# Distributed User-Level Private Mean Estimation

Antonious M. Girgis, Deepesh Data, and Suhas Diggavi
Email: amgirgis@ucla.edu, deepesh.data@gmail.com, suhas@ee.ucla.edu.

***Abstract*—Traditionally, an *item-level* differential privacy framework has been studied for applications in distributed learning. However, when a client has multiple data samples, and might want to also hide its potential participation, a more appropriate notion is that of user-level privacy [1]. In this paper, we develop a distributed private optimization framework that studies the trade-off between *user-level local* differential privacy guarantees and performance. This is enabled by a novel distributed user-level private mean estimation algorithm using distributed private heavy-hitter estimation. We use this result to develop the privacy-performance trade-off for distributed optimization.**

## I. Introduction

Differential privacy (DP) [2] has become the *de facto* standard for measuring the privacy guarantees. When applying to distributed learning settings where data is stored at several client devices (each client may have multiple data points) and a server aims to learn a model, the traditional DP literature focuses on making neighboring datasets indistinguishable, where two datasets are neighbors if they differ in a single data point at a single user. This is called *item-level* DP. However, in distributed learning, a client may not even want to reveal whether it participated or not, which is equivalent to requiring the privacy of its entire local dataset (not just of a single data point). This is called *user-level* DP, which has recently seen some attention [1], [3]–[6].

We can obtain user-level DP from item-level DP by using *group privacy* [7], but this degrades the privacy parameter by a multiplicative factor of the number of data points in a local dataset, which may be impractical. We can achieve a significantly better user-level privacy guarantee by assuming concentration of gradients [5], which essentially reduces their sensitivity and thereby the required noise magnitude.

**Our contributions:** Our distributed learning algorithm is based on distributed private mean estimation that enables clients to privatize their gradients and the server to aggregate them for use in iterative optimization. We present these novel distributed private mean estimation algorithms with user-level privacy, for the scalar case (Algorithm 1), which is used as a building block for the vector case (Algorithm 5). At the core of our algorithms is a method of privately estimating the range of the gradients using the idea of private heavy-hitter estimation. We give the user-level privacy-accuracy trade-off in Theorem 1 (scalar case) & Theorem 2 (vector case). We present its application to distributed learning in Theorem 3. This can be extended using privacy amplification methods (*e.g.,* shuffling or secure aggregation) along with composition theorems (see [8] and Remark 5).

All authors are with the University of California, Los Angeles, USA.
This work was supported in part by NSF grants 2139304 and 2007714.

**Related work:** There has been a lot of recent work in applying *item-level* DP to machine learning algorithms (see [9]–[12] and references therein), and much less work on user-level privacy, with notable exceptions in [1], [3]–[6]. Our algorithms are inspired from that in [5], but with an important distinction that [5] only provide user-level *central* DP guarantees, whereas, our algorithms provide user-level *local* DP guarantees; in distributed learning with an untrusted server, clients need local DP guarantees. Our algorithm is based on distributed private heavy-hitter estimation, whereas it is not clear how the median-based mechanism in [5], could be made distributed.

**Paper organization:** We formulate the problem of mean estimation with user-level LDP and give some preliminaries in Section II. We present our private mean estimation algorithms and the results (both scalar and vector case) in Section III and apply these to an optimization framework in Section IV. We provide the proof outlines of our private mean estimation results in Section V. Proof details are provided in appendices of the full version [8].

## II. Preliminaries and Problem Formulation

Consider a set of $n$ users, each having a local dataset of $m$ samples. Let $\mathcal{D}_i = \{x_1^{(i)}, \dots, x_m^{(i)}\}$ denote the local dataset at the $i$-th user for $i \in [n]$, where $x_j^{(i)} \in \mathcal{X}$ and $\mathcal{X} \subset \mathbb{R}^d$. We define $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_n) \in (\mathcal{X}^m)^n$ as the entire dataset. The users are connected to an untrusted server who wants to estimate the mean $\overline{x} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} x_j^{(i)}$. Users want to preserve the privacy of their local datasets while minimizing the worst-case expected error for estimating $\overline{x}$; see (2).

We first define differential privacy (DP) and the difference between user-level and item-level privacy. We say that two datasets $\mathcal{D}$, $\mathcal{D}'$ are neighboring with respect to distance metric dis if we have $\mathsf{dis}(\mathcal{D}, \mathcal{D}') \leq 1$.

**Definition 1.** (Differential Privacy) Let $\epsilon, \delta \geq 0$. A randomized mechanism $\mathsf{M} : \mathcal{D} \to \Theta$ is said to be $(\epsilon, \delta)$-DP with respect to dis if for any neighboring datasets $\mathcal{D}, \mathcal{D}'$ and any measurable set $\theta \subseteq \Theta$, we have

$$\Pr\left(\mathsf{M}\left(\mathcal{D}\right) \in \theta\right) \leq e^{\epsilon} \Pr\left(\mathsf{M}\left(\mathcal{D}'\right) \in \theta\right) + \delta. \tag{1}$$

If $\delta = 0$, then the privacy is referred to as pure DP.

**Remark 1** ((Central) item-level DP vs (central) user-level DP [5])**.** When we have more than one user (i.e., $n > 1$) and a space $\mathcal{D} \triangleq (\mathcal{X}^m)^n$, by choosing $\mathsf{dis}\left(\mathcal{D}, \mathcal{D}'\right) =$

**Algorithm 1** $\mathsf{Mean}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0, \delta)$: Distributed Private Mean Estimation for Scalars

1: **Inputs:** $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n)$, $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $x_j^{(i)} \in [-B, B]$, concentration radius $\tau$, and user-level LDP parameters $\epsilon_0, \delta$.
2: $[a, b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0/2)$ (Algorithm 3).
3: **for** User $i \in [n]$ **do**
4: $\quad z_i \leftarrow \mathsf{Mean}^{\text{user}}_{\text{scalar}}\left(\mathcal{D}_i, [a, b], \frac{\epsilon_0}{2}, \delta\right)$
5: **Return:** $\hat{x} = \frac{1}{n}\sum_{i=1}^n z_i$.

$\sum_{i=1}^n \sum_{j=1}^m \mathbb{1}\{x_j^{(i)} \neq x_j'^{(i)}\}$, we recover the standard definition of the DP [2], [7], which we call *(central) item-level* DP. In the central item-level DP, two datasets $\mathcal{D}$, $\mathcal{D}'$ are neighboring if they differ in a single item. On the other hand, by choosing $\mathsf{dis}(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^n \mathbb{1}\{\mathcal{D}_i \neq \mathcal{D}_i'\}$, we call it *(central) user-level* DP, where two datasets $\mathcal{D}, \mathcal{D}' \in (\mathcal{X}^m)^n$ are neighboring when they differ in a local dataset of any single user. Observe that when each user has a single item ($m = 1$), then both item-level and user-level privacy are equivalent.

**Remark 2** (User-level Local Differential Privacy (LDP))**.** When we have a single user (i.e., $n = 1$ and $\mathcal{D} = \mathcal{X}^m$), by choosing $\mathsf{dis}(\mathcal{D}, \mathcal{D}') = \mathbb{1}\{\mathcal{D} \neq \mathcal{D}'\}$ for $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^m$, we call it *user-level LDP*. In this case each user privatize her own local dataset using a private mechanism.

Our objective is to design user-level LDP mechanisms $\mathsf{M}_i : \mathcal{X}^m \to \Theta_i$ for $i \in [n]$ and an estimator $\hat{x} : \Theta_1 \times \ldots \times \Theta_n \to \mathcal{X}$ to minimize the worst-case expected error:

$$R_{\epsilon,\delta} = \inf_{\{\mathsf{M}_i \in \mathcal{M}_{\epsilon,\delta}\}} \inf_{\hat{x}} \sup_{\mathcal{D} \in (\mathcal{X}^m)^n} \mathbb{E}\left[\|\hat{x} - \overline{x}\|^2\right], \tag{2}$$

where $\mathcal{M}_{\epsilon,\delta}$ denotes the set of all possible user-level $(\epsilon, \delta)$-LDP mechanisms, and the expectation is taken over the randomness in $\mathsf{M}_1, \ldots, \mathsf{M}_n$ and $\hat{x}$.

As mentioned in Section I, we can significantly improve the user-level privacy guarantees (beyond what can be achieved by applying the group privacy) by assuming concentration of the input vectors.

Now, we define the concentration condition for a set of samples and the sub-Gaussian random vector.

