Shuffled Model of Differential Privacy in Federated Learning

Antonious M. Girgis
UCLA

Deepesh Data
UCLA

Suhas Diggavi
UCLA

Peter Kairouz
Google

Ananda Theertha Suresh
Google

Abstract

We consider a distributed empirical risk minimization (ERM) optimization problem with communication efficiency and privacy requirements, motivated by the federated learning (FL) framework. We propose a distributed communication-efficient and local differentially private stochastic gradient descent (CLDP-SGD) algorithm and analyze its communication, privacy, and convergence trade-offs. Since each iteration of the CLDP-SGD aggregates the client-side local gradients, we develop (optimal) communication-efficient schemes for mean estimation for several ℓ_p spaces under local differential privacy (LDP). To overcome performance limitation of LDP, CLDP-SGD takes advantage of the inherent privacy amplification provided by client subsampling and data subsampling at each selected client (through SGD) as well as the recently developed shuffled model of privacy. For convex loss functions, we prove that the proposed CLDP-SGD algorithm matches the known lower bounds on the *centralized* private ERM while using a finite number of bits per iteration for each client, *i.e.,* effectively getting communication efficiency for "free". We also provide preliminary experimental results supporting the theory.

1 Introduction

We consider a federated learning (FL) framework (*e.g.,* Kairouz et al. [2019]), where data is generated across m *distributed* clients, and the server builds a machine learning model by solving the empirical risk minimization (ERM) problem:

$$\arg\min_{\theta\in\mathcal{C}} \left(F(\theta) := \frac{1}{m}\sum_{i=1}^{m} F_i(\theta) \right), \qquad (1)$$

where $F_i(\theta)$ is a local loss function dependent on the local dataset $\mathcal{D}_i$ at client i, comprising r data points, and $\mathcal{C} \subset \mathbb{R}^d$ is a closed convex set, where d denotes the model dimension; see Section 3 for details on the setup. The goal is to solve this problem while satisfying the FL requirements: (i) give privacy guarantees on the data $\mathcal{D}_i$ at client i, (ii) compress (as efficiently as possible) the communication between clients and the server, and (iii) work with a dynamic client population in each round of communication between the server and the clients – in FL only a small fraction of clients are sampled at each communication round; see Figure 1.

A challenge is that local differential privacy (LDP),*e.g.,* [Beimel et al., 2008, Warner, 1965], is known to give poor learning performance [Duchi et al., 2013, Kairouz et al., 2016, Kasiviswanathan et al., 2011]. Recently, a new privacy framework, the *shuffled model* [Balle et al., 2019a,b,c, 2020a, Cheu et al., 2019, Erlingsson et al., 2019, Ghazi et al., 2019a,b, 2020], enables significantly better privacy-utility performance by amplifying privacy (scaling with the number of clients as $\frac{1}{\sqrt{m}}$ with respect to LDP) through anonymization. Another technique to amplify privacy is through randomized subsampling [Beimel et al., 2010]. This naturally arises in a stochastic gradient descent (SGD) framework for optimizing (1), since clients do mini-batch sampling of local data; moreover clients themselves are sampled in each iteration motivated by the FL setup.

In this paper, we analyse privacy amplification for the FL problem using both forms of amplification: shuffling and subsampling (data and clients). Note that privacy amplification by subsampling (both data and clients)

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

happens automatically,[1] while the secure shuffling is performed explicitly adding an additional layer of privacy transferring local privacy guarantees to central privacy guarantees.

Another important aspect is communication efficiency instantiated through compression of the gradients computed by each active client. There has been a significant recent progress on this topic (see [Alistarh et al., 2017, 2018, Basu et al., 2019, Karimireddy et al., 2019, Singh et al., 2019, 2020, Stich et al., 2018] and references therein). However, there has been less work in combining privacy and compression in the optimization/learning framework, with the notable exception of [Agarwal et al., 2018], which we will elaborate on shortly. One question that we address is whether one pays a price to do compression in terms of the privacy-performance trade-off.

In this paper, we solve the main problem of learning a model with communication constraints, with reasonable learning performance while giving strong privacy guarantees. We believe that this is the first result that analyses the optimization performance with schemes devised using compressed gradient exchange, mini-batch SGD while giving privacy guarantees for clients using a shuffled framework. Here are our main contributions.

• We prove that one can get communication efficiency "for free" by demonstrating schemes that use $O(\log d)$ bits per gradient to obtain the same privacy-utility operating point as full precision gradient exchange.[2]

• One ingredient of our main result is showing that we can compose amplification by sampling (client data through mini-batch SGD and clients themselves in federated sampling) along with amplification by shuffling. Note that sampling of clients and data points together give overall non-uniform sampling of data points, so we cannot use the existing results on privacy amplification by subsampling, necessitating a privacy proof that composes sampling and shuffling techniques.

• At each round of the iterative optimization, one needs to privately aggregate the gradients in a communication efficient manner. For this, we develop new private and compressed vector mean estimation techniques in a minimax estimation framework, that are (order optimal) under several ℓ_p geometries. We develop both lower bounds and matching schemes for this problem. These results may also be of independent interest.

Related work: There has been a lot of work on privacy in the context of FL (see [Kairouz et al., 2019] and references therein) and also on compression for private mean estimation (see [Acharya and Sun, 2019, Balle et al., 2019c, 2020a, Cheu et al., 2019] and references therein). Our focus in this paper is on *distributed* learning with local differential privacy guarantees, which has fewer results, especially in the shuffled privacy framework. We give an extensive account of the related work in Appendix A of the supplementary material, and due to space constraints we will focus on the two most related papers to our work [Agarwal et al., 2018, Erlingsson et al., 2020], which we describe below.

Erlingsson et al. [2020] proposed a distributed local differentially private gradient descent algorithm, where all clients participate in each iteration. They use LDP on gradients as well as the shuffled framework [Balle et al., 2019c]. However, their proposed algorithm sends the full-precision gradient without compression. Our work is different from [Erlingsson et al., 2020] in multiple ways: (i) we propose a communication efficient mechanism for each client that requires $O(\log d)$ bits per client, which can be significant for large d; (ii) our algorithm performs data sampling (using SGD at each client) and client sampling *i.e.*, not all clients are selected at each iteration, as motivated by the FL setup. This requires a careful combination of compression and privacy analysis; see Remark 2, where we recover the convergence result of [Erlingsson et al., 2020] as a special case of our general results.

Agarwal et al. [2018] proposed a communication-efficient algorithm for learning models with local differential privacy. They proposed cp-SGD, a communication efficient algorithm, where clients need to send $O(\log(1+\frac{d}{n}\epsilon_0^2)+\log\log\log\frac{nd}{\epsilon_0\delta})$ bits of communication *per coordinate*, *i.e.*, $O\left(d\left\{\log(1+\frac{d}{n}\epsilon_0^2)+\log\log\log\frac{nd}{\epsilon_0\delta}\right\}\right)$ bits per gradient to achieve the same local differential privacy guarantees of the Gaussian mechanism. In contrast, we achieve better compression in terms of number of bits per gradient, and our framework converts the LDP algorithm to central differential privacy guarantees.

Paper organization. In Section 2, we establish some background results. In Section 3, we set up the problem including the formulation for private mean-estimation and describe our algorithm. We state our results in Section 4 and also give some interpretations. In Section 5, we provide brief proof outlines for some of the results. Section 6 provides preliminary evaluation of the algorithm in terms of communication-privacy-performance operating points on the MNIST dataset. Many of the proof details as well as some additional results are provided in the supplementary material.

[1] In this paper, we use an abstraction for the federated learning model, where clients are sampled randomly. In practice, there are many more complicated considerations for sampling, including availability, energy usage, time-of-day, etc., which we do not model.

[2] Our work focuses on symmetric, private-randomness mechanisms. We do not assume the existence of public randomness in this work as we use the shuffled model.

2 Preliminaries

In this section, we state some preliminary definitions that we use throughout the paper; we give a more detailed exposition of the background in Appendix B of the supplementary material.

Since we are interested in communication constrained privacy of the client, we define a two parameter LDP with privacy and communication budget, generalizing the standard LDP privacy definition (see Definition 3 in Appendix B.1 of supplementary material).

Definition 1 (Local Differential Privacy with Communication Budget - CLDP)**.** For $\epsilon_0 \geq 0$ and $b \in \mathbb{N}^+$, a randomized mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is said to be (ϵ_0, b)-communication-limited-local differentially private (in short, (ϵ_0, b)-CLDP), if $\mathcal{R}(\boldsymbol{x})$ can be represented using b bits and for every pair $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, we have

$$\Pr[\mathcal{R}(\boldsymbol{x}) = \boldsymbol{y}] \leq \exp(\epsilon_0) \Pr[\mathcal{R}(\boldsymbol{x}') = \boldsymbol{y}], \ \forall \boldsymbol{y} \in \mathcal{Y}. \quad (2)$$

Here, ϵ_0 captures the privacy level, lower the ϵ_0, higher the privacy. When we are not concerned about the communication budget, we succinctly denote the corresponding (ϵ_0, ∞)-CLDP, by its correspondence to the classical LDP as ϵ_0-LDP [Kasiviswanathan et al., 2011].

We define $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and $\mathcal{D}' = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n\}$ as neighboring datasets if they differ in one data point.

Definition 2 (Central Differential Privacy - DP [Dwork and Roth, 2014])**.** For $\epsilon, \delta \geq 0$, a randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to be (ϵ, δ)-differentially private (in short, (ϵ, δ)-DP), if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ and every subset $\mathcal{E} \subseteq \mathcal{Y}$, we have

$$\Pr\left[\mathcal{M}\left(\mathcal{D}\right) \in \mathcal{E}\right] \leq \exp(\epsilon) \Pr\left[\mathcal{M}\left(\mathcal{D}'\right) \in \mathcal{E}\right] + \delta. \quad (3)$$

We will propose an iterative algorithm to solve the optimization problem (1) under privacy and communication constraints. Hence, we need the strong composition theorem [Dwork et al., 2010] (we describe it in detail in Appendix B.2 for completeness) to compute the final privacy guarantees of the proposed algorithm. Furthermore, in order to overcome the poor performance of LDP, we need to use privacy amplification provided by subsampling (data and clients) as well as through the shuffled model; both of these are described in detail in Appendix B.3.

3 Problem Formulation and Solution Overview

In this section, first we present the problem formulation and describe our algorithm for solving the empirical risk minimization (ERM) problem under the constraints of privacy, communication, and dynamic client population. Then we give an overview of our approach to analyze this algorithm and briefly describe the challenges faced. In the end, we describe one of the main ingredients in our algorithm, which is a method of private mean estimation using compressed updates.

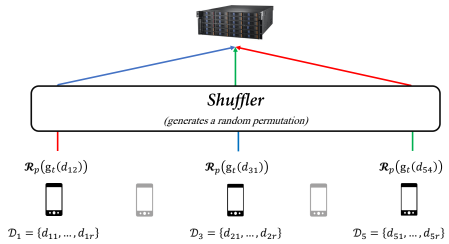
Shuffler
(generates a random permutation)
$\mathcal{R}_p(g_t(d_{12}))$
$\mathcal{R}_p(g_t(d_{31}))$
$\mathcal{R}_p(g_t(d_{54}))$
$\mathcal{D}_1 = \{d_{11}, \ldots, d_{1r}\}$
$\mathcal{D}_3 = \{d_{21}, \ldots, d_{2r}\}$
$\mathcal{D}_5 = \{d_{51}, \ldots, d_{5r}\}$

Figure 1: An example of 5 clients, where each client has r data points. At the current iteration, 3 clients are chosen at random. Each client chooses one data point at random to send the compressed and private gradient $\{\mathcal{R}_p\left(g_t\left(d_{ij}\right)\right)\}$ to the secure shuffler that permutes the private gradients before sending them to the server.

Problem formulation: We have a set of m clients, where each client has a local dataset $\mathcal{D}_i = \{d_{i1}, \ldots, d_{ir}\}$ comprising r data points drawn from a universe $\mathfrak{S}$. Let $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i$ denote the entire dataset and $n = mr$ denote the total number of data points in the system. The clients are connected to an untrusted server in order to solve the ERM problem described in (1). In (1), $F_i(\theta, \mathcal{D}_i) = \frac{1}{r}\sum_{j=1}^{r} f\left(\theta, d_{ij}\right)$ is a local loss function dependent on the local dataset $\mathcal{D}_i$ at client i evaluated at the model parameters $\theta \in \mathcal{C}$.

As described in Section 1, solving the ERM problem (1) in the FL framework introduces several unique challenges, such as the locally residing data $\{\mathcal{D}_i\}$ at all clients need to kept private, the low-bandwidth links between clients and the server necessitates compressed communication exchange between them, and only a small fraction of clients are sampled in each round of communication. Our goal is to solve (1) while preserving privacy on the training dataset $\mathcal{D}$ and minimizing the total number of bits for communication between clients and the server, while dealing with a dynamic client population in each iteration.

Our algorithm CLDP-SGD: In order to solve (1) in the presence of the above challenges in the FL setting, we propose CLDP-SGD, a differentially-private SGD algorithm that works with compressed updates and dynamic client population. The procedure is described in Algorithm 1; also see Figure 1 for a pictorial description of our algorithm. In each step of CLDP-

Algorithm 1 $\mathcal{A}_{\text{cldp}}$: CLDP-SGD

```
1: Inputs: Datasets $\mathcal{D} = \bigcup_{i\in[m]} \mathcal{D}_i$, $\mathcal{D}_i = \{d_{i1}, \ldots, d_{ir}\}$, loss function $F(\theta) = \frac{1}{mr}\sum_{i=1}^{m}\sum_{j=1}^{r} f(\theta; d_{ij})$, LDP privacy parameter $\epsilon_0$, gradient norm bound $C$, and learning rate $\eta_t$.
2: Initialize: $\theta_0 \in \mathcal{C}$
3: for $t \in [T]$ do
4:     Sampling of clients: The secure shuffler chooeses a random set $\mathcal{U}_t$ of $k$ clients.
5:     for clients $i \in \mathcal{U}_t$ do
6:         Sampling of data: Client $i$ chooses uniformly at random a set $\mathcal{S}_{it}$ of $s$ samples.
7:         for Samples $j \in \mathcal{S}_{it}$ do
8:             $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$
9:             $\tilde{\mathbf{g}}_t(d_{ij}) \leftarrow \mathbf{g}_t(d_{ij}) / \max\left\{1, \frac{\|\mathbf{g}_t(d_{ij})\|_p}{C}\right\}$[3]
10:            $\mathbf{q}_t(d_{ij}) \leftarrow \mathcal{R}_p(\tilde{\mathbf{g}}_t(d_{ij}))$
11:        Client $i$ sends $\{\mathbf{q}_t(d_{ij})\}_{j\in\mathcal{S}_{it}}$ to the shuffler.
12:    Shuffling: The shuffler randomly shuffles the elements in $\{\boldsymbol{q}_t(d_{ij}) : i \in \mathcal{U}_t,\ j \in \mathcal{S}_{it}\}$ and sends them to the server.
13:    Aggregate: $\overline{\mathbf{g}}_t \leftarrow \frac{1}{ks}\sum_{i\in\mathcal{U}_t,\ j\in\mathcal{S}_{it}} \boldsymbol{q}_t(d_{ij})$.
14:    Gradient Descent $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}(\theta_t - \eta_t\overline{\mathbf{g}}_t)$
15: Output: The final model parameters $\theta_T$.
```

SGD, the secure shuffler chooses uniformly at random a set $\mathcal{U}_t$ of $k \leq m$ clients out of m clients. Each client $i \in \mathcal{U}_t$ computes the gradient $\nabla_{\theta_t} f(\theta_t; d_{ij})$ for a random subset $\mathcal{S}_{it}$ of $s \leq r$ samples. The i'th client clips the ℓ_p-norm of the gradient $\nabla_{\theta_t} f(\theta_t; d_{ij})$ for each $j \in \mathcal{S}_{it}$ and applies the LDP-compression mechanism $\mathcal{R}_p$, where $\mathcal{R}_p : \mathcal{B}_p^d \to \{0,1\}^b$ is an (ϵ_0, b)-CLDP mechanism when inputs come from an ℓ_p-norm ball. In this paper, we describe (ϵ_0, b)-CLDP mechanisms $\mathcal{R}_p$ for several values of $p \in [1, \infty]$; see Section 5. After that, each client i sends the set of s LDP-compressed gradients $\{\mathcal{R}_p(\mathbf{g}_t(d_{ij}))\}_{j\in\mathcal{S}_{it}}$ in a communication-efficient manner to the secure shuffler. The shuffler randomly shuffles (i.e., outputs a random permutation of) the received ks gradients and sends them to the server. Finally, the server takes the average of the received gradients and updates the parameter vector.

Observe that our CLDP-SGD algorithm provides privacy guarantees against any adversary that can observe the output of the secure shuffler including the untrusted server. Furthermore, we assume that the trusted shuffler samples the clients, and this sampled set is unknown to the server; see line 4 in Algorithm 1. For future work, one could enable client self-sampling and self-anonymization. For example, the authors in Balle et al. [2020b] proposed a new sampling scheme called random check-in, in which each client independently chooses which time slot to participate in the training process.

Overview of our approach for analyzing CLDP-SGD: CLDP-SGD has the following components, which need to be analyzed together: (i) sampling of clients, necessitated by FL; (ii) sampling of data at each client for mini-batch SGD; (iii) compressing the gradients at each client for communication efficiency; (iv) privatizing the gradients at each client to prevent information leakage – the (compressed) gradients received by the server may leak information about the datasets; and (v) shuffling. The two main technical ingredients needed for the analysis are (a) Privacy analysis of coupled sampling and shuffling (b) Commununication efficient private mean estimatioon.

Privacy of coupled sampling and shuffling: As explained in Section 1, client and data sampling as well as shuffling contribute to privacy amplification. However, there are several challenges in analyzing the overall privacy amplification: Firstly, both types of sampling together induce non-uniform sampling of data, so we cannot use the existing privacy amplification from subsampling results (see Section B.3.1) directly to analyze the privacy gain in CLDP-SGD just by subsampling; and secondly, the privacy amplification by shuffling has not been analyzed together with that by subsampling. In this paper, we give one unifying proof that analyzes the privacy amplification by both types of subsampling (that induces non-uniform sampling of data points) as well as shuffling; see Section F.1 for more details.

Communication-efficient private mean estimation: For compressing and privatizing the gradients, we design communication-efficient local differentially private mechanisms $\mathcal{R}_p$ for $p \in [0, \infty]$ to estimate the mean of a set of bounded ℓ_p-norm gradients. These mechanisms $\mathcal{R}_p$'s are in fact more generally applicable for private mean estimation of a set of vectors, each having a bounded ℓ_p-norm and coming from a different client in a communication efficient manner. We study the mean estimation problem in the minimax framework and derive matching lower and upper bounds on the minimax risk for several ℓ_p geometries. This privacy mechanism is composed with the sampling and shuffling to provide the overall privacy analysis. Next, we formulate the compressed and private mean estimation problem as of independent interest.

Compressed and private mean estimation via minimax risk: Now we formulate the generic minimax estimation framework for mean estimation of a given set of n vectors that preserves privacy and is also

[3] Let ℓ_g denote the dual norm of ℓ_p norm, where $\frac{1}{p} + \frac{1}{g} = 1$ and $p, g \geq 1$. Thus, when the loss function $f(\theta, d_{ij})$ is convex and L-Lipschitz continuous with respect to ℓ_g-norm, then the gradient $\nabla_\theta f(\theta; .)$ has a bounded ℓ_p norm [Shalev-Shwartz et al., 2012, Lemma 2.6]. In this case, we do not need the clipping step.

communication-efficient. We then apply that method at the server in each SGD iteration for aggregating the gradients. We derive upper and lower bounds for various ℓ_p geometries for $p \geq 1$ including the ℓ_∞-norm.

The setup is as follows. For any $p \geq 1$ and $d \in \mathbb{N}$, let $\mathcal{B}_p^d(a) = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \leq a\}$ denote the p-norm ball with radius a centered at the origin in $\mathbb{R}^d$, where $\|\boldsymbol{x}\|_p = \left(\sum_{j=1}^d |\boldsymbol{x}_j|^p\right)^{1/p}$. Each client $i \in [n]$ has an input vector $\boldsymbol{x}_i \in \mathcal{B}_p^d(a)$ and the server wants to estimate the mean $\overline{\boldsymbol{x}} := \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$. We have two constraints: (i) each client has a communication budget of b bits to transmit the information about its input vector to the server, and (ii) each client wants to keep its input vector private from the server. Our objective is to design private-quantization mechanisms $\mathcal{R}_i : \mathcal{B}_p^d(a) \to \{0,1\}^b$ for all $i \in [n]$ and also a (stochastic) decoding function $\widehat{\boldsymbol{x}} : \left(\{0,1\}^b\right)^n \to \mathcal{B}_p^d$ that minimizes the worst-case expected error $\sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d} \mathbb{E}\|\overline{\boldsymbol{x}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\|^2$ and characterize the following.

$$r^{p,d}_{\epsilon_0,b,n}(a) = \inf_{\{\mathcal{R}_i \in \mathcal{Q}_{(\epsilon_0,b)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d(a)} \mathbb{E}\left\|\overline{\boldsymbol{x}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2, \tag{4}$$

where $\mathcal{Q}_{(\epsilon_0,b)}$ is the set of all (ϵ_0, b)-CLDP mechanisms, and the expectation is taken over the randomness of $\{\mathcal{R}_i : i \in [n]\}$ and the estimator $\widehat{\boldsymbol{x}}$. Note that in this setup we do not consider any probabilistic assumptions on the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. We also provide additional results when the inputs are sampled from a distribution in Appendix B.4 in the supplementary material.

4 Main Results

In this section, we first state our results on convergence, privacy, and communication bits of the proposed CLDP-SGD algorithm. We also discuss their implications. Then, we present the results on compressed and private mean estimation in Section 4.2.

4.1 Optimization

In the next theorem, we state the privacy guarantees, the communication cost per client, and the privacy-convergence trade-offs for the CLDP-SGD Algorithm. Let $n = mr$ denote the total number of data points in the dataset $\mathcal{D}$. Observe that the probability that an arbitrary data point $d_{ij} \in \mathcal{D}$ is chosen at time $t \in [T]$ is given by $q = \frac{ks}{mr}$.

Theorem 1. *Let the set $\mathcal{C}$ be convex with diameter D,[4] and the function $f(\theta; .) : \mathcal{C} \to \mathbb{R}$ be convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of the ℓ_p-norm.[5] For $s = 1$ and $q = \frac{k}{mr}$, if we run Algorithm $\mathcal{A}_{cldp}$, then we have*

1. ***Privacy:*** *For $\epsilon_0 = \mathcal{O}(1)$, $\mathcal{A}_{cldp}$ is (ϵ, δ)-DP, where $\delta > 0$ is arbitrary, and*

$$\epsilon = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{qT \log(2qT/\delta) \log(2/\delta)}{n}}\right). \tag{5}$$

2. ***Communication:*** *Our algorithm $\mathcal{A}_{cldp}$ requires $\frac{k}{m} \times b$ bits of communication in expectation[6] per client per iteration, where expectation is taken with respect to the sampling of clients. Here, $b = \log(d) + 1$ if $p \in \{1, \infty\}$ and $b = d(\log(e) + 1)$ otherwise.*

3. ***Convergence:*** *If we run $\mathcal{A}_{cldp}$ with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = L^2 \max\{d^{1-\frac{2}{p}}, 1\}\left(1 + \frac{cd}{qn}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\right)$, then*

$$\mathbb{E}\left[F(\theta_T)\right] - F(\theta^*) \leq$$
$$\mathcal{O}\left(\frac{LD\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}}, 1\}}{\sqrt{T}}\sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right). \tag{6}$$

where $c = 4$ if $p \in \{1, \infty\}$ and $c = 14$ otherwise.

We prove Theorem 1 in Section 5.3. Note that the privacy bound (49) holds when $\epsilon_0 = \mathcal{O}(1)$; the general result for $\epsilon_0 = \mathcal{O}\left(\log\left(\frac{qn}{\log(T/\delta)}\right)\right)$ is presented in Section 5.3.1.

