UNIVERSITY OF CALIFORNIA

Los Angeles

Communication-Efficient and Private Distributed Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical & Computer Engineering

by

Antonious Mamdouh Girgis Bebawy

2023

ABSTRACT OF THE DISSERTATION

Communication-Efficient and Private Distributed Learning

by

Antonious Mamdouh Girgis Bebawy
Doctor of Philosophy in Electrical & Computer Engineering
University of California, Los Angeles, 2023
Professor Suhas N. Diggavi, Chair

We are currently facing a rapid growth of data contents originating from edge devices. These data resources offer significant potential for learning and extracting complex patterns in a range of distributed learning applications, such as healthcare, recommendation systems, and financial markets. However, the collection and processing of such extensive datasets through centralized learning procedures imposes potential challenges. As a result, there is a need for the development of distributed learning algorithms. Furthermore, This raises two principal challenges within the realm of distributed learning. The first challenge is to provide privacy guarantees for clients' data, as it may contain sensitive information that can be potentially mishandled. The second challenge involves addressing communication constraints, particularly in cases where clients are connected to a coordinator through wireless/band-limited networks.

In this thesis, our objective is to develop fundamental information-theoretic bounds and devise distributed learning algorithms with privacy and communication requirements while maintaining the overall utility performance. We consider three different adversary models for differential privacy: (1) central model, where the exists a trusted server applies a private

mechanism after collecting the raw data; (2) local model, where each client randomizes her own data before making it public; (3) shuffled model, where there exists a trusted shuffler that randomly permutes the randomized data before publishing them. The contributions of this thesis can be summarized as follows

- We propose communication-efficient algorithms for estimating the mean of bounded ℓ_p-norm vectors under privacy constraints in the local and shuffled models for $p \in [1, \infty]$. We also provide information-theoretic lower bounds showing that our algorithms have order-optimal privacy-communication-performance trade-offs. In addition, we present a generic algorithm for distributed mean estimation under user-level privacy constraints when each client has more than one data point.

- We propose a distributed optimization algorithm to solve the empirical risk minimization(ERM) problem with communication and privacy guarantees and analyze its communication-privacy-convergence trade-offs. We extend our distributed algorithm for a client-self-sampling scheme that fits federated learning frameworks, where each client independently decides to contribute at each round based on tossing a biased coin. We also propose a user-level private algorithm for personalized federated learning.

- We characterize the rényi differential privacy (RDP) of the shuffled model by proposing closed-form upper and lower bounds for general local randomized mechanisms. RDP is a useful privacy notion that enables a much tighter composition for interactive mechanisms. Furthermore, we characterize the RDP of the subsampled shuffled model that combines privacy amplification via shuffling and amplification by subsampling.

- We propose differentially private algorithms for the problem of stochastic linear bandits in the central, local, and shuffled models. Our algorithms achieve almost the same regret as the optimal non-private algorithms in the central and shuffled models, which means we get privacy for free.

- We study successive refinement of privacy by providing hierarchical access to the raw data with different privacy levels. We provide (order-wise) tight characterizations of privacy-utility-randomness trade-offs in several cases of discrete distribution estimation.

The dissertation of Antonious Mamdouh Girgis Bebawy is approved.

Lin Yang

Lieven Vandenberghe

Christina Panagio Fragouli

Suhas N. Diggavi, Committee Chair

University of California, Los Angeles

2023

To my beloved family . . .

whose support and guidance extend beyond measure

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

VITA

2014	B.S. Electrical and Electronic Engineering, Cairo University, Egypt
2014-2018	Research Assistant, Wireless Intelligent Network Center (WINC), Nile University, Egypt
2014-2018	Masters fellowship by School of Communications and Information Technology, Nile University, Egypt.
2018	M.S. Electrical and Computer Engineering, Nile University, Egypt
2021	Teaching Assistant, ECE Department, UCLA
2022	Amazon Ph.D. Fellowship.
2022	Machine Learning Research Engineer, Google.
2023	Machine Learning Student Researcher, Google.

SELECTED PUBLICATIONS

Antonious M. Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh "Shuffled Model of Federated Learning: Privacy, Communication and Accuracy Trade-offs"*IEEE Transactions on Journal on Selected Areas in Information Theory (JSAIT), vol.2, no.1, pp:464–478, 2021.*

Antonious M. Girgis, Deepesh Data, Kamalika Chaudhuri, Christina Fragouli and Suhas

Diggavi "Successive Refinement of Privacy" *IEEE Transactions on Journal on Selected Areas in Information Theory (JSAIT), vol.1, no.3, pp:745–759, 2020.*

Osama A Hanna[1], **Antonious M. Girgis**[1], Christina Fragouli, and Suhas Diggavi "Differentially Private Stochastic Linear Bandits:(Almost) for Free" *arXiv preprint arXiv:2207.03445, 2022*

Antonious M. Girgis, Deepesh Data, and Suhas Diggavi "Distributed User-Level Private Mean Estimation" *IEEE International Symposium On Information Theory (ISIT),2022.*

Kaan Ozkara[1], **Antonious M. Girgis**[1], Deepesh Data, and Suhas Diggavi "A Statistical Framework for Personalized Federated Learning and Estimation: Theory, Algorithms, and Privacy" *in International Conference on Learning Representations (ICLR), 2023*

Antonious M. Girgis, Deepesh Data, and Suhas Diggavi "Renyi Differential Privacy of the Subsampled Shuffle Model in Distributed Learning" *in Neural Information Processing Systems (NeurIPS), 2021*

Antonious M. Girgis, Deepesh Data, Suhas Diggavi, Ananda Theertha Suresh, and Peter Kairouz "On the Renyi Differential Privacy of the Shuffle Model" *ACM Symposium on Computer and Communications Security (CCS), pp:2321–2341, 2021*–**Best paper award**

Antonious M. Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh "Shuffled Model of Differential Privacy in Federated Learning" *In International Conference on Artificial Intelligence and Statistics (AISTATS), pp:2521–2529, 2021*

[1] Equal contribution.

CHAPTER 1

Introduction

The exponential growth of data at edge devices has reached unprecedented levels, transforming the way we perceive and harness information. The massive volume and variety of data generated daily can be exploited for several learning applications, however, it poses substantial challenges for traditional, centralized algorithms. The advent of massive data, often referred to as "big data," has surpassed the capacity of conventional systems, making the central computing paradigms struggle to keep pace with data collection, processing, and computations. Consider the scenario of a global social media platform, where billions of users generate vast amounts of data through posts, likes, and interactions in real time. Traditional algorithms and systems struggle when tasked with processing and analyzing such massive datasets within reasonable timeframes. Another scenario is e-commerce: consider the challenges faced by e-commerce giants handling millions of transactions daily, customer preferences, and purchase histories. The analysis of these datasets demands not only sophisticated algorithms but also a scalable infrastructure capable of parallel processing to expedite computations. Centralized systems prove inadequate in meeting these demands, necessitating a paradigm shift toward distributed systems.

1.1 Communication-Privacy-Utility Trade-offs

Two major challenges appear in distributed systems. The first of these challenges revolves around the imperative to ensure the privacy and security of the data owned by individual clients. In the era of massive data, where information is not just abundant but often sensitive, the potential mishandling of personal or confidential data is a significant concern. Distributed learning systems operate in a decentralized fashion, with data residing across multiple nodes or devices. Consider, for instance, a scenario in healthcare where patient data is dispersed across various medical institutions. Preserving the privacy of individual health records is non-negotiable, yet collaborative learning is crucial for advancing medical research and improving treatment outcomes. Distributed learning systems in this context must navigate the intricacies of data privacy to enable collaborative research without compromising sensitive patient information. Similarly, in the financial sector, distributed learning finds applications in fraud detection and risk assessment. Financial institutions collaborate to enhance their models, leveraging insights from diverse datasets. However, stringent regulations and the need to protect customer financial information underscore the necessity for privacy-preserving mechanisms in distributed learning algorithms.

The second challenge in the realm of distributed learning is communication constraints, especially in scenarios where clients are connected to a central coordinator through wireless or band-limited networks. The nature of these networks introduces latency, bandwidth limitations, and potential connectivity issues that become a bottleneck in the exchange of information between the distributed nodes. Overcoming these communication constraints becomes crucial for the smooth functioning of distributed learning systems. Consider a scenario of Federated Learning (FL), where mobile devices contribute to a high dimensional model but are constrained by slow and unstable internet connections. Addressing these challenges not only requires innovative algorithmic approaches but also a deeper understanding of the trade-offs between privacy, communication, and performance in distributed learning. In this thesis, we study the fundamental limits of communication-privacy-utility trade-offs for several distributed learning approaches.

Differential privacy [DMN06] – a cryptographically motivated notion of privacy – has recently emerged as the gold standard in privacy-preserving data analysis. Privacy is provided by guaranteeing that the participation of a single person in a dataset does not change the probability of any outcome by much. In this thesis, we consider three adversarial models of differential privacy. In the *central DP model*, we assume that there exists a trusted server that collects the entire raw data and applies a private mechanism, where the adversary has only access to the output of the private mechanism. To accommodate the privacy of *locally* held data, a more appropriate notion is that of local differential privacy (LDP) [KLN11, DWJ13]. In the *local DP model*, each (distributed) client holding local data, individually randomizes its interactions with the (untrusted) server, where the adversary has access to the randomized report of each client. Recently, such LDP mechanisms have been deployed by companies such as Google [EPK14], Apple [Gre16], and Microsoft [DKY17]. However, LDP mechanisms suffer from poor performance in comparison with the centralized DP mechanisms, making their applicability limited [DWJ13,KLN11,KBR16]. To address this, a new privacy framework using anonymization has been proposed in the so-called *shuffled model* [EFM19, CSU19, BBG19d]. In the *shuffled DP model*, each client sends her randomized interaction message to a secure shuffler that randomly permutes all the received messages before forwarding them to the server[1], where the adversary has only access to the output of the secure shuffler. This shuffled model enables significantly better privacy-utility performance compared to the local DP model [EFM19, CSU19, BBG19d].

1.2 Outline and Contributions

The goal of this thesis is to study communication-privacy-utility trade-offs in several distributed learning structures. In particular, the contributions of this thesis can be summarized as follows:

In Chapter 3, we study the distributed mean estimation under privacy and communication constraints in the local DP model and the shuffled model for bounded ℓ_p-norm vectors, where $p \in [1, \infty]$. We propose an achievable scheme based on Hadamard transformation

[1] Such a shuffling can be enabled through anonymization techniques [BEM17, EFM19, EFM20a].

and a lower bound on the minimax risk for estimating bounded ℓ_1-norm vectors under LDP constraints. Next, we propose an achievable scheme for estimating binary vectors, bounded ℓ_∞-norm vectors, and bounded ℓ_2-norm vectors. Our proposed schemes are based on compressing the real vectors using finite-bit binary representations and coordinate sampling. Then, we privatize the binary vectors using non-uniform privacy allocation. Furthermore, we present a lower bound on the minimax risk of estimating the mean of bounded ℓ_2-norm vectors under LDP constraints. We show that our proposed schemes are order optimal and match the lower bounds for both the local DP model and the multi-message shuffled model. We also distributed mean estimation under user-level local differential privacy when each client has multiple vectors, where the goal is to protect the entire local datasets. The results of this chapter are based on our work in [GDD21d, GD23, GDD22].

In Chapter 4, we study the empirical risk minimization (ERM) problem in the federated learning framework under privacy and communication constraints in the shuffled model. We propose CLDP-SGD algorithm, where at each round a subset of clients are randomly chosen to contribute. Each of the sampled clients runs stochastic gradient descent (SGD) by sampling her local dataset and sends compressed and private gradients to a secure shuffler. The server aggregates the received updates from the shuffler and takes a descent step. We analyze the communication-privacy-convergence of the proposed CLDP-SGD algorithm. We also extend our work to client-self sampling, where each client decides to contribute at each round by tossing an independent biased coin. The client-self sampling raises challenges in analyzing the total privacy and the conference of the private algorithm, as the number of clients contributed at each round is a random variable. Next, we propose a user-level LDP mechanism for personalized federated learning to learn an individual model for each client via private collaboration. The results of this chapter are based on our work in [GDD21a, GDD21b, OGD22].

In Chapter 5, we characterize the Rényi differential privacy (RDP) of the shuffled model by proposing upper and lower bounds for general LDP mechanisms. RDP is a useful privacy notion that enables a much tighter composition for interactive mechanisms. Furthermore, we characterize the RDP of the subsampled shuffled model that combines privacy amplification via shuffling and amplification by subsampling. To achieve these results, we propose a novel analysis technique by reducing any general neighboring datasets to special case neighboring datasets that can be analyzed in a closed-form solution. We use our RDP bounds to give tighter privacy analysis for our proposed federated learning algorithms in Chapter 4. The results of this chapter are based on our work in [GDD21e, GDD21c].

In Chapter 6, we study stochastic linear bandits under privacy constraints. Stochastic linear bandits offer a sequential decision framework where a learner interacts with an environment over rounds, and decides what is the optimal (from a potentially infinite set) action to play so as to achieve the best possible reward. This model has been widely adopted both in theory but also in a number of applications, including recommendation systems, healthcare, online education, and resource allocation [MGP15, BRC17, RYW18, BR19]. We propose differentially private algorithms for stochastic linear bandits in the central, local, and shuffled models. In the central and shuffled models, we achieve almost the same regret as the optimal non-private algorithms, which means we get privacy for free. The results of this chapter are based on joint work with Osama Hanna in [HGF22].

In Chapter 7, we examine a novel question: how much randomness is needed to achieve local differential privacy (LDP)? A motivating scenario is providing *multiple levels of privacy* to multiple analysts, either for distribution or for heavy hitter estimation, using the *same* (randomized) output. We call this setting *successive refinement of privacy*, as it provides hierarchical access to the raw data with different privacy levels. For example, the same randomized output could enable one analyst to reconstruct the input, while another can only estimate the distribution subject to LDP requirements. This extends the classical Shannon (wiretap) security setting to local differential privacy. We provide (order-wise) tight characterizations of privacy-utility-randomness trade-offs in several cases for distribution estimation, including the standard LDP setting under a randomness constraint. We also provide a non-trivial privacy mechanism for multi-level privacy. Furthermore, we show that we cannot reuse random keys over time while preserving privacy of each user. The results of this chapter are based on our work in [GDC20].

In Chapter 8, we draw conclusions from our work and delve into a prospective discussion for future exploration within the scope of our work.

CHAPTER 2

Background

In this chapter, we set up backgrounds and state some preliminary definitions that will be used throughout the thesis. We state the formal definitions of (central) differential privacy (DP) in Section 2.1.1. We state the formal definitions of the local differential privacy (LDP) in Section 2.2. We present the shuffled model of differential privacy in Section 2.3. We also present the binary randomized response mechanism in Section 2.4.

2.1 Privacy Definitions

Our goal is to understand how to measure privacy for a specific mechanism. Let $\mathcal{D} = (d_1, \ldots, d_n)$ be a dataset collected from n clients, where $d_i \in \mathcal{X}$ for $i \in [n]$. Let Alice's data d_1 be in the dataset $\mathcal{D}$. Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be a mechanism that is a function of the clients' dataset $\mathcal{D}$. An intuitive definition of privacy is to say that the mechanism $\mathcal{M}$ is private if an adversary cannot learn much from observing the output of the mechanism $\mathcal{M}$ beyond whatever side information it has access to. In other words, the adversary cannot distinguish whether the mechanism uses dataset $\mathcal{D}$ or $\mathcal{D}' = (d'_1, d_2, \ldots, d_n)$ from observing the output of the mechanism $\mathcal{M}$, where $\mathcal{D}'$ is a neighboring dataset by replacing Alice's data d_1 with arbitrary data $d'_1 \in \mathcal{X}$.

The traditional privacy approach is obtained via anonymity [Swe02]. However, such deterministic approaches for privacy are vulnerable to recovery attacks using side-information.

For example, Sweeney in [Swe97] was able to re-identify individuals from anonymized medical records by linking them to public voting registration records. Also, Narayanan et al. in [NS08] proposed a de-anonymization attack that can recover the anonymized *Netflix* prize dataset (containing more than $500,000$ anonymous movie ratings) using the public *IMDb* dataset as side-information. There are more examples of de-anonymization attacks of deterministic mechanisms including *Kaggle IJCNN 2011* social network challenge [NSR11] and AOL search log release in 2006 [BZH06]. As a result, we expect that the mechanism $\mathcal{M}$ is randomized to give more protection to any input dataset $\mathcal{D} \in \mathcal{X}^n$. Thus, for given dataset $\mathcal{D}$, the mechanism $\mathcal{M}$ induces a distribution on the output set $\mathcal{Y}$.

