilon)}(md)^{O(1)}$ time. It remains open whether we can get an algorithm whose running time is polynomial in the *worst* case. As discussed in the introduction, it was not known whether excess risk bounds that we achieve were obtainable (even with inefficient algorithms) before.

Another question is whether we can get tight dependency on δ, ε. Specifically, our ERM excess risk bound in the convex case has a factor of $O\left(\frac{(\log(1/\delta))^2}{\varepsilon^{2.5}}\right)$, while previous bounds only had $\widetilde{O}\left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon}\right)$. Note that our larger dependency is indeed due to the generic output perturbation algorithm (DelOutputPert), which requires the noise scale σ to be inflated by a factor of $\kappa = O\left(\frac{\log(1/\delta)}{\varepsilon}\right)$, and the union bound performed for Corollary 3.7 which includes another $\sqrt{\kappa}$ factor. Therefore, this question may be related to the previous question.

Acknowledgements

P.K. would like to thank Gene Li, Ohad Shamir and Nathan Srebro for helpful discussions about stochastic convex optimization. P.M. would also like to thank Adam Sealfon for useful discussions and for pointers to DP graph analysis literature.

References

Agarwal, A., Bartlett, P. L., Ravikumar, P., and Wainwright, M. J. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Trans. Inf. Theory*, 58(5):3235–3249, 2012.

Asi, H., Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: Optimal rates in ℓ_1 geometry. In *ICML*, pp. 393–403, 2021.

Bassily, R., Smith, A. D., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pp. 464–473, 2014.

Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. G. Private stochastic convex optimization with optimal rates. In *NeurIPS*, pp. 11279–11288, 2019.

Chaudhuri, K. and Monteleoni, C. Privacy-preserving logistic regression. In *NIPS*, pp. 289–296, 2008.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, 2011.

Chen, S. and Zhou, S. Recursive mechanism: towards node differential privacy and unrestricted joins. In *SIGMOD*, pp. 653–664, 2013.

Dwork, C. and Lei, J. Differential privacy and robust statistics. In *STOC*, pp. 371–380, 2009.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pp. 486–503, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. Calibrating noise to sensitivity in private data analysis. In *TCC*, pp. 265–284, 2006b.

Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: optimal rates in linear time. In *STOC*, pp. 439–449, 2020.

Ghazi, B., Kumar, R., and Manurangsi, P. User-level differentially private learning via correlated sampling. In *NeurIPS*, pp. 20172–20184, 2021.

Gopi, S., Lee, Y. T., and Liu, D. Private convex optimization via exponential mechanism. In *COLT*, pp. 1948–1989, 2022.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Found. Trends Machine Learning*, 14(1-2), 2021.

Kamath, G., Li, J., Singhal, V., and Ullman, J. R. Privately learning high-dimensional distributions. In *COLT*, pp. 1853–1902, 2019.

Kifer, D., Smith, A. D., and Thakurta, A. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *COLT*, pp. 25.1–25.40, 2012.

Kohli, N. and Laskowsk, P. Differential privacy for black-box statistical analyses. In *TPDP*, 2021.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *Ann. Stat.*, 28 (5):1302 – 1338, 2000.

Levy, D., Sun, Z., Amin, K., Kale, S., Kulesza, A., Mohri, M., and Suresh, A. T. Learning with user-level privacy. In *NeurIPS*, pp. 12466–12479, 2021.

Liu, Y., Suresh, A. T., Yu, F. X., Kumar, S., and Riley, M. Learning discrete distributions: user vs item-level privacy. In *NeurIPS*, 2020.

McSherry, F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commun. ACM*, 53(9):89–97, 2010.

Narayanan, S., Mirrokni, V. S., and Esfandiari, H. Tight and robust private mean estimation with few users. In *ICML*, pp. 16383–16412, 2022.

Nissim, K., Raskhodnikova, S., and Smith, A. D. Smooth sensitivity and sampling in private data analysis. In *STOC*, pp. 75–84, 2007.

Raskhodnikova, S. and Smith, A. D. Differentially private analysis of graphs. In *Encyclopedia of Algorithms*, pp. 543–547. Springer, 2016.

Sambale, H. and Sinulis, A. Concentration inequalities on the multislice and for sampling without replacement. *Journal of Theoretical Probability*, 35:2712–2737, 2022.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *COLT*, 2009.

Vadhan, S. P. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pp. 347–450. Springer International Publishing, 2017.

Wang, D., Ye, M., and Xu, J. Differentially private empirical risk minimization revisited: Faster and more general. In *NIPS*, pp. 2722–2731, 2017.

A. Proof of Lemma 3.8

Lemma 3.8 follows quite immediately as an application of a special case of Proposition 1 in Sambale & Sinulis (2022) as stated below. Let S_N denote the set of all permutations over $[N]$ and for any $\pi \in S_N$, let $\pi^{i\leftrightarrow j}$ denote the permutation with ith and jth entries swapped.