**Definition 2** (Concentration)**.** A set of (random) vectors $y^n = (y_1, \ldots, y_n)$, each taken from $[-B, B]^d$ is $(\tau, \gamma)$-concentrated if there exists $y_0 \in [-B, B]^d$ such that with probability at least $1 - \gamma$,

$$\max_{i \in [n]} \|y_i - y_0\|_2 \leq \tau. \tag{3}$$

**Definition 3** (Sub-Gaussian random vector)**.** A random vector $x \in \mathbb{R}^d$ is said to be sub-Gaussian with proxy variance $\sigma^2$ if for any $u \in \mathbb{R}^d$ with $\|u\|^2 = 1$, the random variable $u^T x$ is sub-Gaussian with proxy variance $\sigma^2$.

Throughout this paper, we assume that the samples $\{x_j^{(i)} : i \in [n], j \in [m]\}$ are drawn from a bounded space $\mathcal{X} \triangleq [-B, B]^d \subset \mathbb{R}^d$ for some $d \geq 1$. Furthermore, we assume

**Algorithm 2** $\mathsf{Mean}^{\text{user}}_{\text{scalar}}(\mathcal{D}, [a, b], \epsilon_0, \delta)$

1: **Inputs:** $\mathcal{D} = (x_1, \ldots, x_m)$, concentration range $[a, b]$, and user-level LDP parameters $\epsilon_0, \delta$.
2: Sample $\nu \sim \mathcal{N}(0, \frac{12(b-a)^2 \log(1.25/\delta)}{\epsilon_0^2})$.
3: **Return:** $z = \prod_{[a,b]} y + \nu$, where $y = \frac{1}{m}\sum_{j=1}^m x_j$ and $\prod_{[a,b]}$ is the projection operator onto $[a, b]$.

that the samples $x_j^{(i)}, i \in [n], j \in [m]$ are i.i.d. sub-Gaussian random vectors with proxy variance $\sigma^2$.

## III. Private Mean Estimation

In this section, we present our distributed user-level LDP mechanism to estimate the mean $\overline{x}$. We start with the scalar case when $d = 1$ in Section III-A. Then, we extend our algorithm for $d$-dimensional space in Section III-B.

### A. Scalar Case

Suppose $x_j^{(i)} \in [-B, B]$ for all $i \in [n]$ and $j \in [m]$. Furthermore, the samples $x_j^{(i)}$ are i.i.d. sub-Gaussian with proxy $\sigma^2$. Let $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ denote the mean of the local samples at the $i$-th user for $i \in [n]$. Thus, $\{y_i\}$ are sub-Gaussian random variables with proxy $\frac{\sigma^2}{m}$ which implies that the set $y^n = (y_1, \ldots, y_n)$ is $(\tau, \gamma)$-concentrated, where $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$ for any $\gamma \in (0, 1)$ (e.g., see [13, Theorem 1.14]).

The mean estimation process works in two stages similar to [5]. In the first stage, the server privately estimates the range in which the means $y_1, \ldots, y_n$ lie with high probability. In the second stage, each user projects her mean value $y_i$ into the determined range from the first step. Then, all users send user-level LDP versions of their projected samples to the central server. The first stage mechanism is denoted by $\mathsf{Range}_{\text{scalar}}$ and is presented in Algorithm 3, and the second stage mechanism is denoted by $\mathsf{Mean}_{\text{scalar}}$ and is presented in Algorithm 1. We give an outline of both these algorithms below.

In $\mathsf{Range}_{\text{scalar}}$ we first divide the original range $[-B, B]$ into $k = B/\tau$ bins, where $\tau$ is the concentration parameter of $y_1, \ldots, y_n$. Then, each user sends a private version of the closest bin to her mean value $y_i$ (using the mechanism $\mathsf{Range}^{\text{user}}_{\text{scalar}}$ as described in Algorithm 4). The server estimates the frequencies (the number of means close to each bin) under user-level LDP constraints. We use a Hadamard Response mechanism similar to the one proposed in [14] to estimate the highest frequency under user-level LDP constraints. Observe that if the means $(y_1, \ldots, y_n)$ lie in radius $\tau$ and the server succeeds to estimate the highest frequency correctly, then we get $y_i \in R \triangleq [a_{\max} - 3\tau, a_{\max} + 3\tau]$ for all $i \in [n]$. In $\mathsf{Mean}_{\text{scalar}}$, each client projects her mean $y_i$ onto the estimated range $R$ from the first stage. The objective of this projection is that the user-level sensitivity will decrease from $2B$ to $2\tau$, where $\tau = \mathcal{O}(\frac{1}{\sqrt{m}})$. In other words, the user-level sensitivity will decrease by increasing the number of samples per user using this projection step. After the projection, each user adds

a Gaussian noise with a variance function of the user-level sensitivity ($\tau$) and LDP parameter $\epsilon_0$ to preserve privacy.

**Theorem 1.** *The mechanism* $\mathsf{Mean}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0, \delta)$ *is user-level* $(\epsilon_0, \delta)$*-LDP. Furthermore, if* $\{x_j^{(i)}\}$ *are sub-Gaussian with proxy* $\sigma^2$*, then* $y^n = (y_1, \ldots, y_n)$ *are* $(\tau, \gamma)$*-concentrated, where* $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ *and* $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$*. With probability at least* $1-\beta$*, we have*

$$\mathcal{E}_1 := \mathbb{E}\left[\left|\frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m x_j^{(i)} - \mathsf{Mean}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0, \delta)\right|^2\right] \leq \mathcal{O}\left(\frac{\tau^2\log(1/\delta)}{n\epsilon_0^2}\right), \quad (4)$$

*where* $\beta = \min\left\{1, \gamma + \frac{2B}{\tau}\exp\left(-\frac{n(e^{\epsilon_0/2}-1)^2}{200(e^{\epsilon_0/2}+1)^2}\right)\right\}$.

We provide a proof of Theorem 1 in Section V. Observe that Theorem 1 provides privacy-utility trade-offs for $\epsilon_0 < 1$. However, we can obtain similar results for general $\epsilon_0$ by adapting the variance of the Gaussian noise using the results in [15], [16]

**Remark 3** (Gaussian vs. Laplace Noise)**.** In $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$, users add Gaussian noise to achieve user-level $(\epsilon_0, \delta)$-LDP. Instead of Gaussian noise, we can add a Laplace noise $\mathsf{Lap}(\frac{12\tau}{\epsilon_0})$ to get a pure user-level $\epsilon_0$-LDP with the same estimation error as (4) in Theorem 1.

**Remark 4** (User-level LDP vs user-level DP)**.** In [5], the authors proposed a (central) user-level DP mean estimation algorithm that achieves estimation error $\mathcal{O}(\frac{\tau^2}{n^2\epsilon^2})$ with probability $(1-\beta_c)$, where $\beta_c = \min\{1, \gamma + \frac{B}{\tau}e^{-\frac{n\epsilon}{8}}\}$ and $\epsilon$ is the (central) DP parameter. Although, the confidence probability $1-\beta$ is almost same for both user-level LDP and user-level DP, it is clear that there is a gap of $\mathcal{O}(n)$ in the estimation error between the central and the local models. This is not surprising as the same gap appears in the item-level DP and LDP as well [17], [18]. In order to amplify the privacy of the user-level LDP to match with that of the user-level DP, we can assume the existence of a trusted shuffler [19]–[21] or secure aggregation [22] between the users and the untrusted server. See Appendix D for more details.

### *B. Vector Case*

In this section, we present the user-level LDP mechanism for general $d$ dimensional spaces. We assume that the samples $x_j^{(i)} \in \mathcal{X} \triangleq [-B, B]^d$ for all $i \in [n]$ and $j \in [m]$. Furthermore, the samples $x_j^{(i)}$ are sub-Gaussian random vector with proxy $\sigma^2$. Let $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ denote the mean of the local samples at the $i$-th user for $i \in [n]$. Thus, $y_i$ are sub-Gaussian random vectors with proxy $\frac{\sigma^2}{m}$ which implies that the set $y^n = (y_1, \ldots, y_n)$ are $(\tau, \gamma)$-concentrated, where $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$ and $\gamma > 0$ is arbitrary [13], [23].