Let d be a measure distance between distributions. Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ denote a dataset comprising n points from $\mathcal{X}$. We say that two datasets $\mathcal{D} = \{d_1, \ldots, d_n\}$ and $\mathcal{D}' = \{d'_1, \ldots, d'_n\}$ are neighboring (and denoted by $\mathcal{D} \sim \mathcal{D}'$) if they differ in one data point, i.e., there exists an $i \in [n]$ such that $d_i \neq d'_i$ and for every $j \in [n], j \neq i$, we have $d_j = d'_j$. We can mathematically measure the privacy of a randomized mechanism by measuring how close the distribution $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ using the measure distance d. When the distributions $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ are close, an adversary cannot distinguish whether the input dataset is $\mathcal{D}$ or $\mathcal{D}'$, and hence, it preserves privacy. Hence, we can define different privacy notions using different distance measures d, e.g., KL divergence, total variation distance, and f-divergence.

2.1.1 Differential Privacy (DP)

Differential privacy (DP) [DMN06] has become the *de facto* standard for measuring privacy guarantees. DP is mainly based on bounding the *max divergence* between the worst-case neighboring datasets.

Definition 2.1.1. (Rényi divergence [EH14]) Let P and Q be probability distributions on the same space $\mathcal{Y}$. For $\alpha \in (1, \infty]$, the Rény divergence of a probability distribution P from a distribution Q is given by:

$$\begin{aligned} D_\alpha\left(P||Q\right) &= \frac{1}{\alpha-1}\log\mathbb{E}_{x\sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^\alpha\right] \qquad \text{for } \alpha\in(1,\infty), \\ D_\infty\left(P||Q\right) &= \lim_{\alpha\to\infty} D_\alpha\left(P||Q\right) = \sup_{x\in\mathcal{Y}}\log\left(\frac{P(x)}{Q(x)}\right) \qquad \text{for } \alpha=\infty, \end{aligned} \tag{2.1}$$

where we call it *max divergence* when $\alpha=\infty$.

Definition 2.1.2 (Central Differential Privacy - DP [DMN06, DR14])**.** For $\varepsilon,\delta\geq 0$, 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 $\mathcal{D}\sim\mathcal{D}'\in\mathcal{X}^n$ and every subset $\mathcal{S}\subseteq\mathcal{Y}$, we have

$$\Pr\left[\mathcal{M}(\mathcal{D})\in\mathcal{S}\right]\leq e^\varepsilon\Pr\left[\mathcal{M}(\mathcal{D}')\in\mathcal{S}\right]+\delta. \tag{2.2}$$

When $\delta=0$, we call it pure DP (in short, ε-DP).

Observe that DP guarantees that the distribution on the output of the mechanism $\mathcal{M}$ does not change much by replacing a single data point from the entire dataset. Here, ε captures the privacy level, the lower the ϵ_0, the higher the privacy. δ can be seen as the failure probability of satisfying privacy that should be of order $\delta=o(\frac{1}{poly}(n))$. DP definition 2.1.2 can be written in the max divergence form as follows. The mechanism $\mathcal{M}$ is (ε,δ)-DP if and only if $D_\infty\left(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')\right)\leq\varepsilon$ for any pair of neighboring datasets $\mathcal{D}\sim\mathcal{D}'$ with probability $1-\delta$ [BS16].

Suppose we have a dataset $\mathcal{D}'=\{d_1,\ldots,d_n\}\in\mathcal{X}^n$ consisting of n elements from a universe $\mathcal{X}$. The subsampling operation $\text{samp}_{n,k}:\mathcal{X}^n\to\mathcal{X}^k$ takes a dataset $\mathcal{D}'\in\mathcal{X}^n$ as an input and selects uniformly at random a subset $\mathcal{D}''$ of $k\leq n$ elements from $\mathcal{D}'$. Note that each element of $\mathcal{D}'$ appears in $\mathcal{D}''$ with probability $q=\frac{k}{n}$. The following result states that the above subsampling procedure amplifies the privacy guarantees of a DP mechanism.

Lemma 2.1.1 (Amplification by Subsampling [KLN11, Ull17])**.** *Let $\mathcal{M}:\mathcal{X}^k\to\mathcal{V}$ be an (ε,δ)-DP mechanism. Then, the mechanism $\mathcal{M}':\mathcal{X}^n\to\mathcal{V}$ defined by $\mathcal{M}'=\mathcal{M}\circ\text{samp}_{n,k}$ is (ε',δ')-DP, where $\varepsilon'=\log(1+q(e^\varepsilon-1))$ and $\delta'=q\delta$ with $q=\frac{k}{n}$. In particular, when $\varepsilon<1$, $\mathcal{M}'$ is $(\mathcal{O}(q\varepsilon),q\delta)$-DP.*

Let $\mathcal{M}_1(\mathcal{D}), \ldots, \mathcal{M}_T(\mathcal{D})$ be a sequence of T DP mechanisms. There are different composition theorems in literature to analyze the privacy guarantees of the composed mechanism $\mathcal{M}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D}), \ldots, \mathcal{M}_T(\mathcal{D}))$; the more accessing the data, the more information leakage. Dwork et al. [DRV10] and Kairouz et al. [KOV15] 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 [DR14].

Lemma 2.1.2 (Strong Composition [DR14, Theorem 3.20])**.** *Let $\mathcal{M}_1, \ldots, \mathcal{M}_T$ be $(\overline{\varepsilon}, \overline{\delta})$-DP mechanisms, where $\overline{\varepsilon}, \overline{\delta} \geq 0$. Then, for any $\delta' > 0$, the composed mechanism $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_T)$ is (ε, δ)-DP, where*

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

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

In interactive applications, we might need to access the data multiple times. For example, 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. Thus, the composition of DP mechanisms is useful for such applications.

2.1.2 Rényi Differential Privacy (RDP)

Now, we define Rényi differential privacy (RDP) by bounding the worst-case Rényi divergence. RDP is useful for composition. Furthermore, we can obtain approximate DP/pure DP from RDP.

Definition 2.1.3 ($(\alpha, \varepsilon(\alpha))$-RDP (Rényi Differential Privacy) [Mir17])**.** A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to have $\varepsilon(\alpha)$-Rényi differential privacy of order $\alpha \in (1, \infty)$ (in short, $(\alpha, \varepsilon(\alpha))$-RDP), if for any neighboring datasets $\mathcal{D} \sim \mathcal{D}' \in \mathcal{X}^n$, we have

$$D_\alpha(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) \leq \varepsilon(\alpha).$$

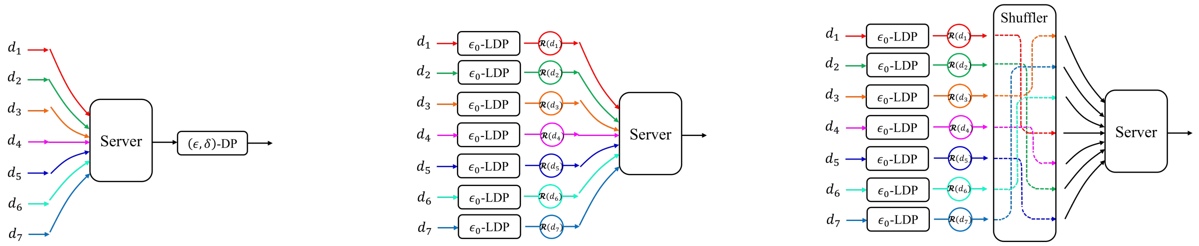


Figure 2.1: Adversary model for (a) central DP, (b) local DP, and (c) shuffled DP.

We use the following result for converting the RDP guarantees of a mechanism to its approximate DP guarantees.

Lemma 2.1.3 (From RDP to DP [CKS20, BBG20a])**.** *Suppose for $\alpha > 1$, a mechanism $\mathcal{M}$ is $(\alpha, \varepsilon(\alpha))$-RDP. Then, the mechanism $\mathcal{M}$ is (ε, δ)-DP, where ε, δ are defined below:*

$$\begin{aligned}
&\textit{For a given } \delta \in (0,1): \\
&\varepsilon = \min_{\alpha} \varepsilon(\alpha) + \frac{\log(1/\delta) + (\alpha-1)\log(1-1/\alpha) - \log(\alpha)}{\alpha - 1} \\
&\textit{For a given } \varepsilon > 0: \\
&\delta = \min_{\alpha} \frac{\exp\left((\alpha-1)(\varepsilon(\alpha) - \varepsilon)\right)}{\alpha - 1}\left(1 - \frac{1}{\alpha}\right)^{\alpha}.
\end{aligned}$$

The following result states that if we adaptively compose two RDP mechanisms in the same order, their privacy parameters add up in the resulting mechanism.

Lemma 2.1.4 (Adaptive composition of RDP [Mir17, Proposition 1])**.** *For any $\alpha > 1$, let $\mathcal{M}_1 : \mathcal{X} \to \mathcal{Y}_1$ be a $(\alpha, \varepsilon_1(\alpha))$-RDP mechanism and $\mathcal{M}_2 : \mathcal{Y}_1 \times \mathcal{X} \to \mathcal{Y}$ be a $(\alpha, \varepsilon_2(\alpha))$-RDP mechanism. Then, the mechanism defined by $(\mathcal{M}_1, \mathcal{M}_2)$ satisfies $(\alpha, \varepsilon_1(\alpha) + \varepsilon_2(\alpha))$-RDP.*

Now, we prove a useful lemma for conversion from RDP to approximate DP.

Lemma 2.1.5. *(Conversion from RDP to approximate DP) For given $\rho > 0$, let a mechanism $\mathcal{M}$ be $(\alpha, \alpha\rho)$-RDP. For any $\delta \in (0,1)$, the mechanism $\mathcal{M}$ satisfies (ε, δ)-DP, where ε is bounded by:*

$$\varepsilon \leq 3 \max\left\{\rho \log(1/\delta), \sqrt{\rho \log(1/\delta)}\right\}. \tag{2.3}$$

Algorithm 2.2.1 : Local Randomizer $\mathcal{R}_p^{2RR}$

```
**Public parameter:** $p$
**Input:** $b \in \{0,1\}$.
Sample $\gamma \leftarrow \mathrm{Ber}(p)$
**if** $\gamma == 0$ **then**
    $y = \frac{b-p}{1-2p}$
**else**
    $y = \frac{1-b-p}{1-2p}$
**Return:** The client sends $y$.
```

2.2 Local Differential Privacy (LDP)

In Section 2.1.1, we present the notion of central DP and RDP, where we assume that there exists a trusted server that can collect the raw dataset and apply the private mechanism $\mathcal{D}$ (see Figure 2.1). However, the existence of such a trusted server might be a strong assumption in distributed systems. In this section, we define local differential privacy (LDP) which is a powerful concept that allows clients to randomize their individual data points before sharing them.

Definition 2.2.1 (Local Differential Privacy - LDP [KLN11])**.** For $\varepsilon_0 \geq 0$, 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 $d, d' \in \mathcal{X}$, we have

$$\Pr[\mathcal{R}(d) \in \mathcal{S}] \leq e^{\varepsilon_0} \Pr[\mathcal{R}(d') \in \mathcal{S}], \qquad \forall \mathcal{S} \subset \mathcal{Y}. \tag{2.4}$$

Observe that the LDP definition in 2.2.1 is similar to the central DP definition in 2.1.2 with an input dataset $\mathcal{D}$ of a single data point.

2.3 Shuffled Model of Differential Privacy

LDP mechanisms suffer from poor performance in comparison with the centralized DP mechanisms, making their applicability limited [DWJ13, KLN11, KBR16]. To address this, a new privacy framework using anonymization has been proposed in the so-called *shuffled model* [EFM19, CSU19, BBG19d] that defined as follows. Consider a set of n clients, where client $i \in [n]$ has a data $d_i \in \mathcal{X}$. Let $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ be an ε_0-LDP mechanism. The i-th client applies $\mathcal{R}$ on her data d_i to get a private message $y_i = \mathcal{R}(d_i)$. There is a secure shuffler $\mathcal{H}_n : \mathcal{Y}^n \to \mathcal{Y}^n$ that receives the set of n messages $(y_1, \ldots, y_n)$ and generates the same set of messages in a uniformly random order, see Figure 2.1. The following lemma states that the shuffling amplifies the privacy of an LDP mechanism by a factor of $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.

Lemma 2.3.1 (Amplification by Shuffling)**.** *Let $\mathcal{R}$ be an ε_0-LDP mechanism. Then, the mechanism $\mathcal{M}(d_1, \ldots, d_n) := \mathcal{H}_n \circ (\mathcal{R}(d_1), \ldots, \mathcal{R}(d_n))$ satisfies (ε, δ)-differential privacy, where*

1. [BBG19d, Corollary 5.3.1]. *If $\varepsilon_0 \leq \frac{\log(n/\log(1/\delta))}{2}$, then for any $\delta > 0$, we have $\varepsilon = \mathcal{O}\left(\min\{\varepsilon_0, 1\} e^{\varepsilon_0} \sqrt{\frac{\log(1/\delta)}{n}}\right)$.*
2. [EFM19, Corollary 9]. *If $\varepsilon_0 < \frac{1}{2}$, then for any $\delta \in (0, \frac{1}{100})$ and $n \geq 1000$, we have $\varepsilon = 12\varepsilon_0 \sqrt{\frac{\log(1/\delta)}{n}}$.*

In Chapter 5, we characterize the RDP of the shuffled model $\mathcal{M}$. Furthermore, we characterize the RDP of the subsampled shuffled model, where we sample a subset of clients before shuffling.

2.4 Binary Randomized Response

In our algorithms, we use an unbiased version of the classical binary randomized response (*2RR*) [War65] whose input is a bit $b \in \{0, 1\}$ and the output is $\frac{b-p}{1-2p}$ w.p. $1-p$ and $\frac{1-b-p}{1-2p}$

w.p. p, where $p \in [0, 1/2)$ controls the privacy-utility trade-off (see Algorithm 2.2.1.

Theorem 2.4.1. *For any $p \in [0, 1/2)$, the 2RR is ε_0-LDP, where $\varepsilon_0 = \log\left(\frac{1-p}{p}\right)$. The output y of the 2RR mechanism is an unbiased estimate of b with bounded MSE:*

$$\mathsf{MSE}^{2RR} = \sup_{b\}\in\{0,1} \mathbb{E}\left[\|b-y\|_2^2\right] = \frac{p(1-p)}{(1-2p)^2}. \tag{2.5}$$

Theorem 2.4.1 gives an upper bound on the mean square error (MSE) of the *2RR* mechanism. For completeness, we present its proof in Section A.2. Next, we present the following lemma which is useful for bounding the privacy parameter (ε_0) of our mechanisms which depend on the binary randomized response. The proof is presented in Section A.3.

Lemma 2.4.1. *(Privacy parameter) For any $v > 0$, by setting $p = \frac{1}{2}\left(1 - \sqrt{\frac{v^2}{v^2+4}}\right)$, the 2RR mechanism with parameter p satisfies ε_0-LDP, where $\varepsilon_0 \leq v$.*

CHAPTER 3

Distributed Mean Estimation (DME)

In this chapter, we study distributed mean estimation (DME) under privacy and communication constraints in the local differential privacy (LDP) and multi-message shuffled privacy frameworks. The DME has wide applications in both federated learning and analytics. We propose communication-efficient algorithms for privately estimating the mean of bounded norm vectors. We also provide lower bounds for mean estimation with privacy and communication constraints for arbitrary ℓ_p-norm spaces. We show that our algorithms have order-optimal privacy-communication-performance trade-offs.

3.1 Introduction

We consider distributed mean estimation (DME) problem, where a set of clients are connected to a (untrusted) server to estimate the average of the clients' data. DME has wide applications including federated learning (FL), in which the central server estimates the mean of the local updates at each round (see e.g., FedAvg [MMR17]). However, DME faces two major challenges in the real world. (i) *Privacy:* the clients' data might contain sensitive information, and hence, each client wants to preserve privacy of her own local data. (ii) *Communication:* the connection between the server and clients might be over wireless/band-limited networks, and hence, the communication becomes a bottleneck for estimation. In this chapter, we focus on the local differential privacy (LDP) model. 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 [CSU19, BBG19d, BBG20b], 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. Thus, we consider the shuffled model of differential privacy (DP), where the clients are connected to the server through a secure shuffler that randomly permutes the clients' responses before passing them to the server [BEM17, EFM19, CSU19].