Proposition A.1 (Proposition 1 in Sambale & Sinulis (2022)). *Let $f : S_N \to \mathbb{R}$ be a real-valued function over S_N, such that $|f(\pi) - f(\pi^{i\leftrightarrow j})| \le c_{i,j}$ for all $\pi \in S_N$ and all $1 \le i < j \le N$ for some $c_{i,j} \ge 0$. For any $t \ge 0$, it holds that*

$$\Pr_{\pi\sim S_N}[f(\pi) - \mathbb{E}[f(\pi)] \ge t] \le \exp\left(-\frac{Nt^2}{4\sum_{1\le i<j\le N} c_{i,j}^2}\right).$$

Proof of Lemma 3.8. Since $\sum_i \boldsymbol{v}_i = 0$, we have for any two vectors $\boldsymbol{u}, \boldsymbol{v}$ sampled randomly without replacement from $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$ that $\langle \boldsymbol{u}, \boldsymbol{v}\rangle < 0$, since $\mathbb{E}[\boldsymbol{u} \mid \boldsymbol{v}] = -\boldsymbol{v}/(N-1)$. Hence, we have

$$\mathbb{E}\left[\left\|\sum_{j\in[m]} \boldsymbol{v}_{i_j}\right\|^2\right] = \sum_{j\in[m]} \mathbb{E}[\|\boldsymbol{v}_{i_j}\|^2] + 2\sum_{j<k} \mathbb{E}\left[\langle \boldsymbol{v}_{i_j}, \boldsymbol{v}_{i_k}\rangle\right] \le mG^2,$$

and hence $\mathbb{E}\left[\left\|\sum_{j\in[m]} \boldsymbol{v}_{i_j}\right\|\right] \le \sqrt{m}G$. Let $f : S_N \to \mathbb{R}$ be defined as $f(\pi) = \|\sum_{j=1}^m \boldsymbol{v}_{\pi(j)}\|$. It follows that $f(\pi) = f(\pi^{i\leftrightarrow j})$ whenever both $i, j \le m$ or both $i, j > m$. Moreover, when $i \le m$ and $j > m$, it holds that

$$|f(\pi) - f(\pi^{i\leftrightarrow j})| = \left\|\sum_{k=1}^m \boldsymbol{v}_{\pi(k)}\right\| - \left\|\sum_{k=1}^m \boldsymbol{v}_{\pi^{i\leftrightarrow j}(k)}\right\| \le \left\|\boldsymbol{v}_{\pi(i)} - \boldsymbol{v}_{\pi(j)}\right\| \le 2G.$$

Thus, using Proposition A.1 with $c_{i,j} = 2G$ when $i \le m < j$ and 0 otherwise, we have that

$$\Pr_{\pi\sim S_N}\left[\left\|\sum_{j\in[m]} \boldsymbol{v}_{\pi(j)}\right\| \ge t + \sqrt{m}G\right] \le \exp\left(-\frac{Nt^2}{16m(N-m)G^2}\right) \le \exp\left(-\frac{t^2}{16mG^2}\right).$$

Choosing $t = 4G\sqrt{m\log(1/\beta)}$ completes the proof. □

B. Proofs of Improved Bounds for Strongly Convex Losses

B.1. Empirical Risk Minimization

Algorithm 5 Strongly-Convex-ERM

Input: Dataset $\boldsymbol{x}$, loss function $\ell : \Theta \times \mathcal{X} \to \mathbb{R}$ that is μ-strongly convex and G-Lipschitz
Parameters: Privacy parameters ε, δ; Target Failure Probability β
$\theta_0 \leftarrow \mathsf{SCOutputPert}_{\varepsilon/2,\delta/2,\beta,G,\mu,\mathcal{K}}(\ell; \boldsymbol{x})$
$R' \leftarrow \sigma(\varepsilon/2, \delta/2, \beta, G, \mu) \cdot \sqrt{d\log 1/\beta}$
$\mathcal{K}' \leftarrow \mathcal{K} \cap \mathcal{B}_d(\theta_0, R')$
$\lambda \leftarrow \frac{G\sqrt{d}}{R'n\sqrt{m}}$
$\widehat{\theta} \leftarrow \mathsf{Phased\text{-}ERM}_{\varepsilon/2,\delta/2,\beta,G,\lambda,\mathcal{K}'}(\ell; \boldsymbol{x})$
return $\widehat{\theta}$

Proof of Theorem 4.3. The mechanism in Algorithm 5, which uses a two-step approach to get stronger rates for strongly convex losses, following a similar reduction in Bassily et al. (2014). It first uses the SCOutputPert algorithm with $(\varepsilon/2, \delta/2)$-DP, which with probability $1-\beta$ returns $\theta_0 := \theta^*(\boldsymbol{x}) + e$ where $e \sim \mathcal{N}(0, \sigma^2 \cdot I)$ for σ specified in Theorem 3.1. From standard concentration, we have that $\Pr[\|e\| \ge C\sigma\sqrt{d\log 1/\beta}] \le \beta$, for a suitable C. Thus, with probability $1-2\beta$, we have that $\theta^*(\boldsymbol{x})$ is indeed contained in $\mathcal{B}_d(\theta_0, R')$ for $R' = C\sigma\sqrt{d\log 1/\beta} = \widetilde{O}_\varepsilon(G\sqrt{d}/(\mu n\sqrt{m}))$; note that this can be much smaller than the diameter of $\mathcal{K}$ which is at most $2G/\mu$. Finally, we use the Phased-ERM algorithm with

$(\varepsilon/2, \delta/2)$-DP over the region $\mathcal{K}' = \mathcal{K} \cap \mathcal{B}_d(\theta_0, R')$. Following the proof of Theorem 4.1, setting $\beta = 1/2n^2m$, we have that

$$\mathbb{E}[\mathcal{L}(\widehat{\theta}; \boldsymbol{x})] - \mathcal{L}(\theta^*; \boldsymbol{x}) \;\leq\; \widetilde{O}_\varepsilon\left(\tfrac{G^2 d}{\mu n^2 m}\right).$$

The value of β was chosen such that $\beta RG \leq O(G^2/(\mu n^2 m))$, where R is the diameter of $\mathcal{K}$, which is at most $2G/\mu$. This is to account for the probability of at most 2β that either SCOutputPert fails or that $\|e\| > C\sigma\sqrt{d \log 1/\beta}$, in which case, the excess risk is at most RG. □

B.2. Stochastic Convex Optimization

We rely on the following proposition, which to the best of our knowledge, is not known in the literature.