We follow similar steps as in the centralized Algorithm presented in [5] for user-level DP mean estimation. The idea

**Algorithm 3** $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0)$: Distributed Private Range Estimation for Scalars

1: **Inputs:** $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n)$, $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $x_j^{(i)} \in [-B, B]$, concentration radius $\tau$, and user-level LDP parameter $\epsilon_0$.
2: All users divide the interval $[-B, B]$ into $k = B/\tau$ disjoint intervals, each with width $2\tau$. Let $\mathcal{T} := \{1, 2, \ldots, k\}$ be the set of middle points of intervals.
3: **for** User $i \in [n]$ **do**
4: $\quad \mathbf{z}_i \leftarrow \mathsf{Range}^{\text{user}}_{\text{scalar}}(\mathcal{D}_i, \tau, \epsilon_0, \mathcal{T})$.
5: $\quad$ Send $\mathbf{z}_i$ to the server – here $\mathbf{z}_i \in \mathbb{R}^k$.
6: The server computes $\overline{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i$. (Here, for any $a \in \mathcal{T}$, $\overline{\mathbf{z}}(a)$ denotes an estimate of the frequency of $a$, i.e., the fraction of $y_i$'s that are closest to $a$).
7: Let $a_{\max} = \arg\max_{a\in\mathcal{T}} \overline{\mathbf{z}}(a)$.
8: **Return:** $R = [a_{\max} - 3\tau, a_{\max} + 3\tau]$

of the private mean estimation Algorithm is to observe that the means $y_1, \ldots, y_n$ are concentrated in $\ell_2$-norm with radius $\tau$. Similar to [5], we first apply an encoding step to bound them in $\ell_\infty$-norm with radius $\mathcal{O}(\frac{\tau}{\sqrt{d}})$. This step can be obtained by applying a random rotation as in [5], [10] or by applying Kashin's representation as in [11]. Then, we apply the scalar Algorithm 3 for each coordinate separately. The private mean estimation for $d$-dimensional vectors is denoted by $\mathsf{Mean}_{\text{vector}}$ and is presented in Algorithm 5.

**Theorem 2.** *The mechanism* $\mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)$ *is user-level* $(\epsilon_0, \delta)$*-LDP. Furthermore, if* $\{x_j^{(i)}\}$ *are sub-Gaussian random vectors with proxy* $\sigma^2$*, then* $y^n = (y_1, \ldots, y_n)$ *are* $(\tau, \gamma)$*-concentrated, where* $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ *and* $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$*. With probability* $1-\beta$*, we have*

$$\mathcal{E}_2 := \mathbb{E}\left[\left\|\frac{1}{mn}\sum_{i=1}^n\sum_{j=1}^m x_j^{(i)} - \mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)\right\|^2\right] \leq \mathcal{O}\left(\frac{\tau^2 d\log(dn/\gamma)\log(1/\delta)}{n\epsilon_0^2}\right), \quad (5)$$

*where* $\beta = \min\left\{1, 2\gamma + \zeta\right\}$, $\epsilon_0' = \frac{\epsilon_0}{2d}$, *and* $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{n(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.

We provide a proof of Theorem 2 in Section V.

## IV. Empirical Risk Minimization

In this section, we present an application of the private mean estimation algorithms under user-level LDP constraints to Federated Learning (FL), where a set of $n$ users are connected to a central server to solve the following empirical risk minimization (ERM) problem:

[1]We assume that $k$ is a power of 2. Otherwise we assume the size of $\mathcal{T}$ is $K = 2^{\lceil\log_2(k)\rceil}$ (the smallest power of 2 larger than $k$).

**Algorithm 4** $\mathsf{Range}^{\text{user}}_{\text{scalar}}\left(\mathcal{D}, \tau, \epsilon_0, \mathcal{T}\right)$

**Inputs:** $\mathcal{D} = (x_1, \ldots, x_m)$, $x_j \in [-B, B]$; concentration radius $\tau$; user-level LDP parameter $\epsilon_0$; $\mathcal{T} = [k]$ be the set of middle points of the intervals.
Compute $y = \frac{1}{m}\sum_{j=1}^{m} x_j$.
Compute $\nu = \arg\min_{j\in[k]} |y - a_j|$ (the index of a point in $\mathcal{T}$ closest to $y$).
Let $\mathbf{H}_k$ be Hadamard matrix.[1]
Compute $\mathbf{m} = \frac{1}{\sqrt{k}}\mathbf{H}_k e_\nu$, where $e_\nu$ denotes the basis vector corresponding to $\nu$.
Sample $j \sim \mathsf{Unif}[k]$ and compute $\mathbf{z}$:

$$\mathbf{z} = \begin{cases} +\mathbf{H}_k(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} + \frac{\sqrt{k}\mathbf{m}(j)}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \\ -\mathbf{H}_k(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} - \frac{\sqrt{k}\mathbf{m}(j)}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \end{cases}$$

**Return:** $\mathbf{z}$

$$\arg\min_{\theta\in\mathcal{C}} \left(F\left(\theta, \mathcal{D}\right) = \frac{1}{n}\sum_{i=1}^{n} F_i\left(\theta, \mathcal{D}_i\right)\right). \tag{6}$$

Here, $\mathcal{C} \subset \mathbb{R}^d$ is a closed convex set. Each user has a local dataset $\mathcal{D}_i = \{d_{i1}, \ldots, d_{im}\}$ of $m$ samples and $F_i\left(\theta, \mathcal{D}_i\right) = \frac{1}{m}\sum_{j=}^{m} f\left(d_{ij}, \theta\right)$ denotes the loss function at the $i$-th user local dataset $\mathcal{D}_i$ w.r.t. the model $\theta$. Our goal is to solve the ERM problem in (6) while providing user-level privacy.

In Algorithm 6, we propose a user-level local DP stochastic gradient descent (ULDP-SGD). At each iteration of the ULDP-SGD, we choose uniformly at random a set of $k$ users. Each user applies a full gradient of the local function $F_i\left(\theta\right)$. Then, the server estimates under user-level LDP constraints the average of gradients $\frac{1}{km}\sum_{i\in\mathcal{U}_t}\sum_{j=1}^{m}\nabla_{\theta_t} f\left(\theta_t; d_{ij}\right)$ and takes a descent step.

We state the (per-iteration) privacy and convergence results of Algorithm 6 below and prove it in Appendix C.

**Theorem 3.** *Let the set $\mathcal{C}$ be convex with diameter $D$ and the function $f\left(\theta; .\right) : \mathcal{C} \to \mathbb{R}$ be convex and $L$-Lipschitz continuous with respect to the $\ell_2$-norm. Let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F\left(\theta\right)$ denote the minimizer of the problem (6). The Algorithm $\mathsf{A}_{\text{uldp}}$ is user-level $(\epsilon_0, \delta)$-LDP per iteration. If we run Algorithm $\mathsf{A}_{\text{uldp}}$ over $T$ iterations with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, then*

$$\mathbb{E}\left[F\left(\theta_{T+1}\right)\right] - F\left(\theta^*\right) = \mathcal{O}\left(DG\frac{\log(T)}{\sqrt{T}}\right) \tag{7}$$

*with probability at least $1 - \beta_T$, where $G = L\sqrt{\left(1 + \frac{cd\log(dn/\gamma)\log(1/\delta)\log(2n/\gamma)}{qnm\epsilon_0^2}\right)}$, $c = 28800$, $q = \frac{k}{n}$, $\gamma = \frac{1}{n^{\log(n)+2}}$, $\beta_T = \min\{1, 2T\gamma + T\zeta\}$, $\epsilon_0' = \frac{\epsilon_0}{2d}$, and $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{qn(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.*

**Remark 5** (Privacy guarantee)**.** Theorem 3 states the per-iteration user-level local DP guarantee of Algorithm 6. With slight modifications in our mean estimation algorithm $\mathsf{Mean}_{\text{vector}}$, we can apply shuffling or secure aggregation to convert these to user-level central DP guarantees [22], [24]. Then, in order to obtain the privacy guarantee of our entire algorithm, we can either use the strong composition theorem [7] or use the Rényi DP guarantees which provides better composition bounds [12], [21]. See Appendix D for more details.

**Algorithm 5** $\mathsf{Mean}_{\text{vector}}\left(\mathcal{D}, \tau, \epsilon_0, \delta\right)$: Distributed Private Mean Estimation for Vectors

**Inputs:** $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n)$, $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $x_j^{(i)} \in [-B, B]^d$, concentration radius $\tau$, and user-level LDP parameters $\epsilon_0, \delta$.
Let $\mathbf{D} = \mathsf{Diag}(w)$, where $w \sim \mathsf{Unif}\{-1, 1\}$.
Compute $\mathbf{U} = \frac{1}{\sqrt{d}}\mathbf{H}_d\mathbf{D}$
Set $\epsilon_0' = \frac{\epsilon_0}{2d}$ and $\tau' = 10\tau\sqrt{\frac{\log(nd/\gamma)}{d}}$.
**for** $l \in [d]$ **do**
  **for** User $i \in [n]$ **do**
    Compute $\hat{x}_j^{(i)}(l) = (\mathbf{U}x_j^{(i)})(l)$ (the $l$-th coordinate of the vector $\mathbf{U}x_j^{(i)}$ ) for $j \in [m]$.
    Let $\overline{\mathcal{D}}_{i,l} = (\hat{x}_1^{(i)}(l), \ldots, \hat{x}_m^{(i)}(l))$.
  Let $\overline{\mathcal{D}}_l = (\overline{\mathcal{D}}_{1,l}, \ldots, \overline{\mathcal{D}}_{n,l})$.
  $R(l) \leftarrow \mathsf{Range}_{\text{scalar}}\left(\overline{\mathcal{D}}_l, \tau', \epsilon_0'\right)$
**for** User $i \in [n]$ **do**
  Sample $j \sim \mathsf{Unif}[d]$.
  Let $z_i := [0, \ldots, 0, d \times z_i(j), 0, \ldots, 0]$, which has a non-zero element in the $j$-th location.
  $z_i(j) \leftarrow \mathsf{Mean}^{\text{user}}_{\text{scalar}}\left(\overline{\mathcal{D}}_{i,j}, R(j), \epsilon_0/2, \delta\right)$.
Compute $\hat{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$
**Return:** $\hat{x} = \mathbf{U}^{-1}\hat{z}$.