We propose mechanisms for DME of bounded binary, ℓ_1-norm, ℓ_∞-norm, and ℓ_2-norm vectors. In addition, we provide information-theoretic lower bounds for estimating the mean of bounded ℓ_p-norm vectors with privacy and communication constraints. We show that our proposed schemes achieve order-optimal privacy-communication-accuracy trade-offs for LDP and shuffled model privacy frameworks. The core technical idea of our proposed scheme consists of three stages as follows. The first stage is the *encoding* by transforming the vectors from the original space to the encoded space. For example, we use Hadamard matrix to encode the bounded ℓ_1-norm vectors and Kashin's representation or matrix rotation to encode the bounded ℓ_2-norm vectors. The second stage is the compression by appropriately sampling the coordinates and using finite number of bits to represent the real vectors. The third stage is the privacy. Our core idea for privacy is to apply *private-waterfilling* to privatize the binary bits, where we allocate unequal privacy for different binary vectors. We allocate increasing privacy with the order of bits, *i.e.,* lower privacy for most significant bits (MSBs); this gives better performance in terms of mean squared error (MSE), as MSBs are more important. This, combined with careful accounting for the composition using RDP, gives our privacy guarantees and performance.

Organization The remainder of this chapter is organized as follows. We present the problem setup in Section 3.2. We provide lower and upper bounds for mean estimation with privacy and communication constraints for bounded ℓ_1-norm vectors in Section 3.4. We present algorithms for privately estimating the mean of binary vectors in Section 3.3.

We provide lower and upper bounds for mean estimation with privacy and communication constraints for bounded ℓ_∞-norm vectors in Section 3.5 and for bounded ℓ_2-norm vectors in Section 3.6. We study DME under user-level privacy in Section 3.7. We give numerical results evaluating the performance of our proposed schemes in Section 3.8. Some proof are delegated to Appendix B.

3.2 Problem Formulation

We consider a set of n clients. Each client has a vector $\mathbf{x}_i \in \mathcal{X}$ for $i \in [n]$, where $\mathcal{X} \subset \mathbb{R}^d$ denotes a bounded subset of all possible inputs. For example, $\mathcal{X} \triangleq \mathbb{B}_2^d(r_2)$ denotes the d dimensional ball with radius r_2, i.e., each vector $\mathbf{x}_i$ satisfies $\|\mathbf{x}_i\|_2 \leq r_2$ for $i \in [n]$. Furthermore, each client has a communication budget of b-bits. The clients are connected to an (untrusted) server that wants to estimate the mean $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$. We consider two distributed privacy models.

LDP Model In the LDP model, we design two mechanisms as depicted in Figure 3.1: (i) Client-side mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ and (ii) Server aggregator $\mathcal{A} : \mathcal{Y}^n \to \mathbb{R}^d$. The local randomizer $\mathcal{R}$ takes an input $\mathbf{x}_i \in \mathcal{X}$ and generates a randomized output $\mathbf{y}_i \in \mathcal{Y}$. In LDP model, the local randomizer $\mathcal{R}$ satisfies privacy and communication constraints as follows. The output $\mathbf{y}_i = \mathcal{R}(\mathbf{x}_i)$ can be represented using only b-bits, as well as, it satisfies ε_0-LDP. Each client sends the output $\mathbf{y}_i$ directly to the server, which applies the aggregator $\mathcal{A}$ to estimate the mean $\hat{\mathbf{x}} = \mathcal{A}(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ such that the estimated mean $\hat{\mathbf{x}}$ is an unbiased estimate of the true mean $\overline{\mathbf{x}}$.

Shuffle Model The multi-message shuffled model is similar to the LDP model but with secure shufflers which anonymize the clients' identities to the server. Precisely, the L-message shuffled model consists of three parameters $(\mathcal{R}, \mathcal{S}, \mathcal{A})$ as depicted in Figure 3.1: (i) *Encode:* a local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}^L$, where the output $\mathbf{y}_i = \mathcal{R}(\mathbf{x}_i) = (\mathbf{y}_i^{(1)}, \ldots, \mathbf{y}_i^{(L)})$ consists of L messages. The local randomizer satisfies communication constraints in which the output $\mathbf{y}_i$ can be represented using b communication bits. (ii) *Shuffle:* a single shuffler $\mathcal{S}^{(k)} : \mathcal{Y}^n \rightarrow \mathcal{Y}^n$, for $k \in [L]$, generates a random permutation of the received n reports: $\mathbf{y}^{(k)} = \mathcal{S}^{(k)}\left(y_1^{(k)}, \ldots, y_n^{(k)}\right)$, where the kth message of each client is sent to the kth shuffler. (iii) *Analyze:* a server aggregator $\mathcal{A} : \left(\mathcal{Y}^L\right)^n \rightarrow \mathbb{R}^d$ is applied to the received messages from the L shufflers to estimate the mean $\hat{\mathbf{x}} = \mathcal{A}\left(\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)}\right)$. We say that the shuffled model is (ε, δ)-DP if the view of the output of the shufflers $\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)}\}$ satisfies (ε, δ)-DP.

Remark 3.2.1 (parallel shufflers vs single shuffler)**.** Observe that we describe the multi-message shuffle model using L independent shufflers, where each shuffler receives a single message from each client. We can also represent the multi-message shuffle model with a single shuffler that receives the total nL messages from all clients by indexing the messages of each client with a slight increase of the communication cost, see [BBG20b] for more details.

In the two privacy models, the performance of the estimator $\hat{\mathbf{x}}$ is measured by:

$$\mathsf{MSE} = \sup_{\{\mathbf{x}_i \in \mathcal{X}\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right], \tag{3.1}$$

where the expectation is taken over the randomness of the private mechanisms. Our goal is to design communication-efficient and private schemes to generate an unbiased estimate of $\overline{x}$ while minimizing the expected loss (3.1). We start by the DME of binary vectors, where $\mathcal{X} \triangleq \{0,1\}^d$. Then, we study the DME for bounded ℓ_∞-norm *i.e.,* $\|\mathbf{x}_i\|_\infty \leq r_\infty$ and bounded ℓ_2-norm vectors, where $\|\mathbf{x}_i\|_2 \leq r_2$.

3.3 DME for Binary Vectors

In this section, we consider binary vectors: $\mathbf{b}_i \in \{0,1\}^d$ for $i \in [n]$. The server wants to estimate the mean $\overline{\mathbf{b}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{b}_i$. The binary vector mechanism is the main building block of the next algorithms. This problem is a generalization to the scalar binary summation problem studied in [CSU19]. A straightforward solution is to apply the scalar mechanism

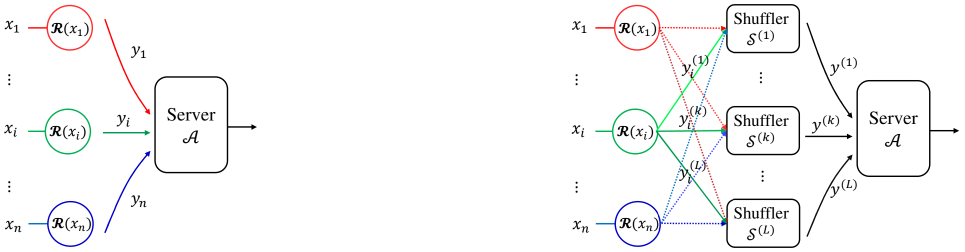


Figure 3.1: DME under privacy and communication constraints: (a) Local differential privacy (LDP) model of n clients. (b) An L-message shuffled (MMS) model of n clients

in [CSU19] per coordinate that requires d bits per client. Our private mechanisms require $\mathcal{O}\left(\min\{\varepsilon_0, d\}\right)$ and $\mathcal{O}\left(\min\{n\min\{\varepsilon^2, \varepsilon\}, d\}\right)$ communication bits per client in the LDP and shuffled models, respectively.

The client-side mechanism is presented in Algorithm 3.3.1, where the parameter s determines the communication budget per client and the parameter v determines the total privacy budget (see Theorem 3.3.1). For given $s \in \{1, \ldots, d\}$, each client splits the binary vector $\mathbf{b}_i$ into s sub-vectors, each with dimension $a = \lceil \frac{d}{s} \rceil$. Then, the client chooses uniformly at random one coordinate from each sub-vector and privatizes its bit using the binary randomized response (*2RR*) Algorithm 2.2.1 in Section 2.4. Observe that the output of Algorithm 3.3.1 can be represented as a sparse d-dimensional vector with only s non-zero coordinates.

When $s = d$, each client applies the *2RR* mechanism on each coordinate separately. On the other hand, when $s = 1$, each client chooses uniformly at random one coordinate and applies the *2RR* mechanism. Thus, we get trade-offs between privacy-communication and accuracy. The server aggregator $\mathcal{A}^{\text{Bin}}$ is simply aggregating the received randomized bits. We present the aggregator $\mathcal{A}^{\text{Bin}}$ in Algorithm 3.3.2.

Below, we state the bound on the MSE of the proposed mechanisms in the LDP and shuffled models. The proofs are presented in Section 3.3.1. Furthermore, we present RDP guarantees of our mechanisms for both LDP and shuffled models in the detailed proofs in Section 3.3.1.

Algorithm 3.3.1 : Local Randomizer $\mathcal{R}_{v,s}^{\text{Bin}}$

```
**Public parameter:** Privacy parameter $v$, and communication budget $s$.
**Input:** $\mathbf{b}_i \in \{0,1\}^d$.
If $\frac{d}{s}$ is not integer, add $(s\lceil \frac{d}{s} \rceil - d)$ dummy zeros to the binary vector $\mathbf{b}$. Let $a \leftarrow \frac{d}{s}$.
$p \leftarrow \frac{1}{2}\left(1 - \sqrt{\frac{v^2/s^2}{v^2/s^2+4}}\right)$
**for** $j \in [s]$ **do**
    Choose uniformly at random one coordinate $a_{ij} \leftarrow \mathsf{Unif}\left(\{(j-1)a+1,\ldots,ja\}\right)$.
    $y_{ij} \leftarrow a\mathcal{R}_p^{2RR}\left(\mathbf{b}_i[a_{ij}]\right)$
**Return:** The client sends $s$ messages $\mathcal{Y}_i \leftarrow \{(a_{i1}, y_{i1}), \ldots, (a_{is}, y_{is})\}$.
```

Theorem 3.3.1 (LDP model). *The output of the local mechanism $\mathcal{R}_{v,s}^{Bin}$ can be represented using $s\left(\log\left(\lceil d/s \rceil\right)+1\right)$-bits. By choosing $v = \varepsilon_0$, the mechanism $\mathcal{R}_{v,s}^{Bin}$ satisfies ε_0-LDP. Let $\hat{\mathbf{b}}$ be the output of the analyzer $\mathcal{A}^{Bin}$. The estimator $\hat{\mathbf{b}}$ is an unbiased estimate of $\overline{\mathbf{b}}$ with MSE:*

$$\mathsf{MSE}_{ldp}^{Bin} = \mathcal{O}\left(\frac{d^2}{n}\max\left\{\frac{1}{s}, \frac{s}{\varepsilon_0^2}\right\}\right). \tag{3.2}$$

Now, we move to the shuffled model, where we assume there exists a secure shuffler that randomly permutes the set of messages $\{\mathcal{Y}_i : i \in [n]\}$ from the n clients.

Theorem 3.3.2 (MMS model). *The output of the local mechanism $\mathcal{R}_{v,s}^{Bin}$ can be represented using $s\left(\log\left(\lceil d/s \rceil\right)+1\right)$ bits. For every $n \in \mathbb{N}$, $\varepsilon \leq s$, and $\delta \in (0,1)$, shuffling the outputs of n mechanisms $\mathcal{R}_{v,s}^{Bin}$ satisfies (ε,δ)-DP by choosing $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$. Let $\hat{\mathbf{b}}$ be the output of the analyzer $\mathcal{A}^{Bin}$. The estimator $\hat{\mathbf{b}}$ is an unbiased estimate of $\overline{\mathbf{b}}$ with MSE:*

$$\mathsf{MSE}_{shuffle}^{Bin} = \mathcal{O}\left(\frac{d^2}{n^2}\max\left\{n\left(\frac{1}{s}-\frac{1}{d}\right), \frac{\log\left(1/\delta\right)}{\min\{\varepsilon^2,\varepsilon\}}\right\}\right). \tag{3.3}$$

Observe that the MSE in (3.2) and (3.3) consists of two terms. The first term represents the communication cost for sending s coordinates out of d coordinates. The second term represents the cost of privacy to randomize the randomly chosen s coordinates. Theorem 3.3.1 shows that each client has to send $s = \min\{\lceil \varepsilon_0 \rceil, d\}$ communication bits to achieve MSE

Algorithm 3.3.2 : Analyzer $\mathcal{A}^{\text{Bin}}$

```
Inputs: $\mathcal{Y}_1,\ldots,\mathcal{Y}_n$, where $\mathcal{Y}_i$ consists of $s$ messages of a pair $(a_{ij},y_{ij})$ for $j\in[s]$ and $i\in[n]$.
$\hat{\mathbf{b}}\leftarrow\mathbf{0}_d$
for $i\in[n]$ do
    for $j\in[s]$ do
        $\hat{\mathbf{b}}[a_{ij}]\leftarrow\hat{\mathbf{b}}[a_{ij}]+y_{ij}$.
$\hat{\mathbf{b}}\leftarrow\frac{1}{n}\hat{\mathbf{b}}$
Return: The server returns $\hat{\mathbf{b}}$.
```

$\mathcal{O}\left(\frac{d^2}{n\min\{\varepsilon_0,\varepsilon_0^2\}}\right)$ in the LDP model. Similarly, Theorem 3.3.2 shows that each client has to send $s=\mathcal{O}\left(\min\{n\{\varepsilon^2,\varepsilon\},d\}\right)$ communication bits to achieve MSE $\mathcal{O}\left(\frac{d^2}{n^2\{\varepsilon^2,\varepsilon\}}\right)$ that matches the MSE of central DP mechanisms. For the scalar case when $d=1$, our results in Theorem 3.3.2 match the optimal MSE as in [CSU19].

3.3.1 Proofs of Theorem 3.3.1 and Theorem 3.3.2 (Binary vectors)

Communication Bound for Theorem 3.3.1 and Theorem 3.3.2 Observe that each client sends s messages; each message consists of a pair (a_{ij},y_{ij}), where a_{ij} is drawn uniformly at random from $\lceil\frac{d}{s}\rceil$ values and y_{ij} is a binary element. Hence, each message requires $\log\left(\lceil\frac{d}{s}\rceil\right)+1$ bits. As a result the total communication bits per client is given by $s\left(\log\left(\lceil\frac{d}{s}\rceil\right)+1\right)$-bits.

Privacy of the LDP model in Theorem 3.3.1 In the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$, each client sends s messages of the *2RR* mechanism $\left((a_{i1},y_{i1}),\ldots,(a_{is},y_{is})\right)$ with parameter $p=\frac{1}{2}\left(1-\sqrt{\frac{\varepsilon_0^2/s^2}{\varepsilon_0^2/s^2+4}}\right)$. Hence, from Lemma 2.4.1, each message is $\frac{\varepsilon_0}{s}$-LDP. As a results, the total mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ is ε_0-LDP from the composition of the DP mechanisms [DR14] when $v=\varepsilon_0$.

In addition, we can bound the RDP of the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ in the LDP model by using the composition of the RDP (see Lemma 2.1.4). From the proof of Theorem 2.4.1 in Section 2.4, the *2RR* mechanism is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha)$ is bounded by:

$$\varepsilon(\alpha) = \frac{1}{\alpha - 1} \log\left(p^{\alpha}(1-p)^{1-\alpha} + p^{1-\alpha}(1-p)^{\alpha}\right), \tag{3.4}$$

In the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$, each client sends s messages of the *2RR* mechanism. Hence, the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ is $(\alpha, s\varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha)$ is given is (3.4).

Privacy of the MMS model in Theorem 3.3.2 In the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$, each client sends s messages of the *2RR* mechanism $((a_{i1}, y_{i1}), \ldots, (a_{is}, y_{is}))$. For simplicity, assume that there exist s shufflers, where the kth shuffler randomly permutes the set of messages $\{(a_{ik}, y_{ik}) : i \in [n]\}$ from the n clients. Hence from composition of the RDP, it is sufficient to bound the RDP of shuffling n outputs of the *2RR* mechanism.

We use the recent results of privacy amplification by shuffling that we will discuss in more details in Chapter 5.

Lemma 3.3.1. *[GDD21e] For any $n \in \mathbb{N}$, $\varepsilon_0 > 0$, and α such that $\alpha^4 e^{5\varepsilon_0} \leq \frac{n}{9}$, the output of shuffling n messages of an ε_0-LDP mechanism is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha)$ is bounded by:*

$$\varepsilon(\alpha) \leq \frac{1}{\alpha - 1} \log\left(1 + \alpha(\alpha-1)\frac{2\left(e^{\varepsilon_0} - 1\right)^2}{n}\right) \leq 2\alpha\frac{\left(e^{\varepsilon_0} - 1\right)^2}{n} \tag{3.5}$$

See Theorem 5.4 in Chapter 5 for more details. Recently [FMT23] improved the dependence on ε_0 of the result in [GDD21e] by showing the following.