Proposition B.1. *For any G-Lipschitz, μ-strongly convex loss ℓ and for any distribution $\mathcal{D}$, it holds for all $\beta < 1/e$ that*

$$\Pr_{\boldsymbol{x} \sim \mathcal{D}^{n \times m}}\left[\|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\| \leq \frac{30G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}}\right] \geq 1 - \beta.$$

Before we prove Proposition B.1, let us see how to use it to prove Theorem 5.3.

Algorithm 6 Strongly-Convex-SCO

Input: Dataset $\boldsymbol{x}$, loss function $\ell : \Theta \times \mathcal{X} \to \mathbb{R}$ that is μ-strongly convex and G-Lipschitz
Parameters: Privacy parameters ε, δ; Target Failure Probability β
$\theta_0 \leftarrow \mathsf{SCOutputPert}_{\varepsilon/2, \delta/2, \beta, G, \mu, \mathcal{K}}(\ell; \boldsymbol{x})$
$R' \leftarrow \sigma(\varepsilon/2, \delta/2, \beta, G, \mu) \cdot \sqrt{d \log 1/\beta} + \frac{G\sqrt{\log 1/\beta}}{\mu\sqrt{nm}}$
$\mathcal{K}' \leftarrow \mathcal{K} \cap \mathcal{B}_d(\theta_0, R')$
$\lambda \leftarrow \frac{G\sqrt{d}}{R' n\sqrt{m}}$
$\widehat{\theta} \leftarrow \mathsf{Phased\text{-}SCO}_{\varepsilon/2, \delta/2, \beta, G, \lambda, \mathcal{K}'}(\ell; \boldsymbol{x})$
return $\widehat{\theta}$

Proof of Theorem 5.3. Algorithm 6 is similar to Algorithm 5, namely, it first uses the SCOutputPert algorithm with $(\varepsilon/2, \delta/2)$-DP, which with probability $1 - \beta$ returns $\theta_0 := \theta^*(\boldsymbol{x}) + e$ where $e \sim \mathcal{N}(0, \sigma^2 \cdot I)$. Using Proposition B.1, we have that with probability at least $1 - \beta$, it holds that $\|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\| \leq O(G\sqrt{\log 1/\beta}/(\mu\sqrt{nm}))$. Thus, we have that $\theta^*(\mathcal{D})$ is contained in $\mathcal{B}_d(\theta_0, R')$ for $R' = O\left(\frac{G}{\mu}\left(\frac{\sqrt{d}}{n\sqrt{m}} + \frac{1}{\sqrt{nm}}\right)\right)$ with probability at least $1 - \beta$. Finally, we use the Phased-SCO algorithm with $(\varepsilon/2, \delta/2)$-DP over the region $\mathcal{K}' = \mathcal{K} \cap \mathcal{B}_d(\theta_0, R')$. We get our desired conclusion by plugging in the bound for R' in Theorem 5.1, again setting $\beta = 1/2n^2m$. □

We suspect that Proposition B.1 might already be known in literature. Since we are unaware of a reference, we include a proof for completeness, which incidentally uses our new result about deletion stability (Theorem 3.6).

Proof of Proposition B.1. First, it is well known from Shalev-Shwartz et al. (2009) that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}^{n \times m}}[\mathcal{L}(\theta^*(\boldsymbol{x}); \mathcal{D})] - \mathcal{L}(\theta^*(\mathcal{D}); \mathcal{D}) \;\leq\; \frac{4G^2}{\mu nm}.$$

On the other hand, from strong convexity we have for all $\boldsymbol{x}$ that

$$\mathcal{L}(\theta^*(\boldsymbol{x}); \mathcal{D}) - \mathcal{L}(\theta^*(\mathcal{D}); \mathcal{D}) \;\geq\; \frac{\mu}{2} \cdot \|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\|^2.$$

Combining the above, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}^{n \times m}}[\|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\|] \;\leq\; \frac{3G}{\mu\sqrt{nm}}. \tag{10}$$

Additionally, from Theorem 3.6 (invoked twice with $m \leftarrow nm$ and $n \leftarrow 2$, followed by the triangle inequality and a union bound), it follows that

$$\Pr_{\boldsymbol{x},\boldsymbol{x}' \sim \mathcal{D}^{n\times m}}\left[\|\theta^*(\boldsymbol{x}) - \theta^*(\boldsymbol{x}')\| \;\leq\; \frac{10G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}}\right] \;\leq\; \beta.$$

By an averaging argument, there exists $\theta_0 = \theta^*(\boldsymbol{x}^{(0)})$ for some $\boldsymbol{x}^{(0)}$, such that

$$\Pr_{\boldsymbol{x} \sim \mathcal{D}^{n\times m}}\left[\|\theta^*(\boldsymbol{x}) - \theta_0\| \;\leq\; \frac{10G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}}\right] \;\leq\; \beta.$$