## V. Proofs of Theorem 1 and Theorem 2

*Proof of Theorem 1.* The algorithm $\mathsf{Mean}_{\text{scalar}}$ is composed of two sub-routines $\mathsf{Range}_{\text{scalar}}$ and $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$. In order to show that $\mathsf{Mean}_{\text{scalar}}$ satisfies user-level $(\epsilon_0, \delta)$-LDP, it suffices to prove that $\mathsf{Range}_{\text{scalar}}$ satisfies user-level $(\epsilon_0/2, 0)$-LDP and $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$ satisfies user-level $(\epsilon_0/2, \delta)$-LDP, and then the result follows by composing these two mechanisms.

- $\mathsf{Range}_{\text{scalar}}$ is user-level $(\epsilon_0/2, 0)$-LDP: We show this along with other results that will be useful to bound the error in the following lemma (which is proved in Appendix A).

**Lemma 1.** $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0)$ *is user-level $\epsilon_0$-LDP. Furthermore, if the samples $x_j^{(i)}$ are sub-Gaussian with proxy $\sigma^2$, then with probability at least $1 - \beta$, we have*

$$y_i \in [a, b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0) \qquad \forall i \in [n] \tag{8}$$

*where $y_i = \frac{1}{m}\sum_{j=1}^{m} x_j^{(i)}$ is the average of local samples at the $i$-th user, and $\beta = \min\left\{1, \gamma + \frac{2B}{\tau}\exp\left(-\frac{n(e^{\epsilon_0/2}-1)^2}{200(e^{\epsilon_0/2}+1)^2}\right)\right\}$.*

This lemma shows that with probability at least $1 - \beta$, the server can privately estimate an interval of length $6\tau$ in which the averages $y_1, \ldots, y_n$ of local samples at all users lie. Thus, each user can project the average of her local samples onto this interval without hurting the estimation accuracy of the

second stage. Furthermore, the sensitivity of replacing a user with another one would be $6\tau = \mathcal{O}(\frac{1}{\sqrt{m}})$ instead of $2B$. As a result, each user adds a noise as a function of $\tau$ that reduces the estimation error.

• $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$ is user-level $(\epsilon_0/2, \delta)$-LDP: Consider any two neighboring local datasets $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $\mathcal{D}'_i = (x_1'^{(i)}, \ldots, x_m'^{(i)})$. Let $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ denotes the average of local samples in $\mathcal{D}$; similarly define $y'_i$. The user-level sensitivity for computing its projection $\prod_{[a,b]} y_i$ is bounded by

$$\Delta_2 y_i = \sup_{\mathcal{D}_i, \mathcal{D}'_i \in [-B,B]^m} \left| \prod_{[a,b]}(y_i) - \prod_{[a,b]}(y'_i) \right| \leq (b-a).$$

Thus, from [7, Theorem 3.22] we get that by setting $\sigma^2 = \frac{12(b-a)^2 \log(1.25/\delta)}{\epsilon_0^2}$, and the output $\mathbf{z}_i = \prod_{[a,b]}(y_i) + \nu_i$ satisfies user-level $\left(\frac{\epsilon_0}{2}, \delta\right)$-LDP.

Bounding the error of $\mathsf{Mean}_{\text{scalar}}$: Let $[a,b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0/2)$ and $\tilde{y}_i = \Pi_{[a,b]} y_i$. Note that $(b-a) = 6\tau$. Let $\hat{x} = \frac{1}{n}\sum_{i=1}^n z_i$ be the estimator of the exact mean $\overline{x} = \frac{1}{n}\sum_{i=1}^n y_i$. Thus, we have

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^n \tilde{y}_i - \frac{1}{n}\sum_{i=1}^n z_i\right|^2\right] = \mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^n \nu_i\right|^2\right] = \frac{\sigma^2}{n}$$
$$= \frac{432\tau^2 \log(1.25/\delta)}{n\epsilon_0^2} = \mathcal{O}\left(\frac{\tau^2 \log(1/\delta)}{n\epsilon_0^2}\right).$$

From Lemma 1, we have that $y_i = \tilde{y}_i$ for all $i \in [n]$ with probability at least $1-\beta$. Thus, we get that, with probability at least $1-\beta$, the error $\mathcal{E}_1$ (defined in (4)) is bounded by $\mathcal{E}_1 = \mathcal{O}\left(\frac{\tau^2 \log(1/\delta)}{n\epsilon_0^2}\right)$, This completes the proof of Theorem 1. ■

*Proof of Theorem 2.* First we prove that $\mathsf{Mean}_{\text{vector}}$ (described in Algorithm 5) is user-level $(\epsilon_0, \delta)$-LDP. Observe that we run $\mathsf{Range}_{\text{scalar}}$ for each coordinate with privacy parameter $\epsilon'_0 = \epsilon_0/2d$. Then, we user chooses uniformly at random one coordinate to apply the Algorithm $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$ with privacy parameters $\epsilon_0/2, \delta$. Thus, by composition, we get that the Algorithm 5 is user-level $(\epsilon_0, \delta)$-LDP.

In order to prove the error bound, we follow the same steps as in [5]. We first state a lemma from [5] about $\ell_\infty$-norm concentration using random rotation. The proof then is obtained by combining this with the scalar case properties.

**Lemma 2** (Random rotation [5]). *Let $\mathbf{U} = \frac{1}{\sqrt{d}}\mathbf{H}_d\mathbf{D}$, where $\mathbf{H}_d$ be Hadamard matrix and $\mathbf{D}$ be a diagonal matrix with i.i.d. uniformly random $\{\pm 1\}$ entries. Let $x_1, \ldots, x_n, x_0 \in \mathbb{R}^d$. With probability at least $1-\gamma$, we have*

$$\max_{i\in[n]} \|\mathbf{U}x_i - \mathbf{U}x_0\|_\infty \leq \frac{10 \max_i \|x_i - x_0\|_2 \sqrt{\log(dn/\gamma)}}{\sqrt{d}}$$

Since $\{x_j^{(i)}\}$ are i.i.d. sub-Gaussian random vectors with proxy variance $\sigma^2$, we have that $y_1, \ldots, y_n$ are $(\tau, \gamma)$-concentrated, where $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$. Thus, from Lemma 2, we have that $\max_{i\in[n]} \|\tilde{y}_i - \tilde{y}_0\|_\infty \leq \tau'$ with probability at least $1-2\gamma$, where $\tilde{y}_i = \frac{1}{m}\sum_{j=1}^m \mathbf{U}x_j^{(i)}$, $\tilde{y}_0 = \mathbf{U}y_0$ ($y_0$ is the center of concentration), and $\tau' = \frac{10\tau\sqrt{\log(dn/\gamma)}}{\sqrt{d}}$. Thus, from Lemma 1, we get that with probability at least $1-\beta'$, we have

$$\tilde{y}_i(l) \in R(l) \leftarrow \mathsf{Range}_{\text{scalar}}(\overline{\mathcal{D}}_l, \tau', \epsilon'_0), \forall i \in [n], l \in [d] \quad (9)$$

where $\beta' = \min\left\{1, 2\gamma + \frac{2\sqrt{d}B}{\tau'} \exp\left(-\frac{n(e^{\epsilon'_0/2}-1)^2}{200(e^{\epsilon'_0/2}+1)^2}\right)\right\}$. Condition on the event that $\tilde{y}_i(l) \in R(l)$ for all $l \in [d]$, $i \in [n]$. Let $z_i$ denote a r.v. taking values in $\{dz_i(1) \cdot e_1, \ldots, dz_i(d) \cdot e_d\}$ uniformly at random, where $z_i(l) = \mathsf{Mean}^{\text{user}}_{\text{scalar}}(\overline{\mathcal{D}}_{i,j}, R(j), \epsilon_0/2, \delta)$ and $e_1, \ldots, e_d$ are the basis vectors in $\mathbb{R}^d$. Note that $dz_i(l) \cdot e_l$ is a length $d$ vector whose $l$'th component is equal to $dz_i(l)$ and all other components are equal to zero.