Lemma 3.3.2. *[FMT23][Corollary 4.3] For any $n \in \mathbb{N}$, $\varepsilon_0 > 0$, and $\alpha \leq \frac{n}{16\varepsilon_0 e^{\varepsilon_0}}$, the output of shuffling n messages of an ε_0-LDP mechanism is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha)$ is bounded by:*

$$\varepsilon(\alpha) \leq \alpha\frac{48\left(e^{\varepsilon_0} - 1\right)^2}{n e^{\varepsilon_0}}. \tag{3.6}$$

From Theorem 2.4.1, each single message of the client is $\varepsilon_0 = \log\left(\frac{1-p}{p}\right)$-LDP. Hence, from Lemma 3.3.2, the output of a single shuffler is $(\alpha, \tilde{\varepsilon}(\alpha))$-RDP, where $\tilde{\varepsilon}(\alpha) \leq 48\alpha\frac{(1-2p)^2}{np(1-p)}$. Thus, from composition, the output of the s shufflers is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha)$ is bounded by:

$$\varepsilon(\alpha) \leq 48\alpha\frac{s(1-2p)^2}{np(1-p)}. \tag{3.7}$$

Observe that (3.7) gives a closed form bound on the RDP of the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ in the shuffled model. However, we can numerically provide better bound on the RDP of the shuffle model using [GDD21e,FMT23]. By setting $p = \frac{1}{2}\left(1 - \sqrt{\frac{v^2/s^2}{v^2/s^2+4}}\right)$, we get that $\frac{(1-2p)^2}{p(1-p)} = v^2/s^2$, and hence, $\varepsilon(\alpha) \leq 48\alpha v^2/(sn)$. Now, we use Lemmas 2.1.5 in Section 2.4 to convert from RDP to approximate DP, where $\rho = 48v^2/(sn)$. For given $\delta > 0$, shuffling the outputs of n mechanisms $\mathcal{R}_{v,s}^{\text{Bin}}$ is (ε, δ)-DP, where ε is bounded by

$$\varepsilon \leq 3\max\left\{\frac{48v^2}{sn}\log(1/\delta), \sqrt{\frac{48v^2}{sn}\log(1/\delta)}\right\}. \tag{3.8}$$

By setting $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$, we can easily show that (3.8) is satisfied, and hence, the output of the shufflers is (ε, δ)-DP.

MSE bound of the local DP model (Theorem 3.3.1) and shuffle model (Theorem 3.3.2) For ease of analysis, we assume in the remaining part that $\frac{d}{s}$ is integer, otherwise, we can add dummy $s\lceil\frac{d}{s}\rceil - d$ zeros to the vector $\mathbf{b}_i$ to make the size of the vector divisible by s. Now, we show that the output of the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ is unbiased estimate of $\mathbf{b}_i$. Let $\mathcal{Y}_i$ be the output of Algorithm 3.3.1 and $a = \frac{d}{s}$. We can represent the output $\mathcal{Y}_i$ as a vector of dimension d that has s non-zero elements as follows: $\mathbf{y}_i = [\mathbf{y}_{i1}, \ldots, \mathbf{y}_{is}]$, where $\mathbf{y}_{ij} = a\mathcal{R}_p^{2RR}(\mathbf{b}_i[a_{ij}])\,\mathbf{e}_{a_{ij}}$ is a sub-vector of a dimensions that has only one non-zero element.

Then, we have

$$\begin{aligned}\mathbb{E}\left[\mathbf{y}_{ij}\right] &= \frac{1}{a}\sum_{a_{ij}=(j-1)a+1}^{ja} a\mathbf{e}_{a_{ij}}\mathbb{E}\left[\mathcal{R}_p^{2RR}\left(\mathbf{b}_i[a_{ij}]\right)\right] \\ &\overset{\text{(a)}}{=} \sum_{a_{ij}=(j-1)a+1}^{ja} \mathbf{e}_{a_{ij}}\mathbf{b}_i[a_{ij}] = \mathbf{b}_i[(j-1)a+1:ja],\end{aligned} \tag{3.9}$$

where $\mathbf{e}_j$ denotes the jth basis vector and (a) follows from the fact that the mechanism $\mathcal{R}_p^{2RR}$ shown in Theorem 2.4.1 is unbiased. $\mathbf{b}_i\left[l:m\right]$ denotes the values of the coordinates $l, l+1, \ldots, m$. As a result, we have that $\mathbb{E}\left[\mathbf{y}_i\right] = \left[\mathbb{E}\left[\mathbf{y}_{i1}\right], \ldots, \mathbb{E}\left[\mathbf{y}_{is}\right]\right] = \mathbf{b}_i$. Hence, Algorithm 3.3.1 is an unbiased estimate of the input $\mathbf{b}_i$. Furthermore, the variance of Algorithm 3.3.1 is bounded by:

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{y}_i-\mathbf{b}_i\|_2^2\right] &= \sum_{j=1}^{s}\mathbb{E}\left[\|\mathbf{y}_{ij}-\mathbf{b}_i[(j-1)a+1:ja]\|_2^2\right] \\ &= \sum_{j=1}^{s}\frac{1}{a}\sum_{a_{ij}=(j-1)a+1}^{ja}\mathbb{E}\left[\|a\mathbf{e}_{a_{ij}}\mathcal{R}_p^{2RR}\left(\mathbf{b}_i[a_{ij}]\right)-\mathbf{b}_i[(j-1)a+1:ja]\|^2\right] \\ &= \frac{1}{a}\sum_{j=1}^{s}\sum_{a_{ij}=(j-1)a+1}^{ja}\mathbb{E}\Bigg[\|\mathbf{e}_{a_{ij}}a\mathcal{R}_p^{2RR}\left(\mathbf{b}_i[a_{ij}]\right)-\mathbf{e}_{a_{ij}}a\mathbf{b}_i[a_{ij}] \\ &\qquad +\mathbf{e}_{a_{ij}}a\mathbf{b}_i[a_{ij}]-\mathbf{b}_i[(j-1)a+1:ja]\|^2\Bigg] \\ &\overset{\text{(a)}}{=} \frac{1}{a}\sum_{j=1}^{s}\sum_{a_{ij}=(j-1)a+1}^{ja}\mathbb{E}\left[\|\mathbf{e}_{a_{ij}}a\mathcal{R}_p^{\text{Bin}}\left(\mathbf{b}_i[a_{ij}]\right)-\mathbf{e}_{a_{ij}}a\mathbf{b}_i[a_{ij}]\|^2\right] \\ &\qquad +\|\mathbf{e}_{a_{ij}}a\mathbf{b}_i[a_{ij}]-\mathbf{b}_i[(j-1)a+1:ja]\|^2 \\ &\overset{\text{(b)}}{=} \frac{sa^2p(1-p)}{(1-2p)^2}+\frac{1}{a}\sum_{j=1}^{d}\left((a-1)^2+(a-1)\right)\mathbf{b}_i^2[j] \\ &= \frac{sa^2p(1-p)}{(1-2p)^2}+\frac{(a-1)\left((a-1)+1\right)}{a}\sum_{j=1}^{d}\mathbf{b}_i^2[j] \\ &= \frac{a^2sp(1-p)}{(1-2p)^2}+(a-1)\|\mathbf{b}_i\|^2 \overset{\text{(c)}}{\leq} \frac{sa^2p(1-p)}{(1-2p)^2}+(a-1)d \\ &\overset{\text{(d)}}{=} \frac{s^3a^2}{v^2}+(a-1)d = d^2\left(\frac{1}{s}-\frac{1}{d}+\frac{s}{v^2}\right),\end{aligned} \tag{3.10}$$

where (a) follows from the fact that the *2RR* mechanism $\mathcal{R}_p^{2RR}$ is unbiased and (b) from the variance of the *2RR* mechanism $\mathcal{R}_p^{2RR}$ (see Theorem 2.4.1). Step (c) follows from the fact that $\|\mathbf{b}_i\|^2 \leq d$. Step (d) follows from the fact that $p = \frac{1}{2}\left(1 - \sqrt{\frac{v^2/s^2}{v^2/s^2+4}}\right)$. Hence, we can bound the MSE in the local DP model and the shuffle model as follows.

MSE for the local DP model (Theorem 3.3.1): Observe that the output of the server $\hat{b} = \mathcal{A}^{\text{Bin}}\left(\mathcal{Y}_1, \ldots, \mathcal{Y}_n\right)$ can be represented as $\hat{b} = \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i$, where $\mathbf{y}_i$ is the sparse representation of the i-th client private message discussed above. By setting $v^2 = \varepsilon_0^2$, we have that:

$$\begin{aligned}
\mathsf{MSE}_{\text{ldp}}^{\text{Bin}} &= \sup_{\{\mathbf{b}_i \in \{0,1\}^d\}} \mathbb{E}\left[\|\hat{\mathbf{b}} - \overline{\mathbf{b}}\|_2^2\right] \\
&\overset{\text{(a)}}{=} \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|\mathbf{y}_i - \mathbf{b}_i\|_2^2\right] \overset{\text{(b)}}{\leq} \frac{d^2}{n}\left(\frac{1}{s} - \frac{1}{d} + \frac{s}{v^2}\right) \\
&\overset{\text{(c)}}{=} \frac{d^2}{n}\left(\left(\frac{1}{s} - \frac{1}{d}\right) + \frac{s}{\varepsilon_0^2}\right) = \mathcal{O}\left(\frac{d^2}{n}\max\left\{\frac{1}{s}, \frac{s}{\varepsilon_0^2}\right\}\right),
\end{aligned} \tag{3.11}$$

where (a) follows from the i.i.d of the random mechanisms $\mathcal{R}_{v,s}^{\text{Bin}}$. Step (b) follows from the variance of the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ in (3.10). Step (c) follows from substituting $v^2 = \varepsilon_0^2$. This completes the proof of Theorem 3.3.1.

MSE for the MMS model (Theorem 3.3.2): Observe that the output of the server $\hat{b} = \mathcal{A}^{\text{Bin}}\left(\mathcal{Y}_1, \ldots, \mathcal{Y}_n\right)$ can be represented as $\hat{b} = \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i$, where $\mathbf{y}_i$ is the sparse representation of the i-th client private message discussed above. By setting $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$, we have that:

$$\begin{aligned}
\mathsf{MSE}_{\text{shuffle}}^{\text{Bin}} &= \sup_{\{\mathbf{b}_i \in \{0,1\}^d\}} \mathbb{E}\left[\|\hat{\mathbf{b}} - \overline{\mathbf{b}}\|_2^2\right] \\
&\overset{\text{(a)}}{=} \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|\mathbf{y}_i - \mathbf{b}_i\|_2^2\right] \overset{\text{(b)}}{\leq} \frac{d^2}{n}\left(\frac{1}{s} - \frac{1}{d} + \frac{s}{v^2}\right) \\
&\overset{\text{(c)}}{=} \frac{d^2}{n^2}\left(n\left(\frac{1}{s} - \frac{1}{d}\right) + \frac{144\log(1/\delta)}{\min\{\varepsilon^2, \varepsilon\}}\right) \\
&= \mathcal{O}\left(\frac{d^2}{n^2}\max\left\{n\left(\frac{1}{s} - \frac{1}{d}\right), \frac{\log(1/\delta)}{\min\{\varepsilon^2, \varepsilon\}}\right\}\right),
\end{aligned} \tag{3.12}$$

where (a) follows from the i.i.d of the random mechanisms $\mathcal{R}_{v,s}^{\text{Bin}}$. Step (b) follows from the variance of the mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ in (3.10). Step (c) follows from substituting $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$.

This completes the proof of Theorem 3.3.2.

3.4 DME for Bounded ℓ_1-norm Vectors

In this Section, we consider the DME problem for bounded ℓ_1-norm vectors, where $\|\mathbf{x}_i\|_1 \leq r_1$ for $i \in [n]$. 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. The proposed mechanism is based on the Hadamard matrix and is inspired from the Hadamard mechanism proposed by Acharya et al. [ASZ19]. 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 $\boldsymbol{x}_i \in \mathbb{B}_1^d\left(r_1\right)$. 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 $[-{r_1}/{\sqrt{d}}, {r_1}/{\sqrt{d}}]$. Client i selects $j \sim \mathsf{Unif}\left[d\right]$ and quantize $\boldsymbol{y}_i[j]$ privately according to (3.13) and obtains $\boldsymbol{z}_i \in \left\{\pm a\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)\right\}$, 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 3.4.1. The server analyzer $\mathcal{A}^{\ell_1}$ is averaging the messages received from the clients.

Theorem 3.4.1 (Local DP model). *The output of the local mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ can be represented using* $\log(d)+1$ *bits. The mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ satisfies ε_0-LDP. Let $\hat{\mathbf{x}}$ be the output of the analyzer $\mathcal{A}^{\ell_1}$. For $\varepsilon 0 \leq 1$, the estimator $\hat{\mathbf{x}}$ is an unbiased estimate of $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ with bounded MSE:*

$$\mathsf{MSE}_{LDP}^{\ell_1} = \sup_{\{\mathbf{x}_i \in \mathbb{B}_1^d(r_1)\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right] = \mathcal{O}\left(\frac{r_1^2 d}{n\varepsilon_0^2}\right). \tag{3.14}$$

Theorem 3.4.2 (MMS model). *The output of the local mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ can be represented using* $\log(d)+1$ *bits. For every $n \in \mathbb{N}$, $\delta \in (0,1)$, and $\varepsilon \leq \sqrt{\frac{\log(1/\delta)}{n}}$, shuffling the outputs of*

Algorithm 3.4.1 Local Randomizer $\mathcal{R}_{\varepsilon_0}^{\ell_1}$

```
1: Input: Vector $\boldsymbol{x} \in \mathbb{B}_1^d(r_1)$, and local privacy level $\varepsilon_0 > 0$.
2: Construct $\boldsymbol{y}_i = \frac{1}{\sqrt{d}}\mathbf{H}_d \boldsymbol{x}_i$
3: Sample $j \sim \mathsf{Unif}[d]$ and quantize $\boldsymbol{y}_i[j]$ as follows:
   $$\boldsymbol{z}_i = \begin{cases} +r_1\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} + \frac{\sqrt{d}\boldsymbol{y}_i[j]}{2r_1}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1} \\ -a\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} - \frac{\sqrt{d}\boldsymbol{y}_i[j]}{2r_1}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1} \end{cases} \quad (3.13)$$
4: Return $\boldsymbol{z}_i$.
```

n mechanisms $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ satisfies (ε, δ)-DP by choosing $\varepsilon_0 = \varepsilon\sqrt{\frac{n}{\log(1/\delta)}}$. Let $\hat{\mathbf{x}}$ be the output of the analyzer $\mathcal{A}^{\ell_1}$. The estimator $\hat{\mathbf{x}}$ is an unbiased estimate of $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ with bounded MSE:

$$\mathsf{MSE}_{MMS}^{\ell_1} = \sup_{\{\mathbf{x}_i \in \mathbb{B}_1^d(r_1)\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right] = \mathcal{O}\left(\frac{r_1^2 d \log(1/\delta)}{n^2\varepsilon^2}\right). \quad (3.15)$$

The proofs of Theorem 3.4.1 and Theorem 3.4.2 are obtained from the following Lemma whose proof is presented in Appendix B.2

Lemma 3.4.1. *The mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ presented in Algorithm 3.4.1 satisfies the following properties, where $\varepsilon_0 > 0$:*

1. *$\mathcal{R}_{\varepsilon_0}^{\ell_1}$ is ε_0-LDP that requires only 1-bit of communication using public randomness and $\mathcal{O}(\log(d))$-bits using private randomness.*

2. *$\mathcal{R}_{\varepsilon_0}^{\ell_1}$ is unbiased and has bounded variance, i.e., for every $\boldsymbol{x} \in \mathbb{B}_1^d(r_1)$, we have $\mathbb{E}\left[\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x})\right] = \boldsymbol{x}$ and*

$$\mathbb{E}\|\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq r_1^2 d \left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)^2.$$

Theorem 3.4.1 is obtained directly from Lemma 3.4.1 and the independent randomness for different clients. Theorem 3.4.2 is obtained from Theorem 3.4.1 and the privacy amplification by shuffling results from [BBG19d, FMT22].

Remark 3.4.1 (Achievable scheme for bounded ℓ_p-norm, $p \in [1,2)$)**.** Observe that from the norm inequality for any $1 \leq p \leq q$, we have that $\|\boldsymbol{x}\|_q \leq \|\boldsymbol{x}\|_p$. Thus, we have that $\mathbb{B}_p^d(r) \subset \mathbb{B}_1^d(r)$ for any $p \in [1,2)$. As a result, we can bound the MSE for general ℓ_p-norm for $p \in [1,2)$ as $\mathsf{MSE}_{\text{LDP}}^{\ell_p} \leq \mathsf{MSE}_{\text{LDP}}^{\ell_1}$ and $\mathsf{MSE}_{\text{MMS}}^{\ell_p} \leq \mathsf{MSE}_{\text{MMS}}^{\ell_1}$

Now, we present lower bound on the MSE of the DME under LDP constraints.