Thus, combining with Equation (10), we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}^{n\times m}} \|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\| \;\geq\; (1-\beta)\cdot\left(\|\theta_0 - \theta^*(\mathcal{D})\| - \frac{10G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}}\right)\\
\implies \qquad &\|\theta_0 - \theta^*(\mathcal{D})\| \;\leq\; \frac{20G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}} \qquad (\text{for } \beta < 1/2).
\end{aligned}$$

Finally by the triangle inequality, we get

$$\Pr_{\boldsymbol{x} \sim \mathcal{D}^{n\times m}}\left[\|\theta^*(\boldsymbol{x}) - \theta^*(\mathcal{D})\| \;\leq\; \frac{30G\sqrt{\log(2/\beta)}}{\mu\sqrt{nm}}\right] \;\leq\; \beta. \qquad \square$$

C. On Speeding up our Algorithms

As stated in Section 6, the time bottleneck of our algorithm is DelOutputPert, which requires computing $\mathcal{X}^{R_1}_{\text{stable}}$. Doing this in a straightforward manner requires enumerating all sets S of size R_1, resulting in a running time of $n^{R_1}(md)^{O(1)} = n^{O(\log(1/\delta)/\varepsilon)}(md)^{O(1)}$. In this section, we sketch an argument that brings the time down to $(nmd)^{O(1)}$ with high probability, while maintaining all the error bounds to within $\widetilde{O}_\varepsilon(1)$ factor. Note that all algorithms invoke DelOutputPert through Theorem 3.1 (i.e., the SCOutputPert algorithm). Therefore, it suffices to argue how to achieve the speed up for SCOutputPert.

The first observation here is that if $\boldsymbol{x}$ belongs to $\mathcal{X}^{R_1}_{\text{stable}}$, then we can just output $\theta^*(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \cdot I)$. Furthermore, we have already shown (Theorem 3.1) that $\boldsymbol{x} \in \mathcal{X}^{R_1}_{\text{stable}}$ with high probability. Thus, if we can give a "certificate" that $\boldsymbol{x} \in \mathcal{X}^{R_1}_{\text{stable}}$, then we would be able to complete skip the check and just output $\theta^*(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \cdot I)$; this means that, whenever we have such a certificate, our algorithm will run in polynomial (in n, m, d) time.

Our certificate is simple: the gradients at θ^* w.r.t. each user. The following lemma (whose proof is similar to part of the proof of Theorem 3.6) relates this certificate to $\overline{\Delta}_r$ (which in turn implies membership in $\mathcal{X}^{R_1}_{\text{stable}}$ for appropriate Δ).

Lemma C.1. *Let $\boldsymbol{x}$ be any dataset and let $\theta^* := \theta^*(\boldsymbol{x})$. Suppose that for all $i \in [n]$, we have $\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}_i)\| \leq \gamma$. Then, we have $\overline{\Delta}_r\, \theta^*(\boldsymbol{x}) \leq \Delta$ for $\Delta = O(\frac{r\gamma}{\mu n})$ for all $r \leq n/2$.*

Proof. Consider any set $S \subseteq [n]$ such that $|S| \leq r$. Let $s := |S|$ and $\theta^*_{-S} := \theta^*(\boldsymbol{x}_{-S})$. Since $\nabla\mathcal{L}(\theta^*; \boldsymbol{x}) = 0$, we have

$$\begin{aligned}
\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}_{-S})\| &= \left\|\frac{1}{n-s}\sum_{i\in S}\nabla\mathcal{L}(\theta^*; \boldsymbol{x}_i)\right\|\\
&\leq \frac{1}{n-s}\sum_{i\in S}\|\nabla\mathcal{L}(\theta^*; \boldsymbol{x}_i)\|\\
&\leq \frac{s\gamma}{n-s}\\
&\leq \frac{r\gamma}{n/2} = O(r\gamma/n).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}\mathcal{L}(\theta^*; \boldsymbol{x}_{-S}) - \mathcal{L}(\theta^*_{-S}; \boldsymbol{x}_{-S}) \;&\le\; \left\langle \nabla \mathcal{L}(\theta^*; \boldsymbol{x}_{-S}), \theta^* - \theta^*_{-S} \right\rangle \\ &\le O(r\gamma/n) \cdot \|\theta^* - \theta^*_{-S}\|.\end{aligned}$$

On the other hand, since ℓ is μ-strongly convex and since θ^*_{-S} is the minimizer for $\mathcal{L}(\cdot; \boldsymbol{x}_{-S})$, we can conclude that

$$\mathcal{L}(\theta^*; \boldsymbol{x}_{-S}) - \mathcal{L}(\theta^*_{-S}; \boldsymbol{x}_{-S}) \;\ge\; \frac{\mu}{2}\|\theta^*_{-S} - \theta^*\|^2$$

Comparing the two bounds above, we get

$$\|\theta^* - \theta^{*,\pi}_{-n}\| \;\le\; O\left(\frac{r\gamma}{\mu n}\right). \qquad \square$$