Remark 1. Using a slightly different sampling procedure, the result in Theorem 1 holds for arbitrary s. We can achieve the same central privacy bound ϵ as stated in Theorem 1 with $q = \frac{ks}{mr}$ (instead of $q = \frac{k}{mr}$) using the following sampling: all clients send s compressed and private gradients corresponding to a uniformly random subset of s data points from their dataset; shuffler selects a uniformly random subset of ks gradients from them and then sends the shuffled output to the server. We analyze the privacy guarantees of our algorithm with this new sampling procedure in Appendix F.4. Note that, in this new sampling procedure, each data point has a probability $q = \frac{ks}{mr}$ of being picked, and we pick $\frac{ks}{m}$ data points (in expectation) from each clients. Note that even for this sampling (which does not yield uniform sampling of ks points from mr points), the privacy amplification of this sampling mechanism does not directly follow from existing results. We provide a

[4] Diameter of a bounded set $\mathcal{C} \subseteq \mathbb{R}^d$ is defined as $\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{x} - \boldsymbol{y}\|$.

[5] For any data point $d \in \mathfrak{S}$, the function $f : \mathcal{C} \to \mathbb{R}$ is L-Lipschitz continuous w.r.t. ℓ_g-norm if for every $\theta_1, \theta_2 \in \mathcal{C}$, we have $|f(\theta_1; d) - f(\theta_2; d)| \leq L\|\theta_1 - \theta_2\|_g$.

[6] A client communicates in an iteration only when that client is selected (sampled) in that iteration.

proof of this in the supplementary material, along with a discussion on other sampling procedures.

Remark 2 (Recovering the Result [Erlingsson et al., 2020, ESA])**.** In Erlingsson et al. [2020], each client has only one data point and all clients participate in each iteration, and gradients have bounded ℓ_2-norm. If we put $p=2$, $T=n/\log^2(n)$, and $q=1$ in (6) and $q=1$ in (49), we recover the convergence and the privacy bound in [Erlingsson et al., 2020, Theorem VI.1].

We want to emphasize that the above privacy-accuracy trade-off in Erlingsson et al. [2020] is achieved by full-precision gradient exchange, whereas, we can achieve the same trade-off with compressed gradients. Moreover, our results are in more general setting, where clients' local datasets have multiple data-points (no bound on that) and our privacy amplification is *effectively* due to two types of sampling, one of data, and the other of clients.

Remark 3 (Optimality of CLDP-SGD for ℓ_2-norm case)**.** Suppose $\epsilon=\mathcal{O}(1)$. Substituting $\epsilon_0 = \epsilon\sqrt{\frac{n}{qT\log(2qT/\delta)\log(2/\delta)}}$, $T=n/q$, and $p=2$ in (6), gives the *optimal* excess risk of central differential privacy, as shown in Bassily et al. [2014]. Note that the results in Bassily et al. [2014] are for centralized SGD with full precision gradients, whereas, our results are for federated learning (which is a distributed setup) with compressed gradient exchange.

4.2 Compressed & Private Mean Estimation

In this subsection, we state our lower and upper bounds on the minimax risk $r^{p,d}_{\epsilon_0,b,n}(a)$ for all $p\in[1,\infty]$. For the lower bounds, we state our results when there is no communication constraints, and for clarity, we denote the corresponding minimax risk by $r^{p,d}_{\epsilon_0,\infty,n}(a)$. Furthermore, we prove that any symmetric private mechanism requires at least $b\geq\log(d)$ bits of communication.

Theorem 2. *For any $d,n\geq 1$, $a,\epsilon_0>0$, and $p\in[1,\infty]$, the minimax risk in* (4) *satisfies*

$$r^{p,d}_{\epsilon_0,\infty,n}(a) \geq \begin{cases}\Omega\left(a^2\min\left\{1,\frac{d}{n\epsilon_0^2}\right\}\right) & \text{if } 1\leq p\leq 2,\\ \Omega\left(a^2 d^{1-\frac{2}{p}}\min\left\{1,\frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right) & \text{if } p\geq 2.\end{cases}$$

Theorem 3. *For any private-randomness, symmetric mechanism $\mathcal{R}$ with communication budget $b<\log(d)$ bits per client, and any decoding function $g:\{0,1\}^b\to\mathbb{R}^d$, when $\widehat{\boldsymbol{x}}=\frac{1}{n}\sum_{i=1}^n g(\mathcal{R}(\boldsymbol{x}_i))$, we have*[7]

$$r^{p,d}_{\epsilon,b,n}(a) > a^2\max\left\{1,d^{1-\frac{2}{p}}\right\}. \tag{7}$$

Theorem 3 shows that it is required at least $\log(d)$ bits per client to design a non-trivial private mechanism $\mathcal{R}$. Though our lower bound results are for arbitrary estimators $\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)$, we can show that the optimal estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)$ is a *deterministic* function of $\boldsymbol{y}^n$; see Lemma 15 in Appendix E.

Theorem 4. *For any $d,n\geq 1$, $a,\epsilon_0>0$, we have*

$$\ell_1: r^{1,d}_{\epsilon_0,b,n}(a) \leq \frac{a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2, \textit{for } b=\log(d)+1,$$

$$\ell_2: r^{2,d}_{\epsilon_0,b,n}(a) \leq \frac{6a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2, \textit{for } b=d\log(e)+1,$$

$$\ell_\infty: r^{\infty,d}_{\epsilon_0,b,n}(a) \leq \frac{a^2 d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2, \textit{for } b=\log(d)+1.$$

Note that when $\epsilon_0=\mathcal{O}(1)$, then the upper and lower bounds on minimax risks match for all $p\in[1,\infty]$. We can give general achievability results for any ℓ_p-norm ball $\mathcal{B}^d_p(a)$ for any $p\in[1,\infty)$. For this, we use standard inequalities between different norms, and probabilistically use the mechanisms for ℓ_1-norm or ℓ_2-norm with expanded radius of the corresponding ball. The main results for this are stated in Appendix D.1.

All the above results are for $r^{p,d}_{\epsilon_0,b,n}(a)$, which is defined for worst cast inputs. Essentially the same results hold when the inputs are sampled from a distribution, and we provide those results in Appendix D.1.

5 Proofs

In this section, first we prove the compressed and private mean estimation results and then prove Theorem 1. Due to the lack of space, we only prove mean estimation results for ℓ_∞-norm case and provide the proofs for other norms in Appendix D.

5.1 Proof of Theorem 2: For ℓ_∞-Norm

The lower bound result presented in this section in fact hold for any ℓ_p-norm for $p\in[2,\infty]$. The main idea of the lower bound is to transform the problem to the private mean estimation when the inputs are sampled from Bernoulli distributions. Let $\mathcal{P}^{\text{Bern}}_{p,d}$ denote the set of Bernoulli distributions on $\{0,{}^1\!/\!_{d^{1/p}}\}^d$, i.e., any element of $\mathcal{P}^{\text{Bern}}_{p,d}$ is a product of d independent Bernoulli distributions, one for each coordinate. For any $\boldsymbol{q}\in\mathcal{P}^{\text{Bern}}_{p,d}$, let $\boldsymbol{\mu}_{\boldsymbol{q}}$ denote the mean of $\boldsymbol{q}$.

Lemma 1. *For any $p\in[2,\infty]$, we have*

$$\inf_{\{\mathcal{M}_i\}\in\mathcal{Q}_{(\epsilon_0,\infty)}}\inf_{\widehat{\boldsymbol{x}}}\sup_{\boldsymbol{q}\in\mathcal{P}^{Bern}_{p,d}}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2 \geq \Omega\left(d^{1-\frac{2}{p}}\min\left\{1,\frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right). \tag{8}$$

[7] Note that Theorem 3 works only when the estimator $\widehat{\boldsymbol{x}}$ applies the decoding function g on individual responses and then takes the average. We leave its extension for arbitrary decoders as a future work.

The proof is straightforward adaptation of the proof of [Duchi and Rogers, 2019, Corollary 3] to our setting; see Appendix 5.1 for more details.

Let $\mathcal{P}_p^d$ denote the set of all distributions on the ℓ_p-norm ball, implying that $\mathcal{P}_{p,d}^{\text{Bern}} \subset \mathcal{P}_p^d$. This together with (8), implies that for every set of private mechanisms $\{\mathcal{M}_i\} \in \mathcal{Q}_{(\epsilon_0,\infty)}$ and estimator $\widehat{\boldsymbol{x}}$, we have

$$\sup_{\boldsymbol{q}\in\mathcal{P}_p^d} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \sup_{\boldsymbol{q}\in\mathcal{P}_{p,d}^{\text{Bern}}} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2$$
$$\geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right), \tag{9}$$

We can now obtain a lower bound on $r_{\epsilon_0,\infty,n}^{p,d}$ by transforming the worst-case lower bound to the average case lower bound as follows. Fix arbitrary private mechanisms $\{\mathcal{M}_1,\ldots,\mathcal{M}_n\}$ and an estimator $\widehat{\boldsymbol{x}}$. It follows from (9) that there exists a distribution $\boldsymbol{q} \in \mathcal{P}_p^d$, such that if we sample $\boldsymbol{x}_i^{(q)} \sim \boldsymbol{q}$, i.i.d. for all $i \in [n]$ and letting $\boldsymbol{y}_i = \mathcal{M}_i(\boldsymbol{x}_i^{(q)})$, we would have $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right)$. We have

$$\sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d} \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2$$
$$\overset{\text{(a)}}{\geq} \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2$$
$$\overset{\text{(b)}}{\geq} \frac{1}{2}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 - \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2$$
$$\overset{\text{(c)}}{\geq} \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right) - \frac{d^{1-\frac{2}{p}}}{n}$$
$$\overset{\text{(d)}}{\geq} \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right) \tag{10}$$

Step (a) holds since the LHS is supremum $\{\boldsymbol{x}_i\} \in \mathcal{B}_p^d$ and the RHS of (a) takes expectation w.r.t. $\{\boldsymbol{x}_i^{(q)}\}$ in $\mathcal{B}_p^d$ and hence lower-bounds the LHS. The inequality (b) follows from the Jensen's inequality. Step (c) follows from $\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \leq \frac{d^{1-\frac{2}{p}}}{n}$, which we show below. Step (d) assumes $\min\{\epsilon_0,\epsilon_0^2\} \leq \mathcal{O}(d)$.

Note that for any vector $\boldsymbol{u} \in \mathbb{R}^d$, we have $\|\boldsymbol{u}\|_2 \leq d^{1/2-1/p}\|\boldsymbol{u}\|_p$, for any $p \geq 2$. Since each $\boldsymbol{x}_i^{(q)} \in \mathcal{B}_p^d$, which implies $\|\boldsymbol{x}_i^{(q)}\|_p \leq 1$, we have that $\|\boldsymbol{x}_i^{(q)}\|_2 \leq d^{\frac{1}{2}-\frac{1}{p}}$. Hence, $\mathbb{E}\|\boldsymbol{x}_i^{(q)}\|_2^2 \leq d^{1-\frac{2}{p}}$ holds for all $i \in [n]$. Now, since $\boldsymbol{x}_i$'s are i.i.d. with $\mathbb{E}[\boldsymbol{x}_i^{(q)}] = \boldsymbol{\mu}_{\boldsymbol{q}}$, we have

$$\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 = \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left\|\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2$$
$$\overset{\text{(a)}}{\leq} \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left\|\boldsymbol{x}_i^{(q)}\right\|_2^2 \leq \frac{1}{n^2}\sum_{i=1}^n d^{1-\frac{2}{p}} = \frac{d^{1-\frac{2}{p}}}{n},$$

where (a) uses $\mathbb{E}\|\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]\|_2^2 \leq \mathbb{E}\|\boldsymbol{x}\|_2^2$, which holds for any random vector $\boldsymbol{x}$.

Taking infimum in (10) over all ϵ-LDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and estimators $\widehat{\boldsymbol{x}}$, we get

$$r_{\epsilon_0,\infty,n}^{p,d} =$$
$$\inf_{\{\mathcal{M}_i\in\mathcal{Q}_{(\epsilon_0,\infty)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d} \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2$$
$$\geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right).$$

5.2 Proof of Theorem 4: For ℓ_∞-Norm

In this section, we propose an (ϵ_0, b)-CLDP mechanism for ℓ_∞-norm ball that requires $b = \mathcal{O}\left(\log\left(d\right)\right)$-bits per client using private randomness and 1-bit of communication per client using public randomness.

Each client i has an input $\boldsymbol{x}_i \in \mathcal{B}_\infty^d\left(a\right)$. It selects $j \sim \mathsf{Unif}[d]$ and quantize $x_{i,j}$ according to (11) and obtains $\boldsymbol{z}_i \in \left\{\pm ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j\right\}$, which can be represented using only 1 bit, where $\boldsymbol{e}_j$ is the j'th standard basis vector in $\mathbb{R}^d$. Client i sends $\boldsymbol{z}_i$ to the server. Server receives n messages $\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\}$ from the clients and outputs their average $\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i$. We present the client-side mechanism in Algorithm 2 and state its properties below, which we show in Appendix D.7.

Algorithm 2 ℓ_∞-MEAN-EST ($\mathcal{R}_\infty$: the client-side algorithm)

```
1: Input: $\boldsymbol{x} \in \mathcal{B}_\infty^d\left(a\right)$ and local privacy level $\epsilon_0 > 0$.
2: Sample $j \sim \mathsf{Unif}[d]$ and quantize $x_j$ as follows:
       $\boldsymbol{z} = \begin{cases} +ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j & \text{w.p. } \frac{1}{2} + \frac{x_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \\ -ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j & \text{w.p. } \frac{1}{2} - \frac{x_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \end{cases}$   (11)
   where $\boldsymbol{e}_j$ is the $j$'th standard basis vector in $\mathbb{R}^d$
3: Return $\boldsymbol{z}$.
```

Lemma 2. *The mechanism $\mathcal{R}_\infty$ presented in Algorithm 2 satisfies the following properties, where $\epsilon_0 > 0$:* (i) *$\mathcal{R}_\infty$ is $(\epsilon_0, \log\left(d\right)+1)$-CLDP and requires only* 1*-bit of communication using public randomness.* (ii) *$\mathcal{R}_\infty$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x} \in \mathcal{B}_\infty^d\left(a\right)$, we have $\mathbb{E}\left[\mathcal{R}_\infty\left(\boldsymbol{x}\right)\right] = \boldsymbol{x}$ and $\mathbb{E}\|\mathcal{R}_\infty\left(\boldsymbol{x}\right) - \boldsymbol{x}\|_2^2 \leq a^2d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$.*

Since the server averages the n received messages, we can easily verify that $r_{\epsilon_0,b,n}^{\infty,d}\left(a\right) \leq \frac{a^2d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$.

5.3 Proof of Theorem 1

We show our results on privacy, communication, and convergence separately in the next three subsections.

5.3.1 Privacy

In Algorithm 1, each client applies the compressed LDP mechanism $\mathcal{R}_p$ (hereafter denoted by $\mathcal{R}$, for simplicity) with privacy parameter ϵ_0 on each gradient, which ensures that the mechanism $\mathcal{A}_{cldp}$ guarantees local differential privacy ϵ_0 for each sample d_{ij} per iteration. Thus, it remains to analyze the central DP of the mechanism $\mathcal{A}_{cldp}$.

Fix an iteration number $t \in [T]$. Let $\mathcal{M}_t\left(\theta_t, \mathcal{D}\right)$ denote the private mechanism at time t that takes the dataset $\mathcal{D}$ and an auxiliary input θ_t (which is the parameter vector at the t'th iteration) and generates the parameter θ_{t+1} as an output. Thus, the mechanism $\mathcal{M}_t$ on an input dataset $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i \in \mathfrak{S}^n$ can be defined as:

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{ks} \circ \mathrm{samp}_{m,k}\left(\mathcal{G}_1, \ldots, \mathcal{G}_m\right), \tag{12}$$

where $\mathcal{G}_i = \mathrm{samp}_{r,s}\left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{ir}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [m], j \in [r]$. Here, $\mathcal{H}_{ks}$ denotes the shuffling operation on ks elements and $\mathrm{samp}_{a,b}$ denotes the sampling operation for choosing a random subset of b elements from a set of a elements.

Now we state the privacy guarantee of the mechanism $\mathcal{M}_t$ for each $t \in [T]$.

Lemma 3. *Let $s = 1$ and $q = \frac{k}{mr}$. Suppose $\mathcal{R}$ is an ϵ_0-LDP mechanism, where $\epsilon_0 \leq \frac{\log\left(qn/\log\left(1/\tilde{\delta}\right)\right)}{2}$ and $\tilde{\delta} > 0$ is arbitrary. Then, for any $t \in [T]$, the mechanism $\mathcal{M}_t$ is $\left(\overline{\epsilon}, \overline{\delta}\right)$-DP, where $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)), \overline{\delta} = q\tilde{\delta}$ with $\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\} e^{\epsilon_0} \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{qn}}\right)$. In particular, if $\epsilon_0 = \mathcal{O}\left(1\right)$, we get $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{n}}\right)$.*

We prove Lemma 3 in Appendix C. In the statement of Lemma 3, we are amplifying the privacy by using the subsampling as well as shuffling ideas. For subsampling, note that we do not pick a uniformly random subset of size ks from $n = mr$ points. So, we cannot directly apply the amplification by subsampling result of Kasiviswanathan et al. [2011] (stated in Lemma 7 in Appendix B.3.1). However, as it turns out that the only property we will need for privacy amplification by subsampling is that each data point is picked with probability $q = \frac{ks}{mr}$, which holds true in our setting.

Consider two neighboring datasets $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i$, $\mathcal{D}' = \mathcal{D}_1' \bigcup (\bigcup_{i=2}^{m} \mathcal{D}_i)$ that are different only in the first data point at the first client d_{11}. The main idea of the proof is to split the probability distribution of the output of the mechanism $\mathcal{M}_t$ into a summation of four conditional probabilities depending on the event whether the first client is picked or not and the first client pick the first data point or not. We use bipartite graphs to get the relation between these events, where each vertex corresponds to one of the possible outputs of the sampling procedure, and each edge connects two neighboring vertices. See Appendix C for more details.

Note that the Algorithm $\mathcal{A}_{cldp}$ is a sequence of T adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_T$, where each $\mathcal{M}_t$ for $t \in [T]$ satisfies the privacy guarantee stated in Lemma 3. Now, we invoke the strong composition theorem from [Dwork and Roth, 2014, Theorem 3.20] (stated in Lemma 6 in Appendix B.2) to obtain the privacy guarantee of the algorithm $\mathcal{A}_{cldp}$ as stated in Theorem 1. We provide the details in Appendix F.

5.3.2 Communication

The (ϵ_0, b)-CLDP mechanism $\mathcal{R}_p : \mathcal{X} \to \mathcal{Y}$ used in Algorithm 1 has output alphabet $\mathcal{Y} = \{1, 2, \ldots, B = 2^b\}$. So, the naïve scheme for any client to send the s compressed and private gradients requires sb bits per iteration. We can reduce this communication cost by using the histogram trick from [Mayekar and Tyagi, 2020] which was applied in the context of non-private quantization. The idea is as follows. Since all clients apply the *same* randomized mechanism $\mathcal{R}_p$ to the s gradients, the output of these s identical mechanisms can be represented accurately using the histogram of the s outputs, which takes value from the set $\mathcal{A}_B^s = \{(n_1, \ldots, n_B) : \sum_{j=1}^{B} n_j = s \text{ and } n_j \geq 0, \forall j \in [B]\}$. Since $|\mathcal{A}_B^s| = \binom{s+B-1}{s} \leq \left(\frac{e(s+B-1)}{s}\right)^s$, it requires at most $s\left(\log\left(e\right) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits to send the s compressed gradients. Since a client is chosen with probability $\frac{k}{m}$ at any time $t \in [T]$, the expected number of bits per client in Algorithm $\mathcal{A}_{cldp}$ is given by $\frac{k}{m} \times T \times s\left(\log\left(e\right) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits, where expectation is taken over the sampling of clients.

5.3.3 Convergence

At iteration $t \in [T]$ of Algorithm 1, server averages the received ks compressed and privatized gradients and obtains $\overline{\mathbf{g}}_t = \frac{1}{ks}\sum_{i \in \mathcal{U}_t}\sum_{j \in \mathcal{S}_{it}} \mathbf{q}_t(d_{ij})$ (line 13 of Algorithm 1) and then updates the parameter vector as $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}\left(\theta_t - \eta_t \overline{\mathbf{g}}_t\right)$. Here, $\mathbf{q}_t(d_{ij}) = \mathcal{R}_p\left(\nabla_{\theta_t} f(\theta_t; d_{ij})\right)$. Since the randomized mechanism $\mathcal{R}_p$ is unbiased, the average gradient $\overline{\mathbf{g}}_t$ is also unbiased, i.e., we have $\mathbb{E}\left[\overline{\mathbf{g}}_t\right] = \nabla_{\theta_t} F\left(\theta_t\right)$, where expectation is taken w.r.t. the sampling of clients and the data points as well as the randomness of the mechanism $\mathcal{R}_p$. Now we show that $\overline{\mathbf{g}}_t$ has a bounded second moment.

Lemma 4. *For any $d \in \mathfrak{S}$, if the function $f\left(\theta; .\right)$:*

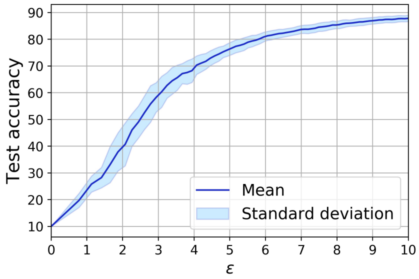

Figure 2: Privacy-Utility trade-offs on the MNIST dataset with ℓ_∞-norm clipping.

$\mathcal{C} \to \mathbb{R}$ is convex and L-Lipschitz continuous w.r.t. the ℓ_g-norm, which is the dual of ℓ_p-norm, then we have

$$\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 \leq L^2 \max\{d^{1-\frac{2}{p}}, 1\} \left(1 + \frac{cd}{qn}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\right),$$

where $c = 4$ if $p \in \{1, \infty\}$ and $c = 14$ otherwise.

Lemma 4 is proved in Appendix F.3.

Now, using the bound on G^2 from Lemma 4 in the following standard SGD convergence results for convex functions proves the third part of Theorem 1; see Appendix F.3 for more details.

Lemma 5 (SGD Convergence [Shamir and Zhang, 2013])**.** *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 $\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq 2DG\left(\frac{2+\log(T)}{\sqrt{T}}\right)$.*

6 Numerical Results

In this section, we present our numerical results to evaluate the proposed CLDP-SGD algorithm for training machine learning models with privacy and communication constraints. We consider the standard MNIST handwritten digit dataset that has $60,000$ training images and $10,000$ test images. We train a simple neural network that was also used in [Erlingsson et al., 2020, Papernot et al., 2020] and described in Table 1. This model has a total number of $d = 13,170$ parameters and achieves an accuracy of 99% for non-private, uncompressed vanilla SGD. In our results, we assume that we have $60,000$ clients, where each client has one sample, i.e., $m = n = 60,000$ and $r = 1$. We present our results for ℓ_∞-norm clipping. At each step of the CLDP-SGD, we choose at random $10,000$ clients. Each client clips the ℓ_∞-norm of the gradient $\nabla_{\theta_t} f(\theta_t; d_i)$ with clipping parameter $C = 1/100$. After that, the client applies the LDP-compression mechanism $\mathcal{R}_\infty$ (presented in Algorithm 2) to the clipped gradient. We run our algorithm for 80 epochs, where we set the learning rate at 0.3 for the first 70 epochs and decrease it to 0.18 in the remaining epochs. We set the local privacy parameters $\epsilon_0 = 2$ and $\delta = 10^{-5}$, while the centralized privacy parameter ϵ is computed numerically from Theorem 1 as follows. We first compute the privacy amplification by shuffling numerically using the expression in [Balle et al., 2019c, Theorem 5.3]. Then, we compute the privacy amplification via subsampling presented in Lemma 3; and finally we use the strong composition stated in Lemma 6 in Appendix B.2 to obtain the central privacy parameter ϵ.