Theorem 3.4.3 (Lower Bound For Local DP model)**.** *Let $n, d \in \mathbb{N}$, $\varepsilon_0 > 0$. For any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{B}_p^d(r_p)$ and $p \in [1,2)$, the MSE is bounded below by:*

$$\mathsf{MSE}_{LDP}^{\ell_p} = \Omega\left(\frac{r_p^2 d}{n\varepsilon_0^2}\right) \tag{3.16}$$

for any unbiased algorithm $\mathcal{R}$ that is ε_0-LDP.

The proof of Theorem 3.4.3 is presented in Section 3.4.1. Note that when $\varepsilon_0 = \mathcal{O}(1)$, then the upper and lower bounds on minimax risks match for estimating the mean of bounded ℓ_p-norm vectors for $p \in [1,2)$.

3.4.1 Lower Bound on MSE for ℓ_1-norm under LDP constraints

In this section, we prove Theorem 3.4.3. Fix an arbitrary $p \in [1,2)$. Let $\mathcal{P}_p^d$ denote the set of all possible distributions on the ℓ_p ball $\mathbb{B}_p^d$. Note that $\|\boldsymbol{x}\|_p \leq \|\boldsymbol{x}\|_1$, which implies that $\mathbb{B}_1^d \subset \mathbb{B}_p^d$, 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 $\mathcal{P}_p^d$. So, in the following, we only lower-bound $\mathsf{MSE}_{\text{LDP}}^{\ell_1}$. 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. Note that $q \in \mathcal{P}_1^d$ may be a continuous distribution supported over all of $\mathbb{B}_1^d$. 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 $q \in \widehat{\mathcal{P}}_1^d$ can be represented 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_0-LDP mechanisms $\mathcal{R}$ and an estimator $\widehat{\boldsymbol{x}}$. Using the above notations and observations, we have:

$$\begin{aligned}\sup_{\boldsymbol{q}\in\mathcal{P}_1^d} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\right\|_2^2 &\geq \sup_{\boldsymbol{q}\in\widehat{\mathcal{P}}_1^d} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\right\|_2^2 \\ &= \sup_{\boldsymbol{q}\in\widehat{\mathcal{P}}_1^d} \mathbb{E}\left\|\boldsymbol{q} - \widehat{\boldsymbol{x}}\right\|_2^2 .\end{aligned} \tag{3.17}$$

In Chapter 7, we lower-bounded the RHS of (3.17) in the context of characterizing a privacy-utility-randomness trade-offs in LDP. When specializing to our setting, where we are not concerned about the amount of randomness used, our lower bound result gives $\mathsf{MSE}_{\text{LDP}}^{\ell_p} \geq \Omega\left(\min\left\{1, \frac{d}{n\varepsilon_0^2}\right\}\right)$. This completes the proof of Theorem 3.4.3.

3.5 DME for Bounded ℓ_∞-norm Vectors

In this Section, we consider the DME problem for bounded ℓ_∞-norm vectors, where $\|\mathbf{x}_i\|_\infty \leq r_\infty$ for $i \in [n]$. For ease of operation, we will scale each vector such that each coordinate becomes bounded in range $[0,1]$, and then re-scale it at the server-side. Let $\mathbf{z}_i = \frac{\mathbf{x}_i + r_\infty}{2r_\infty}$, where the operations are done coordinate-wise. Thus, we have that $\mathbf{z}_i[j] \in [0,1]$ for all $j \in [d]$ and $i \in [n]$, where $\mathbf{z}_i[j]$ denotes the jth coordinate of the vector $\mathbf{z}_i$. Observe that the vector $\mathbf{z}_i$ can be decomposed into a weighted summation of binary vectors $\mathbf{b}_i^{(k)} \in \{0,1\}^d, \forall k \geq 1$ as follows:

$$\mathbf{z}_i = \sum_{k=1}^{\infty} \mathbf{b}_i^{(k)} 2^{-k}, \tag{3.18}$$

where $\mathbf{b}_i^{(k)} = \lfloor 2^k \left(\mathbf{z}_i - \mathbf{z}_i^{(k-1)}\right) \rfloor, k \geq 1$ such that $\mathbf{z}_i^{(0)} = \mathbf{0}$ and $\mathbf{z}_i^{(k)} = \sum_{l=1}^{k} \mathbf{b}_i^{(l)} 2^{-l}$. To make our mechanism communication efficient, each client approximates the vector $\mathbf{z}_i$ by using the first m binary vectors $\{\mathbf{b}_i^{(k)} : 1 \leq k \leq m\}$. Note that the first m binary vectors together give an approximation to the real vector $\mathbf{z}_i$ with error $\|\mathbf{z}_i - \mathbf{z}_i^{(m)}\|_2^2 \leq d/4^m$, where $\mathbf{z}_i^{(m)} = \sum_{k=1}^{m} \mathbf{b}_i^{(k)} 2^{-k}$. However, this mechanism creates a biased estimate of $\mathbf{z}_i$. Hence, to design an unbiased mechanism, the client approximates the vector $\mathbf{z}_i$ using the first $m-1$ binary vectors $\{\mathbf{b}_i^{(k)} : 1 \leq k \leq m-1\}$ of the binary representation above and the last binary vector $(\mathbf{u}_i)$ is reserved for unbiasedness as follows:

$$\mathbf{u}_i[j] = \mathsf{Bern}\left(2^{m-1}(\mathbf{z}_i[j] - \mathbf{z}_i^{(m-1)}[j])\right), \tag{3.19}$$

where $\mathbf{z}_i^{(m-1)} = \sum_{k=1}^{m-1} \mathbf{b}_i^{(k)} 2^{-k}$ and $\mathsf{Bern}(p)$ denotes Bernoulli random variable with bias p. For completeness, we prove some properties of this quantization scheme in Section 3.5.1. Then, we estimate the mean of binary vectors $\{\mathbf{b}_i^{(k)} \in \{0,1\}^d : i \in [n]\}$ using Algorithm 3.3.1 with different privacy guarantees for each level $k \in [m]$, where we allocate lower privacy (higher privacy parameter v_k) for the most significant bits (MSBs) (lower k) in order to get better performance in terms of the MSE.

The private DME mechanism is given in Algorithm 3.5.1, where v controls the total privacy of the mechanism. There are two communication parameters: m controls the number of levels for quantization and s controls the number of dimensions used to represent each binary vector. In Theorems 3.5.1 and 3.5.2, we present how the privacy and communication parameters v, m, s affects the accuracy of the mechanism. The server aggregator $\mathcal{A}^{\ell_\infty}$ is presented in Algorithm 3.5.2, where the server first estimates the mean of each binary vectors $\{b_i^{(k)} : i \in [n]\}$ for $k \in [m-1]$ and decodes the messages to generate an estimate of the true mean $\overline{\mathbf{z}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i$. Then, the server scales the vector $\overline{\mathbf{z}}$ to generate an unbiased estimate of the mean $\overline{\mathbf{x}}$. We prove the bound on the MSE of our proposed mechanism for the LDP and MMS models in the following theorems. We defer the proofs to Section 3.5.2.

Theorem 3.5.1 (Local DP model). *The output of the local mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ can be represented using $ms(\log(\lceil d/s \rceil) + 1)$ bits. By choosing $v = \varepsilon_0$, the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ satisfies ε_0-LDP. Let $\hat{\mathbf{x}}$ be the output of the analyzer $\mathcal{A}^{\ell_\infty}$. The estimator $\hat{\mathbf{x}}$ is an unbiased*

Algorithm 3.5.1 : Local Randomizer $\mathcal{R}^{\ell_\infty}_{v,m,s}$

```
Public parameter: Privacy budget $v$, communication levels $m$, and coordinate sampling per level $s$.
Input: $\mathbf{x}_i \in \mathbb{B}^d_\infty(r_\infty)$.
$\mathbf{z}_i \leftarrow (\mathbf{x}_i + r_\infty)/2r_\infty$
$\mathbf{z}_i^{(0)} \leftarrow 0$
for $k = 1, \ldots, m-1$ do
    $\mathbf{b}_i^{(k)} \leftarrow \lfloor 2^k(\mathbf{z}_i - \mathbf{z}_i^{(k-1)}) \rfloor$
    $v_k \leftarrow \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$
    $\mathcal{Y}_i^{(k)} \leftarrow \mathcal{R}^{\mathrm{Bin}}_{v_k,s}(\mathbf{b}_i^{(k)})$
    $\mathbf{z}_i^{(k)} \leftarrow \mathbf{z}_i^{(k-1)} + \mathbf{b}_i^{(k)} 2^{-k}$
Sample $\mathbf{u}_i \leftarrow \mathsf{Bern}\left(2^{m-1}\left(\mathbf{z}_i - \mathbf{z}_i^{(m-1)}\right)\right)$
$v_m \leftarrow \frac{4^{\frac{-m+1}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$
$\mathcal{Y}_i^{(m)} \leftarrow \mathcal{R}^{\mathrm{Bin}}_{v_m,s}(\mathbf{u}_i)$
Return: The client sends $\mathcal{Y}_i \leftarrow \left\{\mathcal{Y}_i^{(1)}, \ldots, \mathcal{Y}_i^{(m)}\right\}$.
```

estimate of $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ *with bounded MSE:*

$$\mathsf{MSE}^{\ell_\infty}_{LDP} = \sup_{\{\mathbf{x}_i \in \mathbb{B}^d_\infty(r_\infty)\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right] = \mathcal{O}\left(\frac{r_\infty^2 d^2}{n} \max\left\{\frac{1}{d4^m}, \frac{1}{s}, \frac{s}{\varepsilon_0^2}\right\}\right). \tag{3.20}$$

Theorem 3.5.2 (MMS model)**.** *The output of the local mechanism* $\mathcal{R}^{\ell_\infty}_{v,m,s}$ *can be represented using* $ms\left(\log\left(\lceil d/s \rceil\right) + 1\right)$ *bits. For every* $n \in \mathbb{N}$, $\varepsilon \leq ms$, *and* $\delta \in (0,1)$, *shuffling the outputs of* n *mechanisms* $\mathcal{R}^{\ell_\infty}_{v,m,s}$ *satisfies* (ε, δ)*-DP by choosing* $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$. *Let* $\hat{\mathbf{x}}$ *be the output of the analyzer* $\mathcal{A}^{\ell_\infty}$. *The estimator* $\hat{\mathbf{x}}$ *is an unbiased estimate of* $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ *with bounded MSE:*

$$\mathsf{MSE}^{\ell_\infty}_{MMS} = \sup_{\{\mathbf{x}_i \in \mathbb{B}^d_\infty(r_\infty)\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right] = \mathcal{O}\left(\frac{r_\infty^2 d^2}{n^2} \max\left\{\frac{n}{d4^m}, n\left(\frac{1}{s} - \frac{1}{d}\right), \frac{\log(1/\delta)}{\min\{\varepsilon^2, \varepsilon\}}\right\}\right). \tag{3.21}$$

Observe that the MSE in (3.20) and (3.21) consists of three terms. The first term is the communication cost of quantizing the real vector $\mathbf{z}_i$ using m binary vectors. The second term represents the communication cost of sending s out of d coordinates from each binary vector. The third term is the privacy cost to randomize the binary vectors. Theorem 3.5.1 shows that each client has to set $m = 1$ and $s = \lceil \varepsilon_0 \rceil$ of total $\mathcal{O}\left(\lceil \varepsilon_0 \rceil\right)$ communication bits to achieve MSE $\mathcal{O}\left(\frac{d^2}{n \min\{\varepsilon_0, \varepsilon_0^2\}}\right)$ when $\varepsilon_0 \leq d$. Similarly, by setting $m = \max\{1, \lceil \log\left(n \min\{\varepsilon^2, \varepsilon\}/d\right)\rceil\}$ and $s = \mathcal{O}\left(\min\{n\{\varepsilon^2, \varepsilon\}, d\}\right)$ in Theorem 3.5.2, the MSE is bounded by $\mathcal{O}\left(\frac{d^2}{n^2 \min\{\varepsilon^2, \varepsilon\}}\right)$, which matches the MSE of central differential privacy mechanisms with total communication cost of $\mathcal{O}\left(d \log\left(\frac{n \min\{\varepsilon^2, \varepsilon\}}{d}\right)\right)$ when $d \leq n \min\{\varepsilon^2, \varepsilon\}$ and $\mathcal{O}\left(n\{\varepsilon^2, \varepsilon\} \log\left(\frac{d}{n\{\varepsilon^2, \varepsilon\}}\right)\right)$ when $d > n\{\varepsilon^2, \varepsilon\}$.

Remark 3.5.1 (Scalar case)**.** When $d = 1$, i.e., scalar case, our MMS algorithm achieves the central DP error $\mathcal{O}\left(\frac{1}{n^2 \min\{\varepsilon^2, \varepsilon\}}\right)$ using $m = \max\{1, \lceil \log\left(n \min\{\varepsilon^2, \varepsilon\}\right)\rceil\}$ bits per client. This result covers the private-communication trade-offs for all privacy regimes. For example, for $\varepsilon = \frac{1}{\sqrt{n}}$, each client needs only a single bit to achieve the central DP error. On the other hand, the multi-message shuffled mechanism based on IKOS protocol [IKO06] proposed in [BBG20b, GKM20] requires $\mathcal{O}\left(\log\left(n\right)\right)$-bits of communication for all privacy regimes, where it doesn't provide any guarantees for any small communication cost [BBG20b, Sec. 1.2]. Even when particular regimes of order-optimality are achieved for the MMS, the communication bound is in expectation [GKM21b], whereas ours is deterministic.

Remark 3.5.2 (Scalar summation with sampling/sketching)**.** Observe that when $d < n \min\{\varepsilon^2, \varepsilon\}$, it is not possible to combine the scalar summation scheme [BBG20b, GKM20] with coordinate sampling due to the following. When each client independently chooses a set of s coordinates, we might lose the amplification gain from shuffling, as not all the n clients will choose the same set of s coordinates. When choosing the same s coordinates for all clients, the MSE is bounded below by $\Omega\left(r_\infty^2(d - s)\right)$. Thus, the scalar summation in MMS cannot be directly combined with coordinate sampling.

Algorithm 3.5.2 : Analyzer $\mathcal{A}^{\ell_\infty}$

```
1: Inputs: $\mathcal{Y}_1, \ldots, \mathcal{Y}_n$, where $\mathcal{Y}_i = \left\{\mathcal{Y}_i^{(1)}, \ldots, \mathcal{Y}_i^{(m)}\right\}$ is a set of $m$ sets.
2: for $k = 1, \ldots, m-1$ do
3:     $\hat{\mathbf{b}}^{(k)} \leftarrow \mathcal{A}^{\text{Bin}}\left(\mathcal{Y}_1^{(k)}, \ldots, \mathcal{Y}_n^{(k)}\right)$
4: $\hat{\mathbf{u}} \leftarrow \mathcal{A}^{\text{Bin}}\left(\mathcal{Y}_1^{(m)}, \ldots, \mathcal{Y}_n^{(m)}\right)$
5: $\hat{\mathbf{z}} \leftarrow \sum_{k=1}^{m-1} \hat{\mathbf{b}}^{(k)} 2^{-k} + \hat{\mathbf{u}} 2^{-m+1}$
6: Return: The server returns $\hat{\mathbf{x}} \leftarrow 2r_\infty \hat{\mathbf{z}} - r_\infty$.
```

3.5.1 Properties of The Quantization Scheme

In this section, we prove some properties of the quantization scheme for vector $\mathbf{z}_i \in [0,1]^d$. We first prove some properties for a scalar case when $x \in [0,1]$, and then, the results of the bounded ℓ_∞ will be obtained directly from repeating the scalar case on each coordinate.

Let $x \in [0,1]$ and $x^{(k)} = \sum_{l=1}^{s} b_l 2^{-l}$ for $k \geq 1$, where $x^{(0)} = 0$ and $b_k = \lfloor 2^k (x - x^{k-1}) \rfloor$. For given $m \geq 1$, we represent x using m bits as follows: $\tilde{x}^{(m)} = \sum_{k=1}^{m-1} b_k 2^{-k} + u 2^{-m+1}$, where $u = \mathsf{Bern}\left(2^{m-1}(x - x^{(m-1)}[j])\right)$. This estimator needs only m communication bits.