Recall also from the proof of Theorem 3.6 that w.h.p. we have $\|\nabla \mathcal{L}(\theta^*; \boldsymbol{x}_i)\| \le \widetilde{O}(G/\sqrt{m})$. When this holds, by computing $\sum_{j\in[m]} \nabla \ell(\theta^*; \boldsymbol{x}_{i,j})$ for all $i \in [n]$, the above lemma means that this is a certificate that $\boldsymbol{x} \in \mathcal{X}^{R_1}_{\text{stable}}$ when we set $\Delta = O(\frac{\kappa\gamma}{\mu n}) = \widetilde{O}\left(\frac{G\cdot\log(1/\delta)}{\varepsilon\mu n\sqrt{m}}\right)$. Plugging this into Theorem 3.3, we arrive at a statement similar to Theorem 3.1 but with

$$\sigma = O\left(\frac{G\sqrt{\log n \log(1/\delta)/\varepsilon + \log(1/\beta)}}{\mu n\sqrt{m}} \cdot \frac{(\log(1/\delta))^{2.5}}{\varepsilon^3}\right),$$

i.e., with an extra factor of $O(\log(1/\delta)/\varepsilon)$. On the other hand, from the discussion about the certificate, we have that this algorithm runs in polynomial time with high probability (whenever $\|\nabla \mathcal{L}(\theta^*; \boldsymbol{x}_i)\| \le \widetilde{O}(\sqrt{m})$).

D. On Lower Bounds for User-Level DP-ERM and DP-SCO

This section discusses lower bounds for user-level DP-SCO and DP-ERM. We start by noting that Levy et al. (2021) already proved a lower bound of $\Omega\left(RG\left(\frac{1}{\sqrt{nm}} + \frac{\sqrt{d}}{\varepsilon n\sqrt{m}}\right)\right)$ for DP-SCO for the convex case assuming $n \ge \Omega\left(\sqrt{d}/\varepsilon\right)$. It can be easily seen that this also implies a lower bound of $\Omega\left(RG \cdot \frac{\sqrt{d}}{\varepsilon n\sqrt{m}}\right)$ for $\Omega\left(\sqrt{d}/\varepsilon\right) \le n \le O\left(d/\varepsilon^2\right)$ (see, e.g., the proof of Theorem D.8 below). In the remainder of this section, we extend their techniques to show the lower bounds for strongly convex losses.

D.1. Preliminaries

Throughout, we will consider the loss $\ell^{\zeta}_{\text{sq}}(\theta; x) := \zeta \cdot \|\theta - x\|^2$ where $\zeta > 0$ is a parameter. We list here a few results that will be useful throughout. We start by defining the (ℓ_2-)truncated version of the Gaussian distribution as follows.

Definition D.1. Let $\mathcal{N}^{\text{tr}}(\chi, \Sigma; B)$ denote the distribution of r.v. Z drawn as follows. First, draw $Z' \sim \mathcal{N}(\chi, \Sigma)$. Then, let $Z = Z' \cdot \mathbf{1}[\|Z'\| \le B]$. We use $\chi^{\text{tr}}(\chi, \Sigma; B)$ to denote the mean of the distribution $\mathcal{N}^{\text{tr}}(\chi, \Sigma; B)$.

As shown in Levy et al. (2021), the means of the truncated Gaussian distribution and the standard (non-truncated) version are very close:

Lemma D.2 (Levy et al. 2021)**.** *For any $\chi \in \mathbb{R}^d, d \in \mathbb{N}, \sigma > 0$, if $\|\chi\|_2 + 100\sqrt{d}\cdot\sigma < B$, then $\|\chi^{\text{tr}}(\chi, \sigma^2 I_d; B) - \chi\|_2 \le O((\sigma + \|\chi\|_2) \cdot e^{-10d})$.*

Since the version of the lemma in Levy et al. 2021 is slightly different than the one we use here, we give a proof sketch of this below[6].

Proof Sketch of Lemma D.2. Due to spherical symmetry, we may assume w.l.o.g. that $\chi_2 = \cdots = \chi_d = 0$ and $\chi_1 \ge 0$. Again, due to symmetry, we have $\chi^{\text{tr}}(\chi, \sigma^2 I_d; B)_2 = \cdots = \chi^{\text{tr}}(\chi, \sigma^2 I_d; B)_d = 0$ and thus $\|\chi^{\text{tr}}(\chi, \sigma^2 I_d; B) - \chi\|_2 = |\chi^{\text{tr}}(\chi, \sigma^2 I_d; B)_1 - \chi_1|$.

To bound this term, observe further that we may view Z_1 as being generated as follows:

[6] More precisely, Levy et al. 2021 is using truncation in a coordinate-by-coordinate manner (i.e. by the ℓ_∞ norm), which results in an extra polylogarithmic factor.

- Sample $Z_1' \sim \mathcal{N}(\chi_1, \sigma^2)$.
- Sample $U \sim \chi^2(d-1)$. (This represents $((Z_2')^2 + \cdots + (Z_d')^2)/\sigma^2$.)
- Let $Z_1 = Z_1' \cdot \mathbf{1}[(Z_1')^2 + \sigma^2 \cdot U \le B^2]$

For $u > 0$, let μ_u denote the mean of Z_1 conditioned on $U = u$. We have

$$\chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B) = \mathop{\mathbb{E}}_{U \sim \chi^2(d-1)}[\mu_U].$$

From symmetry, it is again simple to see that $0 \le \mu_U \le \chi_1$. As such, we have

$$|\chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)_1 - \chi_1| \le \mathop{\mathbb{E}}_{U \sim \chi^2(d-1)}[|\mu_U - \chi_1|] = \mathop{\mathbb{E}}_{U \sim \chi^2(d-1)}[\chi_1 - \mu_U].$$