Layer	Parameters
Convolution	16 filters of 8×8, Stride 2
Max-Pooling	2×2
Convolution	32 filters of 4×4, Stride 2
Max-Pooling	2×2
Fully connected	32 units
Softmax	10 units

Table 1: Model Architecture for MNIST

Figure 2 demonstrates the mean and the standard deviation of privacy-accuracy plot averaged over 10 runs. It shows that we can achieve an accuracy 76.7% (± 2) for total privacy $\epsilon = 5$ and an accuracy 87.9% (± 1) for total privacy $\epsilon = 10$. Furthermore, observe that our proposed CLDP-SGD algorithm preserves a local privacy of $\epsilon_0 = 2$ per sample per epoch. In addition, the private mechanism $\mathcal{R}_\infty$ requires only $\lceil \log(d) \rceil + 1$ bits per gradient, while the full precision gradient requires $32 \times d$ bits per gradient. Thus, the proposed private mechanism saves in communication bits a factor of $28096\times$ in comparison with the full precision gradient.

In [Papernot et al., 2020], the authors achieve a test accuracy of 98% on MNIST with central privacy parameters $\epsilon = 3$ and $\delta = 10^{-5}$ using a DP centralized algorithm by adding Gaussian noise to the aggregated gradients in each iteration. However, Papernot et al. [2020] do not offer any local differential privacy guarantees, which can be thought of as $\epsilon_0 = \infty$. Although, Theorem 1 and Remark 3 show that our proposed algorithm matches theoretically the results of the centralized SGD with full precision gradients, the numerical results show that there is a gap between the accuracy of our algorithm and the test accuracy of the centralized algorithm in Papernot et al. [2020]. We believe that the privacy parameters of our algorithm can be improved by analyzing the Renyi differential privacy of the shuffled model, which is an important open question of the ongoing investigation.

Acknowledgements

This was supported by the NSF grant #1740047 and by the UC-NL grant LFR-18-548554. This work was also supported in part through the Google Faculty Research Award.

References

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

J. Acharya and Z. Sun. Communication complexity in locally private distribution estimation and heavy hitters. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 2019.

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*, pages 1120–1129, 2019.

N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, pages 5973–5983, 2018.

B. Balle, J. Bell, A. Gascon, and K. Nissim. Differentially private summation with multi-message shuffling. *arXiv preprint arXiv:1906.09116*, 2019a.

B. Balle, J. Bell, A. Gascón, and K. Nissim. Improved summation from shuffling. *arXiv preprint arXiv:1909.11225*, 2019b.

B. Balle, J. Bell, A. Gascón, and K. Nissim. The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, pages 638–667. Springer, 2019c.

B. Balle, J. Bell, A. Gascón, and K. Nissim. Private summation in the multi-message shuffle model. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 657–676. ACM, 2020a. doi: 10.1145/3372297.3417242.

B. Balle, P. Kairouz, B. McMahan, O. D. Thakkar, and A. Thakurta. Privacy amplification via random check-ins. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b.

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

D. Basu, D. Data, C. Karakus, and S. Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, pages 14695–14706, 2019.

A. Beimel, K. Nissim, and E. Omri. Distributed private data analysis: Simultaneously solving how and what. In *Annual International Cryptology Conference*, pages 451–468. Springer, 2008.

A. Beimel, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. In *Theory of Cryptography Conference*, pages 437–454. Springer, 2010.

A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

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

W. Chen, P. Kairouz, and A. Özgür. Breaking the communication-privacy-accuracy trilemma. In *Advances in Neural Information Processing Systems*, 2020.

A. Cheu, A. D. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology - EUROCRYPT 2019*, volume 11476, pages 375–403. Springer, 2019.

J. C. Duchi and R. Rogers. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory (COLT)*, pages 1161–1191, 2019.

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

J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

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

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

C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 51–60, 2010.

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 *SODA*, pages 2468–2479. SIAM, 2019.

Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, S. Song, K. Talwar, and A. Thakurta. Encode, shuffle, analyze privacy revisited: formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. *Information Systems*, 29(4):343–364, 2004.

V. Gandikota, D. Kane, R. K. Maity, and A. Mazumdar. vqsgd: Vector quantized stochastic gradient descent. *arXiv preprint arXiv:1911.07971*, 2019.

B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. On the power of multiple anonymous messages. *IACR Cryptol. ePrint Arch.*, 2019:1382, 2019a.

B. Ghazi, R. Pagh, and A. Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *arXiv preprint arXiv:1906.08320*, 2019b.

B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *International Conference on Machine Learning (ICML)*, pages 3505–3514, 2020.

A. M. Girgis, D. Data, K. Chaudhuri, C. Fragouli, and S. N. Diggavi. Successive refinement of privacy. *IEEE Journal on Selected Areas in Information Theory*, 1(3):745–759, 2020. doi: 10.1109/JSAIT.2020.3040403.

P. Kairouz, K. Bonawitz, and D. Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning, ICML*, pages 2436–2444, 2016.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *ICML*, pages 3252–3261, 2019.

S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40 (3):793–826, 2011.

P. Mayekar and H. Tyagi. Limits on gradient compression for stochastic optimization. *IEEE International Symposium on Information Theory (ISIT)*, 2020.

N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *arXiv preprint arXiv:2007.14191*, 2020.

S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79, 2013.

N. Singh, D. Data, J. George, and S. Diggavi. Sparq-sgd: Event-triggered and compressed communication in decentralized stochastic optimization. *arXiv preprint arXiv:1910.14280*, 2019.

N. Singh, D. Data, J. George, and S. Diggavi. Squarm-sgd: Communication-efficient momentum sgd for decentralized optimization. *arXiv preprint arXiv:2005.07041*, 2020.

S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3329–3337. JMLR. org, 2017.

J. Ullman. Cs7880. rigorous approaches to data privacy. 2017. URL `http://www.ccs.neu.edu/home/jullman/cs7880s17/HW1sol.pdf`.

S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Supplementary Material

A Related Work

Among the several main challenges in the recently developed FL framework (see Kairouz et al. [2019] and references therein), we focus in this paper on the combination of privacy and communication efficiency, and examining its impact on model learning. We briefly review some of the main developments in related papers on these topics below.

A.1 Communication-Privacy Trade-offs

Distributed mean estimation and its use in training learning models has been studied extensively in the literature (see [Alistarh et al., 2017, Gandikota et al., 2019, Mayekar and Tyagi, 2020, Suresh et al., 2017] and references therein). In [Suresh et al., 2017], the authors have proposed a communication efficient scheme for estimating the mean of set a of vectors distributed over multiple clients. Acharya et al. [2019] studied the discrete distribution estimation under LDP. They proposed a randomized mechanism based on Hadamard coding which is optimal for all privacy regime and requires $\mathcal{O}\left(\log\left(d\right)\right)$ bits per client, where d denotes the support size of the discrete distribution. In [Acharya and Sun, 2019], the authors consider both private and public coin mechanisms, and show that the Hadamard mechanism is near optimal in terms of communication for both distribution and frequency estimation. Recently, Chen et al. [2020] proposed a communication efficient scheme for mean estimation under local differential privacy constraints. This work is is done concurrently and independently of our work. Furthermore, it focuses on mean estimation for bounded ℓ_2-norm vectors, in contrast to our optimization approach, privacy amplification through sampling and shuffling. Also, this work considers the existence of public randomness, while we do not need public randomness.

LDP mechanisms suffer from the utility degradation that motivates other work to find alternative techniques to improve the utility under LDP. One of new developments in privacy is the use of anonymization to amplify the privacy by using secure shuffler. In [Balle et al., 2019c, 2020a, Cheu et al., 2019], the authors studied the mean estimation problem under LDP with secure shuffler, where they show that the shuffling provides better utility than the LDP framework without shuffling.

A.2 Private Optimization

Chaudhuri et al. [2011] studied *centralized* privacy-preserving machine learning algorithms for convex optimization problem. The authors proposed a new idea of perturbing the objective function to preserve privacy of the training dataset. Bassily et al. [2014] derived lower bounds on the empirical risk minimization under *central* differential privacy constraints. Furthermore, they proposed a differential privacy SGD algorithm that matches the lower bound for convex functions. In [Abadi et al., 2016], the authors have generalized the private SGD algorithm proposed in [Bassily et al., 2014] for non-convex optimization framework. In addition, the authors have proposed a new analysis technique, called moment accounting, to improve on the strong composition theorems to compute the central differential privacy guarantee for iterative algorithms. However, the works mentioned, Abadi et al. [2016], Bassily et al. [2014], Chaudhuri et al. [2011], assume that there exists a trusted server that collects the clients' data. This motivates other works to design a distributed SGD algorithms, where each client perturbs her own data without needing a trusted server. For this, the natural privacy framework is *local* differential privacy or LDP (*e.g.,* see [Bhowmick et al., 2018, Duchi et al., 2013, Evfimievski et al., 2004, Warner, 1965]). However, it is well understood that LDP does not give good performance guarantees as it requires significant local randomization to give privacy guarantees [Duchi et al., 2013, Kairouz et al., 2016, Kasiviswanathan et al., 2011]. The two most related papers to our work are [Agarwal et al., 2018, Erlingsson et al., 2020] which we describe below.

Erlingsson et al. [2020] proposed a distributed local-differential-privacy gradient descent algorithm, where each client has one sample. In their proposed algorithm, each client perturbs the gradient of her sample using an LDP mechanism. To improve upon the LDP performance guarantees, they use the newly proposed anonymization/shuffling framework [Balle et al., 2019c]. Therefore in their work, gradients of all clients are passed through a secure shuffler that eliminates the identities of the clients to amplify the central privacy guarantee. However, their proposed algorithm is not communication efficient, where each client has to send the full-precision gradient without compression. Our work is different from [Erlingsson et al., 2020], as we propose a communication efficient mechanism for each client that requires $O(\log d)$ bits per client, which can be significant for large d. Furthermore, our algorithm consider multiple data samples at client, which is accessed through a mini-batch random sampling at each iteration of the optimization. This requires a careful combination of compression and privacy analysis in order to preserve the variance reduction of mini-batch as well as privacy.[8] In addition we obtain a gain in privacy by using the fact that (anonymized) clients are sampled (*i.e.,* not all clients are selected at each iteration) as motivated by the federated learning framework.

Agarwal et al. [2018] proposed a communication-efficient algorithm for learning models with central differential privacy. Let n be the number of clients per round and d be the dimensionality of the parameter space. They proposed cp-SGD, a communication efficient algorithm, where clients need to send $O(\log(1+\frac{d}{n}\epsilon^2)+\log\log\log\frac{nd}{\epsilon\delta})$ bits of communication *per coordinate*, *i.e.,* $O\left(d\left\{\log(1+\frac{d}{n}\epsilon^2)+\log\log\log\frac{nd}{\epsilon\delta}\right\}\right)$ bits per round to achieve the same local differential privacy guarantees of ϵ_0 as the Gaussian mechanism. Their algorithm is based on a Binomial noise addition mechanism and secure aggregation. In contrast, we propose a generic framework to convert any LDP algorithm to a central differential privacy guarantee and further use recent results on amplification by shuffling, that also achieves better compression in terms of number of bits per client.

B Background tools

B.1 Differential Privacy

In this section, we formally define local differential privacy (LDP) and (central) differential privacy (DP). First we recall the standard definition of LDP [Kasiviswanathan et al., 2011].

Definition 3 (Local Differential Privacy - LDP [Kasiviswanathan et al., 2011])**.** For $\epsilon_0 \geq 0$ and $b \in \mathbb{N}^+ := \{1,2,3,\ldots\}$, a randomized mechanism $\mathcal{R}:\mathcal{X}\to\mathcal{Y}$ is said to be ϵ_0-local differentially private (in short, ϵ_0-LDP), if for every pair of inputs $\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}$, we have

$$\Pr[\mathcal{R}(\boldsymbol{x})=\boldsymbol{y}] \leq \exp(\epsilon)\Pr[\mathcal{R}(\boldsymbol{x}')=\boldsymbol{y}], \qquad \forall \boldsymbol{y}\in\mathcal{Y}. \tag{13}$$

In our problem formulation, since each client has a communication budget on what it can send in each SGD iteration while keeping its data private, it would be convenient for us to define two parameter LDP with privacy and communication budget.

Definition 4 (Local Differential Privacy with Communication Budget - CLDP)**.** For $\epsilon_0 \geq 0$ and $b\in\mathbb{N}^+$, a randomized mechanism $\mathcal{R}:\mathcal{X}\to\mathcal{Y}$ is said to be (ϵ_0,b)-communication-limited-local differentially private (in short, (ϵ_0,b)-CLDP), if for every pair of inputs $\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}$, we have

$$\Pr[\mathcal{R}(\boldsymbol{x})=\boldsymbol{y}] \leq \exp(\epsilon)\Pr[\mathcal{R}(\boldsymbol{x}')=\boldsymbol{y}], \qquad \forall \boldsymbol{y}\in\mathcal{Y}. \tag{14}$$

Furthermore, the output of $\mathcal{R}$ can be represented using b bits.

Here, ϵ_0 captures the privacy level, lower the ϵ_0, higher the privacy. When we are not concerned about the communication budget, we succinctly denote the corresponding (ϵ_0,∞)-CLDP, by its correspondence to the classical LDP as ϵ_0-LDP [Kasiviswanathan et al., 2011].

Let $\mathcal{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ denote a dataset comprising n points from $\mathcal{X}$. We say that two datasets $\mathcal{D}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ and $\mathcal{D}'=\{\boldsymbol{x}'_1,\ldots,\boldsymbol{x}'_n\}$ are neighboring if they differ in one data point. In other words, $\mathcal{D}$ and $\mathcal{D}'$ are neighboring if there exists an index $i\in[n]$ such that $\boldsymbol{x}_i\neq\boldsymbol{x}'_i$ and $\boldsymbol{x}_j=\boldsymbol{x}'_j$ for all $j\neq i$.

Definition 5 (Central Differential Privacy - DP [Dwork and Roth, 2014, Dwork et al., 2006])**.** For $\epsilon,\delta\geq 0$, a randomized mechanism $\mathcal{M}:\mathcal{X}^n\to\mathcal{Y}$ is said to be (ϵ,δ)-differentially private (in short, (ϵ,δ)-DP), if for all neighboring datasets $\mathcal{D},\mathcal{D}'\in\mathcal{X}^n$ and every subset $\mathcal{E}\subseteq\mathcal{Y}$, we have

$$\Pr\left[\mathcal{M}\left(\mathcal{D}\right)\in\mathcal{E}\right] \leq \exp(\epsilon)\Pr\left[\mathcal{M}\left(\mathcal{D}'\right)\in\mathcal{E}\right]+\delta. \tag{15}$$

Remark 4. For any ϵ_0-LDP mechanism $\mathcal{R}:\mathcal{X}\to\mathcal{Y}$, it is easy to verify that the randomized mechanism $\mathcal{M}:\mathcal{X}^n\to\mathcal{Y}$ defined by $\mathcal{M}\left(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\right):=\left(\mathcal{R}\left(\boldsymbol{x}_1\right),\ldots,\mathcal{R}\left(\boldsymbol{x}_n\right)\right)$ is $(\epsilon_0,0)$-DP.

[8] The naive method of quantizing the aggregated mini-batch gradient will fail to preserve the required variance reduction.

Remark 5. Note that in this paper we make a clear distinction between the notation used for central differential privacy, denoted by (ϵ, δ)-DP (see Definition 5), local differential privacy ϵ_0-LDP (see definition 3) and communication limited local differential privacy, denoted by (ϵ_0, b)-CLDP (see Definition 4).

The main objective of this paper is to make SGD differentially private and communication-efficient, suitable for federated learning. For that we compress and privatize gradients in each SGD iteration. Since the parameter vectors in any iteration depend on the previous iterations, so do the gradients, which makes this procedure a sequence of many adaptive DP mechanisms. We can calculate the final privacy guarantees achieved at the end of this procedure by using composition theorems.

B.2 Strong Composition [Dwork et al., 2010]

Let $\mathcal{M}_1\left(\mathcal{I}_1, \mathcal{D}\right), \ldots, \mathcal{M}_T\left(\mathcal{I}_T, \mathcal{D}\right)$ be a sequence of T adaptive DP mechanisms, where $\mathcal{I}_i$ denotes the auxiliary input to the ith mechanism, which may depend on the previous mechanisms' outputs and the auxiliary inputs $\{(\mathcal{I}_j, \mathcal{M}_j(\mathcal{I}_j, \mathcal{D})) : j < i\}$. There are different composition theorems in literature to analyze the privacy guarantees of the composed mechanism $\mathcal{M}(\mathcal{D}) = \left(\mathcal{M}_1\left(\mathcal{I}_1, \mathcal{D}\right), \ldots, \mathcal{M}_T\left(\mathcal{I}_T, \mathcal{D}\right)\right)$.

Dwork et al. [2010] provided a strong composition theorem (which is stronger than the basic composition theorem in which the privacy parameters scale linearly with T) where the privacy parameter of the composition mechanism scales as $\sqrt{T}$ with some loss in δ. Below, we provide a formal statement of that result from Dwork and Roth [2014].

Lemma 6 (Strong Composition, [Dwork and Roth, 2014, Theorem 3.20])**.** *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be T adaptive $(\overline{\epsilon}, \overline{\delta})$-DP mechanisms, where $\overline{\epsilon}, \overline{\delta} \geq 0$. Then, for any $\delta' > 0$, the composed mechanism $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_T)$ is (ϵ, δ)-DP, where*

$$\epsilon = \sqrt{2T \log\left(1/\delta'\right)}\overline{\epsilon} + T\overline{\epsilon}\left(e^{\overline{\epsilon}} - 1\right), \quad \delta = T\overline{\delta} + \delta'.$$

In particular, when $\overline{\epsilon} = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta')}{T}}\right)$, we have $\epsilon = \mathcal{O}\left(\overline{\epsilon}\sqrt{T \log\left(1/\delta'\right)}\right)$.

Note that training large-scale machine learning models (e.g., in deep learning) typically requires running SGD for millions of iterations, as the dimension of the model parameter is quite large. We can make it differentially private by adding noise to the gradients in each iteration, and appeal to the strong composition theorem to bound the privacy loss of the entire process (which in turn dictates the amount of noise to be added in each iteration).

B.3 Privacy Amplification

In this section, we describe the techniques that can be used for privacy amplification. The first one amplifies privacy by subsampling the data (to compute stochastic gradients) as well as the clients (as in FL), and the other one amplifies privacy by shuffling.

B.3.1 Privacy Amplification by Subsampling

Suppose we have a dataset $\mathcal{D}' = \{U_1, \ldots, U_{r_1}\} \in \mathcal{U}^{r_1}$ consisting of r_1 elements from a universe $\mathcal{U}$. A subsampling procedure takes a dataset $\mathcal{D}' \in \mathcal{U}^{r_1}$ and subsamples a subset from it as formally defined below.

Definition 6 (Subsampling)**.** The subsampling operation $\text{samp}_{r_1, r_2} : \mathcal{U}^{r_1} \to \mathcal{U}^{r_2}$ takes a dataset $\mathcal{D}' \in \mathcal{U}^{r_1}$ as input and selects uniformly at random a subset $\mathcal{D}''$ of $r_2 \leq r_1$ elements from $\mathcal{D}'$. Note that each element of $\mathcal{D}'$ appears in $\mathcal{D}''$ with probability $q = \frac{r_2}{r_1}$.

The following result states that the above subsampling procedure amplifies the privacy guarantees of a DP mechanism.

Lemma 7 (Amplification by Subsampling, [Kasiviswanathan et al., 2011])**.** *Let $\mathcal{M} : \mathcal{U}^{r_2} \to \mathcal{V}$ be an (ϵ, δ)-DP mechanism. Then, the mechanism $\mathcal{M}' : \mathcal{U}^{r_1} \to \mathcal{V}$ defined by $\mathcal{M}' = \mathcal{M} \circ \text{samp}_{r_1, r_2}$ is (ϵ', δ')-DP, where $\epsilon' = \log(1 + q(e^{\epsilon} - 1))$ and $\delta' = q\delta$ with $q = \frac{r_2}{r_1}$. In particular, when $\epsilon < 1$, $\mathcal{M}'$ is $(\mathcal{O}(q\epsilon), q\delta)$-DP.*

Note that in the case of subsampling the data for computing stochastic gradients, where client i selects a mini-batch of size s from its local dataset $\mathcal{D}_i$ that has r data points, we take $\mathcal{D}' = \mathcal{D}_i$, $r_1 = r$, and $r_2 = s$. In the case of subsampling the clients, k clients are randomly selected from the m clients, we take $\mathcal{D}' = \{1, 2, \ldots, m\}$, $r_1 = m$, and $r_2 = k$. An important point is that such a sub-sampling is not uniform overall (i.e., this does not imply that any subset of ks data points is chosen with equal probability) and we cannot directly apply the above result. We need to revisit the proof of Lemma 7 to adapt it to our case, and we do it in Lemma 3, which is proved in Appendix C. In fact, the proof of Lemma 3 is more general than just adapting the amplification by subsampling to our setting, it also incorporates the amplification by shuffling, which is crucial for obtaining strong privacy guarantees. We describe it next.

B.3.2 Privacy Amplification by Shuffling

Consider a set of m clients, where client $i \in [m]$ has a data $\boldsymbol{x}_i \in \mathcal{X}$. Let $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ be an ϵ_0-LDP mechanism. The i-th client applies $\mathcal{R}$ on her data $\boldsymbol{x}_i$ to get a private message $\boldsymbol{y}_i = \mathcal{R}(\boldsymbol{x}_i)$. There is a secure shuffler $\mathcal{H}_m : \mathcal{Y}^m \to \mathcal{Y}^m$ that receives the set of m messages $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$ and generates the same set of messages in a uniformly random order.

The following lemma states that the shuffling amplifies the privacy of an LDP mechanism by a factor of $\frac{1}{\sqrt{m}}$.

Lemma 8 (Amplification by Shuffling)**.** *Let $\mathcal{R}$ be an ϵ_0-LDP mechanism. Then, the mechanism $\mathcal{M}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) := \mathcal{H}_m \circ (\mathcal{R}(\boldsymbol{x}_1), \ldots, \mathcal{R}(\boldsymbol{x}_m))$ satisfies (ϵ, δ)-differential privacy, where*

1. [Balle et al., 2019c, Corollary 5.3.1]. *If $\epsilon_0 \leq \frac{\log(m/\log(1/\delta))}{2}$, then for any $\delta > 0$, we have* $\epsilon = \mathcal{O}\left(\min\{\epsilon_0, 1\} e^{\epsilon_0} \sqrt{\frac{\log(1/\delta)}{m}}\right)$.
2. [Erlingsson et al., 2019, Corollary 9]. *If $\epsilon_0 < \frac{1}{2}$, then for any $\delta \in (0, \frac{1}{100})$ and $m \geq 1000$, we have* $\epsilon = 12\epsilon_0 \sqrt{\frac{\log(1/\delta)}{m}}$.

In our proposed algorithm, only $k \leq m$ clients send messages and each client sends a mini-batch of s gradients. So, in total, shuffler applies the shuffling operation on ks gradients. In our algorithm, though sampling and shuffling are applied one after another (first k clients are sampled, then each client samples s data points, and then shuffling of these ks data points is performed), we analyze the privacy amplification we get using both of these techniques by analyzing them together; see Lemma 3 proved in Appendix C.

B.4 Compressed and Private Mean Estimation via Minimax Risk

Recall that in each SGD iteration, server sends the current parameter vector to all clients, upon receiving which they compute stochastic gradients from their local datasets and send them to the server, who then computes the average/mean of received gradients and updates the parameter vector. Note that these gradients (over the entire execution of algorithm) may also leak information about the datasets. As mentioned in Section 1, we also compress the gradients to mitigate the communication bottleneck.