Lemma 3.5.1. *For given $x \in [0,1]$, let $\tilde{x}^{(m)}$ be the quantization of x presented above. We have that $\tilde{x}^{(m)}$ is an unbiased estimate of x with bounded MSE:*

$$\mathsf{MSE}_{scalar}^{quan} = \sup_{x \in [0,1]} \mathbb{E}\left[\|x - \tilde{x}^{(m)}\|_2^2\right] \leq \frac{1}{4^m}, \tag{3.22}$$

where the expectation is taken over the randomness in the quantization scheme.

Proof. First, we show that $\tilde{x}^{(m)}$ is an unbiased estimate of x:

$$\begin{aligned}\mathbb{E}\left[\tilde{x}^m\right] &= \sum_{k=1}^{m-1} b_k 2^{-k} + \mathbb{E}\left[u\right] 2^{-m+1} \\ &\overset{\text{(a)}}{=} \sum_{k=1}^{m-1} b_k 2^{-k} + 2^{m-1}(x - x^{(m-1)}) 2^{-m+1} = x_i,\end{aligned} \tag{3.23}$$

where step (a) is obtained from the fact that u is a Bernoulli random variable with bias $p = 2^{m-1}(x - x^{(m-1)})$. We show that the estimator $\tilde{x}^{(m)}$ has a bounded MSE by 4^{-m}:

$$\begin{aligned}\mathsf{MSE}_{\text{scalar}}^{\text{quan}} &= \sup_{x\in[0,1]} \mathbb{E}\left[\|x - \tilde{x}^{(m)}\|_2^2\right] \\ &= \sup_{x\in[0,1]} \mathbb{E}\left[\|x - x^{(m-1)} - u2^{-m+1}\|^2\right] \\ &= \sup_{x\in[0,1]} 4^{-(m-1)}\mathbb{E}\left[\|2^{-(m-1)}(x - x^{(m-1)}) - u\|^2\right] \overset{\text{(a)}}{\leq} \frac{1}{4^m},\end{aligned} \tag{3.24}$$

where the inequality (a) is obtained from the fact that u is a Bernoulli random variable, and hence, it has a variance less that 1/4. This completes the proof of Lemma 3.5.1. ■

Corollary 3.5.1. For given a vector $\mathbf{z}_i \in [0,1]^d$, let $\tilde{\mathbf{z}}_i^{(m)}$ be the quantization of $\mathbf{z}_i$ by applying the above scalar quantization scheme on each coordinate $\mathbf{z}_i[j]$ for $j \in [d]$. Then, $\tilde{\mathbf{z}}_i^{(m)}$ is an ubiased estimate of $\mathbf{z}_i$ with bounded MSE:

$$\mathsf{MSE}_{\text{vector}}^{\text{quan}} = \sup_{\mathbf{z}_i\in[0,1]^d} \mathbb{E}\left[\|\mathbf{z}_i - \tilde{\mathbf{z}}_i^{(m)}\|_2^2\right] \leq \frac{d}{4^m}, \tag{3.25}$$

where the expectation is taken over the randomness in the quantization scheme.

3.5.2 Proofs of Theorem 3.5.1 and Theorem 3.5.2 (Bounded ℓ_∞-norm vectors)

Communication cost for Theorem 3.5.1 and Theorem 3.5.2 In the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$, the client sends m binary vectors $\mathbf{b}_i^{(1)}, \ldots, \mathbf{b}_i^{(m-1)}, \mathbf{u}_i$ using the private mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$. From Theorem 3.3.1 and Theorem 3.3.2, the private mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ needs $\log\left(\lceil \frac{d}{s} \rceil\right) + 1$ bits for communication. Thus, the total communication of the private mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ is $ms\left(\log\left(\lceil \frac{d}{s} \rceil\right) + 1\right)$-bits.

Privacy of the local DP model in Theorem 3.5.1 In the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$, each client sends m messages from the private mechanism $\mathcal{R}_{v,s}^{\text{Bin}}$ as follows:

$$\left\{\mathcal{R}_{v_1,s}^{\text{Bin}}(\mathbf{b}_i^{(1)}), \ldots, \mathcal{R}_{v_{m-1},s}^{\text{Bin}}(\mathbf{b}_i^{(m-1)}), \mathcal{R}_{v_m,s}^{\text{Bin}}(\mathbf{u}_i)\right\},$$

where $v_k = \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$ for $k \in [m-1]$ and $v_m = \frac{4^{\frac{-m+1}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$. Hence, from Theorem 3.3.1, the k-th message $\mathcal{R}_{v_k,s}^{\text{Bin}}(\mathbf{b}_i^{(k)})$ is $\varepsilon_0^{(k)}$-LDP, where $\varepsilon_0^{(k)} = v_k$ for $k \in [m]$. As a results, the total mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ is bounded by:

$$\varepsilon_0 = \sum_{k=1}^{m} \varepsilon_0^{(k)} = \sum_{k=1}^{m} v_k = \sum_{k=1}^{m-1} \left\{ \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v \right\} + \frac{4^{\frac{-m+1}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v = v, \tag{3.26}$$

from the composition of the DP mechanisms [DR14]. Observe that we choose $v = \varepsilon_0$, and hence, the bound in (3.26) is satisfied. In addition, we can bound the RDP of the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ in the local DP model by using the composition of the RDP (see Lemma 2.1.4). From the proof of Theorem 3.3.1 in Section 3.3.1, the mechanism $\mathcal{R}_{v_k,s}^{\text{Bin}}$ is $\left(\alpha, \varepsilon^{(k)}(\alpha)\right)$-RDP, where $\varepsilon^{(k)}(\alpha)$ is bounded by:

$$\varepsilon^{(k)}(\alpha) = \frac{s}{\alpha - 1} \log\left(p_k^{\alpha}(1-p_k)^{1-\alpha} + p_k^{1-\alpha}(1-p_k)^{\alpha}\right), \tag{3.27}$$

where $p_k = \frac{1}{2}\left(1 - \sqrt{\frac{v_k^2/s^2}{v_k^2/s^2+4}}\right)$. Hence, the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha) = \sum_{k=1}^{m} \varepsilon^{(k)}(\alpha)$.

Privacy of the MMS model in Theorem 3.5.2 In the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$, each client sends m messages from the private mechanism $\mathcal{R}_{p,s}^{\text{Bin}}$ as follows:

$$\left\{\mathcal{R}_{v_1,s}^{\text{Bin}}(\mathbf{b}_i^{(1)}), \ldots, \mathcal{R}_{v_{m-1},s}^{\text{Bin}}(\mathbf{b}_i^{(m-1)}), \mathcal{R}_{v_m,s}^{\text{Bin}}(\mathbf{u}_i)\right\},$$

where $v_k = \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$ for $k \in [m-1]$ and $v_m = \frac{4^{\frac{-m+1}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$.

From the proof of Theorem 3.3.2 in Section 3.3.1, shuffling the outputs of n mechanisms $\mathcal{R}_{v_k,s}^{\text{Bin}}$ is $\left(\alpha, \varepsilon^{(k)}(\alpha)\right)$, where $\varepsilon^{(k)}(\alpha)$ is bounded by:

$$\varepsilon^{(k)}(\alpha) \leq 48\alpha \frac{v_k^2}{sn}, \tag{3.28}$$

from (3.7) by substituting $p_k = \frac{1}{2}\left(1 - \sqrt{\frac{v_k^2/s^2}{v_k^2/s^2+4}}\right)$. From Lemma 2.1.4 of the RDP composition, we get that the total RDP of the mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ is bounded by:

$$\varepsilon\left(\alpha\right)=\sum_{k=1}^{m}\varepsilon^{(k)}\left(\alpha\right)=\alpha\frac{48}{sn}\sum_{k=1}^{m}v_k^2=\alpha\frac{48v^2}{sn}\sum_{k=1}^{m}f_k^2\leq\alpha\frac{48v^2}{sn}, \tag{3.29}$$

where $f_k=\frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1}4^{\frac{-l}{3}}+4^{\frac{-m+1}{3}}\right)}$ for $k\in[m]$ and $f_m=\frac{4^{\frac{-m+1}{3}}}{\left(\sum_{l=1}^{m-1}4^{\frac{-l}{3}}+4^{\frac{-m+1}{3}}\right)}$. The last inequality is obtained from the fact that $\sum_{k=1}^{m}f_k=1$ and hence $\sum_{k=1}^{m}f_k^2\leq 1$. Now, we use Lemma 2.1.5 in Section 2.4 to convert from RDP to approximate DP, where $\rho=48v^2/(sn)$. For given $\delta>0$, shuffling the outputs of n mechanisms $\mathcal{R}_{v,m,s}^{\ell_\infty}$ is (ε,δ)-DP, where ε is bounded by

$$\varepsilon\leq 3\max\left\{\frac{48v^2}{sn}\log\left(1/\delta\right),\sqrt{\frac{48v^2}{sn}\log\left(1/\delta\right)}\right\}. \tag{3.30}$$

By setting $v^2=\frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$, we can easily show that (3.30) is satisfied, and hence, the output of the shufflers is (ε,δ)-DP.

MSE bound of the local DP model (Theorem 3.5.1) and MMS model (Theorem 3.5.2) We first present some notations to simplify the analysis. For given $\mathbf{x}_i\in\mathbb{B}_\infty^d\left(r_\infty\right)$, we define $\mathbf{z}_i=\frac{\mathbf{x}_i+r_\infty}{2r_\infty}$, where the operations are done coordinate-wise. Thus, we have that $\mathbf{z}_i\in[0,1]^d$. For given $\mathbf{z}_i\in[0,1]^d$ and $m\geq 1$, we define $\tilde{\mathbf{z}}_i^{(m)}=\sum_{k=1}^{m-1}\mathbf{b}_i^{(k)}2^{-k}+\mathbf{u}_i2^{-m+1}$, where $\mathbf{b}_i^{(k)}=\lfloor 2^k\left(\mathbf{z}_i-\mathbf{z}_i^{(k-1)}\right)\rfloor$ and $\mathbf{z}_i^{(0)}=\mathbf{0}$ and $\mathbf{z}_i^{(k)}=\sum_{l=1}^{k}\mathbf{b}_i^{(l)}2^{-l}$ for $k\geq 1$. Furthermore, $\mathbf{u}_i$ is a Bernoulli vector defined by $\mathbf{u}_i=\mathsf{Bern}\left(2^{m-1}\left(\mathbf{z}_i-\mathbf{z}_i^{(m-1)}\right)\right)$. Let $\overline{\mathbf{b}}^{(k)}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{b}_i^{(k)}$, $\overline{\mathbf{u}}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{u}_i$, and $\overline{\tilde{\mathbf{z}}}^{(m)}=\frac{1}{n}\sum_{i=1}^{n}\tilde{\mathbf{z}}_i^{(m)}$.

MSE for the local DP model (Theorem 3.5.1): Observe that the output of the server $\hat{\mathbf{x}}=\mathcal{A}^{\ell_\infty}\left(\mathcal{Y}_1,\ldots,\mathcal{Y}_n\right)=2r_\infty\hat{\mathbf{z}}-r_\infty$, where $\hat{\mathbf{z}}=\sum_{k=1}^{m-1}\hat{\mathbf{b}}^{(k)}2^{-k}+\hat{\mathbf{u}}2^{-m+1}$. Thus, we have that:

$$\begin{aligned}\mathsf{MSE}_{\text{ldp}}^{\ell_\infty}&=\sup_{\{\mathbf{x}_i\in\mathbb{B}_\infty^d(r_\infty)\}}\mathbb{E}\left[\|\hat{\mathbf{x}}-\overline{\mathbf{x}}\|_2^2\right]\\&\overset{\text{(a)}}{=}4r_\infty^2\sup_{\{\mathbf{z}_i\in[0,1]^d\}}\mathbb{E}\left[\|\hat{\mathbf{z}}-\overline{\mathbf{z}}\|_2^2\right]\\&=4r_{\infty^2}\sup_{\{\mathbf{z}_i\in[0,1]^d\}}\mathbb{E}\left[\|\hat{\mathbf{z}}-\overline{\tilde{\mathbf{z}}}^{(m)}+\overline{\tilde{\mathbf{z}}}^{(m)}-\overline{\mathbf{z}}\|_2^2\right]\end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{=} 4r_\infty^2 \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \left(\mathbb{E}\left[\|\hat{\mathbf{z}} - \bar{\tilde{\mathbf{z}}}^{(m)}\|_2^2\right] + \mathbb{E}\left[\|\bar{\tilde{\mathbf{z}}}^{(m)} - \overline{\mathbf{z}}\|_2^2\right]\right) \\
&\overset{(c)}{\leq} 4r_\infty^2 \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \left(\mathbb{E}\left[\|\sum_{k=1}^{m-1}\hat{\mathbf{b}}^{(k)}2^{-k} + \hat{\mathbf{u}}2^{-m+1} - \sum_{k=1}^{m-1}\overline{\mathbf{b}}^{(k)}2^{-k} + \overline{\mathbf{u}}2^{-m+1}\|_2^2\right] + \frac{d}{n4^m}\right) \\
&\overset{(d)}{\leq} 4r_\infty^2 \left(\sum_{k=1}^{m-1}\frac{d^2 4^{-k}}{n}\left(\frac{1}{s} + \frac{s}{v_k^2}\right) + \frac{d^2 4^{-m+1}}{n}\left(\frac{1}{s} + \frac{s}{v_m^2}\right) + \frac{d}{n4^m}\right) \\
&\overset{(e)}{\leq} 4r_\infty^2 \left(\frac{d^2}{ns}\left(\sum_{k=1}^{m-1}4^{-k} + 4^{-m+1}\right) + \frac{d^2 s}{nv^2}\left(\sum_{k=1}^{m-1}4^{-k/3} + 4^{-(m-1)/3}\right)^3 + \frac{d}{n4^m}\right) \\
&\overset{(f)}{\leq} 4r_\infty^2 \left(\frac{4d^2}{3ns} + \frac{5d^2 s}{n\varepsilon_0^2} + \frac{d}{n4^m}\right) \qquad (3.31) \\
&= \mathcal{O}\left(\frac{r_\infty^2 d^2}{n}\max\left\{\frac{1}{d4^m}, \frac{1}{s}, \frac{s}{\varepsilon_0^2}\right\}\right), \qquad (3.32)
\end{aligned}
$$

where (a) follows from the fact that $\mathbf{z}_i$ is a linear transformation of $\mathbf{x}_i$. Step (b) follows from the fact that $\bar{\tilde{\mathbf{z}}}^{(m)}$ is an unbiased estimate of $\overline{\mathbf{z}}$ from Corollary 3.5.1. Step (c) from the bound of the MSE of the quantization scheme $\bar{\tilde{\mathbf{z}}}^{(m)}$ in Corollary 3.5.1. Step (d) follows from the MSE of the private mean estimation of binary vectors in Theorem 3.3.1. Step (e) follows from substituting $v_k = \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1}4^{\frac{-l}{3}}+4^{\frac{-m+1}{3}}\right)}v$. Step (f) follows from the geometric series bound. This completes the proof of Theorem 3.5.1.