Now, using standard concentration of $\chi^2(d-1)$ distribution (see e.g., (Laurent & Massart, 2000)), we have $\Pr_{U \sim \chi^2(d-1)}[U \ge 70\sqrt{d}] \le e^{-10d}$. From this, we have

$$\begin{aligned}|\chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)_1 - \chi_1| &\le \mathop{\mathbb{E}}_{U \sim \chi^2(d-1)}[\chi_1 - \mu_U \mid U \le 70\sqrt{d}] + \chi_1 \cdot \Pr_{U \sim \chi^2(d-1)}[U \ge 70\sqrt{d}] \\ &\le \max_{u \in [0, 70\sqrt{d}]} (\chi_1 - \mu_u) + \chi_1 \cdot e^{-10d}.\end{aligned}$$

To bound the first term, observe that for a fixed u, we simply have $Z_1 = Z_1' \mathbf{1}[|Z_1'| \le B_u]$ where $B_u := \sqrt{B^2 - \sigma^2 u} \ge \|\chi\|_2 + 70\sqrt{d} \cdot \sigma$. Thus, we have

$$\mu_u = \Pr[|Z_1'| \le B_u]\, \mathbb{E}[Z_1' \mid |Z_1'| \le B_u] \ge (1 - e^{-10d}) \cdot \mathbb{E}[Z_1' \mid |Z_1'| \le B_u],$$

where the probability bound on $\Pr[|Z_1'| > B_u]$ is based on standard concentrations of a (single-variate) Gaussian.

Finally, $\mathbb{E}[Z_1' \mid |Z_1'| \le B_u]$ is simply the expectation of the truncated single-variate Gaussian distribution, which has a closed-form formula, described below. Here ψ, Φ denote the PDF and CDF of the standard normal distribution respectively, and let $\alpha = \left(\frac{-B_u - \chi_1}{\sigma}\right), \beta = \left(\frac{B_u - \chi_1}{\sigma}\right)$. Note that we have $\beta \ge 70\sqrt{d}$.

$$\mathbb{E}[Z_1' \mid |Z_1'| \le B_u] = \chi_1 + \sigma \left(\frac{\psi(\alpha) - \psi(\beta)}{\Phi(\beta) - \Phi(\alpha)}\right) \ge \chi_1 - \sigma \cdot O(\psi(\beta)) \ge \chi_1 - \sigma \cdot O(e^{-10d}).$$

Plugging the previous three bounds together, we have

$$|\chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)_1 - \chi_1| \le O((\sigma + \chi_1) \cdot e^{-10d}).$$ □

More importantly, Levy et al. (2021) make the following crucial observation, which allows us to reduce any user-level DP algorithm for Gaussian distribution back to an item-level DP algorithm, albeit with the variance that is m times smaller.

Lemma D.3 (User-to-Item Level Reduction, Levy et al. 2021)**.** *Let $\mathcal{A}_{\mathrm{user}}$ be any user-level (ε, δ)-DP algorithm. Then, there exists an item-level (ε, δ)-DP algorithm $\mathcal{A}_{\mathrm{item}}$ such that, for any Gaussian distribution $\mathcal{D} = \mathcal{N}(\chi, \sigma^2 I_d)$, $\mathcal{A}_{\mathrm{user}}(\mathcal{D}^{n \times m})$ has exactly the same distribution as $\mathcal{A}_{\mathrm{item}}(\tilde{\mathcal{D}}^n)$ where $\tilde{\mathcal{D}} = \mathcal{N}\left(\chi, \frac{\sigma^2}{m} I_d\right)$.*

Finally, we will use the following "fingerprinting lemma for Gaussians" result due to Kamath et al. (2019), which gives a lower bound for any DP algorithm for estimating the mean of a spherical Gaussian.

Theorem D.4 (Kamath et al. 2019)**.** *For any $\psi \in (0,1), \sigma > 0, n, d \in \mathbb{N}$ and $\varepsilon \in (0,1], \delta \in (0, 1/2]$ such that $\delta \le \frac{\sqrt{d}}{100\psi n \sqrt{\log(100\psi n/\sqrt{d})}}$, if there exists an item-level (ε, δ)-DP mechanism $\mathcal{M}$ such that, for any Gaussian distribution $\mathcal{D} = \mathcal{N}(\chi, \sigma^2 I_d)$ where χ is unknown with $-\psi\sigma \le \chi \le \psi\sigma$ it satisfies*

$$\mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\mathcal{D}^n)}[\|\hat{\chi} - \chi\|^2] \le \alpha^2 \le \frac{d\sigma^2\psi^2}{6},$$

then we must have $n \ge \frac{d\sigma}{24\alpha\varepsilon}$.

Combining Lemma D.3 and Theorem D.4, we immediately arrive at the following lower bound for the user-level DP setting.