It is easy to see that $\mathbb{E}[z_i] = \tilde{y}_i$ for all $i \in [n]$ and that $z_i$ has bonded variance, i.e., $\mathbb{E}\|z_i - \tilde{y}_i\|^2 = \mathcal{O}\left(\frac{d^2(\tau')^2 \log(1/\delta)}{\epsilon_0^2}\right)$ – proof is in Appendix B.

Since $\tilde{y}_1, \ldots, \tilde{y}_n$ are independent we have that

$$\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n (z_i - \tilde{y}_i)\right\|^2 = \mathcal{O}\left(\frac{d^2(\tau')^2 \log(1/\delta)}{n\epsilon_0^2}\right) \quad (10)$$

Thus, we have that the error $\mathcal{E}_2$ (defined in (5)) is bounded by

$$\mathcal{E}_2 = \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n (\tilde{y}_i - z_i)\right\|^2 \leq \mathcal{O}\left(\frac{\tau^2 d \log(dn/\gamma)\log(1/\delta)}{n\epsilon_0^2}\right).$$

From the union bound, we have that $\tilde{y}_i(l) \in R(l)$ for all $l \in [d]$, $i \in [n]$ with probability at least $1-\beta$, where $\beta = \min\left\{1, 2\gamma + \zeta\right\}$, $\epsilon'_0 = \frac{\epsilon_0}{2d}$, and $\zeta = \frac{2d^2 B\sqrt{\log(dn/\gamma)}}{\tau} \exp\left(-\frac{n(e^{\epsilon'_0}-1)^2}{200(e^{\epsilon'_0}+1)^2}\right)$. This completes the proof of Theorem 2. ■

**Algorithm 6** $\mathsf{A}_{\text{uldp}}$: ULDP-SGD

**Inputs:** Datasets $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n)$, where $\mathcal{D}_i = \{d_{i1}, \ldots, d_{im}\}$ for $i \in [n]$; user-level LDP privacy parameters $\epsilon_0, \delta$; gradient norm bound $C$; and learning rate schedule $\{\eta_t\}$.
**Initialize:** $\theta_0 \in \mathcal{C}$
**for** $t \in [T]$ **do**
  **Sampling of users:** A uniformly random set $\mathcal{U}_t$ of $k$ users is chosen.
  **for** users $i \in \mathcal{U}_t$ **do**
    **for** $j = 1, 2, \ldots, m$ **do**
      Compute gradient: $x_j^{(i)} \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij}) \,/\, \max\left\{1, \frac{\|\nabla_{\theta_t} f(\theta_t; d_{ij})\|_2}{C}\right\}$
    Let $\tilde{\mathcal{D}}_i := \left(x_1^{(i)}, \ldots, x_m^{(i)}\right)$
  Let $\tilde{\mathcal{D}} = \left(\tilde{\mathcal{D}}_i : i \in \mathcal{U}_t\right)$
  Let $\gamma \leftarrow \frac{1}{n^{\log(n)+2}}$ and $\tau \leftarrow \frac{C\sqrt{\log(2n/\gamma)}}{m}$
  **Aggregate:** $\overline{\mathbf{g}}_t \leftarrow \mathsf{Mean}_{\text{vector}}\left(\tilde{\mathcal{D}}, \tau, \epsilon_0, \delta\right)$
  **Gradient Descent** $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}\left(\theta_t - \eta_t \overline{\mathbf{g}}_t\right)$, where $\prod_{\mathcal{C}}$ denotes the projection operator onto $\mathcal{C}$.
**Output:** The model $\theta_T$.

## REFERENCES


[1] H. B. McMahan, G. Andrew, U. Erlingsson, S. Chien, I. Mironov, N. Papernot, and P. Kairouz, "A general approach to adding differential privacy to iterative training procedures," *arXiv preprint arXiv:1812.06210*, 2018.

[2] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, 2006, pp. 265–284.

[3] Y. Liu, A. T. Suresh, F. X. X. Yu, S. Kumar, and M. Riley, "Learning discrete distributions: user vs item-level privacy," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 965–20 976, 2020.

[4] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2512–2520.

[5] D. Levy, Z. Sun, K. Amin, S. Kale, A. Kulesza, M. Mohri, and A. T. Suresh, "Learning with user-level privacy," *arXiv preprint arXiv:2102.11845*, 2021.

[6] B. Ghazi, R. Kumar, and P. Manurangsi, "User-level differentially private learning via correlated sampling," in *Advances in Neural Information Processing Systems*, 2021.

[7] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[8] A. M. Girgis, D. Data, and S. Diggavi, "Distributed user-level private mean estimation," 2022, available online on arXiv.

[9] A. M. Girgis, D. Data, S. N. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of federated learning: Privacy, accuracy and communication trade-offs," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 464–478, 2021. [Online]. Available: https://doi.org/10.1109/JSAIT.2021.3056102

[10] A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3329–3337.

[11] W. Chen, P. Kairouz, and A. Özgür, "Breaking the communication-privacy-accuracy trilemma," in *Annual Conference on Neural Information Processing Systems*, 2020.

[12] A. Girgis, D. Data, and S. Diggavi, "Renyi differential privacy of the subsampled shuffle model in distributed learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[13] P. Rigollet, "High dimensional statistics," *Lecture Notes, Cambridge, MA, USA: MIT Open-CourseWare*, 2015.

[14] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1120–1129.

[15] B. Balle and Y.-X. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *International Conference on Machine Learning*. PMLR, 2018, pp. 394–403.

[16] S. A. Vinterbo, "A closed form scale bound for the $(\epsilon, \delta)$-differentially private gaussian mechanism valid for all privacy regimes," *arXiv preprint arXiv:2012.10523*, 2020.

[17] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: Simultaneously solving how and what," in *Annual International Cryptology Conference*. Springer, 2008, pp. 451–468.

[18] T. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *European Symposium on Algorithms*. Springer, 2012, pp. 277–288.

[19] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2019, pp. 2468–2479.

[20] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," *arXiv preprint arXiv:2012.12803*, 2020, open source implementation of privacy https://github.com/apple/ml-shuffling-amplification

[21] A. M. Girgis, D. Data, S. N. Diggavi, A. T. Suresh, and P. Kairouz, "On the rényi differential privacy of the shuffle model," in *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 2021, pp. 2321–2341. [Online]. Available: https://doi.org/10.1145/3460120.3484794

[22] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," *arXiv preprint arXiv:2102.06387*, 2021.

[23] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018, vol. 47.

[24] A. M. Girgis, D. Data, S. N. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, 2021, pp. 2521–2529. [Online]. Available: http://proceedings.mlr.press/v130/girgis21a.html

[25] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[26] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International conference on machine learning*, 2013, pp. 71–79.

[27] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," in *Annual International Cryptology Conference*. Springer, 2019, pp. 638–667.

[28] J. Ullman, "Cs7880. rigorous approaches to data privacy," 2017. [Online]. Available: http://www.ccs.neu.edu/home/jullman/cs7880s17/HW1sol.pdf

## APPENDIX A
## OMITTED DETAILS FROM SECTION V: PROOF OF LEMMA 1

**Lemma** (Restating Lemma 1)**.** $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0)$ *is user-level* $\epsilon_0$*-LDP. Furthermore, if the samples* $x_j^{(i)}$ *are sub-Gaussian with proxy* $\sigma^2$*, then with probability at least* $1-\beta$*, we have*

$$y_i \in [a,b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0) \qquad \forall i \in [n] \tag{11}$$

*where* $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ *is the average of local samples at the* $i$*-th user, and* $\beta = \min\left\{1, \gamma + \frac{2B}{\tau}\exp\left(-\frac{n(e^{\epsilon_0/2}-1)^2}{200(e^{\epsilon_0/2}+1)^2}\right)\right\}$.

*Proof.* In order to prove that $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0)$ is user-level $\epsilon_0$-LDP, it suffices to show that $\mathsf{Range}^{\text{user}}_{\text{scalar}}(\mathcal{D}, \tau, \epsilon_0)$ is user-level $\epsilon_0$-LDP.