MSE for the MMS model (Theorem 3.5.2): Observe that the output of the server $\hat{\mathbf{x}} = \mathcal{A}^{\ell_\infty}(\mathcal{Y}_1, \ldots, \mathcal{Y}_n) = 2r_\infty\hat{\mathbf{z}} - r_\infty$, where $\hat{\mathbf{z}} = \sum_{k=1}^{m-1}\hat{\mathbf{b}}^{(k)} + \hat{\mathbf{u}}2^{-m+1}$. Thus, we have that:

$$
\begin{aligned}
\mathsf{MSE}_{\text{shuffle}}^{\ell_\infty} &= \sup_{\{\mathbf{x}_i\in\mathbb{B}_\infty^d(r_\infty)\}} \mathbb{E}\left[\|\hat{\mathbf{x}} - \overline{\mathbf{x}}\|_2^2\right] \\
&\overset{(a)}{=} 4r_\infty^2 \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \mathbb{E}\left[\|\hat{\mathbf{z}} - \overline{\mathbf{z}}\|_2^2\right] \\
&= 4r_{\infty^2} \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \mathbb{E}\left[\|\hat{\mathbf{z}} - \bar{\tilde{\mathbf{z}}}^{(m)} + \bar{\tilde{\mathbf{z}}}^{(m)} - \overline{\mathbf{z}}\|_2^2\right] \\
&\overset{(b)}{=} 4r_\infty^2 \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \left(\mathbb{E}\left[\|\hat{\mathbf{z}} - \bar{\tilde{\mathbf{z}}}^{(m)}\|_2^2\right] + \mathbb{E}\left[\|\bar{\tilde{\mathbf{z}}}^{(m)} - \overline{\mathbf{z}}\|_2^2\right]\right) \\
&\overset{(c)}{\leq} 4r_\infty^2 \sup_{\{\mathbf{z}_i\in[0,1]^d\}} \left(\mathbb{E}\left[\|\sum_{k=1}^{m-1}\hat{\mathbf{b}}^{(k)}2^{-k} + \hat{\mathbf{u}}2^{-m+1} - \sum_{k=1}^{m-1}\overline{\mathbf{b}}^{(k)}2^{-k} + \overline{\mathbf{u}}2^{-m+1}\|_2^2\right] + \frac{d}{n4^m}\right)
\end{aligned}
$$

$$
\begin{aligned}
&\overset{\text{(d)}}{\leq} 4r_\infty^2 \left(\sum_{k=1}^{m-1} \frac{d^2 4^{-k}}{n} \left(\left(\frac{1}{s} - \frac{1}{d} \right) + \frac{s}{v_k^2} \right) + \frac{d^2 4^{-m+1}}{n} \left(\left(\frac{1}{s} - \frac{1}{d} \right) + \frac{s}{v_m^2} \right) + \frac{d}{n 4^m} \right) \\
&\overset{\text{(e)}}{\leq} 4r_\infty^2 \frac{d^2}{n} \left(\frac{1}{s} - \frac{1}{d} \right) \left(\sum_{k=1}^{m-1} 4^{-k} + 4^{-m+1} \right) \\
&\qquad + 4r_\infty^2 \left(\frac{d^2 s}{n v^2} \left(\sum_{k=1}^{m-1} 4^{-k/3} + 4^{-(m-1)/3} \right)^3 + \frac{d}{n 4^m} \right) \qquad (3.33) \\
&\overset{\text{(f)}}{\leq} 4r_\infty^2 \left(\frac{4d^2}{3n} \left(\frac{1}{s} - \frac{1}{d} \right) + \frac{5d^2 \log\left(1/\delta\right)}{n^2 \min\{\varepsilon^2, \varepsilon\}} + \frac{d}{n 4^m} \right) \qquad (3.34) \\
&= \mathcal{O}\left(\frac{r_\infty^2 d^2}{n^2} \max\left\{ \frac{n}{d 4^m}, n \left(\frac{1}{s} - \frac{1}{d} \right), \frac{\log\left(1/\delta\right)}{\min\{\varepsilon^2, \varepsilon} \right\} \right), \qquad (3.35)
\end{aligned}
$$

where (a) follows from the fact that $\mathbf{z}_i$ is a linear transformation of $\mathbf{x}_i$. Step (b) follows from the fact that $\overline{\overline{\mathbf{z}}}^{(m)}$ is an unbiased estimate of $\overline{\mathbf{z}}$ from Corollary 3.5.1. Step (c) from the bound of the MSE of the quantization scheme $\overline{\overline{\mathbf{z}}}^{(m)}$ in Corollary 3.5.1. Step (d) follows from the MSE of the private mean estimation of binary vectors in Theorem 3.3.2. Step (e) follows from substituting $v_k = \frac{4^{\frac{-k}{3}}}{\left(\sum_{l=1}^{m-1} 4^{\frac{-l}{3}} + 4^{\frac{-m+1}{3}}\right)} v$. Step (f) follows from the geometric series bound. This completes the proof of Theorem 3.5.2.

3.6 DME for Bounded ℓ_2-norm Vectors

In this section, we consider the DME problem for bounded ℓ_2-norm vectors, where $\|\mathbf{x}_i\|_2 \leq r_2$ for $i \in [n]$. We first use the random rotation proposed in [SFK17] to bound the ℓ_∞-norm of the vector with radius $r_\infty = \mathcal{O}\left(\frac{r_2}{\sqrt{d}}\right)$. Then, we apply the bounded ℓ_∞-norm algorithm in Section 3.5. The client-side scheme is presented in Algorithm 3.6.1 and the server-side scheme is presented in Algorithm 3.6.2.

Theorem 3.6.1 (LDP model)**.** *The output of the local mechanism $\mathcal{R}_{v,m,s}^{\ell_2}$ can be represented using $ms\left(\log\left(\lceil d/s \rceil\right) + 1\right)$ bits. By choosing $v = \varepsilon_0$, the mechanism $\mathcal{R}_{v,m,s}^{\ell_2}$ satisfies ε_0-LDP. Let $\hat{\mathbf{x}}$ be the output of the analyzer $\mathcal{A}^{\ell_2}$. With probability at least $1 - \beta$, the estimator $\hat{\mathbf{x}}$ is an*

Algorithm 3.6.1 : Local Randomizer $\mathcal{R}^{\ell_2}_{v,m,s}$

```
**Public parameter:** Privacy budget $v$, communication levels $m$, coordinate sampling per level $s$, and confidence term $\beta$.
**Input:** $\mathbf{x}_i \in \mathbb{B}^d_2(r_2)$.
Let $U = \frac{1}{\sqrt{d}}\mathbf{H}D$, where $\mathbf{H}$ denotes a Hadamard matrix and $D$ is a diagonal matrix with i.i.d. uniformly random $\{\pm 1\}$ entries.
$\mathbf{w}_i \leftarrow W\mathbf{x}_i$
$r_\infty \leftarrow 10r_2\sqrt{\frac{\log(dn/\beta)}{d}}$
**for** $j = 1, \ldots, d$ **do**
    $\mathbf{w}_i[j] = \min\{r_\infty, \max\{\mathbf{w}_i(j), -r_\infty\}\}$
$\mathcal{Y}_i \leftarrow \mathcal{R}^{\ell_\infty}_{v,m,s}(\mathbf{w}_i)$
**Return:** The client sends $\mathcal{Y}_i$.
```

unbiased estimate of $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ *with bounded MSE:*

$$\mathsf{MSE}^{\ell_2}_{LDP} = \tilde{\mathcal{O}}\left(\frac{r_2^2 d}{n}\max\left\{\frac{1}{d4^m}, \frac{1}{s}, \frac{s}{\varepsilon_0^2}\right\}\right), \tag{3.36}$$

where $\tilde{\mathcal{O}}$ *hides* $\log(nd/\beta)$ *factor.*

Theorem 3.6.2 (MMS model)**.** *The output of the local mechanism* $\mathcal{R}^{\ell_2}_{v,m,s}$ *can be represented using* $ms(\log(\lceil d/s\rceil) + 1)$ *bits. For every* $n \in \mathbb{N}$*,* $\varepsilon \leq ms$*, and* $\delta \in (0,1)$*, the shuffling the outputs of* n *mechanisms* $\mathcal{R}^{\ell_2}_{v,m,s}$ *satisfies* (ε,δ)*-DP by choosing* $v^2 = \frac{sn\min\{\varepsilon^2,\varepsilon\}}{144\log(1/\delta)}$*. Let* $\hat{\mathbf{x}}$ *be the output of the analyzer* $\mathcal{A}^{\ell_2}$*. With probability at least* $1-\beta$*, the estimator* $\hat{\mathbf{x}}$ *is an unbiased estimate of* $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ *with bounded MSE:*

$$\mathsf{MSE}^{\ell_2}_{MMS} = \tilde{\mathcal{O}}\left(\frac{r_2^2 d}{n^2}\max\left\{\frac{n}{d4^m}, n\left(\frac{1}{s} - \frac{1}{d}\right), \frac{\log(1/\delta)}{\min\{\varepsilon^2,\varepsilon\}}\right\}\right), \tag{3.37}$$

where $\tilde{\mathcal{O}}$ *hides* $\log(nd\beta)$ *factor.*

Remark 3.6.1 (Kashin's representation)**.** Observe that the MSE in (3.36) and in (3.37) is achievable with probability $(1-\beta)$, and has a factor of $(\log(nd/\beta))$ due to the random rotation matrix. We can remove this factor by using the Kashin's representation [Kas77] to transform the bounded ℓ_2-norm vector into a bounded ℓ_∞-norm vector with radius $r_\infty = \frac{cr_2}{\sqrt{d}}$, where c is constant (see e.g., [LV10, CKM18, CKO20]). However, Kashin's representation has large constants in practice [FT21].

Next we present a lower bound for the MSE of the DME under privacy and communication constraints.

Theorem 3.6.3 (Lower Bound For Local DP model)**.** *Let $n, d \in \mathbb{N}$, $\varepsilon_0 > 0$. For any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{B}_p^d(r_p)$ and $p \geq 2$, the MSE is bounded below by:*

$$\mathsf{MSE}_{LDP}^{\ell_p} = \Omega\left(\frac{r_p^2 d^{2-\frac{2}{p}}}{n \min\left\{\varepsilon_0, \varepsilon_0^2\right\}}\right) \tag{3.38}$$

for any unbiased algorithm $\mathcal{R}$ that is ε_0-LDP.

The proof of Theorem 3.6.3 is presented in Section 3.6.2. Observe that Theorem 3.6.3 shows that our achievable MSE in Theorem 3.6.1 and Theorem 3.5.1 are order optimal for all privacy regimes by choosing $s = \max\{1, \varepsilon_0\}$.

Theorem 3.6.4 (Lower Bound For central DP model [CCK22]& [BUV14])**.** *Let $n, d \in \mathbb{N}$, $\varepsilon = \mathcal{O}(1)$, $r_2 \geq 1$, and $\delta = o(\frac{1}{n})$. For any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{B}_2^d(r_2)$, the MSE is bounded below by:*

$$\mathsf{MSE}_{central}^{\ell_2} = \Omega\left(\frac{r_2^2 d}{n^2} \max\left\{\frac{\log(1/\delta)}{\varepsilon^2}, \frac{n}{d4^{b/d}}\right\}\right) \tag{3.39}$$

for any unbiased algorithm $\mathcal{M}$ that is (ε, δ)-DP with $b > d$-bits of communication per client. Furthermore, when $b < d$ bits per client, the MSE is bounded below by:

$$\mathsf{MSE}_{central}^{\ell_2} = \Omega\left(\frac{r_2^2 d}{n^2} \max\left\{\frac{\log(1/\delta)}{\varepsilon^2}, \frac{n}{b}\right\}\right). \tag{3.40}$$

Remark 3.6.2. (Optimality of our mechanism) When the communication budget $b > d$, we can see that our MSE in Theorem 3.6.2 matches the lower bound in Theorem 3.6.4 (up to logarithmic factor) by choosing $s = d$ and $m = b/d$. Furthermore, when the communication budget $b < d$, our algorithm achieve the lower bound by choosing $s = b$ and $m = 1$. Thus, our algorithm for MMS is order optimal for all privacy-communication regimes.

	MMS model (this work)	MMS (Cheu *et al.* [CJM22])	MMS (Chang *et al.* [CGK21])	SecAgg ([KLS21, CCK22])
$d < n\varepsilon^2$	$\mathcal{O}\left(d\log\left(\frac{n\varepsilon^2}{d}\right)\right)$	$\mathcal{O}\left(d\sqrt{n}\right)$	$\mathcal{O}\left(d\log(n)\right)$	$\mathcal{O}\left(d\log(n)\right)$
$n\varepsilon^2 < d < n^2\varepsilon^2$	$\mathcal{O}\left(n\varepsilon^2\log\left(\frac{d}{n\varepsilon^2}\right)\right)$	$\mathcal{O}\left(d\sqrt{n}\right)$	$\mathcal{O}\left(d\log(n)\right)$	$\mathcal{O}\left(d\log(n)\right)$
$d > n^2\varepsilon^2$	$\mathcal{O}\left(n\varepsilon^2\log\left(\frac{d}{n\varepsilon^2}\right)\right)$	$\mathcal{O}\left(d\sqrt{n}\right)$	$\mathcal{O}\left(d\log(n)\right)$	$\mathcal{O}\left(n^2\varepsilon^2\log(d)\right)$

Table 3.1: Comparison on the communication cost of several schemes to design (ε,δ)-DP mechanism achieving MSE $\mathcal{O}\left(\frac{r_2^2 d}{n^2\varepsilon^2}\right)$ for $\varepsilon = \mathcal{O}(1)$.

Algorithm 3.6.2 : Analyzer $\mathcal{A}^{\ell_2}$

```
**Inputs:** $\mathcal{Y}_1,\ldots,\mathcal{Y}_n$, where $\mathcal{Y}_i = \left\{\mathcal{Y}_i^{(1)},\ldots,\mathcal{Y}_i^{(m)}\right\}$ is a set of $m$ sets.
$\hat{\mathbf{w}} \leftarrow \mathcal{A}^{\ell_\infty}\left(\mathcal{Y}_1,\ldots,\mathcal{Y}_n\right)$
**Return:** The server returns $\hat{\mathbf{x}} \leftarrow U^{-1}\hat{\mathbf{w}}$.
```

Remark 3.6.3 (Comparison with SecAgg)**.** When $d < n\varepsilon^2$, our MMS algorithm requires $\mathcal{O}\left(d\log\left(\frac{n\varepsilon^2}{d}\right)\right)$ bits per client to achieve the central DP error $\mathcal{O}\left(\frac{d}{n^2\varepsilon^2}\right)$. Furthermore, it requires only $\mathcal{O}\left(n\varepsilon^2\log\left(\frac{d}{n\varepsilon^2}\right)\right)$-bits when $d > n\varepsilon^2$. In contrast, the DDG algorithm [KLS21] needs $\mathcal{O}\left(d\log\left(n\right)\right)$-bits when $d < n^2\varepsilon^2$ and $\mathcal{O}\left(n^2\varepsilon^2\log\left(d\right)\right)$-bits when $d > n^2\varepsilon^2$ [CCK22] to achieve the same order MSE. Hence, the MMS saves communication in comparison with SecAgg.

In Table 3.1, we present comparison on the communication cost of several schemes in the literature to design (ε,δ)-DP mechanism and to achieve MSE $\mathcal{O}\left(\frac{r_2^2 d}{n^2\varepsilon^2}\right)$ that matches the optimal MSE of the central DP mechanisms. We can see that our proposed mechanism saves a significant amount of communication cost when $d > n\varepsilon^2$ comparing to the MMS schemes in [CJM22, CGK21]. Furthermore, our MMS mechanism saves a gain of $\mathcal{O}\left(n\right)$ of communication cost comparing with the secure aggregation scheme [CCK22] when $d > n\varepsilon^2$.

3.6.1 Proofs of Theorem 3.6.1 and Theorem 3.6.2 (Bounded ℓ_2-norm vectors)

In the mechanism $\mathcal{R}^{\ell_2}_{v,m,s}$, each client applies random rotation to her vector $\mathbf{x}_i$ and then applies the private mechanism $\mathcal{R}^{\ell_\infty}_{v,m,s}$ to the bounded ℓ_∞-norm vector $\mathbf{w}_i$. Hence the communication and privacy are the same as the private mechanism $\mathcal{R}^{\ell_\infty}_{v,m,s}$. Thus, it remains to prove the MSE bound for both local DP model and shuffle model. The proofs are obtained directly from the MSE of the bounded ℓ_∞-norm vector in Theorem 3.5.1 and Theorem 3.5.2 with the following Theorem about the random rotation matrix.

Theorem 3.6.5. *[LSA21] Let $U = \frac{1}{\sqrt{d}}\mathbf{H}D$, where $\mathbf{H}$ denotes Hadamard matrix and D is a diagonal matrix with i.i.d. uniformly ranodom $\{\pm 1\}$ entries. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{B}^d_2(r_2)$ be bounded ℓ_2-norm vectors and*

$mathbfw_i = U\mathbf{x}_i$. With probability at least $1-\beta$, we have that

$$\max_{i\in[n]} \|\mathbf{w}_i\|_\infty = \max_{i\in[n]} \|U\mathbf{x}_i\|_\infty \leq 10r_2\sqrt{\frac{\log(\frac{nd}{\beta})}{d}}. \tag{3.41}$$

From Lemma 3.6.5, the vector $\mathbf{w}_i = U\mathbf{x}_i$ is bounded ℓ_∞-norm of radius $r_\infty = 10r_2\sqrt{\frac{\log(\frac{nd}{\beta})}{d}}$ with probability at least $1-\beta$. Hence, by plugging the radius $r_\infty = 10r_2\sqrt{\frac{\log(\frac{nd}{\beta})}{d}}$ into Theorem 3.6.1, we obtained the MSE in Theorem 3.6.1. Similarly, by plugging the radius $r_\infty = 10r_2\sqrt{\frac{\log(\frac{nd}{\beta})}{d}}$ into Theorem 3.5.2, we obtained the MSE in Theorem 3.6.2.

3.6.2 Lower Bound on MSE for ℓ_2-norm under LDP constraints

In this section, we prove Theorem 3.6.3. 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 $\mathcal{P}^d_p$ denote the set of all distributions on the ℓ_p-norm ball $\mathbb{B}^d_p$. 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 MSE when the input distribution belongs to $\mathcal{P}^{\text{Bern}}_{p,d}$.