In this section, we formulate the generic mimimax estimation framework for mean estimation of a given set of n vectors that preserves privacy and is also communication-efficient. We then apply that method at the server in each SGD iteration for aggregating the gradients. We derive upper and lower bounds for various ℓ_p geometries for $p \geq 1$ including the ℓ_∞-norm. Let us setup the problem. For any $p \geq 1$ and $d \in \mathbb{N}$, let $\mathcal{B}_p^d(a) = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \leq a\}$ denote the p-norm ball with radius a centered at the origin in $\mathbb{R}^d$,[9] where $\|\boldsymbol{x}\|_p = \left(\sum_{j=1}^d |\boldsymbol{x}_j|^p\right)^{1/p}$. Each client $i \in [n]$ has an input vector $\boldsymbol{x}_i \in \mathcal{B}_p^d(a)$ and the server wants to estimate the mean $\overline{\boldsymbol{x}} := \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$. We have two constraints: (i) each client has a communication budget of b bits to transmit the information about its input vector to the server, and (ii) each client wants to keep its input vector private from the server. We develop *private-quantization* mechanisms to simultaneously address these constraints. Specifically, we design mechanisms $\mathcal{M}_i : \mathcal{B}_p^d(a) \to \{0,1\}^d$ for $i \in [n]$ that are quantized in the sense that they produce a b-bit output and are also locally differentially private. In other words, $\mathcal{M}_i$ is (ϵ_0, b)-LDP for some $\epsilon_0 \geq 0$ (see Definition 4).

[9] Assuming that the ball is centered at origin is without loss of generaility; otherwise, we can translate the ball to origin and work with that.

The procedure goes as follows. client $i \in [n]$ applies a private-quantization mechanism $\mathcal{M}_i$ on her input $\boldsymbol{x}_i$ and obtains a private output $\boldsymbol{y}_i = \mathcal{M}_i(\boldsymbol{x}_i)$ and sends it to the server. Upon receiving $\boldsymbol{y}^n = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$, server applies a decoding function to estimate the mean vector $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$. Our objective is to design private-quantization mechanisms $\mathcal{M}_i : \mathcal{B}_p^d(a) \to \{0,1\}^d$ for all $i \in [n]$ and also a (stochastic) decoding function $\widehat{\boldsymbol{x}} : \left(\{0,1\}^b\right)^n \to \mathcal{B}_p^d$ that minimizes the worst-case expected error $\sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d} \mathbb{E}\|\overline{\boldsymbol{x}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\|^2$. In other words, we are interested in characterizing the following quantity.

$$r_{\epsilon,b,n}^{p,d}(a) = \inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,b)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d(a)} \mathbb{E}\left\|\overline{\boldsymbol{x}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2, \tag{16}$$

where $\mathcal{Q}_{(\epsilon,b)}$ is the set of all (ϵ, b)-LDP mechanisms, and the expectation is taken over the randomness of $\{\mathcal{M}_i : i \in [n]\}$ and the estimator $\widehat{\boldsymbol{x}}$. Note that in (16) we do not assume any probabilistic assumptions on the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

Now we extend the formulation in (16) to a probabilistic model. Let $\mathcal{P}_p^d(a)$ denote the set of all probability density functions on $\mathcal{B}_p^d(a)$. For every distribution $\boldsymbol{q} \in \mathcal{P}_p^d(a)$, let $\boldsymbol{\mu}_{\boldsymbol{q}}$ denote its mean. Since the support of each distribution $\boldsymbol{q} \in \mathcal{P}_p^d$ is $\mathcal{B}_p^d(a)$ and ℓ_p is a norm, we have that $\boldsymbol{\mu}_{\boldsymbol{q}} \in \mathcal{B}_p^d(a)$. For a given unknown distribution $\boldsymbol{q} \in \mathcal{P}_p^d(a)$, client $i \in [n]$ observes $\boldsymbol{x}_i$, where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are i.i.d. according to $\boldsymbol{q}$, and the goal for the server is to estimate $\boldsymbol{\mu}_{\boldsymbol{q}}$, while satisfying the same two constraints as above, i.e., only b bits of communication is allowed from any client to the server while preserving the privacy of clients' inputs. Analogous to (16), we are interested in characterizing the following quantity.

$$R_{\epsilon,b,n}^{p,d}(a) = \inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,b)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q}\in\mathcal{P}_p^d(a)} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2, \tag{17}$$

where the expectation is taken over the randomness of the output $\boldsymbol{y}^n$ and the estimator $\widehat{\boldsymbol{x}}$.

In this paper, we design private-quantization mechanisms $\{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ such that they are symmetric (i.e., $\mathcal{M}_i$'s are same for all $i \in [n]$) and any client uses only private source of randomness that is not accessible by any other party in the system.

C Proof of Lemma 3

This entire section is devoted to proving Lemma 3. For convenience, we restate the lemma below.

Lemma (Restating Lemma 3)**.** *Let $s = 1$ and $q = \frac{k}{mr}$. Suppose $\mathcal{R}$ is an ϵ_0-LDP mechanism, where $\epsilon_0 \leq \frac{\log\left(qn/\log\left(1/\tilde{\delta}\right)\right)}{2}$ and $\tilde{\delta} > 0$ is arbitrary. Then, for any $t \in [T]$, the mechanism $\mathcal{M}_t$ is $\left(\overline{\epsilon}, \overline{\delta}\right)$-DP, where $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)), \overline{\delta} = q\tilde{\delta}$ with $\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\}e^{\epsilon_0}\sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{qn}}\right)$. In particular, if $\epsilon_0 = \mathcal{O}(1)$, we get $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{n}}\right)$.*

Recall that the input dataset at client $i \in [m]$ is denoted by $\mathcal{D}_i = \{d_{i1}, d_{i2}, \ldots, d_{ir}\} \in \mathfrak{S}^r$ and $\mathcal{D} = \bigcup_{i=1}^m \mathcal{D}_i$ denotes the entire dataset. Recall from (12) that the mechanism $\mathcal{M}_t$ on input dataset $\mathcal{D}$ can be defined as:

$$\mathcal{M}_t(\mathcal{D}) = \mathcal{H}_{ks} \circ \text{samp}_{m,k}\left(\mathcal{G}_1, \ldots, \mathcal{G}_m\right), \tag{18}$$

where $\mathcal{G}_i = \text{samp}_{r,s}\left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{ir}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [m], j \in [r]$. We define a mechanism $\mathcal{Z}\left(\mathcal{D}^{(t)}\right) = \mathcal{H}_{ks}\left(\mathcal{R}\left(\boldsymbol{x}_1^t\right), \ldots, \mathcal{R}\left(\boldsymbol{x}_{ks}^t\right)\right)$ which is a shuffling of ks outputs of local mechanism $\mathcal{R}$, where $\mathcal{D}^{(t)}$ denotes an arbitrary set of ks data points and we index $\boldsymbol{x}_i^t$'s from $i = 1$ to ks just for convenience. From the amplification by shuffling result [Balle et al., 2019c, Corollary 5.3.1] (also see Lemma 8), the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, where $\tilde{\delta} > 0$ is arbitrary, and, if $\epsilon_0 \leq \frac{\log\left(ks/\log\left(1/\tilde{\delta}\right)\right)}{2}$, then

$$\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\}e^{\epsilon_0}\sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{ks}}\right). \tag{19}$$

Furthermore, when $\epsilon_0 = \mathcal{O}(1)$, we get $\tilde{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{\log(1/\tilde{\delta})}{ks}}\right)$.

Let $\mathcal{T} \subseteq \{1, \ldots, m\}$ denote the identities of the k clients chosen at iteration t, and for $i \in \mathcal{T}$, let $\mathcal{T}_i \subseteq \{1, \ldots, r\}$ denote the identities of the s data points chosen at client i at iteration t.[10] For any $\mathcal{T} \in \binom{[m]}{k}$ and $\mathcal{T}_i \in \binom{[r]}{s}, i \in \mathcal{T}$, define $\overline{\mathcal{T}} = (\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T})$, $\mathcal{D}^{\mathcal{T}_i} = \{d_{ij} : j \in \mathcal{T}_i\}$ for $i \in \mathcal{T}$, and $\mathcal{D}^{\overline{\mathcal{T}}} = \{\mathcal{D}^{\mathcal{T}_i} : i \in \mathcal{T}\}$. Note that $\mathcal{T}$ and $\mathcal{T}_i, i \in \mathcal{T}$ are random sets, where randomness is due to the sampling of clients and of data points, respectively. The mechanism $\mathcal{M}_t$ can be equivalently written as $\mathcal{M}_t = \mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}})$.

Observe that our sampling strategy is different from subsampling of choosing a uniformly random subset of ks data points from the entire dataset $\mathcal{D}$. Thus, we revisit the proof of privacy amplification by subsampling (see, for example, Ullman [2017]) – which is for uniform sampling – to compute the privacy parameters of the mechanism $\mathcal{M}_t$, where sampling is non-uniform. Define a dataset $\mathcal{D}' = (\mathcal{D}'_1) \bigcup (\cup_{i=2}^{m} \mathcal{D}_i) \in \mathfrak{S}^n$, where $\mathcal{D}'_1 = \{d'_{11}, d_{12}, \ldots, d_{1r}\}$ is different from the dataset $\mathcal{D}_1$ in the first data point d_{11}. Note that $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets – where, we assume, without loss of generality, that the differing elements are d_{11} and d'_{11}.

In order to show that $\mathcal{M}_t$ is $(\overline{\epsilon}, \overline{\delta})$-DP, we need show that for an arbitrary subset $\mathcal{S}$ of the range of $\mathcal{M}_t$, we have

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] \leq e^{\overline{\epsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + \overline{\delta} \tag{20}$$

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] \leq e^{\overline{\epsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] + \overline{\delta} \tag{21}$$

Note that both (20) and (21) are symmetric, so it suffices to prove only one of them. We prove (20) below.

Let $q = \frac{ks}{mr}$. We define conditional probabilities as follows:

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1\right] \\
A'_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1\right] \\
A_{10} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1\right] \\
A_{0} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right]
\end{aligned}$$

Let $q_1 = \frac{k}{m}$ and $q_2 = \frac{s}{r}$, and hence $q = q_1 q_2$. Thus, we have

$$\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= qA_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0 \\
\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] &= qA'_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0
\end{aligned}$$

Note that the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP. Therefore, we have

$$A_{11} \leq e^{\tilde{\epsilon}} A'_{11} + \tilde{\delta} \tag{22}$$

$$A_{11} \leq e^{\tilde{\epsilon}} A_{10} + \tilde{\delta} \tag{23}$$

Here (22) is straightforward, but proving (23) requires a combinatorial argument, which we give at the end of this proof.

We prove (20) separately for two cases, first when $s = 1$ and other when $s > 1$; k is arbitrary in both cases.

C.1 For $s = 1$ and arbitrary $k \in [m]$

Since the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, in addition to (22)-(23), since $s = 1$, we also have the following inequality:

$$A_{11} \leq e^{\tilde{\epsilon}} A_0 + \tilde{\delta} \tag{24}$$

Similar to (23), proving (24) requires a combinatorial argument, which we will give at the end of this proof. Note that (24) only holds for $s = 1$ and may not hold for arbitrary s.

[10] Though $\mathcal{T}$ and $\mathcal{T}_i, i \in \mathcal{T}$ may be different at different iteration t, for notational convenience, we suppress the dependence on t here.

Inequalities (22)-(24) together imply $A_{11} \leq e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}, A_0\} + \tilde{\delta}$. Now we prove (20) for $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)$ and $\overline{\delta} = q\tilde{\delta}$. Note that when $s = 1$, we have $q_1 = \frac{k}{m}$, $q_2 = \frac{1}{r}$, and $q = \frac{k}{mr}$.

$$\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= qA_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0 \\
&\leq q\left(e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}, A_0\} + \tilde{\delta}\right) + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0 \\
&= q\left((e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}, A_0\} + \min\{A'_{11}, A_{10}, A_0\}\right) + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0 + q\tilde{\delta} \\
&\overset{\text{(a)}}{\leq} q(e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}, A_0\} + qA'_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0 + q\tilde{\delta} \\
&\overset{\text{(b)}}{\leq} q(e^{\tilde{\epsilon}} - 1)\left(qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0\right)\right) + \left(qA'_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0\right) + q\tilde{\delta} \\
&= \left(1 + q\left(e^{\tilde{\epsilon}} - 1\right)\right)\left(qA'_{11} + q_1\left(1 - q_2\right)A_{10} + \left(1 - q_1\right)A_0\right) + q\tilde{\delta} \\
&= e^{\ln(1 + q(e^{\tilde{\epsilon}} - 1))} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + q\tilde{\delta}.
\end{aligned}$$

Here, (a) follows from $\min\{A'_{11}, A_{10}, A_0\} \leq A'_{11}$, and (b) follows from the fact that minimum is upper-bounded by the convex combination. By substituting the value of $\tilde{\epsilon}$ from (19) and using $ks = qn$, we get that for $\epsilon_0 = \mathcal{O}\left(1\right)$, we have $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{n}}\right)$.

C.2 For $s > 1$ and arbitrary $k \in [m]$

Note that (22)-(23) together imply $A_{11} \leq e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}\} + \tilde{\delta}$. Now we prove (20) for $\overline{\epsilon} = \ln(1 + q_2(e^{\tilde{\epsilon}} - 1))$ and $\overline{\delta} = q\tilde{\delta}$.

$$\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= qA_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 \\
&\leq q\left(e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}\} + \tilde{\delta}\right) + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 \\
&= q\left((e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}\} + \min\{A'_{11}, A_{10}\}\right) + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 + q\tilde{\delta} \\
&\overset{\text{(a)}}{\leq} q\left(e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}\}\right) + qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0 + q\tilde{\delta} \\
&\overset{\text{(b)}}{\leq} q\left((e^{\tilde{\epsilon}} - 1)(q_2A'_{11} + (1 - q_2)A_{10})\right) + (qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0) + q\tilde{\delta} \\
&= q_2\left((e^{\tilde{\epsilon}} - 1)(q_1q_2A'_{11} + q_1(1 - q_2)A_{10})\right) + (qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0) + q\tilde{\delta} \\
&\overset{\text{(c)}}{\leq} q_2\left((e^{\tilde{\epsilon}} - 1)(qA'_{11} + q_1(1 - q_2)A_{10}) + (1 - q_1)A_0\right) + (qA'_{11} + q_1(1 - q_2)A_{10} + (1 - q_1)A_0) + q\tilde{\delta} \\
&= \left(1 + q_2\left((e^{\tilde{\epsilon}} - 1)\right)(qA'_{11} + q_1(1 - q_2)A_{10}) + (1 - q_1)A_0\right) + q\tilde{\delta} \\
&= e^{\ln(1 + q_2(e^{\tilde{\epsilon}} - 1))} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + q\tilde{\delta}
\end{aligned}$$

Here, (a) follows from $\min\{A'_{11}, A_{10}\} \leq A'_{11}$, (b) follows from the fact that minimum is upper-bounded by the convex combination, and (c) holds because $(1 - q_1)A_0 \geq 0$. By substituting the value of $\tilde{\epsilon}$ from (19) and using $ks = qn$, we get that for $\epsilon_0 = \mathcal{O}\left(1\right)$, we have $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{q_2\log\left(1/\tilde{\delta}\right)}{q_1n}}\right)$. Note that when $q_1 = 1$ (i.e., we select all the clients in each iteration), then this gives the desired privacy amplification of $q = q_2$.

The proof of Lemma 3 is complete, except for that we have to prove (23) and (24). Before proving (23) and (24), we state an important remark about the privacy amplification in both the cases.

Remark 6. Note that when $s = 1$ and $\epsilon_0 = \mathcal{O}(1)$, we have $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)) = \mathcal{O}(q\tilde{\epsilon})$. So we get a privacy amplification by a factor of $q = \frac{ks}{mr}$ – the sampling probability of each data point from the entire dataset. Here, we get a privacy amplification from both types of sampling, of clients as well of data points.

On the other hand, when $s > 1$ and $\epsilon_0 = \mathcal{O}(1)$, we have $\overline{\epsilon} = \ln(1 + q_2(e^{\tilde{\epsilon}} - 1)) = \mathcal{O}(q_2\tilde{\epsilon})$, which, unlike the case of $s = 1$, only gives the privacy amplification by a factor of $q_2 = \frac{s}{r}$ – the sampling probability of each data point from a client. So, unlike the case of $s = 1$, here we only get a privacy amplification from sampling of data points, not from sampling of clients. Note that when $k = m$ and any $s \in [r]$ (which implies $q_1 = 1$ and $q = q_2$), we have $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{q_2\log\left(1/\tilde{\delta}\right)}{n}}\right)$, which gives the desired amplification when we select all the clients in each iteration.

Proof of (23)**.** First note that the number of subsets $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \in \mathcal{T}_1$ is equal to $\binom{r-1}{s-1}$ and the number of subsets $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \notin \mathcal{T}_1$ is equal to $\binom{r-1}{s}$. It is easy to verify that $(r-s)\binom{r-1}{s-1} = s\binom{r-1}{s}$.

Consider the following bipartite graph $G = (V_1 \cup V_2, E)$, where the left vertex set V_1 has $\binom{r-1}{s-1}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \in \mathcal{T}_1$, the right vertex set V_2 has $\binom{r-1}{s}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \notin \mathcal{T}_1$, and the edge set E contains all the edges between neighboring vertices, i.e., if $(\boldsymbol{u}, \boldsymbol{v}) \in V_1 \times V_2$ is such that $\boldsymbol{u}$ and $\boldsymbol{v}$ differ in only one element, then $(\boldsymbol{u}, \boldsymbol{v}) \in E$. Observe that each vertex of V_1 has $(r-s)$ neighbors in V_2 – the neighbors of $\mathcal{T}_1 \in V_1$ will be $\{(\mathcal{T}_1 \setminus \{1\}) \cup \{i\} : i \in [m] \setminus \mathcal{T}_1\} \in V_2$. Similarly, each vertex of V_2 has s neighbors in V_1 – the neighbors of $\mathcal{T}_1 \in V_2$ will be $\{(\mathcal{T}_1 \setminus \{i\}) \cup \{1\} : i \in \mathcal{T}_1\} \in V_1$.

Now, fix any $\mathcal{T} \in \binom{[m]}{k}$ s.t. $1 \in \mathcal{T}$, and for $i \in \mathcal{T} \setminus \{1\}$, fix any $\mathcal{T}_i \in \binom{[r]}{s}$, and consider an arbitrary $(\boldsymbol{u}, \boldsymbol{v}) \in E$. Since the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, we have

$$\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T}, \mathcal{T}_1 = \boldsymbol{u}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\}\right] \leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T}, \mathcal{T}_1 = \boldsymbol{v}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\}\right] + \tilde{\delta}. \tag{25}$$

Now we are ready to prove (23).

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1\right] \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1] \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] \\
&\overset{\text{(a)}}{=} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\} | 1 \in \mathcal{T}] \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} \Pr[\mathcal{T}_1 | 1 \in \mathcal{T}_1] \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\} | 1 \in \mathcal{T}] \frac{1}{(r-s)\binom{r-1}{s-1}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} (r-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\} | 1 \in \mathcal{T}] \frac{1}{s\binom{r-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} (r-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] \\
&\overset{\text{(b)}}{\leq} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\} | 1 \in \mathcal{T}] \frac{1}{s\binom{r-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1} s \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\} | 1 \in \mathcal{T}] \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1} \Pr[\mathcal{T}_1 | 1 \notin \mathcal{T}_1] \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&\overset{\text{(c)}}{=} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in \mathcal{T} \setminus \{1\}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1] \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&\leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1\right] + \tilde{\delta} \\
&= e^{\tilde{\epsilon}} A_{10} + \tilde{\delta}.
\end{aligned}$$

Here, (a) and (c) follow from the fact that clients sample the data points independent of each other, and (b) follows from (25) together with the fact that there are $(r-s)\binom{r-1}{s-1} = s\binom{r-1}{s}$ edges in the bipartite graph $G = (V_1 \cup V_2, E)$, where degree of vertices in V_1 is $(r-s)$ and degree of vertices in V_2 is s.

Proof of (24)**.** First note that the number of subsets $\mathcal{T} \in [m]$ such that $|\mathcal{T}| = k, 1 \in \mathcal{T}$ is equal to $\binom{m-1}{k-1}$ and the number of subsets $\mathcal{T} \subset [m]$ such that $|\mathcal{T}| = k, 1 \notin \mathcal{T}$ is equal to $\binom{m-1}{k}$. It is easy to verify that $(m-k)\binom{m-1}{k-1} = k\binom{m-1}{k}$.

Consider the following bipartite graph $G = (V_1 \cup V_2, E)$, where the left vertex set V_1 has $\binom{m-1}{k-1}r^{k-1}$ vertices, one for each configuration of $(\mathcal{T}, \mathcal{T}_i : i \in \mathcal{T})$ such that $\mathcal{T} \subset [m]$, $|\mathcal{T}| = k, 1 \in \mathcal{T}$ and $\mathcal{T}_1 = 1$, the right vertex set V_2 has $\binom{m-1}{k}r^k$ vertices, one for each configuration of $(\mathcal{T}, \mathcal{T}_i : i \in \mathcal{T})$ such that $\mathcal{T} \subset [m]$, $|\mathcal{T}| = k, 1 \notin \mathcal{T}$, and the edge set E contains all the edges between neighboring vertices, i.e., if $(\boldsymbol{u}, \boldsymbol{v}) \in V_1 \times V_2$ is such that $\boldsymbol{u}$ and $\boldsymbol{v}$ differ in only one element, then $(\boldsymbol{u}, \boldsymbol{v}) \in E$. Observe that each vertex of V_1 has $r(m-k)$ neighbors in V_2. Similarly, each vertex of V_2 has k neighbors in V_1.

Consider an arbitrary edge $(\boldsymbol{u}, \boldsymbol{v}) \in E$. By construction, there exists $\mathcal{T} \in \binom{[m]}{k}$ with $1 \in \mathcal{T}$ and $\mathcal{T}_i \in [r], i \in \mathcal{T}$ such that $\boldsymbol{u} = (\mathcal{T}, \mathcal{T}_i : i \in \mathcal{T})$ and $\mathcal{T}' \in \binom{[m]}{k}$ with $1 \notin \mathcal{T}'$ and $\mathcal{T}'_i \in [r], i \in \mathcal{T}'$ such that $\boldsymbol{v} = (\mathcal{T}', \mathcal{T}'_i : i \in \mathcal{T}')$. Note that, since $(\boldsymbol{u}, \boldsymbol{v}) \in E$, $(\mathcal{T}_i : i \in \mathcal{T})$ and $(\mathcal{T}'_i : i \in \mathcal{T}')$ have $k-1$ elements common. Now, since the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, we have

$$\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}\right] \leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}'}}) \in \mathcal{S} | \mathcal{T}', \mathcal{T}'_i, i \in \mathcal{T}'\right] + \tilde{\delta}. \tag{26}$$

Now we are ready to prove (24).

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } \mathcal{T}_1 = 1\right] \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} | 1 \in \mathcal{T} \text{ and } \mathcal{T}_1 = 1] \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] \\
&= \frac{1}{\binom{m-1}{k-1} r^{k-1}} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] \\
&= \frac{1}{(m-k)\binom{m-1}{k-1} r^{k}} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} r(m-k) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] \\
&\overset{\text{(a)}}{=} \frac{1}{k\binom{m-1}{k} r^{k}} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} r(m-k) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] \\
&\overset{\text{(b)}}{\leq} \frac{1}{k\binom{m-1}{k} r^{k}} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T}}} k \left(e^{\epsilon} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] + \tilde{\delta}\right) \\
&= \frac{1}{\binom{m-1}{k} r^{k}} \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T}}} \left(e^{\epsilon} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] + \tilde{\delta}\right) \\
&= \sum_{\substack{\mathcal{T} \in \binom{[m]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [r] \text{ for } i \in \mathcal{T}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} | 1 \notin \mathcal{T}] \left(e^{\epsilon} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}] + \tilde{\delta}\right) \\
&= e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right] + \tilde{\delta} \\
&= e^{\tilde{\epsilon}} A_0 + \tilde{\delta}
\end{aligned}$$

Here, (a) uses $(m-k)\binom{m-1}{k-1} = k\binom{m-1}{k}$, and (b) follows from (26) together with the fact that there are $r(m-k)\binom{m-1}{k-1}r^{k-1} = k\binom{m-1}{k}r^k$ edges in the bipartite graph $G = (V_1 \cup V_2, E)$, where degree of vertices in V_1 is $r(m-k)$ and degree of vertices in V_2 is k.