Consider an arbitrary user $i \in [n]$ and two local datasets $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $\mathcal{D}_i' = (x_1^{'(i)}, \ldots, x_m^{'(i)})$. Let $\mathcal{Z} = \{\pm \mathbf{H}_k(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) : j \in \{1, \cdots, k\}\}$ denote all possible outputs of the mechanism $\mathsf{R}_{range}$. Thus, we get

$$\begin{aligned}
\sup_{\mathcal{D}_i, \mathcal{D}_i' \in [-B,B]^m} \sup_{z \in \mathcal{Z}} \frac{\Pr\left[\mathsf{Range}_{\text{scalar}}\left(\mathcal{D}_i\right) = z\right]}{\Pr\left[\mathsf{Range}_{\text{scalar}}\left(\mathcal{D}_i'\right) = z\right]} &\leq \sup_{\mathcal{D}_i, \mathcal{D}_i' \in [-B,B]^m} \frac{\frac{1}{k}\sum_{j=1}^k \frac{1}{2} + \frac{\sqrt{k}|\mathbf{m}_i(j)|}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}}{\frac{1}{k}\sum_{j=1}^k \frac{1}{2} - \frac{\sqrt{k}|\mathbf{m}_i'(j)|}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}} \\
&\overset{\text{(a)}}{\leq} \frac{\frac{1}{k}\sum_{j=1}^k \frac{1}{2} + \frac{1}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}}{\frac{1}{k}\sum_{j=1}^k \frac{1}{2} - \frac{1}{2}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}} \\
&\leq e^{\epsilon_0}
\end{aligned} \tag{12}$$

where the step (a) is obtained from the fact that $\mathbf{m}_i(j), m_i'(j) \in \{\pm\frac{1}{\sqrt{k}}, \}$. Thus, the private range mechanism $\mathsf{Range}^{\text{user}}_{\text{scalar}}$ is user level $(\epsilon_0, 0)$-LDP.

Now, suppose that $\{x_j^{(i)}\}$ are $\sigma^2$ sub-Gaussian. Thus, $y^n = (y_1, \ldots, y_n)$ are $(\tau, \gamma)$-concentrated, where $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ and $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$ (e.g., see [13, Theorem 1.14]). We show that with probability $1-\beta$, we have $y_i \in [a,b]$ for all $i \in [n]$, where $[a,b] \leftarrow \mathsf{Range}_{\text{scalar}}\left(\mathcal{D}, \tau, \epsilon_0, \delta\right)$. Condition on the event that $y^n = (y_1, \ldots, y_n)$ are concentrated with radius $\tau$. Hence, there exists $y_0 \in [-B, B]$ such that $|y_i - y_0| \leq \tau$ for all $i \in [n]$. In Algorithm 4, we split the interval $[-B, B]$ into $T = \frac{B}{\tau}$ interval each with width $2\tau$, where $\mathcal{T}$ denotes the set of middle points of intervals. For each $i \in [n]$, let $\nu_i = \arg\min_{a \in \mathcal{T}} |y_i - a|$ be the closest bin in $\mathcal{T}$ to the exact value $y_i$. We define $f(a) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\left(\nu_i = a\right)$ as the fraction (frequency) of elements in $y^n$ that are close to the bin $a$ for each bin $a \in \mathcal{T}$. Observe that when $y^n$ are concentrated with radius $\tau$, we expect that $f(a) = 0$ for all $a \in \mathcal{T}$ except two adjacent bins.

Let $\mathbf{z}_i \leftarrow \mathsf{Range}^{\text{user}}_{\text{scalar}}$ of the $i$-th user. Thus, we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{z}_i\right] &= \frac{1}{d}\sum_{j=1}^k \mathbf{H}_k(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\left[\sqrt{k}\mathbf{m}(j)\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right] \\
&= \frac{1}{d}\sum_{j=1}^k \mathbf{H}_k(j)\sqrt{k}\mathbf{m}_i(j) \\
&\overset{\text{(a)}}{=} \frac{1}{d}\sum_{j=1}^k \mathbf{H}_k(j)\mathbf{H}_k^T(j)e_{\nu_i} \overset{\text{(b)}}{=} e_{\nu_i},
\end{aligned} \tag{13}$$

where step (a) follows from $\mathbf{m}_i = \mathbf{H}_k e_{\nu_i}$ and step (b) follows from $\sum_{j=1}^k \mathbf{H}_k(j)\mathbf{H}_k^T(j) = \mathbf{H}_k\mathbf{H}_k^T = k \times \mathbb{I}_k$. Thus, $\overline{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i$ is unbiased estimate of $\mathbf{f} = [f(a_1), \ldots, f(a_k)]$, i.e., $\mathbb{E}\left[\overline{\mathbf{z}}\right] = \mathbf{f}$.

Observe that $\overline{\mathbf{z}}(j)$ is a sum of i.i.d. Bernoulli random variables for $j \in [k]$. Thus, $\overline{\mathbf{z}}(j)$ is a sub-Gaussian with proxy $\frac{4\left(e^{\epsilon_0^2}+1\right)^2}{n\left(e^{\epsilon_0^2}-1\right)^2}$ and $\mathbb{E}[\overline{\mathbf{z}}(j)] = f(a_j)$. Hence, from [13, Theorem 1.14], we get that

$$\Pr[\max_{j \in [k]} |\overline{\mathbf{z}}(j) - f(a_j)| > t] \leq 2k\exp\left(-\frac{t^2 n\left(e^{\epsilon_0^2}-1\right)^2}{8\left(e^{\epsilon_0^2}+1\right)^2}\right) \tag{14}$$

By setting $t = \frac{1}{5}$, with probability at least $1 - 2k\exp\left(-\frac{n\left(e^{\epsilon_0^2}-1\right)^2}{200\left(e^{\epsilon_0^2}+1\right)^2}\right)$, we get

$$\max_{j \in [k]} |\overline{\mathbf{z}}(j) - f(a_j)| \leq \frac{1}{5}. \tag{15}$$

With probability $1-\gamma$, since there are only two adjacent bins of non-zero frequencies, one of them has a frequency $f(a) \geq \frac{1}{2}$. Let $a_{\max}$ be the bin that has the maximum estimated frequency. Conditioned on the event (15), the $a_{\max}$ will be equal one of these two non-zero bins that has non-zero frequencies. This can be seen as follows:

Let $j_1, j_2 \in [k]$ be such that $f(a_{j_1}), f(a_{j_2}) > 0$ and we know that one of them, say, $j_1$, has $f(a_{j_1}) \geq \frac{1}{2}$. Since $a_{\max} = \arg\max_{j\in[k]} \overline{z}(j)$, by (15), we have $\overline{z}(j_1), \overline{z}(j_2) \in [\frac{3}{10}, \frac{7}{10}]$ and $\overline{z}(j_l) < \frac{1}{5}, \forall l \in [k] \setminus \{j_1, j_2\}$. Hence, $a_{\max} \in \{j_1, j_2\}$.

This implies that each $y_i$ lies within $3\tau$ of $a_{\max}$. Thus, from union bound we conclude that $y_i \in [a_{\max} - 3\tau, a_{\max} + 3\tau]$ for all $i \in [n]$ with probability at least $1-\beta$. This completes the proof of Lemma 1. ■

## Appendix B
## Omitted Details from Section V: Remaining Details in the Proof of Theorem 2

**Variance of $z_i$:** Recall that $z_i$ is a r.v. that takes values in $\{dz_i(1)\cdot e_1, \ldots, dz_i(d)\cdot e_d\}$ uniformly at random, where $z_i(l) = \mathsf{Mean}^{\text{user}}_{\text{scalar}}(\overline{\mathcal{D}}_{i,j}, R(j), \epsilon_0/2, \delta)$ and $e_1, \ldots, e_d$ are the basis vectors in $\mathbb{R}^d$. Note that $dz_i(l)\cdot e_l$ is a length $d$ vector whose $l$'th component is equal to $dz_i(l)$ and all other components are equal to zero.

$$
\begin{aligned}
\mathbb{E}\left[\|z_i - \tilde{y}_i\|^2\right] &= \mathbb{E}\left[\|z_i\|^2\right] - \mathbb{E}\left[\|\tilde{y}_i\|^2\right] \\
&= \frac{1}{d}\sum_{l=1}^{d} d^2\mathbb{E}\left[z_i(l)^2\right] - \|\tilde{y}_i\|^2 \\
&= d\sum_{l=1}^{d} \mathbb{E}\left[z_i(l)^2\right] - \|\tilde{y}_i\|^2 \\
&\overset{(a)}{=} d\sum_{l=1}^{d} \left(\tilde{y}_i(l)^2 + \mathbb{E}[\nu^2]\right) - \|\tilde{y}_i\|^2 \\
&\overset{(b)}{=} d\|\tilde{y}_i\|^2 + d^2\mathbb{E}[\nu^2] - \|\tilde{y}_i\|^2 \\
&= d^2\frac{12(6\tau')^2\log(1.25/\delta)}{\epsilon_0^2} + (d-1)\|\tilde{y}_i\|^2 \\
&\overset{(c)}{\leq} \mathcal{O}\left(\frac{d^2\left(\tau'\right)^2\log(1/\delta)}{\epsilon_0^2}\right) + d(d-1)(\tau')^2 \\
&\overset{(d)}{=} \mathcal{O}\left(\frac{d^2\left(\tau'\right)^2\log(1/\delta)}{\epsilon_0^2}\right)
\end{aligned}
$$