Lemma D.5. *For any $\psi \in (0,1), \sigma > 0, m, n, d \in \mathbb{N}$ and $\varepsilon \in (0,1], \delta \in (0,1/2]$ such that $\delta \le \frac{\sqrt{d}}{100\psi n\sqrt{\log(100\psi n/\sqrt{d})}}$, if there exists a user-level (ε,δ)-DP mechanism $\mathcal{M}$ such that, for any Gaussian distribution $\mathcal{D} = \mathcal{N}(\chi, \sigma^2 I_d)$ where χ is unknown with $-\frac{\psi\sigma}{\sqrt{m}} \le \chi \le \frac{\psi\sigma}{\sqrt{m}}$ it satisfies*

$$\mathop{\mathbb{E}}_{\hat{\chi}\leftarrow\mathcal{M}(\mathcal{D}^{n\times m})}[\|\hat{\chi}-\chi\|^2] \le \alpha^2 \le \frac{d\sigma^2\psi^2}{6m},$$

then we must have $n \ge \frac{d\sigma}{24\alpha\varepsilon\sqrt{m}}$.

Furthermore, combining the above with Lemma D.2, we arrive at the following lower bound where the only change is from Gaussian distributions to truncated Gaussian distributions.

Lemma D.6. *For any $\psi \in (\Omega(e^{-d}), 1), B, \sigma > 0, m, n, d \in \mathbb{N}$ and $\varepsilon \in (0,1], \delta \in (0,1/2]$ such that $\delta \le \frac{\sqrt{d}}{100\psi n\sqrt{\log(100\psi n/\sqrt{d})}}$ and $B > \frac{\psi\sigma\sqrt{d}}{\sqrt{m}} + 100\sqrt{d}\sigma$, if there exists a user-level (ε,δ)-DP mechanism $\mathcal{M}$ such that, for any truncated Gaussian distribution $\mathcal{D} = \mathcal{N}^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)$ where χ is unknown with $-\frac{\psi\sigma}{\sqrt{m}} \le \chi \le \frac{\psi\sigma}{\sqrt{m}}$ it satisfies*

$$\mathop{\mathbb{E}}_{\hat{\chi}\leftarrow\mathcal{M}(\mathcal{D}^{n\times m})}[\|\hat{\chi}-\chi^{\mathrm{tr}}(\chi,\sigma^2 I_d; B)\|^2] \le \alpha^2 \le \frac{d\sigma^2\psi^2}{12m},$$

then we must have $n \ge \frac{d\sigma}{50\alpha\varepsilon\sqrt{m}}$.

D.2. Lower Bounds for Strongly Convex Losses

D.2.1. DP-SCO

We can now prove the lower bound for DP-SCO in the strongly convex case in a relatively straightforward manner, as optimizing for the loss ℓ_{sq} is equivalent to mean estimation with ℓ_2^2-error.

Theorem D.7. *For any $\varepsilon \in (0,1], \delta \in (0,1/2]$ and any sufficiently large $d, n, m \in \mathbb{N}$ such that $n \ge \sqrt{d}/\varepsilon$ and $\delta \le \frac{\sqrt{d}}{200\sqrt{n\sqrt{\log n}}}$, there exists a μ-strongly convex G-Lipschitz loss function ℓ such that for any (ε,δ)-DP algorithm, we have*

$$\sup_{\mathcal{D}}\left(\mathop{\mathbb{E}}_{\widehat{\theta}\leftarrow\mathcal{M}(\mathcal{D}^n)}\left[\mathcal{L}(\widehat{\theta};\mathcal{D})\right] - \mathcal{L}(\theta^*;\mathcal{D})\right) \;\ge\; \Omega\left(\frac{G^2}{\mu}\left(\frac{1}{nm} + \frac{d}{\varepsilon^2 n^2 m}\right)\right).$$

We note that the condition $n \ge \sqrt{d}/\varepsilon$ may be unnecessary. However, a slightly weaker condition $n\sqrt{m} \ge \Omega(\sqrt{d}/\varepsilon)$ is necessary because outputting, e.g., the origin already achieves an error of G^2/μ. Therefore, the second term $\frac{G^2}{\mu}\cdot\frac{d}{\varepsilon^2 n^2 m}$ cannot be present in this case.

Proof of Theorem D.7. The first term of $\Omega\left(\frac{G^2}{\mu}\frac{1}{nm}\right)$ is simply the statistical excess risk bound that holds even without any privacy considerations (Agarwal et al., 2012). We will only focus on the second term here.

Consider $\ell = \ell_{\mathrm{sq}}^{\zeta}$ for $\zeta = \mu/2$ and the parameter space $\mathcal{K} = \mathcal{B}_d(0, G/\mu)$. The loss is μ-strongly convex and is G-Lipschitz in $\mathcal{K}$. Set the parameters as follows: $B = \frac{G}{\mu}$, $\sigma = \frac{B}{1000\sqrt{d}}$, $\psi = 1$. Let $\alpha = \frac{d\sigma}{100\varepsilon n\sqrt{m}}$; note that when $n \ge \sqrt{d}/\varepsilon$, we also have $\alpha^2 \le \frac{d\sigma^2\psi^2}{12m}$ as desired. Thus, we may apply Lemma D.6 with these parameters. This implies that, for any user-level (ε,δ)-DP mechanism $\mathcal{M}$, there must be some truncated Gaussian distribution $\mathcal{D} = \mathcal{N}(\chi, \sigma^2 I_d; B)$ such that

$$\mathop{\mathbb{E}}_{\hat{\chi}\leftarrow\mathcal{M}(\mathcal{D}^{n\times m})}[\|\hat{\chi}-\chi^{\mathrm{tr}}(\chi,\sigma^2 I_d; B)\|^2] \ge \Omega(\alpha^2) = \Omega\left(\frac{G^2}{\mu^2}\cdot\frac{d}{\varepsilon^2 n^2 m}\right).$$