This completes the proof of Lemma 3.

D Compressed and Private Mean Estimation

In this section, we provide additional results on compressed and private mean estimation and also prove the results stated in Section 4.2.

D.1 Main Results

Theorem (Restating Theorem 2)**.** *For any $d, n \geq 1$, $a, \epsilon_0 > 0$, and $p \in [1,\infty]$, the minimax risk in* (4) *satisfies*

$$r_{\epsilon,\infty,n}^{p,d}(a) \geq \begin{cases} \Omega\left(a^2 \min\left\{1, \frac{d}{n\epsilon_0^2}\right\}\right) & \text{if } 1 \leq p \leq 2, \\ \Omega\left(a^2 d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right) & \text{if } p \geq 2. \end{cases}$$

Theorem 5. *For any $d, n \geq 1$, $a, \epsilon_0 > 0$, and $p \in [1,\infty]$, we have the minimax risk in* (17) *satisfies*

$$R_{\epsilon,\infty,n}^{p,d}(a) \geq \begin{cases} \Omega\left(a^2 \min\left\{1, \frac{d}{n\epsilon_0^2}\right\}\right) & \text{if } 1 \leq p \leq 2, \\ \Omega\left(a^2 d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon_0,\epsilon_0^2\}}\right\}\right) & \text{if } p \geq 2. \end{cases}$$

Theorem (Restating Theorem 3)**.** *For any private-randomness, symmetric mechanism $\mathcal{R}$ with communication budget $b < \log(d)$ bits per client, and any decoding function $g : \{0,1\}^b \to \mathbb{R}^d$, when $\widehat{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} g\left(\mathcal{R}\left(\boldsymbol{x}_i\right)\right)$, we have*[11]

$$r_{\epsilon,b,n}^{p,d}(a) > a^2 \max\left\{1, d^{1-\frac{2}{p}}\right\}.$$

For convenience, we will write Theorem 4 in three separate theorems.

Theorem 6 (ℓ_1-norm)**.** *For any $d, n \geq 1$, $a, \epsilon_0 > 0$, we have*

$$r_{\epsilon_0,b,n}^{1,d}\left(a\right) \leq \frac{a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2 \quad \text{and} \quad R_{\epsilon_0,b,n}^{1,d}\left(a\right) \leq \frac{4a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2,$$

for $b = \log(d) + 1$.

Theorem 7 (ℓ_2-norm)**.** *For any $d, n \geq 1$, $a, \epsilon_0 > 0$, we have*

$$r_{\epsilon_0,b,n}^{2,d}\left(a\right) \leq \frac{6a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2 \quad \text{and} \quad R_{\epsilon_0,b,n}^{2,d}\left(a\right) \leq \frac{14a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2,$$

for $b = d\log(e) + 1$.

Theorem 8 (ℓ_∞-norm)**.** *For any $d, n \geq 1$, $a, \epsilon_0 > 0$, we have*

$$r_{\epsilon_0,b,n}^{\infty,d}(a) \leq \frac{a^2 d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2 \quad \text{and} \quad R_{\epsilon_0,b,n}^{\infty,d}(a) \leq \frac{4a^2 d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2,$$

for $b = \log(d) + 1$.

Note that when $\epsilon_0 = \mathcal{O}(1)$, then the upper and lower bounds on minimax risks match for $p \in [1,2]$. Furthermore, when $\epsilon_0 \leq 1$, then they match for all $p \in [1,\infty]$.

Now we give a general achievability result for any ℓ_p-norm ball $\mathcal{B}_p^d(a)$ for any $p \in [1,\infty)$. For this, we use standard inequalities between different norms, and probabilistically use the mechanisms for ℓ_1-norm or ℓ_2-norm with expanded radius of the corresponding ball. We assume that every work can pick any mechanisms with the same probability $\bar{p} \in [0,1]$. This gives the following result, which we prove in Section D.8.

Corollary 1 (General ℓ_p-norm, $p \in [1,\infty)$)**.** Suppose clients pick the mechanism for ℓ_1-norm with probability $\bar{p} \in [0,1]$. Then, for any $d, n \geq 1$, $a, \epsilon_0 > 0$, we have:

$$r_{\epsilon_0,b,n}^{p,d}(a) \;\leq\; \bar{p}\, d^{2-\frac{2}{p}} \cdot r_{\epsilon_0,b,n}^{1,d}(a) + (1-\bar{p}) \max\left\{d^{1-\frac{2}{p}}, 1\right\} \cdot r_{\epsilon_0,b,n}^{2,d}(a), \tag{27}$$

[11] Note that Theorem 3 works only when the estimator $\widehat{\boldsymbol{x}}$ applies the decoding function g on individual responses and then takes the average. We leave its extension for arbitrary decoders as a future work.

$$R^{p,d}_{\epsilon_0,b,n}(a) \;\leq\; \bar{p}\, d^{2-\frac{2}{p}} \cdot R^{1,d}_{\epsilon_0,b,n}(a) + (1-\bar{p}) \max\left\{d^{1-\frac{2}{p}}, 1\right\} \cdot R^{2,d}_{\epsilon_0,b,n}(a). \tag{28}$$

for $b = \bar{p}\log(d) + (1-\bar{p})d\log(e) + 1$. Note that this communication is in expectation, which is taken over the sampling of selecting ℓ_1 or ℓ_2 mechanisms.

We can recover Theorem 6 by setting $p = 1$ and $\bar{p} = 1$ and Theorem 7 by setting $p = 2$ and $\bar{p} = 0$.

In this section, we study the private mean-estimation problem in the minimax framework given in Section B.4. Note that in this section we focus on giving (ϵ_0, b)-CLDP) privacy-communication guarantees for the mean-estimation problem and give the performance of schemes in terms of the associated minimax risk. This framework is applied at each round of the optimization problem, and is then converted to the eventual central DP privacy guarantees using the shuffling framework in Section 5.3, yielding the main result Theorem 1 stated in Section 4.

We prove the lower bound results (Theorems 5, 2) in the first two subsections and the achievable results (Theorems 6, 7, 8, and Corollary 1) in the last four subsections, respectively.

We prove lower bounds for private mechanisms with no communication constraints, and for clarity, we denote such mechanisms by (ϵ, ∞)-CLDP mechanisms. Our achievable schemes use finite amount of randomness.

For lower bounds, for simplicity, we assume that the inputs come from an ℓ_p-norm ball of unit radius – the bounds will be scaled by the factor of a^2 if inputs come from an ℓ_p-norm ball of radius a. For convenience, we denote $\mathcal{B}^d_p(1), \mathcal{P}^d_p(1), r^{p,d}_{\epsilon,b,n}(1)$, and $R^{p,d}_{\epsilon,b,n}(1)$ by $\mathcal{B}^d_p, \mathcal{P}^d_p, r^{p,d}_{\epsilon,b,n}$, and $R^{p,d}_{\epsilon,b,n}$, respectively.

D.2 Lower Bound on $R^{p,d}_{\epsilon,\infty,n}$: Proof of Theorem 5

Theorem 5 states separate lower bounds on $R^{p,d}_{\epsilon,\infty,n}$ depending on whether $p \geq 2$ or $p \leq 2$ (at $p = 2$, both bounds coincide), and we prove them below in Section D.2.1 and Section D.2.2, respectively.

D.2.1 Lower bound for $p \in [2, \infty]$

The main idea of the lower bound is to transform the problem to the private mean estimation when the inputs are sampled from Bernoulli distributions. Recall that $\mathcal{P}^d_p$ denote the set of all distributions on the p-norm ball $\mathcal{B}^d_p$. Let $\mathcal{P}^{\text{Bern}}_{p,d}$ denote the set of Bernoulli distributions on $\left\{0, \frac{1}{d^{1/p}}\right\}^d$, i.e., any element of $\mathcal{P}^{\text{Bern}}_{p,d}$ is a product of d independent Bernoulli distributions, one for each coordinate. We first prove a lower bound on $R^{p,d}_{\epsilon,\infty,n}$ when the input distribution belongs to $\mathcal{P}^{\text{Bern}}_{p,d}$.

Lemma 9. *For any $p \in [2, \infty]$, we have*

$$\inf_{\{\mathcal{M}_i\}\in\mathcal{Q}_{(\epsilon,\infty)}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q}\in\mathcal{P}^{Bern}_{p,d}} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right). \tag{29}$$

Proof. The proof is straightforward from the proof of [Duchi and Rogers, 2019, Corollary 3]. In their setting, $\mathcal{P}^{\text{Bern}}_{p,d}$ is supported on $\{0,1\}^d$, and they proved a lower bound of $\Omega\left(\min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right)$. In our setting, since $\mathcal{P}^{\text{Bern}}_{p,d}$ is supported on $\left\{0, \frac{1}{d^{1/p}}\right\}^d$, we can simply scale the elements in the support of $\mathcal{P}^{\text{Bern}}_{p,d}$ by a factor of $1/d^{1/p}$, which will also scale the mean $\boldsymbol{\mu}_{\boldsymbol{q}}$ by the same factor. Note that the best estimator $\widehat{\boldsymbol{x}}$ will be equal to the scaled version of the best estimator from [Duchi and Rogers, 2019, Corollary 3] with the same value $1/d^{1/p}$. This proves Lemma 9. ■

In order to use Lemma 9, first observe that for every $\boldsymbol{x} \in \mathcal{P}^{\text{Bern}}_{p,d}$, we have $\|\boldsymbol{x}\|_p \leq 1$, which implies that $\boldsymbol{x} \in \mathcal{P}^d_p$. Thus we have $\mathcal{P}^{\text{Bern}}_{p,d} \subset \mathcal{P}^d_p$. Now our bound on $R^{p,d}_{\epsilon,\infty,n}$ trivially follows from the following inequalities:

$$\begin{aligned} R^{p,d}_{\epsilon,\infty,n} = \inf_{\{\mathcal{M}_i\}\in\mathcal{Q}_{(\epsilon,\infty)}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q}\in\mathcal{P}^d_p} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 &\geq \inf_{\{\mathcal{M}_i\}\in\mathcal{Q}_{(\epsilon,\infty)}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q}\in\mathcal{P}^{\text{Bern}}_{p,d}} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \\ &\geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right), \end{aligned} \tag{30}$$

where the last inequality follows from (29).

D.2.2 Lower bound for $p \in [1,2]$

Fix an arbitrary $p \in [1,2]$. Note that $\|\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_1$, which implies that $\mathcal{B}_1^d \subset \mathcal{B}_p^d$, and therefore, we have $\mathcal{P}_1^d \subset \mathcal{P}_p^d$. These imply that the lower bound derived for $\mathcal{P}_1^d$ also holds for $\mathcal{P}_p^d$, i.e., $R_{\epsilon,\infty,n}^{p,d} \geq R_{\epsilon,\infty,n}^{1,d}$ holds for any $p \in [1,2]$. So, in the following, we only lower-bound $R_{\epsilon,\infty,n}^{1,d}$.

The main idea of the lower bound is to transform the problem to the private discrete distribution estimation when the inputs are sampled from a discrete distribution taken from a simplex in d dimensions. Recall that $\mathcal{P}_1^d$ denotes all probability density functions q over the 1-norm ball $\mathcal{B}_1^d$. Note that q may be a continuous distribution supported over all of $\mathcal{B}_1^d$. Let $\widehat{\mathcal{P}}_1^d$ denote a set of all discrete distributions $\boldsymbol{q}$ supported over the d standard basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$, *i.e.,* the distribution has support on $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\}$. Since $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} \subset \mathcal{B}_1^d$, we have $\widehat{\mathcal{P}}_1^d \subset \mathcal{P}_1^d$. Moreover, since any $q \in \widehat{\mathcal{P}}_1^d$ is a discrete distribution, by abusing notation, we describe q through a $d-$dimensional vector $\boldsymbol{q}$ of its probability mass function. Note that, for any $\boldsymbol{q} \in \widehat{\mathcal{P}}_1^d$, the average over this distribution is $\boldsymbol{\mu}_{\boldsymbol{q}} = \mathbb{E}_{\boldsymbol{q}}[\mathbf{U}]$, where $\mathbb{E}_{\boldsymbol{q}}[\cdot]$ denotes the expectation over the distribution $\boldsymbol{q}$ for a discrete random variable $\mathbf{U} \sim \boldsymbol{q}$, where we denote $q_i = \Pr[\mathbf{U} = \boldsymbol{e}_i]$. Therefore we have $\boldsymbol{\mu}_{\boldsymbol{q}} = \sum_{i=1}^d q_i \boldsymbol{e}_i = (q_1, \ldots, q_d)^T = \boldsymbol{q}$, for every $\boldsymbol{q} \in \widehat{\mathcal{P}}_1^d$. Let Δ_d denote the probability simplex in d dimensions. Since the discrete distribution $q \in \widehat{\mathcal{P}}_1^d$ is representable as $\boldsymbol{q} \in \Delta_d$, we have an isomorphism between Δ_d and $\widehat{\mathcal{P}}_1^d$, *i.e.,* we can equivalently think of $\widehat{\mathcal{P}}_1^d = \Delta_d$. Fix arbitrary (ϵ, ∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and an estimator $\widehat{\boldsymbol{x}}$. Using the above notations and observations, we have:

$$\sup_{\boldsymbol{q} \in \mathcal{P}_1^d} \mathbb{E} \left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \sup_{\boldsymbol{q} \in \widehat{\mathcal{P}}_1^d} \mathbb{E} \left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 = \sup_{\boldsymbol{q} \in \widehat{\mathcal{P}}_1^d} \mathbb{E} \left\|\boldsymbol{q} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2. \tag{31}$$

Using $\widehat{\mathcal{P}}_1^d = \Delta_d$, and taking the infimum in (31) over all (ϵ, ∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and estimators $\widehat{\boldsymbol{x}}$, we get

$$\inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,\infty)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q} \in \mathcal{P}_1^d} \mathbb{E} \left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,\infty)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q} \in \Delta_d} \mathbb{E} \left\|\boldsymbol{q} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2. \tag{32}$$

Girgis et al. [Girgis et al., 2020, Theorem 1] lower-bounded the RHS of (32) in the context of characterizing a privacy-utility-randomness tradeoff in LDP. When specializing to our setting, where we are not concerned about the amount of randomness used, their lower bound result gives $\inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,\infty)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q} \in \Delta_d} \mathbb{E} \left\|\boldsymbol{q} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right)$. Substituting this in (32) gives

$$R_{\epsilon,\infty,n}^{1,d} = \inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,\infty)}\}} \inf_{\widehat{\boldsymbol{x}}} \sup_{\boldsymbol{q} \in \mathcal{P}_1^d} \mathbb{E} \left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right). \tag{33}$$

D.3 Lower Bound on $r_{\epsilon,\infty,n}^{p,d}$: Proof of Theorem 2

Similar to Section D.2, we prove the lower bound on $r_{\epsilon,\infty,n}^{p,d}$ separately depending on whether $p \geq 2$ or $p \leq 2$ (at $p = 2$, both bounds coincide) below in Section D.3.1 and Section D.3.2, respectively. In both the proofs, the main idea is to transform the worst-case lower bound to the average case lower bound and then use relation between different norms.

D.3.1 Lower bound for $p \in [2, \infty]$

Fix arbitrary (ϵ, ∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and an estimator $\widehat{\boldsymbol{x}}$. It follows from (30) that there exists a distribution $\boldsymbol{q} \in \mathcal{P}_p^d$, such that if we sample $\boldsymbol{x}_i^{(q)} \sim \boldsymbol{q}$, i.i.d. for all $i \in [n]$ and letting $\boldsymbol{y}_i = \mathcal{M}_i(\boldsymbol{x}_i^{(q)})$, we would have $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n \min\{\epsilon, \epsilon^2\}}\right\}\right)$. We have

$$\sup_{\{\boldsymbol{x}_i\} \in \mathcal{B}_p^d} \mathbb{E} \left\|\frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \overset{\text{(a)}}{\geq} \mathbb{E} \left\|\frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2$$

$$\overset{\text{(b)}}{\geq} \frac{1}{2}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 - \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \tag{34}$$

$$\overset{\text{(c)}}{\geq} \Omega\left(d^{1-\frac{2}{p}}\min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right) - \frac{d^{1-\frac{2}{p}}}{n}$$

$$\overset{\text{(d)}}{\geq} \Omega\left(d^{1-\frac{2}{p}}\min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right) \tag{35}$$

In the LHS of (a), the expectation is taken over the randomness of the mechanisms $\{\mathcal{M}_i\}$ and the estimator $\widehat{\boldsymbol{x}}$; whereas, in the RHS of (a), in addition, the expectation is also taken over sampling $\boldsymbol{x}_i$'s from the distribution $\boldsymbol{q}$. Moreover (a) holds since the LHS is supremum $\{\boldsymbol{x}_i\} \in \mathcal{B}_p^d$ and the RHS of (a) takes expectation w.r.t. a distribution over $\mathcal{B}_p^d$ and hence lower-bounds the LHS. The inequality (b) follows from the Jensen's inequality $2\|\mathbf{u}\|_2^2 + 2\|\mathbf{v}\|_2^2 \geq \|\boldsymbol{u}+\mathbf{v}\|_2^2$ by setting $\boldsymbol{u} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)$ and $\mathbf{v} = \boldsymbol{\mu}_{\boldsymbol{q}} - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)}$. In (c) we used $\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \leq \frac{d^{1-\frac{2}{p}}}{n}$, which we show below. In (d), we assume $\min\{\epsilon,\epsilon^2\} \leq \mathcal{O}(d)$.

Note that for any vector $\boldsymbol{u} \in \mathbb{R}^d$, we have $\|\boldsymbol{u}\|_2 \leq d^{\frac{1}{2}-\frac{1}{p}}\|\boldsymbol{u}\|_p$, for any $p \geq 2$. Since each $\boldsymbol{x}_i^{(q)} \in \mathcal{B}_p^d$, which implies $\|\boldsymbol{x}_i^{(q)}\|_p \leq 1$, we have that $\|\boldsymbol{x}_i^{(q)}\|_2 \leq d^{\frac{1}{2}-\frac{1}{p}}$. Hence, $\mathbb{E}\|\boldsymbol{x}_i^{(q)}\|_2^2 \leq d^{1-\frac{2}{p}}$ holds for all $i \in [n]$. Now, since $\boldsymbol{x}_i$'s are i.i.d. with $\mathbb{E}[\boldsymbol{x}_i^{(q)}] = \boldsymbol{\mu}_{\boldsymbol{q}}$, we have

$$\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \overset{\text{(a)}}{\leq} \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_i^{(q)}\right\|_2^2 \leq \frac{1}{n^2}\sum_{i=1}^{n}d^{1-\frac{2}{p}} = \frac{d^{1-\frac{2}{p}}}{n}, \tag{36}$$

where (a) uses $\mathbb{E}\|\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]\|_2^2 \leq \mathbb{E}\|\boldsymbol{x}\|_2^2$, which holds for any random vector $\boldsymbol{x}$.

Taking supremum in (35) over all (ϵ,∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and estimators $\widehat{\boldsymbol{x}}$, we get

$$r_{\epsilon,\infty,n}^{p,d} = \inf_{\{\mathcal{M}_i \in \mathcal{Q}_{(\epsilon,\infty)}\}}\inf_{\widehat{\boldsymbol{x}}}\sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(d^{1-\frac{2}{p}}\min\left\{1, \frac{d}{n\min\{\epsilon,\epsilon^2\}}\right\}\right). \tag{37}$$

D.3.2 Lower bound for $p \in [1,2]$

Similar to the argument given in Section D.2.2, since $r_{\epsilon,\infty,n}^{p,d} \geq r_{\epsilon,\infty,n}^{1,d}$ holds for any $p \in [1,2]$, it suffices to lower-bound $r_{\epsilon,\infty,n}^{1,d}$.

Fix arbitrary (ϵ,∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and an estimator $\widehat{\boldsymbol{x}}$. It follows from (33) that there exists a distribution $\boldsymbol{q} \in \mathcal{P}_p^d$, such that if we sample $\boldsymbol{x}_i^{(q)} \sim \boldsymbol{q}$, i.i.d. for all $i \in [n]$ and letting $\boldsymbol{y}_i = \mathcal{M}_i(\boldsymbol{x}_i^{(q)})$, we would have $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right)$. Now, by the same reasoning using which we obtained (34), we have

$$\sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_p^d}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 \geq \frac{1}{2}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\left(\boldsymbol{y}^n\right)\right\|_2^2 - \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2$$

$$\overset{\text{(a)}}{\geq} \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right) - \frac{1}{n} \overset{\text{(b)}}{\geq} \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right) \tag{38}$$

In (a) we used

$$\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \leq \frac{1}{n}, \tag{39}$$

which can be obtained by first noting that for any $\boldsymbol{u} \in \mathbb{R}^d$, we have $\|\boldsymbol{u}\|_2 \leq \|\boldsymbol{u}\|_p$ for $p \in [1,2]$, and then using this in the set of inequalities which give (36). In (b), we assume $\epsilon \leq \mathcal{O}(\sqrt{d})$.

Taking supremum in (35) over all (ϵ,∞)-CLDP mechanisms $\{\mathcal{M}_i : i \in [n]\}$ and estimators $\widehat{\boldsymbol{x}}$, we get $r_{\epsilon,\infty,n}^{1,d} \geq \Omega\left(\min\left\{1, \frac{d}{n\epsilon^2}\right\}\right)$.

D.4 Lower Bound on $r_{\epsilon,b,n}^{p,d}$: Proof of Theorem 3

Let $M = 2^b < d$ be the total number of possible outputs of the mechanism $\mathcal{R}$. Let $\{o_1, o_2, \ldots, o_M\}$ be the set of M possible outputs of $\mathcal{R}$. For every $i \in [M]$, let $q_i = g(o_i)$. We can write the M possible outputs of $\mathcal{R}$ as columns of a $d \times M$ matrix $Q = [q_1, \ldots, q_M]$. Since $M < d$, the rank of the matrix Q is at most M. Let $\boldsymbol{x} \in \mathbb{R}^d$ be a vector in the null space of the matrix Q, i.e., $\boldsymbol{x}^T q_j = 0$ for all $j \in [M]$. Then, we set the sample of each client by $\boldsymbol{x}_i = \overline{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_p}$ for all $i \in [n]$, and hence, $\boldsymbol{x}_i \in \mathcal{B}_p^d$. Observe that the estimator $\hat{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n g\left(\mathcal{R}\left(\boldsymbol{x}_i\right)\right)$ is in the column space of the matrix Q. Thus, we get

$$r_{\epsilon,b,n}^{p,d} \geq \mathbb{E}\left\|\overline{\boldsymbol{x}} - \frac{1}{n}\sum_{i=1}^n g\left(\mathcal{R}\left(\boldsymbol{x}_i\right)\right)\right\|_2^2 \overset{(a)}{=} \|\overline{\boldsymbol{x}}\|_2^2 + \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n g\left(\mathcal{R}\left(\boldsymbol{x}_i\right)\right)\right\|_2^2 \geq \max\left\{1, d^{1-\frac{2}{p}}\right\}$$

where step (a) follows from the fact that $\overline{\boldsymbol{x}}$ is in the null space of Q, while the estimator $\hat{\boldsymbol{x}}$ is in the column space of Q. This completes the proof of Theorem 3.