In (a), we used $z_i(l) = \tilde{y}_i(l) + \nu$ for every $l \in [d]$, where $\nu \sim \mathcal{N}(0, \frac{12(b-a)^2\log(1.25/\delta)}{\epsilon_0^2})$, with $(b-a) = 6\tau'$ (see $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$) and also the fact that $\tilde{y}_i(l)$ and $\nu$ are independent. In (b) we used $\|\tilde{y}_i\|^2 = \sum_{l=1}^{d}\tilde{y}_i(l)^2$. In (c), we used that each coordinate of $\tilde{y}_i$ is bounded by $\tau'$. In (d) we assumed $\log(1/\delta) \geq \Omega(\epsilon_0^2)$.

## Appendix C
## Proof of Theorem 3

**Theorem** (Restating Theorem 3)**.** *Let the set $\mathcal{C}$ be convex with diameter $D$ and the function $f(\theta;.) : \mathcal{C} \to \mathbb{R}$ be convex and $L$-Lipschitz continuous with respect to the $\ell_2$-norm. Let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F(\theta)$ denote the minimizer of the problem (6). The Algorithm $\mathsf{A}_{uldp}$ is user-level $(\epsilon_0, \delta)$-LDP per user per iteration. If we run Algorithm $\mathsf{A}_{uldp}$ over $T$ iterations with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, then*

$$
\mathbb{E}\left[F\left(\theta_{T+1}\right)\right] - F\left(\theta^*\right) = \mathcal{O}\left(DG\frac{\log(T)}{\sqrt{T}}\right) \tag{16}
$$

*with probability at least $1-\beta_T$, where $G = L\sqrt{\left(1 + \frac{cd\log(dn/\gamma)\log(1/\delta)\log(2n/\gamma)}{qnm\epsilon_0^2}\right)}$, $q = \frac{k}{n}$, $\gamma = \frac{1}{n^{\log(n)+2}}$, $\beta_T = \min\{1, 2T\gamma + T\zeta\}$, $\epsilon_0' = \frac{\epsilon_0}{2d}$, and $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{qn(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.*

*Proof.* The privacy is directly obtained from the fact that at each iteration the private gradients are obtained from the Algorithm private mean estimation $\mathsf{Mean}_{\text{vector}}$ which is user-level $(\epsilon_0, \delta)$-LDP. Thus, the ULDP-SGD Algorithm is user-level $(\epsilon_0, \delta)$-LDP per iteration.

Observe that if the function $f(\theta, d)$ is $L$-Lipschitz continuous with respect to $\ell_2$-norm, then the gradient $\|\nabla_\theta f(\theta, d)\|_2 \leq L$ (see e.g., [25, Lemma 2.6]) . At any iteration $t \in [T]$, let $x_j^{(i)} = \nabla_{\theta_t} f(\theta_t, d_{ij})$, and $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ for $i \in \mathcal{U}_t$, $j \in [m]$. Thus, we get that $\{x_i^{(j)}\}$ are sub-Gaussian with proxy $L^2$. Thus, the samples $(y_i : i \in \mathcal{U}_t)$ are $(\tau, \gamma)$ concentrated, where $\tau = L\sqrt{\frac{\log(2n/\gamma)}{k}}$.

At iteration $t \in [T]$ of Algorithm 6, the server receives the average of $km$ private gradients $\{x_j^{(i)} : i \in \mathcal{U}_t, j \in [m]\}$ using the private mean estimation Algorithm $\mathsf{Mean}_{\text{vector}}$. Since Algorithm $\mathsf{Mean}_{\text{vector}}$ is an unbiased estimate of $\frac{1}{km}\sum_{i\in\mathcal{U}_t}\sum_{j=1}^m x_j^{(i)}$ with probability at least $1-\beta$ (see Theorem 2), we get that $\mathbb{E}\left[\mathsf{Mean}_{\text{vector}}(\tilde{D})\right] = \nabla F(\theta_t, \mathcal{D})$ where the expectation with respect to the randomness of user sampling and the randomness in the Algorithm $\mathsf{Mean}_{\text{vector}}$. Now we show that $\overline{\mathbf{g}}_t$ has a bounded second moment.

**Lemma 3.** *If the function $f(\theta; .) : \mathcal{C} \to \mathbb{R}$ is convex and $L$-Lipschitz continuous with respect to the $\ell_2$-norm, then we have*

$$\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 \leq L^2\left(1 + \frac{cd\log(dn/\gamma)\log(1/\delta)\log(2n/\gamma)}{qnm\epsilon_0^2}\right), \tag{17}$$

*with probability at least $1-\beta$, where $c = 28800$ is a global constant, $\beta = \min\left\{1, 2\gamma + \zeta\right\}$, $\epsilon_0' = \frac{\epsilon_0}{2d}$, and $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{qn(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.*

*Proof.* Under the conditions of the lemma, we have from [25, Lemma 2.6] that $\|\nabla_\theta f(\theta; d)\| \leq L$, which implies that $\|\nabla_\theta F(\theta)\| \leq L$. Thus, we have

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 &= \|\mathbb{E}[\overline{\mathbf{g}}_t]\|_2^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}[\overline{\mathbf{g}}_t]\|_2^2 \\
&\overset{(a)}{\leq} L^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}[\overline{\mathbf{g}}_t]\|_2^2 \\
&\overset{(b)}{\leq} L^2 + \frac{cL^2 d\log(nd/\gamma)\log(1/\delta)\log(2n/\gamma)}{km\epsilon_0^2}
\end{aligned}$$

with probability at least $1-\beta$, where $c = 28800$ is a global constant. Step $(a)$ follows from the fact that $\|\nabla_{\theta_t} F(\theta_t)\| \leq L$. Step $(b)$ follows from Theorem 2. ∎

Thus, from union bound and Lemma 3, with probability $1-\beta_T$, we have $\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 \leq G^2$ for all $t \in [T]$, where $\beta_T = T\beta$ and $G = L\sqrt{\left(1 + \frac{cd\log(dn/\gamma)\log(1/\delta)\log(2n/\gamma)}{qnm\epsilon_0^2}\right)}$ and $q = k/n$. Now, we can use standard SGD convergence results for convex functions. In particular, we use the following result from [26].

**Lemma 4** (SGD Convergence [26])**.** *Let $F(\theta)$ be a convex function, and the set $\mathcal{C}$ has diameter $D$. Consider a stochastic gradient descent algorithm $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}(\theta_t - \eta_t \mathbf{g}_t)$, where $\mathbf{g}_t$ satisfies $\mathbb{E}[\mathbf{g}_t] = \nabla_{\theta_t} F(\theta_t)$ and $\mathbb{E}\|\mathbf{g}_t\|_2^2 \leq G^2$. By setting $\eta_t = \frac{D}{G\sqrt{t}}$, we get*

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq 2DG\frac{2+\log(T)}{\sqrt{T}} = \mathcal{O}\left(DG\frac{\log(T)}{\sqrt{T}}\right). \tag{18}$$

As a result, Algorithm 6 satisfies the premise of Lemma 4. Now, using the bound on $G^2$ from Lemma 3, we have that the output $\theta_T$ of Algorithm 6 satisfies

$$\mathbb{E}[F(\theta_{T+1})] - F(\theta^*) = \mathcal{O}\left(DG\frac{\log(T)}{\sqrt{T}}\right) \tag{19}$$

This completes the proof of Theorem 3. ∎

## Appendix D
## User-Level LDP in The Shuffle Model

In this section, we extend our proposed algorithms for private mean estimation and empirical risk minimization to the shuffle model to provide a strong (central) user-level DP as well as user-level LDP. In the shuffle model, each user applies a user-level LDP mechanisms to preserve privacy of her own local dataset. We assume there exists a secure shuffler between the users and the central server. The shuffler receives the users' reports (the output of the user-level LDP mechanism) and randomly permutes them before passing them to the server.