Moreover, the excess (population) risk can be expanded as

$$\mathop{\mathbb{E}}_{\widehat{\theta}\leftarrow\mathcal{M}(\mathcal{D}^{n\times m})}\left[\mathcal{L}(\widehat{\theta};\mathcal{D})\right] - \mathcal{L}(\theta^*;\mathcal{D}) = \frac{\mu}{2}\cdot\mathop{\mathbb{E}}_{\widehat{\theta}\leftarrow\mathcal{M}(\mathcal{D}^{n\times m})}[\|\widehat{\theta}-\chi^{\mathrm{tr}}(\chi,\sigma^2 I_d; B)\|^2] \ge \Omega\left(\frac{G^2}{\mu}\cdot\frac{d}{\varepsilon^2 n^2 m}\right). \qquad \square$$

D.2.2. DP-ERM

The proof for DP-ERM is similar to above, except that we now have to account for the error between the population mean and the empirical mean. We enforce the parameters in such a way that this error is dominated by the lower bound given by Theorem D.7.

Theorem D.8. *There exists a sufficiently small constant $c > 0$ such that the following holds. For any $\varepsilon \in (0, 1], \delta \in (0, 1/2]$ and any sufficiently large $d, n, m \in \mathbb{N}$ such that $cd/\varepsilon^2 \le n \ge \sqrt{d}/\varepsilon$ and $\delta \le \frac{\sqrt{d}}{200\sqrt{n\sqrt{\log n}}}$, there exists a μ-strongly convex G-Lipschitz loss function ℓ such that for any (ε, δ)-DP algorithm, we have*

$$\sup_{\mathcal{D}} \left(\mathop{\mathbb{E}}_{\boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}, \widehat{\theta} \leftarrow \mathcal{M}(\boldsymbol{x})} \left[\mathcal{L}(\widehat{\theta}; \boldsymbol{x}) - \mathcal{L}(\theta^*; \boldsymbol{x}) \right] \right) \ge \Omega\left(\frac{G^2}{\mu} \cdot \frac{d}{\varepsilon^2 n^2 m} \right).$$

In addition to the assumption $n \ge \sqrt{d}/\varepsilon$ as in Theorem D.7, this theorem also requires the assumption $n \le O(d/\varepsilon^2)$. This assumption is required for the error between the empirical mean and the population mean to be small enough to be dominated by the error term $\Omega\left(\frac{G^2}{\mu} \cdot \frac{d}{\varepsilon^2 n^2 m}\right)$.

Proof of Theorem D.8. Let $\ell, B, \sigma, \psi, \alpha$ be exactly as in the setting of Theorem D.7. Similarly, there must exist some truncated Gaussian distribution $\mathcal{D} = \mathcal{N}(\chi, \sigma^2 I_d; B)$ such that, for any (ε, δ)-DP algorithm $\mathcal{M}$, we have

$$\mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\mathcal{D}^{n \times m})} [\|\hat{\chi} - \chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)\|^2] \ge \Omega(\alpha^2) = \Omega\left(\frac{G^2}{\mu^2} \cdot \frac{d}{\varepsilon^2 n^2 m} \right).$$

Let $\hat{\chi}(\boldsymbol{x})$ denote the empirical mean of the dataset $\boldsymbol{x}$. The left hand side can be further expanded as

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\hat{\chi} - \chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)\|^2] \\
&\le 2 \left(\mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\hat{\chi} - \hat{\chi}(\boldsymbol{x})\|^2] + \mathop{\mathbb{E}}_{\boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\hat{\chi}(\boldsymbol{x}) - \chi^{\mathrm{tr}}(\chi, \sigma^2 I_d; B)\|^2] \right) \\
&= 2 \mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\hat{\chi} - \hat{\chi}(\boldsymbol{x})\|^2] + O\left(\frac{B^2}{nm} \right) \\
&= 2 \mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\hat{\chi} - \hat{\chi}(\boldsymbol{x})\|^2] + O\left(\frac{G^2}{\mu^2} \cdot \frac{1}{nm} \right).
\end{aligned}$$

Since we assume that $n \le cd/\varepsilon^2$, we have $\frac{1}{nm} \le c \cdot \frac{d}{\varepsilon^2 n^2 m}$. Therefore, when c is sufficiently small, we can combine the previous two inequalities to conclude that

$$\mathop{\mathbb{E}}_{\hat{\chi} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\chi - \hat{\chi}(\boldsymbol{x})\|^2] \ge \Omega\left(\frac{G^2}{\mu^2} \cdot \frac{d}{\varepsilon^2 n^2 m} \right). \tag{11}$$

Finally, the excess (empirical) risk can be expanded as

$$\mathop{\mathbb{E}}_{\widehat{\theta} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} \left[\mathcal{L}(\widehat{\theta}; \boldsymbol{x}) - \mathcal{L}(\theta^*; \boldsymbol{x}) \right] = \frac{\mu}{2} \cdot \mathop{\mathbb{E}}_{\widehat{\theta} \leftarrow \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \leftarrow \mathcal{D}^{n \times m}} [\|\widehat{\theta} - \hat{\chi}(\boldsymbol{x})\|^2] \overset{(11)}{\ge} \Omega\left(\frac{G^2}{\mu} \cdot \frac{d}{\varepsilon^2 n^2 m} \right).$$

□