D.5 Achievability for ℓ_1-norm Ball: Proof of Theorem 6

In this section, we propose an ϵ_0-LDP mechanism that requires $\mathcal{O}\left(\log(d)\right)$-bits of communication per client using private randomness and 1-bit of communication per client using public randomness. In other words we can guarantee $(\epsilon_0, \mathcal{O}\left(\log(d)\right))$-CLDP with private randomness and $(\epsilon_0, 1)$-CLDP using public randomness. The proposed mechanism is based on the Hadamard matrix and is inspired from the Hadamard mechanism proposed by Acharya et al. [2019]. We assume that d is a power of 2. Let $\mathbf{H}_d$ denote the Hadamard matrix of order d, which can be constructed by the following recursive mechanism:

$$\mathbf{H}_d = \begin{bmatrix} \mathbf{H}_{d/2} & \mathbf{H}_{d/2} \\ \mathbf{H}_{d/2} & -\mathbf{H}_{d/2} \end{bmatrix} \qquad \mathbf{H}_1 = \begin{bmatrix} 1 \end{bmatrix}$$

Client i has an input $\boldsymbol{x}_i \in \mathcal{B}_1^d\left(a\right)$. It computes $\boldsymbol{y}_i = \frac{1}{\sqrt{d}}\mathbf{H}_d\boldsymbol{x}_i$. Note that each coordinate of $\boldsymbol{y}_i$ lies in the interval $[-a/\sqrt{d}, a/\sqrt{d}]$. Client i selects $j \sim \mathsf{Unif}\left[d\right]$ and quantize $y_{i,j}$ privately according to (40) and obtains $\boldsymbol{z}_i \in \left\{\pm a\mathbf{H}_d(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right\}$, which can be represented using only 1-bit. Here, $\mathbf{H}_d(j)$ denotes the j-th column of the Hadamard matrix $\mathbf{H}_d$. Server receives the n messages $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ from the clients and outputs their average $\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i$. We present this mechanism in Algorithm 3 – we only present the client-side part of the algorithm, as server only averages the messages received from the clients.

Algorithm 3 ℓ_1-MEAN-EST ($\mathcal{R}_1$: the client-side algorithm)

```
1: Input: Vector $\boldsymbol{x} \in \mathcal{B}_1^d\left(a\right)$, and local privacy level $\epsilon_0 > 0$.
2: Construct $\boldsymbol{y} = \frac{1}{\sqrt{d}}\mathbf{H}_d\boldsymbol{x}$
3: Sample $j \sim \mathsf{Unif}[d]$ and quantize $y_j$ as follows:
       $$\boldsymbol{z} = \begin{cases} +a\mathbf{H}_d\left(j\right)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} + \frac{\sqrt{d}y_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \\ -a\mathbf{H}_d\left(j\right)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} - \frac{\sqrt{d}y_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1} \end{cases} \qquad (40)$$
4: Return $\boldsymbol{z}$.
```

Lemma 10. *The mechanism $\mathcal{R}_1$ presented in Algorithm 3 satisfies the following properties, where $\epsilon_0 > 0$:*

1. *$\mathcal{R}_1$ is $(\epsilon_0, \log\left(d\right)+1)$-CLDP and requires only 1-bit of communication using public randomness.*

2. *$\mathcal{R}_1$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x} \in \mathcal{B}_1^d\left(a\right)$, we have*

$$\mathbb{E}\left[\mathcal{R}_1\left(\boldsymbol{x}\right)\right] = \boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\|\mathcal{R}_1\left(\boldsymbol{x}\right) - \boldsymbol{x}\|_2^2 \leq a^2 d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

Proof. We show these properties one-by-one below.

1. Observe that the output of the mechanism $\mathcal{R}_1$ can be represented using the index $j \in [d]$ and one bit of the sign of $\{\pm a\mathbf{H}_d\left(j\right)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\}$. Hence, it requires only $\log\left(d\right)+1$ bits for communication. Furthermore, the randomness $j \sim \mathsf{Unif}\left[d\right]$ is independent of the input $\boldsymbol{x}$. Thus, if the client has access to a public randomness j, then the client needs only to send one bit to represent its sign. Now, we show that the mechanism $\mathcal{R}_1$ is ϵ_0-LDP. Let $\mathcal{Z} = \big\{ \pm\, a\mathbf{H}_d(j)\big(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\big) : j = 1, 2, \ldots, d\big\}$ denote all possible $2d$ outputs of the mechanism $\mathcal{R}_1$. We get

$$\begin{aligned}\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_1^d(a)}\sup_{\boldsymbol{z}\in\mathcal{Z}}\frac{\Pr[\mathcal{R}_1(\boldsymbol{x})=\boldsymbol{z}]}{\Pr[\mathcal{R}_1(\boldsymbol{x}')=\boldsymbol{z}]} &\leq \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_1^d(a)}\frac{\frac{1}{d}\sum_{j=1}^d\left(\frac{1}{2}+\frac{\sqrt{d}|y_j|}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)}{\frac{1}{d}\sum_{j=1}^d\left(\frac{1}{2}-\frac{\sqrt{d}|y'_j|}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)}\\ &= \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_1^d(a)}\frac{\frac{1}{d}\sum_{j=1}^d\left(a(e^{\epsilon_0}+1)+\sqrt{d}|y_j|(e^{\epsilon_0}-1)\right)}{\frac{1}{d}\sum_{j=1}^d\left(a(e^{\epsilon_0}+1)-\sqrt{d}|y'_j|(e^{\epsilon_0}-1)\right)}\\ &\overset{\text{(a)}}{\leq} \frac{2ae^{\epsilon_0}}{2a} = e^{\epsilon_0},\end{aligned}$$

where (a) uses the fact that for every $j \in [d]$, we have $|y_j| \leq {}^{a}/_{\sqrt{d}}$ and $|y'_j| \leq {}^{a}/_{\sqrt{d}}$.

2. Fix an arbitrary $\boldsymbol{x} \in \mathcal{B}_1^d\left(a\right)$.

$$\begin{aligned}\text{Unbiasedness:}\quad \mathbb{E}\left[\mathcal{R}_1\left(\boldsymbol{x}\right)\right] &= \frac{1}{d}\sum_{j=1}^d a\mathbf{H}_d\left(j\right)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\left(\frac{\sqrt{d}y_j}{a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)\\ &= \frac{1}{d}\sum_{j=1}^d\mathbf{H}_d\left(j\right)\sqrt{d}y_j \overset{\text{(b)}}{=} \frac{1}{d}\sum_{j=1}^d\mathbf{H}_d\left(j\right)\mathbf{H}_d^T(j)\boldsymbol{x} \overset{\text{(c)}}{=} \boldsymbol{x}\end{aligned}$$

where (b) uses $\boldsymbol{y} = \frac{1}{\sqrt{d}}\mathbf{H}_d\boldsymbol{x}$ and (c) uses $\sum_{j=1}^d \mathbf{H}_d(j)\mathbf{H}_d^T(j) = \mathbf{H}_d\mathbf{H}_d^T = d\mathbf{I}_d$.

$$\begin{aligned}\text{Bounded variance:}\quad \mathbb{E}\|\mathcal{R}_1\left(\boldsymbol{x}\right)-\boldsymbol{x}\|_2^2 &\leq \mathbb{E}\|\mathcal{R}_1(\boldsymbol{x})\|^2 = \mathbb{E}[\mathcal{R}_1(\boldsymbol{x})^T\mathcal{R}_1(\boldsymbol{x})]\\ &= \frac{1}{d}\sum_{j=1}^d a^2\mathbf{H}_d(j)^T\mathbf{H}_d(j)\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\\ &= a^2d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2 \qquad (\text{Since } \mathbf{H}_d(j)^T\mathbf{H}_d(j) = d, \forall j \in [d])\end{aligned}$$

This completes the proof of Lemma 10. ■

Now we are ready to prove Theorem 6. Let $\mathcal{R}_1(\boldsymbol{x})$ denote the output of Algorithm 3 on input $\boldsymbol{x}$. As mentioned above, the server employs a simple estimator that simply averages the n received messages, i.e., the server outputs $\widehat{\boldsymbol{x}}(\boldsymbol{z}^n) = \frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i = \frac{1}{n}\sum_{i=1}^n \mathcal{R}_1(\boldsymbol{x}_i)$. In the following, first we show the bound on $r_{\epsilon_0,b,n}^{1,d}\left(a\right)$ and then on $R_{\epsilon_0,b,n}^{1,d}\left(a\right)$ for $b = \log(d)+1$.

$$\begin{aligned}\text{For } r_{\epsilon_0,b,n}^{1,d}\left(a\right):\qquad \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_1^d(a)}\mathbb{E}\left\|\overline{\boldsymbol{x}}-\widehat{\boldsymbol{x}}(\boldsymbol{z}^n)\right\|_2^2 &= \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_1^d(a)}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n\left(\boldsymbol{x}_i-\mathcal{R}_1(\boldsymbol{x}_i)\right)\right\|_2^2\\ &\overset{\text{(a)}}{=} \sup_{\{\boldsymbol{x}_i\}\in\mathcal{B}_1^d(a)}\frac{1}{n^2}\sum_{i=1}^n\mathbb{E}\left\|\boldsymbol{x}_i-\mathcal{R}_1(\boldsymbol{x}_i)\right\|_2^2 \overset{\text{(b)}}{\leq} \frac{a^2d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2, \qquad (41)\end{aligned}$$

where (a) uses the fact that all clients use independent private randomness (which makes the random variables $\boldsymbol{x}_i - \mathcal{R}_1(\boldsymbol{x}_i)$ independent for different i's and also that $\mathcal{R}_1$ is unbiased. (b) uses that $\mathcal{R}_1$ has bounded variance. Taking infimum in (41) over all (ϵ_0, b)-CLDP mechanisms (where $b = \log(d)+1$) and estimators $\widehat{\boldsymbol{x}}$, we have that $r_{\epsilon_0,b,n}^{1,d}\left(a\right) \leq \frac{a^2d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$, which is $\mathcal{O}\left(\frac{a^2d}{n\epsilon_0^2}\right)$ when $\epsilon_0 = \mathcal{O}(1)$.

$$\text{For } R^{1,d}_{\epsilon_0,b,n}\left(a\right): \qquad \sup_{\boldsymbol{q}\in\mathcal{P}^d_1(a)} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\widehat{\boldsymbol{x}}(\boldsymbol{z}^n)\right\|_2^2 \overset{\text{(c)}}{\leq} \sup_{\boldsymbol{q}\in\mathcal{P}^d_1(a)}\left[2\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\overline{\boldsymbol{x}}\right\|_2^2+2\mathbb{E}\left\|\overline{\boldsymbol{x}}-\widehat{\boldsymbol{x}}(\boldsymbol{z}^n)\right\|_2^2\right]$$

$$\overset{\text{(d)}}{\leq} \frac{2a^2}{n}+\frac{2a^2d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2 \tag{42}$$

In the LHS of (c), for any $\boldsymbol{q}\in\mathcal{P}^d_1\left(a\right)$, first we generate n i.i.d. samples $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ and then compute $\boldsymbol{z}_i=\mathcal{R}_1(\boldsymbol{x}_i)$ for all $i\in[n]$. We use the Jensen's inequality in (c). We used $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\overline{\boldsymbol{x}}\right\|_2^2\leq\frac{a^2}{n}$ (see (39)) in (d). Taking infimum in (42) over all (ϵ_0,b)-CLDP mechanisms (where $b=\log(d)+1$) and estimators $\widehat{\boldsymbol{x}}$, we have that $R^{1,d}_{\epsilon_0,b,n}\left(a\right)\leq\frac{2a^2}{n}+\frac{2a^2d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$, which is $\mathcal{O}\left(\frac{a^2d}{n\epsilon_0^2}\right)$ when $\epsilon_0=\mathcal{O}(1)$.

This completes the proof of Theorem 6.

D.6 Achievability for ℓ_2-norm Ball: Proof of Theorem 7

In this section, we propose an ϵ_0-LDP mechanism that requires $\mathcal{O}\left(d\right)$-bits of communication per client using private randomness. Our proposed mechanism is a combination of the private-mechanism Priv from[Duchi et al., 2018, Section 4.2.3] and the non-private quantization mechanism Quan from [Mayekar and Tyagi, 2020, Section 4.2]. For completeness, we describe both these mechanisms in Algorithm 5 and Algorithm 6, respectively, and our proposed mechanism in Algorithm 4. Each client i first privatize its input $\boldsymbol{x}_i\in\mathcal{B}^d_2\left(a\right)$ using Priv and then quantize the privatized result using Quan and sends the final result $\boldsymbol{z}_i=\mathsf{Quan}(\mathsf{Priv}(\boldsymbol{x}_i))$ to the server, which outputs the average of all the received n messages. Since the server is only taking an average of the received messages, we only present the client side of our mechanism in Algorithm 4.

Algorithm 4 ℓ_2-MEAN-EST ($\mathcal{R}_2$: the client-side algorithm)

```
1: Input: Vector $\boldsymbol{x}\in\mathcal{B}^d_2\left(a\right)$, and local privacy level $\epsilon_0>0$.
2: Apply the randomized mechanism $\boldsymbol{y}=\mathsf{Priv}\left(\boldsymbol{x}\right)$.
3: Return $\boldsymbol{z}=\mathsf{Quan}\left(\boldsymbol{y}\right)$.
```

Algorithm 5 Priv (a private mechanism from Duchi et al. [2018])

```
1: Input: Vector $\boldsymbol{x}\in\mathcal{B}^d_2\left(a\right)$, and local privacy level $\epsilon_0>0$.
2: Compute $\widetilde{\boldsymbol{x}}=\begin{cases}+a\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} & \text{w.p. } \frac{1}{2}+\frac{\|\boldsymbol{x}\|_2}{2a}\\ -a\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} & \text{w.p. } \frac{1}{2}-\frac{\|\boldsymbol{x}\|_2}{2a}\end{cases}$
3: Sample $U\sim\text{Bernoulli}\left(\frac{e^{\epsilon_0}}{e^{\epsilon_0}+1}\right)$
4: $M\triangleq a\frac{\sqrt{\pi}}{2}\frac{\Gamma\left(\frac{d-1}{2}+1\right)}{\Gamma\left(\frac{d}{2}+1\right)}\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}$
5: $\boldsymbol{z}=\begin{cases}\mathsf{Unif}\left(\boldsymbol{y}:\boldsymbol{y}^T\widetilde{\boldsymbol{x}}>0,\|\boldsymbol{y}\|_2=M\right) & \text{if } U=1\\ \mathsf{Unif}\left(\boldsymbol{y}:\boldsymbol{y}^T\widetilde{\boldsymbol{x}}\leq0,\|\boldsymbol{y}\|_2=M\right) & \text{if } U=0\end{cases}$
6: Return $\boldsymbol{z}$.
```

Lemma 11 ([Duchi et al., 2018, Appendix *I*.2])**.** *The mechanism* Priv *presented in Algorithm 5 is unbiased and outputs a bounded length vector, i.e., for every* $\boldsymbol{x}\in\mathcal{B}^d_2\left(a\right)$*, we have*

$$\mathbb{E}[\mathsf{Priv}(\boldsymbol{x})]=\boldsymbol{x} \quad \textit{and} \quad \|\mathsf{Priv}(\boldsymbol{x})\|_2^2=M^2\leq a^2d\left(\frac{3\sqrt{\pi}}{4}\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

Lemma 12 ([Mayekar and Tyagi, 2020, Theorem 4.2])**.** *The mechanism* Quan *presented in Algorithm 6 is unbiased and has bounded variance, i.e., for every* $\boldsymbol{x}\in\mathcal{B}^d_2(a)$*, we have*

$$\mathbb{E}[\mathsf{Quan}(\boldsymbol{x})]=\boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\|\mathsf{Quan}(\boldsymbol{x})-\boldsymbol{x}\|_2^2\leq2\|\boldsymbol{x}\|^2\leq2a^2.$$

Furthermore, it requires $d\left(\log(e)+1\right)$*-bits to represent its output.*

Algorithm 6 Quan (a quantization mechanism from Mayekar and Tyagi [2020])

```
1: Input: Vector $\boldsymbol{x} \in \mathcal{B}_2^d(a)$, where $a$ is the radius of the ball.
2: Compute $\widetilde{\boldsymbol{x}} = \begin{cases} \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1} & \text{w.p. } \frac{1+\|\boldsymbol{x}\|_1}{2a\sqrt{d}} \\ -\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1} & \text{w.p. } \frac{1-\|\boldsymbol{x}\|_1}{2a\sqrt{d}} \end{cases}$
3: Generate a discrete distribution $\boldsymbol{\mu} = (|\widetilde{x}_1|, \ldots, |\widetilde{x}_d|)$ where $\Pr[\boldsymbol{\mu} = i] = |\widetilde{x}_i|$.
4: Construct a $d$-dimensional vector $\boldsymbol{y}$ by sampling $y_j \sim \boldsymbol{\mu}$ for $j \in [d]$
5: Return $\boldsymbol{z} = \frac{1}{d}\sum_{j=1}^{d}\left(a\sqrt{d} \cdot \text{sgn}(\widetilde{x}_{y_j}) \cdot \boldsymbol{e}_{y_j}\right)$.
```

Note that the radius a in Lemma 12 is equal to the length of any output of Priv, which is M (see line 4 of Algorithm 5).

Lemma 13. *The mechanism $\mathcal{R}_2$ presented in Algorithm 4 satisfies the following properties, where $\epsilon_0 > 0$:*

1. *$\mathcal{R}_2$ is $(\epsilon_0, d(\log(e)+1))$-CLDP.*
2. *$\mathcal{R}_2$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x} \in \mathcal{B}_2^d(a)$, we have*

$$\mathbb{E}\left[\mathcal{R}_2(\boldsymbol{x})\right] = \boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\|\mathcal{R}_2(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq 6a^2 d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

Proof. We prove these properties one-by-one below.

1. It was shown in [Duchi et al., 2018, Section 4.2.3] that Priv is an ϵ_0-LDP mechanism. Now, since $\mathcal{R}_2 =$ Quan $\circ$ Priv is a post-processing of a differentially-private mechanism Priv and post-processing preserves differential privacy, we have that $\mathcal{R}_2$ is also ϵ_0-LDP. The claim that $\mathcal{R}_2$ uses $d(\log(e)+1)$ bits of communication follows because $\mathcal{R}_2$ outputs the result of Quan, which produces an output which can be represented using $d(\log(e)+1)$ bits; see [Mayekar and Tyagi, 2020].

2. Unbiasedness of $\mathcal{R}_2$ follows because $\mathcal{R}_2 =$ Quan $\circ$ Priv and both Priv and Quan are unbiased. To prove that variance is bounded, fix an $\boldsymbol{x} \in \mathcal{B}_2^d(a)$.

$$\begin{aligned}\mathbb{E}\|\mathcal{R}_2(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 &= \mathbb{E}\|\mathsf{Quan}\left(\mathsf{Priv}(\boldsymbol{x})\right) - \boldsymbol{x}\|_2^2 \\ &= \mathbb{E}\|\mathsf{Quan}\left(\mathsf{Priv}(\boldsymbol{x})\right) - \mathsf{Priv}(\boldsymbol{x}) + \mathsf{Priv}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\ &\overset{\text{(a)}}{=} \mathbb{E}\|\mathsf{Quan}\left(\mathsf{Priv}(\boldsymbol{x})\right) - \mathsf{Priv}(\boldsymbol{x})\|_2^2 + \mathbb{E}\|\mathsf{Priv}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\ &\overset{\text{(b)}}{\leq} 2\|\mathsf{Priv}(\boldsymbol{x})\|^2 + \mathbb{E}\|\mathsf{Priv}(\boldsymbol{x})\|^2 \\ &\overset{\text{(c)}}{\leq} 3\|\mathsf{Priv}(\boldsymbol{x})\|^2 \overset{\text{(d)}}{\leq} 6d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.\end{aligned}$$

   In (a) we used the fact that Quan and Priv are unbiased, which implies that the cross multiplication term is zero. In (b) we used Lemma 12 to write $\mathbb{E}\|\mathsf{Quan}\left(\mathsf{Priv}(\boldsymbol{x})\right) - \mathsf{Priv}(\boldsymbol{x})\|_2^2 \leq 2\|\mathsf{Priv}(\boldsymbol{x})\|^2$ and used the unbiasedness of Priv together with the fact that variance is bounded by the second moment to write $\mathbb{E}\|\mathsf{Priv}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq \mathbb{E}\|\mathsf{Priv}(\boldsymbol{x})\|_2^2$. In (c) we used that the length of Priv on any input remains fixed, i.e., $\mathbb{E}\|\mathsf{Priv}(\boldsymbol{x})\|^2 = \|\mathsf{Priv}(\boldsymbol{x})\|^2 = M^2$ (where M is from the line 4 of Algorithm 5) holds for any $\boldsymbol{x} \in \mathcal{B}_2^d(a)$. In (d) we used the bound on $\|\mathsf{Priv}(\boldsymbol{x})\|_2^2$ from Lemma 11.

This completes the proof of Lemma 13. ■

Now we are ready to prove Theorem 7. In order to bound $r_{\epsilon_0,b,n}^{2,d}(a)$ for $b = d(\log(e)+1)$, we follow exactly the same steps that we used to bound $r_{\epsilon_0,b,n}^{1,d}(a)$ and arrived at (41). This would give $r_{\epsilon_0,b,n}^{2,d}(a) \leq \frac{6a^2 d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$, which is $\mathcal{O}\left(\frac{a^2 d}{n\epsilon_0^2}\right)$ when $\epsilon_0 = \mathcal{O}(1)$. To bound $R_{\epsilon_0,b,n}^{2,d}(a)$, first note that when $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{B}_2^d(a)$, then we have from (39) that $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\overline{\boldsymbol{x}}\right\|_2^2 \leq \frac{a^2}{n}$. Here $\boldsymbol{q}\in\mathcal{P}_2^d\left(a\right)$ and $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ are sampled from $\boldsymbol{q}$ i.i.d. Now, following exactly the same steps that we used to bound $R_{\epsilon_0,b,n}^{1,d}\left(a\right)$ and arrived at (42). This would give $R_{\epsilon_0,b,n}^{2,d}\left(a\right)\leq\frac{2a^2}{n}+\frac{12a^2d}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$ for $b=d(\log(e)+1)$. Note that $R_{\epsilon_0,b,n}^{2,d}\left(a\right)=\mathcal{O}\left(\frac{a^2d}{n\epsilon_0^2}\right)$ when $\epsilon_0=\mathcal{O}(1)$.

This completes the proof of Theorem 7.

D.7 Achievability for ℓ_∞-norm Ball: Proof of Theorem 8

In this section, we propose an ϵ_0-LDP mechanism that requires $\mathcal{O}\left(\log\left(d\right)\right)$-bits per client using private randomness and 1-bit of communication per client using public randomness. Each client i has an input $\boldsymbol{x}_i\in\mathcal{B}_\infty^d\left(a\right)$. It selects $j\sim\mathsf{Unif}[d]$ and quantize $x_{i,j}$ according to (43) and obtains $\boldsymbol{z}_i\in\left\{\pm ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j\right\}$, which can be represented using only 1 bit, where $\boldsymbol{e}_j$ is the j'th standard basis vector in $\mathbb{R}^d$. Client i sends $\boldsymbol{z}_i$ to the server. Server receives the n messages $\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\}$ from the clients and outputs their average $\frac{1}{n}\sum_{i=1}^n\boldsymbol{z}_i$. We present this mechanism in Algorithm 7 – we only present the client-side part of the algorithm, as server only averages the messages received from the clients.