Let $\mathsf{M}: \mathcal{X}^m \to \Theta$ be user-level LDP mechanism, $\mathcal{D}_i \in \mathcal{X}^m$ be the local dataset at the $i$-th client, and $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n)$. Let $\mathcal{H}_n : \Theta^n \to \Theta^n$ denote the shuffling operation that takes $n$ inputs and outputs their uniformly random permutation. We define the shuffling mechanism as follow:

$$\mathsf{M}_{\text{shuffle}}(\mathcal{D}) = \mathcal{H}_n\left(\mathsf{M}(\mathcal{D}_1), \ldots, \mathsf{M}(\mathcal{D}_n)\right). \tag{20}$$

In section D-A, we extend the private mean estimation results in the shuffle model and in Section D-B, we present the results of the ERM in the shuffle model.

### A. Private Mean Estimation

In order to use the results of privacy amplification by shuffling [12], [19]–[21], [27], it is required that the local mechanism $\mathsf{M}$ to be pure user-level $\epsilon_0$-LDP. Thus, we replace the Gaussian noise added in line 2 in Algorithm 2 with Laplace noise to give pure user-level LDP. In other words, we suppose in mechanism $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$, we sample $\nu \sim \mathsf{Lap}\left(2\frac{(b-a)}{\epsilon_0}\right)$. Thus, the mechanism $\mathsf{Mean}^{\text{user}}_{\text{scalar}}$ is pure user-level $\epsilon_0$-LDP.

**Theorem 4.** *In the shuffle model, the mechanism* $\mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)$ *is user-level* $(\epsilon_0, 0)$*-LDP. For* $\epsilon_0 \leq 1$*, the output of the shuffler is (central) user-level* $(\epsilon, \delta)$*-DP, where* $\delta \in (0,1)$ *and* $\epsilon$ *is given by:*

$$\epsilon = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{\log(1/\delta)}{n}}\right). \tag{21}$$

*Furthermore, if* $\{x_j^{(i)}\}$ *are sub-Gaussian random vectors with proxy* $\sigma^2$*, then* $y^n = (y_1, \ldots, y_n)$ *are* $(\tau, \gamma)$*-concentrated, where* $y_i = \frac{1}{m}\sum_{j=1}^m x_j^{(i)}$ *and* $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$*. With probability* $1-\beta$*, we have*

$$\mathcal{E}_2 := \mathbb{E}\left[\left\|\frac{1}{mn}\sum_{i=1}^n\sum_{j=1}^m x_j^{(i)} - \mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)\right\|^2\right] \leq \mathcal{O}\left(\frac{\tau^2 d\log(dn/\gamma)}{n\epsilon_0^2}\right), \tag{22}$$

*where* $\beta = \min\left\{1, 2\gamma + \zeta\right\}$*,* $\epsilon_0' = \frac{\epsilon_0}{2d}$*, and* $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{n(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.

*Proof.* It is clear that the mechanism $\mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)$ is user-level $(\epsilon_0, 0)$-LDP as we replaced the Gaussian noise with Laplace noise. since each user LDP mechanism is identical and are $\epsilon_0$-LDP, from privacy amplification by shuffling [27, Corollary 5.3.1], we get that the output of the shuffler is user-level $(\epsilon, \delta)$, where $\delta \in (0,1)$ and $\epsilon$ is given by:

$$\epsilon = \mathcal{O}\left(\min\{1, \epsilon_0\}e^{\epsilon_0}\sqrt{\frac{\log(1/\delta)}{n}}\right). \tag{23}$$

when $\epsilon_0 \leq 1$, we get that $\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{\log(1/\delta)}{n}}\right)$. The estimation error analysis is exactly the same as the Gaussian noise case given in Appendix B. This completes the proof of Theorem 4. ■

**Remark 6** (Achieving the mean estimation of the (central) user-level DP)**.** Observe that by substituting $\epsilon = \mathcal{O}\left(\min\{1, \epsilon_0\}e^{\epsilon_0}\sqrt{\frac{\log(1/\delta)}{n}}\right)$ from (23) into the mean estimation error in (22), we get that

$$\mathcal{E}_2 := \mathbb{E}\left[\left\|\frac{1}{mn}\sum_{i=1}^n\sum_{j=1}^m x_j^{(i)} - \mathsf{Mean}_{\text{vector}}(\mathcal{D}, \tau, \epsilon_0, \delta)\right\|^2\right] \leq \mathcal{O}\left(\frac{\tau^2 d\log(dn/\gamma)\log(1/\delta)}{n^2\epsilon^2}\right).$$

This matches the mean square error of the central user-level DP Algorithm proposed in [5]. However, our Algorithm additionally provides LDP guarantees for the users.

### B. Private ERM

In this section, we extend the empirical risk minimization problem presented in Section IV to the shuffle model. The main difference in Algorithm 6 is that each user follows the changes in the private mean estimation Algorithm proposed in Section D-A and we assume that there exists a trusted shuffler between the users and the server.

**Theorem 5.** *Let the set $\mathcal{C}$ be convex with diameter $D$ and the function $f(\theta;.) : \mathcal{C} \to \mathbb{R}$ be convex and L-Lipschitz continuous with respect to the $\ell_2$-norm. Let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F(\theta)$ denote the minimizer of the problem (6). The Algorithm $\mathsf{A}_{\text{ulpd}}$ is user-level $(\epsilon_0, 0)$-LDP per iteration. Furthermore, for $\epsilon - 0 \leq 1$, the Algorithm $\mathsf{A}_{\text{ulpd}}$ is (central) user-level $(\epsilon, \delta)$-DP, where $\delta \in (0, 1)$ and $\epsilon$ is given by*

$$\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{n}}\right) \tag{24}$$

*If we run Algorithm $\mathsf{A}_{\text{uldp}}$ over $T$ iterations with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, then*

$$\mathbb{E}\left[F(\theta_{T+1})\right] - F(\theta^*) = \mathcal{O}\left(DG\frac{\log(T)}{\sqrt{T}}\right) \tag{25}$$

*with probability at least $1 - \beta_T$, where $G = L\sqrt{\left(1 + \frac{cd\log(dn/\gamma)\log(2n/\gamma)}{qnm\epsilon_0^2}\right)}$, $c = 28800$, $q = \frac{k}{n}$, $\gamma = \frac{1}{n^{\log(n)+2}}$, $\beta_T = \min\{1, 2T\gamma + T\zeta\}$, $\epsilon_0' = \frac{\epsilon_0}{2d}$, and $\zeta = \frac{2d^2B\sqrt{\log(dn/\gamma)}}{\tau}\exp\left(-\frac{qn(e^{\epsilon_0'}-1)^2}{200(e^{\epsilon_0'}+1)^2}\right)$.*

*Proof.* It is straightforward that the Algorithm $\mathsf{A}_{\text{ulpd}}$ is user-level $(\epsilon_0, 0)$-LDP per iteration as we use the private mean estimation $\mathsf{Mean}_{\text{vector}}$ at each iteration. Furthermore, at each iteration, a set of $k$ users are choosen at random. Thus from Theorem 4, the output of the shuffler is (central) user-level $\left(\tilde{\epsilon}_t, \frac{\delta}{2qT}\right)$-DP, where $\tilde{\epsilon}_t = \mathcal{O}\left(\epsilon_0\sqrt{\frac{\log(2T/\delta)}{qn}}\right)$. Thus, from uniform sampling [28], we get that the Algorithm $\mathsf{A}_{\text{ulpd}}$ is (central) user-level $\left(\epsilon_t, \frac{\delta}{2T}\right)$-DP per iteration, where $\epsilon_t = q\tilde{\epsilon}_t$. From composition theorem [7, Corollary 3.21], we get that the Algorithm $\mathsf{A}_{\text{ulpd}}$ is (central) user-level $(\epsilon, \delta)$-DP, where $\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{n}}\right)$.

The convergence results follows exactly from our proofs in Appendix C. This completes the proof of Theorem 5. ∎

**Remark 7.** Observe that substituting $\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{n}}\right)$, $T = \frac{n}{q}$, and $q = \frac{k}{n}$, we get that

$$\mathbb{E}\left[F(\theta_{T+1})\right] - F(\theta^*) = \mathcal{O}\left(DL\frac{\log(T/\delta)^3\log(nd/\gamma)\sqrt{d\log(1/\delta)}}{n\sqrt{m}\epsilon}\right) \tag{26}$$

with probability at least $1 - \beta_T$. Furthermore, when $\gamma = \frac{1}{n^{\log(n)+2}}$ and the number of users $n$ is arbitrary large, we get that $\beta_T \leq \frac{1}{n^{\log(n)}} + n^2e^{-n\epsilon_0^2/400} \to 0$ as $\epsilon_0 \leq 1$ and $n \to \infty$.