Algorithm 7 ℓ_∞-MEAN-EST ($\mathcal{R}_\infty$: the client-side algorithm)

```
Input: Vector $\boldsymbol{x}\in\mathcal{B}_\infty^d\left(a\right)$, and local privacy level $\epsilon_0>0$.
Sample $j\sim\mathsf{Unif}[d]$ and quantize $x_j$ as follows:
    $\boldsymbol{z}=\begin{cases}+ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j & \text{w.p. } \frac{1}{2}+\frac{x_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\\ -ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j & \text{w.p. } \frac{1}{2}-\frac{x_j}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\end{cases}$    (43)
    where $\boldsymbol{e}_j$ is the $j$'th standard basis vector in $\mathbb{R}^d$
Return $\boldsymbol{z}$.
```

Lemma 14. *The mechanism $\mathcal{R}_\infty$ presented in Algorithm 7 satisfies the following properties, where $\epsilon_0>0$:*

1. *$\mathcal{R}_\infty$ is $(\epsilon_0,\log\left(d\right)+1)$-CLDP and requires only 1-bit of communication using public randomness.*

2. *$\mathcal{R}_\infty$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x}\in\mathcal{B}_\infty^d\left(a\right)$, we have*

$$\mathbb{E}\left[\mathcal{R}_\infty\left(\boldsymbol{x}\right)\right]=\boldsymbol{x}\quad\textit{and}\quad\mathbb{E}\|\mathcal{R}_\infty\left(\boldsymbol{x}\right)-\boldsymbol{x}\|_2^2\leq a^2d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

Proof. We prove these properties one-by-one below.

1. Observe that the output of the mechanism $\mathcal{R}_\infty$ can be represented using the index $j\in[d]$ and one bit for the sign of $\left\{\pm ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j\right\}$. Hence, it requires only $\log\left(d\right)+1$ bits for communication. Furthermore, the randomness $j\sim\mathsf{Unif}\left[d\right]$ is independent of the input $\boldsymbol{x}$. Thus, if the client has access to a public randomness j, then the client needs only to send one bit for its sign. Now, we show that the mechanism $\mathcal{R}_\infty$ is ϵ_0-LDP. Let $\mathcal{Z}=\left\{\pm ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\boldsymbol{e}_j : j=1,2,\ldots,d\right\}$ denote all possible $2d$ outputs of the mechanism $\mathcal{R}_\infty$. We get

$$\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_\infty^d(a)}\ \sup_{\boldsymbol{z}\in\mathcal{Z}}\frac{\Pr\left[\mathcal{R}_\infty\left(\boldsymbol{x}\right)=\boldsymbol{z}\right]}{\Pr\left[\mathcal{R}_\infty\left(\boldsymbol{x}\right)=\boldsymbol{z}\right]}\leq\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_\infty^d(a)}\frac{\frac{1}{d}\sum_{i=1}^d\left(\frac{1}{2}+\frac{|x_j|}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)}{\frac{1}{d}\sum_{i=1}^d\left(\frac{1}{2}-\frac{|x_j'|}{2a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)}\tag{44}$$

$$=\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_\infty^d}\frac{\frac{1}{d}\sum_{i=1}^d\left(a(e^{\epsilon_0}+1)+|x_j|(e^{\epsilon_0}-1)\right)}{\frac{1}{d}\sum_{i=1}^d\left(a(e^{\epsilon_0}+1)-|x_j'|(e^{\epsilon_0}-1)\right)}\tag{45}$$

$$\overset{\text{(a)}}{\leq}\frac{2ae^{\epsilon_0}}{2a}=e^{\epsilon_0},\tag{46}$$

where in (a) we used the fact that for every $j\in[d]$, we have $|x_j|\leq a$ and $|x_j'|\leq a$.

2. Fix an arbitrary $\boldsymbol{x} \in \mathcal{B}_{\infty}^{d}$.

$$\begin{aligned}\text{Unbiasedness:} \quad \mathbb{E}\left[\mathcal{R}_{\infty}\left(\boldsymbol{x}\right)\right] &= \frac{1}{d}\sum_{j=1}^{d}\boldsymbol{e}_{j}ad\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\left(\frac{x_j}{a}\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}+1}\right)\\ &= \sum_{j=1}^{d}\boldsymbol{e}_{j}x_{j}\\ &= \boldsymbol{x}\end{aligned}$$

$$\begin{aligned}\text{Bounded variance:} \quad \mathbb{E}\|\mathcal{R}_{\infty}(\boldsymbol{x})-\boldsymbol{x}\|_2^2 \leq \mathbb{E}\|\mathcal{R}_{\infty}(\boldsymbol{x})\|^2 &= \mathbb{E}[\mathcal{R}_{\infty}(\boldsymbol{x})^T\mathcal{R}_{\infty}(\boldsymbol{x})]\\ &= \frac{1}{d}\sum_{j=1}^{d}a^2d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\\ &= a^2d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\end{aligned}$$

This completes the proof of Lemma 14. ■

Now we are ready to prove Theorem 8. In order to bound $r_{\epsilon_0,b,n}^{\infty,d}\left(a\right)$ for $b=\log\left(d\right)+1$, we follow exactly the same steps that we used to bound $r_{\epsilon_0,b,n}^{1,d}\left(a\right)$ and arrived at (41). This would give $r_{\epsilon_0,b,n}^{\infty,d}\left(a\right) \leq \frac{a^2d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$, which is $\mathcal{O}\left(\frac{a^2d^2}{n\epsilon_0^2}\right)$ when $\epsilon_0=\mathcal{O}(1)$. To bound $R_{\epsilon_0,b,n}^{\infty,d}\left(a\right)$, first note that when $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \in \mathcal{B}_{\infty}^{d}\left(a\right)$, then we have from (36) (by substituting $p=\infty$) that $\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}}-\overline{\boldsymbol{x}}\right\|_2^2 \leq \frac{a^2d}{n}$. Here $\boldsymbol{q}\in\mathcal{P}_{\infty}^{d}\left(a\right)$ and $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ are sampled from $\boldsymbol{q}$ i.i.d. Now, following exactly the same steps that we used to bound $R_{\epsilon_0,b,n}^{1,d}\left(a\right)$ and arrived at (42). This would give $R_{\epsilon_0,b,n}^{\infty,d}\left(a\right) \leq \frac{2a^2d}{n}+\frac{2a^2d^2}{n}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^{2,d}$ for $b=\log\left(d\right)+1$. Note that $R_{\epsilon_0,b,n}^{\infty,d}\left(a\right)=\mathcal{O}\left(\frac{a^2d^2}{n\epsilon_0^2}\right)$ when $\epsilon_0=\mathcal{O}(1)$. This completes the proof of Theorem 8.

D.8 Achievability for ℓ_p-norm Ball for $p\in[1,\infty)$: Proof of Corollary 1

In this section, first we propose two ϵ_0-LDP mechanisms for ℓ_p-norm ball $\mathcal{B}_p^d(a)$ for $p\in[1,\infty)$ based on the inequalities between different norms, and our final mechanism will be chosen probabilistically from these two. The first mechanism, which we denote by $\mathcal{R}_p^{(1)}$, is based on the private mechanism $\mathcal{R}_1$ (presented in Algorithm 3) that requires $\mathcal{O}\left(\log\left(d\right)\right)$ bits per client. The second mechanism, which we denote by $\mathcal{R}_p^{(2)}$ is based on the private mechanism $\mathcal{R}_2$ (presented in Algorithm 4) that requires $\mathcal{O}\left(d\right)$ bits per client. Observe that for any $1\leq p\leq q\leq\infty$, using the relation between different norms ($\|\boldsymbol{u}\|_q \leq \|\boldsymbol{u}\|_p \leq d^{\frac{1}{p}-\frac{1}{q}}\|\boldsymbol{u}\|_q$), we have

$$\mathcal{B}_q^d\left(a\right) \subseteq \mathcal{B}_p^d\left(a\right) \subseteq \mathcal{B}_q^d\left(ad^{\frac{1}{p}-\frac{1}{q}}\right). \tag{47}$$

1. *Description of the private mechanism $\mathcal{R}_p^{(1)}$:* Each client has a vector $\boldsymbol{x}_i\in\mathcal{B}_p^d\left(a\right)\subseteq\mathcal{B}_1^d\left(ad^{1-\frac{1}{p}}\right)$. Thus, each client runs the private mechanism $\mathcal{R}_1\left(\boldsymbol{x}_i\right)$ presented in Algorithm 3 with radius $ad^{1-\frac{1}{p}}$. Thus, the mechanism $\mathcal{R}_p^{(1)}$ for $p\in[1,\infty)$ satisfies the following properties, where $\epsilon_0>0$:
   - $\mathcal{R}_p^{(1)}$ is $(\epsilon_0,\log\left(d\right)+1)$-CLDP and requires only 1-bit of communication using public randomness.
   - $\mathcal{R}_p^{(1)}$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x}\in\mathcal{B}_p^d\left(a\right)$, we have

$$\mathbb{E}\left[\mathcal{R}_p^{(1)}\left(\boldsymbol{x}\right)\right]=\boldsymbol{x} \quad \text{and} \quad \mathbb{E}\|\mathcal{R}_p^{(1)}\left(\boldsymbol{x}\right)-\boldsymbol{x}\|_2^2 \leq a^2d^{3-\frac{2}{p}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

2. *Description of the private mechanism $\mathcal{R}_p^{(2)}$:* Each client has a vector $\boldsymbol{x}_i\in\mathcal{B}_p^d\left(a\right)\subseteq\mathcal{B}_2^d\left(a\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}\right)$. Thus, each client runs the private mechanism $\mathcal{R}_2\left(\boldsymbol{x}_i\right)$ presented in Algorithm 4 with radius $a\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}$. Thus, the mechanism $\mathcal{R}_p^{(2)}$ for $p\in[1,\infty)$ satisfies the following properties, where $\epsilon_0>0$:

- $\mathcal{R}_p^{(2)}$ is $(\epsilon_0, d(\log(e)+1))$-CLDP.
- $\mathcal{R}_p^{(2)}$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x} \in \mathcal{B}_p^d(a)$, we have

$$\mathbb{E}\left[\mathcal{R}_p^{(2)}(\boldsymbol{x})\right] = \boldsymbol{x} \quad \text{and} \quad \mathbb{E}\|\mathcal{R}_p^{(2)}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq 6a^2 \max\{d^{2-\frac{2}{p}}, d\} \left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2.$$

Note that $\mathcal{R}_p^{(1)}$ requires low communication and has high variance, whereas, $\mathcal{R}_p^{(2)}$ requires high communication and has low variance: $\mathcal{R}_p^{(2)}$ requires exponentially more communication than $\mathcal{R}_p^{(1)}$, whereas, $\mathcal{R}_p^{(1)}$ has a factor of d more variance than $\mathcal{R}_p^{(2)}$.

To define our final mechanism $\mathcal{R}_p$ for any norm $p \in [1,\infty)$, we choose $\mathcal{R}_p^{(1)}$ with probability $\bar{p}$ and $\mathcal{R}_p^{(2)}$ with probability $(1-\bar{p})$, where $\bar{p}$ is any number in $[0,1]$. Note that $\mathcal{R}_p$ is ϵ_0-LDP and requires $\bar{p}\log(d)+(1-\bar{p})d\log(e)+1$ expected communication, where expectation is taken over the sampling of choosing $\mathcal{R}_p^{(1)}$ or $\mathcal{R}_p^{(2)}$. We have the following bounds on $r_{\epsilon_0,b,n}^{p,d}(a)$ and $R_{\epsilon_0,b,n}^{p,d}(a)$:

$$\begin{aligned} r_{\epsilon_0,b,n}^{p,d}(a) &\leq \bar{p}\, d^{2-\frac{2}{p}} r_{\epsilon_0,b,n}^{1,d}(a) + (1-\bar{p}) \max\{d^{1-\frac{2}{p}}, 1\} r_{\epsilon_0,b,n}^{2,d}(a) \\ \text{For } R_{\epsilon_0,b,n}^{p,d}(a) &\leq \bar{p}\, d^{2-\frac{2}{p}} R_{\epsilon_0,b,n}^{1,d}(a) + (1-\bar{p}) \max\{d^{1-\frac{2}{p}}, 1\} R_{\epsilon_0,b,n}^{2,d}(a) \end{aligned}$$

This completes the proof of Corollary 1.

E Minimax Risk Estimation

Lemma 15. *For the minimax problems* (16) *and* (17)*, the optimal estimator* $\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)$ *is a deterministic function. In other words, the randomized decoder does not help in reducing the minimax risk.*

Proof. Towards a contradiction, suppose that the optimal estimator $\widehat{\boldsymbol{x}}$ is a randomized decoder defined as follows. For given clients' responses $\boldsymbol{y}^n$, let the probabilistic estimator generate an estimate $\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)$ whose mean and trace of the covariance matrix are given by $\boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} = \mathbb{E}[\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)]$ and $\sigma^2_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} = \mathbb{E}\left[\left\|\widehat{\boldsymbol{x}}(\boldsymbol{y}^n) - \mu_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}\right\|_2^2 \middle| Y^n\right]$, respectively, where expectation is taken with respect to the randomization of the decoder, conditioned of Y^n.

$$\begin{aligned} \mathbb{E}\left[\|\overline{\boldsymbol{x}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\|_2^2 \middle| \boldsymbol{y}^n\right] &= \mathbb{E}\left[\left\|\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} + \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2 \middle| \boldsymbol{y}^n\right] \\ &= \mathbb{E}\left[\left\|\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}\right\|_2^2 \middle| \boldsymbol{y}^n\right] + \mathbb{E}\left[\left\|\boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\right\|_2^2 \middle| \boldsymbol{y}^n\right] \\ &\quad + 2\mathbb{E}\left\langle \overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}, \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n) \middle| \boldsymbol{y}^n\right\rangle \\ &\overset{\text{(a)}}{=} \mathbb{E}\left[\left\|\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}\right\|_2^2 \middle| \boldsymbol{y}^n\right] + \sigma^2_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} \\ &> \mathbb{E}\left[\left\|\overline{\boldsymbol{x}} - \boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}\right\|_2^2 \middle| \boldsymbol{y}^n\right] \end{aligned}$$

In (a), we used that $\boldsymbol{\mu}_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)} = \mathbb{E}[\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)]$ to eliminate the last term. Similarly, we can prove that $\mathbb{E}\left[\|\boldsymbol{\mu}_{\mathbf{q}} - \widehat{\boldsymbol{x}}(\boldsymbol{y}^n)\|_2^2 \middle| \boldsymbol{y}^n\right] > \mathbb{E}\left[\|\boldsymbol{\mu}_{\mathbf{q}} - \boldsymbol{\mu}_{\boldsymbol{y}^n}\|_2^2 \middle| \boldsymbol{y}^n\right]$. Hence, the deterministic estimator $\widehat{\boldsymbol{x}}(\boldsymbol{y}^n) = \mu_{\widehat{\boldsymbol{x}}(\boldsymbol{y}^n)}$ has a lower minimax risk than the probabilistic estimator. ■

F Optimization: Privacy, Communication, and Convergence Analyses

In this section, we establish the privacy, communication, and convergence guarantees of Algorithm 1 and prove Theorem 1. We show these three results on privacy, communication, and convergence separately in the next three subsections.

F.1 Proof of Theorem 1: Privacy

We have already proven Lemma 3 in Appendix C. Now we use that to prove our final privacy parameter of our entire algorithm $\mathcal{A}_{cldp}$.

Note that the Algorithm $\mathcal{A}_{cldp}$ is a sequence of T adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_T$, where each $\mathcal{M}_t$ for $t \in [T]$ satisfies the privacy guarantee stated in Lemma 3. Now, we invoke the strong composition stated in Lemma 6 to obtain the privacy guarantee of the algorithm $\mathcal{A}_{cldp}$. We can conclude that for any $\delta' > 0$, $\mathcal{A}_{cldp}$ is (ϵ, δ)-DP for

$$\epsilon = \sqrt{2T \log\left(1/\delta'\right)}\overline{\epsilon} + T\overline{\epsilon}\left(e^{\overline{\epsilon}} - 1\right), \quad \delta = qT\tilde{\delta} + \delta',$$

where $\overline{\epsilon}$ is from Lemma 3. We have from Lemma 6 that if $\overline{\epsilon} = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta')}{T}}\right)$, then $\epsilon = \mathcal{O}\left(\overline{\epsilon}\sqrt{T\log\left(1/\delta'\right)}\right)$. If $\epsilon_0 = \mathcal{O}(1)$, then we can satisfy this condition on $\overline{\epsilon}$ by choosing $\epsilon_0 = \mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{qT\log(1/\tilde{\delta})}}\right)$. By substituting the bound on $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{n}}\right)$ from Lemma 3, we have $\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log\left(1/\tilde{\delta}\right)\log(1/\delta')}{n}}\right)$. By setting $\tilde{\delta} = \frac{\delta}{2qT}$ and $\delta' = \frac{\delta}{2}$, we get $\epsilon_0 = \mathcal{O}\left(\sqrt{\frac{n\log(2/\delta)}{qT\log(2qT/\delta)}}\right)$ and $\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{n}}\right)$. This completes the proof of the privacy part of Theorem 1.

F.2 Proof of Theorem 1: Communication

The (ϵ_0, b)-CLDP mechanism $\mathcal{R}_p : \mathcal{X} \to \mathcal{Y}$ used in Algorithm 1 has output alphabet $\mathcal{Y} = \{1, 2, \ldots, B = 2^b\}$. So, the output of $\mathcal{R}_p$ on any input can be represented by b bits. Therefore, the naïve scheme for any client to send the s compressed and private gradients requires sb bits per iteration. We can reduce this communication cost by using the histogram trick from Mayekar and Tyagi [2020] which was applied in the context of non-private quantization. The idea is as follows. Since any client applies the *same* randomized mechanism $\mathcal{R}_p$ to the s gradients, the output of these s identical mechanisms can be represented accurately using the histogram of the s outputs, which takes value from the set $\mathcal{A}_B^s = \{(n_1, \ldots, n_B) : \sum_{j=1}^{B} n_j = s \text{ and } n_j \geq 0, \forall j \in [B]\}$. Since the cardinality of this set is $\binom{s+B-1}{s} \leq \left(\frac{e(s+B-1)}{s}\right)^s$, it requires at most $s\left(\log\left(e\right) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits to send the s compressed gradients. Since the probability that the client is chosen at any time $t \in [T]$ is given by $\frac{k}{m}$, the expected number of bits per client in Algorithm $\mathcal{A}_{cldp}$ is given by $\frac{k}{m} \times T \times s\left(\log\left(e\right) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits, where expectation is taken over the sampling of k out of m clients in all T iterations.

This completes the proof of the second part of Theorem 1.

F.3 Proof of Theorem 1 : Convergence

First we prove Lemma 4 and then using that we prove the convergence.

Proof of Lemma 4. Under the conditions of the lemma, we have from [Shalev-Shwartz et al., 2012, Lemma 2.6] that $\|\nabla_\theta f\left(\theta; d\right)\|_p \leq L$ for all $d \in \mathfrak{S}$, which implies that $\|\nabla_\theta F(\theta)\|_p \leq L$. Thus, we have

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 &= \|\mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 \\
&\overset{(a)}{\leq} \max\{d^{1-\frac{2}{p}}, 1\}L^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 \\
&\overset{(b)}{\leq} \max\{d^{1-\frac{2}{p}}, 1\}L^2 + \frac{cL^2\max\{d^{2-\frac{2}{p}}, d\}}{ks}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2,
\end{aligned}$$

where c is a global constant, and $c = 4$ if $p \in \{1, \infty\}$ and $c = 14$ otherwise. Step (a) follows from the fact that $\|\nabla_{\theta_t} F\left(\theta_t\right)\|_p \leq L$ together with the norm inequality $\|\boldsymbol{u}\|_q \leq \|\boldsymbol{u}\|_p \leq d^{\frac{1}{p}-\frac{1}{q}}\|\boldsymbol{u}\|_q$ for $1 \leq p \leq q \leq \infty$. The claim follows by substituting $q = \frac{ks}{n}$. ■

Using the bound on G^2 from Lemma 4, we have that the output θ_T of Algorithm 1 satisfies

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}}, 1\}}{\sqrt{T}}\left(1 + \sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right)\right), \tag{48}$$

Algorithm 8 $\mathcal{A}_{\text{cldp}}$: CLDP-SGD with New Sampling

```
Inputs: Datasets $\mathcal{D} = \bigcup_{i\in[m]} \mathcal{D}_i$, $\mathcal{D}_i = \{d_{i1},\ldots,d_{ir}\}$, loss function $F(\theta) = \frac{1}{mr}\sum_{i=1}^{m}\sum_{j=1}^{r} f(\theta; d_{ij})$, LDP privacy parameter $\epsilon_0$, gradient norm bound $C$, and learning rate $\eta_t$.
Initialize: $\theta_0 \in \mathcal{C}$
for $t \in [T]$ do
    for each client $i \in [m]$ do
        Sampling 1: Client $i$ chooses uniformly at random a set $\mathcal{S}_{it}$ of $s$ samples.
        for Samples $j \in \mathcal{S}_{it}$ do
            $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$
            $\tilde{\mathbf{g}}_t(d_{ij}) \leftarrow \mathbf{g}_t(d_{ij}) / \max\left\{1, \frac{\|\mathbf{g}_t(d_{ij})\|_p}{C}\right\}$
            $\mathbf{q}_t(d_{ij}) \leftarrow \mathcal{R}_p(\tilde{\mathbf{g}}_t(d_{ij}))$
        Client $i$ sends $\{\mathbf{q}_t(d_{ij})\}_{j\in\mathcal{S}_{it}}$ to the shuffler.
    Sampling 2: The shuffler selects a uniformly random subset of $ks$ elements from $\{\mathbf{q}_t(d_{ij})\}_{j\in\mathcal{S}_{it}}$. Let $\mathcal{U} \subseteq \{(i,j) : i\in[m], j\in\mathcal{S}_{it}\}$ denote the indices of these selected $ks$ elements.
    Shuffling: The shuffler randomly shuffles the elements in $\{\boldsymbol{q}_t(d_{ij}) : (i,j)\in\mathcal{U}\}$ and sends them to the server.
    Aggregate: $\overline{\mathbf{g}}_t \leftarrow \frac{1}{ks}\sum_{(i,j)\in\mathcal{U}} \boldsymbol{q}_t(d_{ij})$.
    Gradient Descent $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}(\theta_t - \eta_t \overline{\mathbf{g}}_t)$
Output: The final model parameters $\theta_T$.
```

where we used the inequality $\sqrt{1+\frac{cd}{qn}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2} \leq \left(1+\sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right)$.

Note that if $\sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) \leq \mathcal{O}(1)$, then we recover the convergence rate of vanilla SGD without privacy. So, the interesting case is when $\sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right) \geq \Omega(1)$, which gives

$$\mathbb{E}\left[F(\theta_T)\right] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}}{\sqrt{T}}\sqrt{\frac{cd}{qn}}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)\right).$$

This completes the proof of the third part of Theorem 1.

F.4 Privacy Guarantee for a New Sampling Procedure

As mentioned in Remark 1, we can show a general privacy amplification by subsampling for $q=\frac{ks}{mr}$ (instead of just by the factor of $q=\frac{k}{mr}$ as in Theorem 1) by using a different sampling procedure, where all clients send s compressed and private gradients corresponding to a uniformly random subset of s data points from their datasets; shuffler selects a uniformly random subset of ks gradients from them and then sends the shuffled output to the server. Note that, in this procedure, each data point has a probability $q=\frac{ks}{mr}$ of being picked, and we pick $\frac{ks}{m}$ data points (in expectation) from each clients. Note that even for this sampling (which does not yield uniform sampling of ks points from mr points), the privacy amplification of this sampling mechanism does not directly follow from existing results.

For convenience, we describe the modified algorithm with this new sampling procedure in Algorithm 8. The final privacy guarantee of this algorithm is given below.

Theorem 9. *Let $q=\frac{ks}{mr}$. Under the above sampling procedure, Algorithm $\mathcal{A}_{cldp}$ satisfies the following privacy guarantee: For $\epsilon_0=\mathcal{O}(1)$, $\mathcal{A}_{cldp}$ is (ϵ,δ)-DP, where $\delta>0$ is arbitrary, and*

$$\epsilon = \mathcal{O}\left(\epsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{n}}\right). \tag{49}$$

Proof. Fix an iteration number $t\in[T]$ in Algorithm 8. Let $\mathcal{M}_t(\theta_t,\mathcal{D})$ denote the private mechanism at time t that takes the dataset $\mathcal{D}$ and an auxiliary input θ_t (which is the parameter vector at the t'th iteration) and generates the parameter θ_{t+1} as an output. Thus, the mechanism $\mathcal{M}_t$ on an input dataset $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i \in \mathfrak{S}^n$ can be defined as:

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{ks} \circ \text{samp}_{ms,ks}\left(\{\mathcal{G}_1, \ldots, \mathcal{G}_m\}\right), \tag{50}$$

where $\mathcal{G}_i = \text{samp}_{r,s}\left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{ir}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [m], j \in [r]$. Here, $\mathcal{H}_{ks}$ denotes the shuffling operation on ks elements and $\text{samp}_{a,b}$ denotes the sampling operation for choosing a random subset of b elements from a set of a elements.

Now we state the privacy guarantee of the mechanism $\mathcal{M}_t$ for each $t \in [T]$.

Lemma 16. *Let $q = \frac{ks}{mr}$. Suppose $\mathcal{R}$ is an ϵ_0-LDP mechanism, where $\epsilon_0 \leq \frac{\log\left(qn/\log\left(1/\tilde{\delta}\right)\right)}{2}$ and $\tilde{\delta} > 0$ is arbitrary. Then, for any $t \in [T]$, the mechanism $\mathcal{M}_t$ is $\left(\overline{\epsilon}, \overline{\delta}\right)$-DP, where $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)), \overline{\delta} = q\tilde{\delta}$ with $\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\} e^{\epsilon_0} \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{qn}}\right)$. In particular, if $\epsilon_0 = \mathcal{O}(1)$, we get $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{q \log\left(1/\tilde{\delta}\right)}{n}}\right)$.*

We prove Lemma 16 next in Appendix F.5.

Analogous to how we proved the privacy guarantee of Theorem 1 from Lemma 3 using strong composition (see Appendix F.1 for details), we can also prove Theorem 9 using Lemma 16, and we omit the details here. ■

F.5 Proof of Lemma 16

This can be proved along the lines of the proof of Lemma 3. For completeness, we give a detailed proof below.

We define a mechanism $\mathcal{Z}\left(\mathcal{D}^{(t)}\right) = \mathcal{H}_{ks}\left(\mathcal{R}\left(\boldsymbol{x}_1^t\right), \ldots, \mathcal{R}\left(\boldsymbol{x}_{ks}^t\right)\right)$ which is a shuffling of ks outputs of local mechanism $\mathcal{R}$, where $\mathcal{D}^{(t)}$ denotes an arbitrary set of ks data points and we index $\boldsymbol{x}_i^t$'s from $i = 1$ to ks just for convenience. From the amplification by shuffling result [Balle et al., 2019c, Corollary 5.3.1] (also see Lemma 8), the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, where $\tilde{\delta} > 0$ is arbitrary, and, if $\epsilon_0 \leq \frac{\log\left(ks/\log\left(1/\tilde{\delta}\right)\right)}{2}$, then

$$\tilde{\epsilon} = \mathcal{O}\left(\min\{\epsilon_0, 1\} e^{\epsilon_0} \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{ks}}\right). \tag{51}$$

Furthermore, when $\epsilon_0 = \mathcal{O}(1)$, we get $\tilde{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{ks}}\right)$.

For $i \in [m]$, let $\mathcal{T}_i \subseteq \{1, \ldots, r\}$ denote the identities of the s data points chosen at client i at iteration t and define $\mathcal{D}^{\mathcal{T}_i} = \{d_{ij} : j \in \mathcal{T}_i\}$. Let $\mathcal{D}^{\mathcal{T}_{[m]}} = \{\mathcal{D}^{\mathcal{T}_i} : i \in [m]\}$, which has ms elements. The shuffler selects ks elements from $\mathcal{D}^{\mathcal{T}_{[m]}}$ uniformly at random,[12] and we denote the resulting set by $\mathcal{D}^{\overline{\mathcal{T}}}$, which has ks elements. Note that $\mathcal{D}^{\overline{\mathcal{T}}}$ is a random set, where randomness is due to the sampling of data points in both stages. The mechanism $\mathcal{M}_t$ can be equivalently written as $\mathcal{M}_t = \mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}})$.

Observe that our sampling strategy is different from subsampling of a uniformly random subset of ks data points from the entire dataset $\mathcal{D}$. Thus, we revisit the proof of privacy amplification by subsampling (see, for example, Ullman [2017]) – which is for uniform sampling – to compute the privacy parameters of the mechanism $\mathcal{M}_t$, where sampling is non-uniform. Define a dataset $\mathcal{D}' = (\mathcal{D}'_1) \bigcup (\cup_{i=2}^{m} \mathcal{D}_i) \in \mathfrak{S}^n$, where $\mathcal{D}'_1 = \{d'_{11}, d_{12}, \ldots, d_{1r}\}$ is different from the dataset $\mathcal{D}_1$ in the first data point d_{11}. Note that $\mathcal{D}$ and $\mathcal{D}'$ are neighboring datasets – where, we assume, without loss of generality, that the differing elements are d_{11} and d'_{11}.

In order to show that $\mathcal{M}_t$ is $(\overline{\epsilon}, \overline{\delta})$-DP, we need show that for an arbitrary subset $\mathcal{S}$ of the range of $\mathcal{M}_t$, we have

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] \leq e^{\overline{\epsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + \overline{\delta} \tag{52}$$

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] \leq e^{\overline{\epsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] + \overline{\delta} \tag{53}$$

[12] Though the shuffler selects ks gradients from the received ms gradients, but effectively, we can assume that it selects ks data points that correspond to these gradients.

Note that both (20) and (21) are symmetric, so it suffices to prove only one of them. We prove (20) below.

We define conditional probabilities as follows:

$$
\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
A'_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
A_{10} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
A_{0} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \notin \mathcal{D}^{\mathcal{T}_1}\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \notin \mathcal{D}^{\mathcal{T}_1}\right]
\end{aligned}
$$

Let $q_1 = \frac{s}{r}, q_2 = \frac{ks}{ms} = \frac{k}{m}$, and $q = q_1 q_2 = \frac{ks}{mr}$. Thus, we have

$$
\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= qA_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0 \\
\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] &= qA'_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0
\end{aligned}
$$

We show the following inequalities:

$$A_{11} \le e^{\tilde{\epsilon}} A'_{11} + \tilde{\delta}, \tag{54}$$

$$A_{11} \le e^{\tilde{\epsilon}} A_{10} + \tilde{\delta}, \tag{55}$$

$$A_{11} \le e^{\tilde{\epsilon}} A_{0} + \tilde{\delta}. \tag{56}$$

Here, (54) is straightforward and follows because the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP. However, proving (55) and (56) is not straightforward and requires a combinatorial argument, which we give after we show our final result below.

Inequalities (54)-(56) together imply $A_{11} \le e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}, A_0\} + \tilde{\delta}$. Now we prove (52) for $\overline{\epsilon} = \ln(1 + q(e^{\tilde{\epsilon}} - 1)$ and $\overline{\delta} = q\tilde{\delta}$.

$$
\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= qA_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0 \\
&\le q\left(e^{\tilde{\epsilon}} \min\{A'_{11}, A_{10}, A_0\} + \tilde{\delta}\right) + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0 \\
&= q\left((e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}, A_0\} + \min\{A'_{11}, A_{10}, A_0\}\right) + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0 + q\tilde{\delta} \\
&\overset{\text{(a)}}{\le} q(e^{\tilde{\epsilon}} - 1)\min\{A'_{11}, A_{10}, A_0\} + qA'_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0 + q\tilde{\delta} \\
&\overset{\text{(b)}}{\le} q(e^{\tilde{\epsilon}} - 1)\left(qA'_{11} + q_1(1-q_2)A_{10} + (1-q_1)A_0)\right) + \left(qA'_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0\right) + q\tilde{\delta} \\
&= \left(1 + q\left(e^{\tilde{\epsilon}} - 1\right)\right)\left(qA'_{11} + q_1\left(1-q_2\right)A_{10} + \left(1-q_1\right)A_0\right) + q\tilde{\delta} \\
&= e^{\ln(1+q(e^{\tilde{\epsilon}}-1))} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + q\tilde{\delta}.
\end{aligned}
$$

Here, (a) follows from $\min\{A'_{11}, A_{10}, A_0\} \le A'_{11}$, and (b) follows from the fact that minimum is upper-bounded by the convex combination. By substituting the value of $\tilde{\epsilon}$ from (19) and using $ks = qn$, we get that for $\epsilon_0 = \mathcal{O}\left(1\right)$, we have $\overline{\epsilon} = \mathcal{O}\left(\epsilon_0 \sqrt{\frac{q \log\left(1/\tilde{\delta}\right)}{n}}\right)$.

Proofs of (55) and (56)

As we see below, the proof of (55) is similar to the proof of (23), as the bipartite graphs in both the proofs have similar structure. However, the proof of (56) is different from these proofs (and also from the proof of (24)), as we prove it using a two stage bipartite graph, where the bipartite graph in the second stage has similar structure as the one in the proof of (23), but the bipartite graph in the first stage is irregular (i.e., different vertices in one side of the vertex set have different degrees), which requires a careful degree analysis.

Proof of (55)**.** Fix any $\mathcal{T}_1, \ldots, \mathcal{T}_m \in \binom{[r]}{s}$ such that $1 \in \mathcal{T}_1$, i.e., $d_{11} \in \mathcal{D}^{\mathcal{T}_1}$. For these fixed subsets, consider the following bipartite graph $G = (V_1 \cup V_2, E)$, where the left vertex set V_1 has $\binom{ms-1}{ks-1}$ vertices, one for each configuration of $\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \ldots, \mathcal{D}^{\mathcal{T}_m}\}$ such that $|\mathcal{D}^{\overline{\mathcal{T}}}| = ks$ and $d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}$, the right vertex set V_2 has $\binom{ms-1}{ks}$

vertices, one for each configuration of $\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}$ such that $|\mathcal{D}^{\overline{\mathcal{T}}}| = ks$ and $d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}$, and the edge set E contains all the edges between neighboring vertices, i.e., if $(\boldsymbol{u}, \boldsymbol{v}) \in V_1 \times V_2$ is such that $\boldsymbol{u}$ and $\boldsymbol{v}$ differ in only one element, then $(\boldsymbol{u}, \boldsymbol{v}) \in E$. Observe that each vertex of V_1 has $(ms - ks)$ neighbors in V_2 – the neighbors of any $\mathcal{D}^{\overline{\mathcal{T}}} \in V_1$ will be $\{(\mathcal{D}^{\overline{\mathcal{T}}} \setminus \{d_{11}\}) \cup \{d\} : d \in \mathcal{D}^{\overline{\mathcal{T}}} \setminus \{d_{11}\}\} \in V_2$. Similarly, each vertex of V_2 has ks neighbors in V_1 – the neighbors of any $\mathcal{D}^{\overline{\mathcal{T}}} \in V_2$ will be $\{(\mathcal{D}^{\overline{\mathcal{T}}} \setminus \{d\}) \cup \{d_{11}\} : d \in \mathcal{D}^{\overline{\mathcal{T}}}\} \in V_1$.

Consider an arbitrary $(\boldsymbol{u}, \boldsymbol{v}) \in E$. Since the mechanism $\mathcal{Z}$ is $(\tilde{\epsilon}, \tilde{\delta})$-DP, we have

$$\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}} = \boldsymbol{u}\right] \leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}} = \boldsymbol{v}\right] + \tilde{\delta}. \tag{57}$$

Now we are ready to prove (55).

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
&= \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}] \\
&= \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \sum_{\substack{\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}: \\ |\mathcal{D}^{\overline{\mathcal{T}}}| = ks, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}}} \frac{1}{\binom{ms-1}{ks-1}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}] \\
&= \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \frac{1}{(ms - ks)\binom{ms-1}{ks-1}} \sum_{\substack{\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}: \\ |\mathcal{D}^{\overline{\mathcal{T}}}| = ks, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}}} (ms - ks) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}] \\
&\overset{\text{(a)}}{=} \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \frac{1}{ks\binom{ms-1}{ks}} \sum_{\substack{\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}: \\ |\mathcal{D}^{\overline{\mathcal{T}}}| = ks, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}}} (ms - ks) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}] \\
&\overset{\text{(b)}}{\leq} \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \frac{1}{ks\binom{ms-1}{ks}} \sum_{\substack{\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}: \\ |\mathcal{D}^{\overline{\mathcal{T}}}| = ks, d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}}} ks \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}] + \tilde{\delta}\right) \\
&= \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \dots, \mathcal{T}_m | 1 \in \mathcal{T}_1] \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \dots, \mathcal{T}_m, d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}] + \tilde{\delta}\right) \\
&= e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta} \\
&= e^{\tilde{\epsilon}} A_{10} + \tilde{\delta}.
\end{aligned}$$

Here, (a) uses the identity $(ms - ks)\binom{ms-1}{ks-1} = ks\binom{ms-1}{ks}$ and (b) follows from (57) together with the fact that there are $(ms - ks)\binom{ms-1}{ks-1} = ks\binom{ms-1}{ks}$ edges in the bipartite graph $G = (V_1 \cup V_2, E)$, where degree of vertices in V_1 is $(ms - ks)$ and degree of vertices in V_2 is ks.

Proof of (56). Fix any $\mathcal{T}_1, \dots, \mathcal{T}_m \in \binom{[r]}{s}$ such that $1 \in \mathcal{T}_1$, i.e., $d_{11} \in \mathcal{D}^{\mathcal{T}_1}$. Let $\mathcal{T}_1' \in \binom{[r]}{s}$ be such that $1 \notin \mathcal{T}_1'$ and $\mathcal{D}^{\mathcal{T}_1}$ & $\mathcal{D}^{\mathcal{T}_1'}$ are neighbors. First we show that

$$\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \dots, \mathcal{T}_m\right] + \tilde{\delta} \tag{58}$$

Note that, in (58), $\mathcal{T}_1', \mathcal{T}_1, \dots, \mathcal{T}_m \in \binom{[r]}{s}$ are fixed subsets such that $1 \in \mathcal{T}_1$, $1 \notin \mathcal{T}_1'$, and $\mathcal{D}^{\mathcal{T}_1}$ & $\mathcal{D}^{\mathcal{T}_1'}$ are neighbors. Since $\mathcal{D}^{\mathcal{T}_1}$ and $\mathcal{D}^{\mathcal{T}_1'}$ are neighbors, we have $|\mathcal{D}^{\mathcal{T}_1} \cap \mathcal{D}^{\mathcal{T}_1'}| = s - 1$. Let d_{1i^*} be such that $\{d_{1i^*}\} = \mathcal{D}^{\mathcal{T}_1'} \setminus \mathcal{D}^{\mathcal{T}_1}$. Note that $\{d_{11}\} = \mathcal{D}^{\mathcal{T}_1} \setminus \mathcal{D}^{\mathcal{T}_1'}$.

In order to show (58), construct the following bipartite graph $G_1 = (V_{11} \cup V_{12}, E_1)$, where the left vertex set V_{11} has $\binom{ms-1}{ks-1}$ vertices, one for each configuration of $\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1}, \dots, \mathcal{D}^{\mathcal{T}_m}\}$ such that $|\mathcal{D}^{\overline{\mathcal{T}}}| = ks$ and $d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}$, the right vertex set V_{12} has $\binom{ms}{ks}$ vertices, one for each configuration of $\mathcal{D}^{\overline{\mathcal{T}}} \subseteq \{\mathcal{D}^{\mathcal{T}_1'}, \mathcal{D}^{\mathcal{T}_2} \dots, \mathcal{D}^{\mathcal{T}_m}\}$ such that $|\mathcal{D}^{\overline{\mathcal{T}}}| = ks$ (note that $d_{11} \notin \mathcal{D}^{\overline{\mathcal{T}}}$ because $d_{11} \notin \mathcal{D}^{\mathcal{T}_1'}$), and the edge set E_1 contains all the edges between neighboring vertices, i.e., if $(\boldsymbol{u}, \boldsymbol{v}) \in V_{11} \times V_{12}$ is such that $\boldsymbol{u}$ and $\boldsymbol{v}$ differ in only one element, then $(\boldsymbol{u}, \boldsymbol{v}) \in E_1$.

Observe that each vertex of V_{11} has $(ms-ks+1)$ neighbors in V_{12} – the neighbors of any $\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}$ are $\{(\mathcal{D}^{\overline{\mathcal{T}}} \setminus \{d_{11}\}) \cup \{d\} : d \in \{\mathcal{D}^{\mathcal{T}_1'}, \mathcal{D}^{\mathcal{T}_2}, \ldots, \mathcal{D}^{\mathcal{T}_m}\} \setminus \mathcal{D}^{\overline{\mathcal{T}}}\} \in V_{12}$. Note that $\{\mathcal{D}^{\mathcal{T}_1'}, \mathcal{D}^{\mathcal{T}_2}, \ldots, \mathcal{D}^{\mathcal{T}_m}\} \setminus \mathcal{D}^{\overline{\mathcal{T}}}$ has $(ms-ks+1)$ elements.

In contrast, vertices in V_{12} have irregular degrees. To see this, we partition V_{12} in two parts $V_{12} = V_{12}' \uplus V_{12}''$ (here $\uplus$ denotes disjoint union), where

$$\begin{aligned} V_{12}' &= \{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12} : d_{1i^*} \in \mathcal{D}^{\overline{\mathcal{T}}}\} \\ V_{12}'' &= \{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12} : d_{1i^*} \notin \mathcal{D}^{\overline{\mathcal{T}}}\}. \end{aligned}$$

Note that $|V_{12}'| = |V_{11}| = \binom{ms-1}{ks-1}$ and $|V_{12}''| = \binom{ms}{ks} - |V_{12}'| = \left(\frac{ms}{ks} - 1\right)\binom{ms-1}{ks-1}$. The vertices in V_{12} have the following degrees:

- Each vertex of V_{12}' has exactly one neighbor in V_{11} (by replacing d_{1i^*} by d_{11}) and vice-versa. This implies (using $(\tilde{\epsilon}, \tilde{\delta})$-DP of the mechanism $\mathcal{Z}$):

$$\sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \leq \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}'} \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta}\right) \tag{59}$$

- Each vertex of V_{12}'' has ks neighbors in V_{11} – the neighbors of any $\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}''$ are $\{(\mathcal{D}^{\overline{\mathcal{T}}} \setminus \{d\}) \cup \{d_{11}\} : d \in \mathcal{D}^{\overline{\mathcal{T}}}\} \in V_{11}$. It can also be verified that each vertex of V_{11} has $(ms-ks)$ neighbors in V_{12}''. This implies

$$\sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} (ms-ks) \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \leq \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}''} ks \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta}\right) \tag{60}$$

Note that $(ms-ks+1)|V_{11}| = |V_{12}'| + ks|V_{12}''|$.

Now we can prove (58).

$$\begin{aligned}
&\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] = \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
&= \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} \frac{1}{\binom{ms-1}{ks-1}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
&= \frac{\frac{ms}{ks}}{\binom{ms}{ks}} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
&= \frac{1}{\binom{ms}{ks}} \left(\sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \frac{1}{ks} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{11}} (ms-ks) \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \right) \\
&\overset{\text{(a)}}{\leq} \frac{1}{\binom{ms}{ks}} \left(\sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}'} \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta}\right) + \frac{1}{ks} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}''} ks \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta}\right) \right) \\
&= \frac{1}{\binom{ms}{ks}} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}} \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] + \tilde{\delta}\right) \\
&= e^{\tilde{\epsilon}} \left(\frac{1}{\binom{ms}{ks}} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m, \mathcal{D}^{\overline{\mathcal{T}}}\right] \right) + \frac{1}{\binom{ms}{ks}} \sum_{\mathcal{D}^{\overline{\mathcal{T}}} \in V_{12}} \tilde{\delta} \\
&= e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m\right] + \tilde{\delta},
\end{aligned}$$

where (a) follows from (59) and (60).

Now consider another bipartite graph $G_2 = (V_{21} \cup V_{22}, E_2)$, where the left vertex set V_{21} has $\binom{r-1}{s-1}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \in \mathcal{T}_1$, the right vertex set V_{22} has $\binom{r-1}{s}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [r]$ such that $|\mathcal{T}_1| = s, 1 \notin \mathcal{T}_1$, and the edge set E_2 contains all the edges between neighboring vertices, i.e., if $(\boldsymbol{u}, \boldsymbol{v}) \in V_{21} \times V_{22}$ is such that $\boldsymbol{u}$ and $\boldsymbol{v}$ differ in only one element, then $(\boldsymbol{u}, \boldsymbol{v}) \in E_2$. Observe that each vertex of V_{21} has $(r-s)$ neighbors in V_{22} – the neighbors of $\mathcal{T}_1 \in V_{21}$ will be $\{(\mathcal{T}_1 \setminus \{1\}) \cup \{i\} : i \in [m] \setminus \mathcal{T}_1\} \in V_{22}$. Similarly, each vertex of V_{22} has s neighbors in V_{21} – the neighbors of $\mathcal{T}_1 \in V_{22}$ will be $\{(\mathcal{T}_1 \setminus \{i\}) \cup \{1\} : i \in \mathcal{T}_1\} \in V_1$.

Fix any $\mathcal{T}_2, \ldots, \mathcal{T}_m \in \binom{[r]}{s}$. For these fixed subsets $\mathcal{T}_2, \ldots, \mathcal{T}_m \in \binom{[r]}{s}$ and any $(\mathcal{T}_1, \mathcal{T}_1') \in E_2$ (note that $1 \in \mathcal{T}_1$ and $1 \notin \mathcal{T}_1'$), we have from (58) that $\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \leq e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m\right] + \tilde{\delta}$. Taking summation over all vertices and (taking into account their degrees), we have

$$\sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} (r-s) \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \leq \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1} s \left(e^{\tilde{\epsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1', \mathcal{T}_2, \ldots, \mathcal{T}_m\right] + \tilde{\delta}\right) \tag{61}$$

Now we are ready to prove (56).

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} \mid d_{11} \in \mathcal{D}^{\mathcal{T}_1} \text{ and } d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}\right] \\
&= \sum_{\mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}} \Pr[\mathcal{T}_2, \ldots, \mathcal{T}_m] \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} \Pr[\mathcal{T}_1 | 1 \in \mathcal{T}_1] \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}] \\
&= \sum_{\mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}} \Pr[\mathcal{T}_2, \ldots, \mathcal{T}_m] \frac{1}{(r-s)\binom{r-1}{s-1}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} (r-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}] \\
&\overset{\text{(b)}}{=} \sum_{\mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}} \Pr[\mathcal{T}_2, \ldots, \mathcal{T}_m] \frac{1}{s\binom{r-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \in \mathcal{T}_1} (r-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m, d_{11} \in \mathcal{D}^{\overline{\mathcal{T}}}] \\
&\overset{\text{(c)}}{\leq} \sum_{\mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}} \Pr[\mathcal{T}_2, \ldots, \mathcal{T}_m] \frac{1}{s\binom{r-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1} s \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&= \sum_{\mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}} \Pr[\mathcal{T}_2, \ldots, \mathcal{T}_m] \sum_{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1} \Pr[\mathcal{T}_1 | 1 \notin \mathcal{T}_1] \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&= \sum_{\substack{\mathcal{T}_1 \in \binom{[r]}{s} : 1 \notin \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[r]}{s} \text{ for } i \in [m] \setminus \{1\}}} \Pr[\mathcal{T}_1, \ldots, \mathcal{T}_m | 1 \notin \mathcal{T}_1] \left(e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}_1, \ldots, \mathcal{T}_m] + \tilde{\delta}\right) \\
&= e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}_1] + \tilde{\delta} \\
&\overset{\text{(d)}}{=} e^{\tilde{\epsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | d_{11} \notin \mathcal{D}^{\mathcal{T}_1}] + \tilde{\delta} \\
&= e^{\tilde{\epsilon}} A_0 + \tilde{\delta}
\end{aligned}$$

Here, (b) uses $(r-s)\binom{r-1}{s-1} = s\binom{r-1}{s}$, (c) follows from (61), and (d) uses the equivalence of $1 \notin \mathcal{T}_1$ and $d_{11} \notin \mathcal{D}^{\mathcal{T}_1}$. This completes the proof of Lemma 16.