Lemma 3.6.1. *For any $p \in [2, \infty]$, we have*

$$\begin{aligned}&\inf_{\mathcal{R}\in\mathcal{Q}_{\varepsilon_0}} \inf_{\hat{\mathbf{x}}} \sup_{\boldsymbol{q}\in\mathcal{P}_{p,d}^{Bern}} \mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \hat{\mathbf{x}}\right\|_2^2 \\ &\quad \geq \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\varepsilon_0, \varepsilon_0^2\}}\right\}\right).\end{aligned} \tag{3.42}$$

The proof of Lemma 3.6.1 is presented in Appendix B.1. In order to use Lemma 3.6.1, first observe that for every $\boldsymbol{x} \in \mathcal{P}_{p,d}^{\text{Bern}}$, we have $\|\boldsymbol{x}\|_p \leq 1$, which implies that $\boldsymbol{x} \in \mathcal{P}_p^d$. For given $\{\boldsymbol{x}_i\} \in \mathbb{B}_p^d$, let $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. Thus we have $\mathcal{P}_{p,d}^{\text{Bern}} \subset \mathcal{P}_p^d$. Now our bound on MSE follows from the following inequalities:

$$\begin{aligned}&\sup_{\{\boldsymbol{x}_i\}\in\mathbb{B}_p^d} \mathbb{E}\left\|\overline{\mathbf{x}} - \widehat{\boldsymbol{x}}\right\|_2^2 \overset{\text{(a)}}{\geq} \sup_{\boldsymbol{q}\in\mathcal{P}_{p,d}} \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \widehat{\boldsymbol{x}}\right\|_2^2 \\ &\overset{\text{(b)}}{\geq} \sup_{\boldsymbol{q}\in\mathcal{P}_{p,d}} \frac{1}{2}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\right\|_2^2 - \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \\ &\overset{\text{(c)}}{\geq} \sup_{\boldsymbol{q}\in\mathcal{P}_{p,d}^{\text{Bern}}} \frac{1}{2}\mathbb{E}\left\|\boldsymbol{\mu}_{\boldsymbol{q}} - \widehat{\boldsymbol{x}}\right\|_2^2 - \frac{d^{1-\frac{2}{p}}}{n} \\ &\overset{\text{(d)}}{\geq} \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\varepsilon_0, \varepsilon_0^2\}}\right\}\right) - \frac{d^{1-\frac{2}{p}}}{n} \quad \overset{\text{(e)}}{\geq} \Omega\left(d^{1-\frac{2}{p}} \min\left\{1, \frac{d}{n\min\{\varepsilon_0, \varepsilon_0^2\}}\right\}\right)\end{aligned} \tag{3.43}$$

In the LHS of (a), the expectation is taken over the randomness of the mechanism $\mathcal{R}$ 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 $\mathbb{B}_p^d$ 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 $\boldsymbol{u} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^{(q)} - \widehat{\boldsymbol{x}}$ 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. (d) follows from Lemma 3.6.1. In (e), we assume $\min\{\varepsilon_0, \varepsilon_0^2\} \leq \mathcal{O}(d)$.

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

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_i^{(q)} - \boldsymbol{\mu}_{\boldsymbol{q}}\right\|_2^2 \\ &\overset{\text{(a)}}{\leq} \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_i^{(q)}\right\|_2^2 \leq \frac{1}{n^2}\sum_{i=1}^{n} d^{1-\frac{2}{p}} = \frac{d^{1-\frac{2}{p}}}{n},\end{aligned} \tag{3.44}$$

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 $\boldsymbol{x}$. This completes the proof of Theorem 3.6.3.

3.7 DME under User-Level Privacy

In the previous sections, we focus on making neighboring datasets indistinguishable, where two datasets are neighbors if they differ in a single data point at a single user. This is called *item-level* DP. However, in distributed systems, each client might have more than one data point. Furthermore, a client may not even want to reveal whether it participated or not, which is equivalent to requiring the privacy of its entire local dataset (not just of a single data point). This is called *user-level* DP, which has recently seen some attention [MAE18, LSY20, WSZ19, LSA21, GKM21a].

In this section, we study distributed mean estimation under user-level local differential privacy. Consider a set of n users, each having a local dataset of m samples. Let $\mathcal{D}_i = \{x_1^{(i)}, \ldots, x_m^{(i)}\}$ denote the local dataset at the i-th user for $i \in [n]$, where $x_j^{(i)} \in \mathcal{X}$ and $\mathcal{X} \subset \mathbb{R}^d$. We define $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_n) \in (\mathcal{X}^m)^n$ as the entire dataset. The server wants to estimate the mean $\overline{x} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} x_j^{(i)}$. Users want to preserve the privacy of their local datasets while minimizing the worst-case expected error for estimating $\overline{x}$. We first define the difference between user-level and item-level privacy. We say that two datasets $\mathcal{D}$, $\mathcal{D}'$ are neighboring with respect to distance metric dis if we have $\mathsf{dis}(\mathcal{D}, \mathcal{D}') \leq 1$.

Definition 3.7.1. (Differential Privacy) Let $\varepsilon, \delta \geq 0$. A randomized mechanism $\mathsf{M} : \mathcal{D} \to \Theta$ is said to be (ε, δ)-DP with respect to dis if for any neighboring datasets $\mathcal{D}, \mathcal{D}'$ and any measurable set $\theta \subseteq \Theta$, we have

$$\Pr\left(\mathsf{M}\left(\mathcal{D}\right) \in \theta\right) \leq e^{\varepsilon} \Pr\left(\mathsf{M}\left(\mathcal{D}'\right) \in \theta\right) + \delta. \tag{3.45}$$

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

Remark 3.7.1 ((Central) item-level DP vs (central) user-level DP [LSA21]). When we have more than one user (i.e., $n > 1$) and a space $\mathcal{D} \triangleq \left(\mathcal{X}^m\right)^n$, by choosing $\mathsf{dis}\left(\mathcal{D}, \mathcal{D}'\right) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}\{x_j^{(i)} \neq x_j'^{(i)}\}$, we recover the standard definition of the DP [DMN06, DR14] (see also Definition 2.1.2), which we call *(central) item-level* DP. In the central item-level DP, two datasets $\mathcal{D}$, $\mathcal{D}'$ are neighboring if they differ in a single item. On the other hand, by choosing $\mathsf{dis}\left(\mathcal{D}, \mathcal{D}'\right) = \sum_{i=1}^{n} \mathbb{1}\{\mathcal{D}_i \neq \mathcal{D}_i'\}$, we call it *(central) user-level* DP, where two datasets $\mathcal{D}, \mathcal{D}' \in \left(\mathcal{X}^m\right)^n$ are neighboring when they differ in a local dataset of any single user. Observe that when each user has a single item ($m = 1$), then both item-level and user-level privacy are equivalent.

Remark 3.7.2 (User-level Local Differential Privacy (LDP)). When we have a single user (i.e., $n = 1$ and $\mathcal{D} = \mathcal{X}^m$), by choosing $\mathsf{dis}\left(\mathcal{D}, \mathcal{D}'\right) = \mathbb{1}\{\mathcal{D} \neq \mathcal{D}'\}$ for $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^m$, we call it *user-level LDP*. In this case each user privatize her own local dataset using a private mechanism.

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

$$R_{\varepsilon,\delta} = \inf_{\{\mathsf{M}_i \in \mathcal{M}_{\varepsilon,\delta}\}} \inf_{\hat{x}} \sup_{\mathcal{D} \in (\mathcal{X}^m)^n} \mathbb{E}\left[\|\hat{x} - \overline{x}\|^2\right], \tag{3.46}$$

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

We can obtain user-level DP from item-level DP by using *group privacy* [DR14], but this degrades the privacy parameter by a multiplicative factor of the number of data points in a local dataset (m), which may be impractical. We can achieve a significantly better user-level privacy guarantees by assuming concentration of data points [LSA21], which essentially reduces their sensitivity and thereby the required noise magnitude. Now, we define the concentration condition for a set of samples and the sub-Gaussian random vector.

Definition 3.7.2 (Concentration)**.** A set of (random) vectors $y^n = (y_1, \ldots, y_n)$, each taken from $[-B, B]^d$ is (τ, γ)-concentrated if there exists $y_0 \in [-B, B]^d$ such that with probability at least $1 - \gamma$,

$$\max_{i \in [n]} \|y_i - y_0\|_2 \leq \tau. \tag{3.47}$$

Definition 3.7.3 (Sub-Gaussian random vector)**.** A random vector $x \in \mathbb{R}^d$ is said to be sub-Gaussian with proxy variance σ^2 if for any $u \in \mathbb{R}^d$ with $\|u\|^2 = 1$, the random variable $u^T x$ is sub-Gaussian with proxy variance σ^2.

We assume that the samples $\{x_j^{(i)} : i \in [n], j \in [m]\}$ are drawn from a bounded space $\mathcal{X} \triangleq [-B, B]^d \subset \mathbb{R}^d$ for some $d \geq 1$. Furthermore, we assume that the samples $x_j^{(i)}, i \in [n], j \in [m]$ are i.i.d. sub-Gaussian random vectors with proxy variance σ^2. We focus on the scalar case when $d = 1$, where the vector case can be obtained by applying our scalar scare coordinate-wise.

3.7.1 Scalar Case $d = 1$

The main idea of our algorithm is the following. Let $y_i = \frac{1}{m}\sum_{j=1}^{m} x_j^{(i)}$ denote the mean of the local samples at the i-th user for $i \in [n]$. Observe that the worst-case sensitivity of replacing a single client is $\max_{y_i, y_i' \in [-B,B]} |y_i - y_i'| = 2B$. However, since the data $\{x_j^{(i)}\}$ are i.i.d. sub-Gaussian, $\{y_i\}$ are sub-Gaussian random variables with proxy $\frac{\sigma^2}{m}$ which implies that the set $y^n = (y_1, \ldots, y_n)$ is (τ, γ)-concentrated, where $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$ for any $\gamma \in (0, 1)$ (e.g., see [RH15, Theorem 1.14]). Thus, the worst-case sensitivity of replacing a user is reduced to $2\tau = \mathcal{O}\left(\sqrt{\frac{\sigma^2}{m}}\right)$ instead of $2B$ that decreases as the number of local samples m increases.

The mean estimation process works in two stages similar to [LSA21]. In the first stage, the server privately estimates the range in which the means $y_1, \ldots, y_n$ lie with high probability.

Algorithm 3.7.1 $\mathsf{Range}^{\text{user}}_{\text{scalar}}\left(\mathcal{D}, \tau, \varepsilon_0, \mathcal{T}\right)$

```
1: Inputs: $\mathcal{D} = (x_1, \ldots, x_m)$, $x_j \in [-B, B]$; concentration radius $\tau$; user-level LDP parameter $\varepsilon_0$; $\mathcal{T} = [k]$ be the set of middle points of the intervals.
2: Compute $y = \frac{1}{m}\sum_{j=1}^{m} x_j$.
3: Compute $\nu = \arg\min_{j\in[k]} |y - a_j|$ (the index of a point in $\mathcal{T}$ closest to $y$).
4: Let $\mathbf{H}_k$ be Hadamard matrix.
5: Compute $\mathbf{m} = \frac{1}{\sqrt{k}}\mathbf{H}_k e_\nu$, where $e_\nu$ denotes the basis vector corresponding to $\nu$.
6: Sample $j \sim \mathsf{Unif}[k]$ and compute $\mathbf{z}$:
       $\mathbf{z} = \begin{cases} +\mathbf{H}_k(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} + \frac{\sqrt{k}\mathbf{m}(j)}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1} \\ -\mathbf{H}_k(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) & \text{w.p. } \frac{1}{2} - \frac{\sqrt{k}\mathbf{m}(j)}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1} \end{cases}$
7: Return: $\mathbf{z}$
```

In the second stage, each user projects her mean value y_i into the determined range from the first step. Then, all users send user-level LDP versions of their projected samples to the central server. The first stage mechanism is denoted by $\mathsf{Range}_{\text{scalar}}$ and is presented in Algorithm 3.7.2, and the second stage mechanism is denoted by $\mathsf{Mean}_{\text{scalar}}$ and is presented in Algorithm 3.7.3. We give an outline of both these algorithms below.

In $\mathsf{Range}_{\text{scalar}}$, we first divide the original range $[-B, B]$ into $k = B/\tau$ bins, where τ is the concentration parameter of $y_1, \ldots, y_n$. Then, each user sends a private version of the closest bin to her mean value y_i (using the mechanism $\mathsf{Range}^{\text{user}}_{\text{scalar}}$ as described in Algorithm 3.7.1). The server estimates the frequencies (the number of means close to each bin) under user-level LDP constraints. We use a Hadamard Response mechanism similar to the one proposed in [ASZ19] to estimate the highest frequency under user-level LDP constraints. Observe that if the means $(y_1, \ldots, y_n)$ lie in radius τ and the server succeeds to estimate the highest frequency correctly, then we get $y_i \in R \triangleq [a_{\max} - 3\tau, a_{\max} + 3\tau]$ for all $i \in [n]$. In $\mathsf{Mean}_{\text{scalar}}$, each client projects her mean y_i onto the estimated range R from the first stage. The objective of this projection is that the user-level sensitivity will decrease from $2B$ to 2τ, where $\tau = \mathcal{O}(\frac{1}{\sqrt{m}})$. In

Algorithm 3.7.2 $\mathsf{Range}_{\text{scalar}}\left(\mathcal{D},\tau,\varepsilon_0\right)$: Distributed Private Range Estimation for Scalars

```
Inputs: $\mathcal{D}=(\mathcal{D}_1,\ldots,\mathcal{D}_n)$, $\mathcal{D}_i=(x_1^{(i)},\ldots,x_m^{(i)})$, $x_j^{(i)}\in[-B,B]$, concentration radius $\tau$, and user-level LDP parameter $\varepsilon_0$.
All users divide the interval $[-B,B]$ into $k=B/\tau$ disjoint intervals, each with width $2\tau$. Let $\mathcal{T}:=\{1,2,\ldots,k\}$ be the set of middle points of intervals.
for User $i\in[n]$ do
    $\mathbf{z}_i\leftarrow\mathsf{Range}^{\text{user}}_{\text{scalar}}\left(\mathcal{D}_i,\tau,\varepsilon_0,\mathcal{T}\right)$.
    Send $\mathbf{z}_i$ to the server – here $\mathbf{z}_i\in\mathbb{R}^k$.
The server computes $\overline{\mathbf{z}}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i$. (Here, for any $a\in\mathcal{T}$, $\overline{\mathbf{z}}(a)$ denotes an estimate of the frequency of $a$, i.e., the fraction of $y_i$'s that are closest to $a$).
Let $a_{\max}=\arg\max_{a\in\mathcal{T}}\overline{\mathbf{z}}(a)$.
Return: $R=[a_{\max}-3\tau,a_{\max}+3\tau]$
```

other words, the user-level sensitivity will decrease by increasing the number of samples per user using this projection step. After the projection, each user applies any LDP mechanism $\mathcal{R}$ with user-level sensitivity (τ) and LDP parameter $\varepsilon_0/2$ to preserve privacy.

Theorem 3.7.1. *Let $\mathcal{R}$ be an unbiased LDP mechanism with MSE $\mathbb{E}\left[\|\mathcal{R}(y)-y\|^2\right]\leq f(\tau,\varepsilon_0)$. The mechanism $\mathsf{Mean}_{\text{scalar}}\left(\mathcal{D},\tau,\varepsilon_0,\delta\right)$ is user-level (ε_0,δ)-LDP. With probability at least $1-\beta$, we have*

$$\mathcal{E}_1:=\mathbb{E}\left[\left|\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}x_j^{(i)}-\mathsf{Mean}_{\text{scalar}}\left(\mathcal{D},\tau,\varepsilon_0,\delta\right)\right|^2\right] \tag{3.48}$$

$$\leq\mathcal{O}\left(\frac{f(\tau,\varepsilon_0)}{n}\right), \tag{3.49}$$

where $\beta=\min\left\{1,\gamma+\frac{2B}{\tau}\exp\left(-\frac{n(e^{\varepsilon_0/2}-1)^2}{200(e^{\varepsilon_0/2}+1)^2}\right)\right\}$.

We provide a proof of Theorem 3.7.1 in Section 3.7.2. Theorem 3.7.1 is presented for any general LDP mechanism $\mathcal{R}$. We can apply our proposed local randomizer in Section 3.6 that

Algorithm 3.7.3 $\mathsf{Mean}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0, \delta)$: Distributed Private Mean Estimation for Scalars

```
**Inputs:** $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_n)$, $\mathcal{D}_i = (x_1^{(i)}, \dots, x_m^{(i)})$, $x_j^{(i)} \in [-B, B]$, concentration radius $\tau$, and user-level LDP parameters $\varepsilon_0, \delta$.
$[a, b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0/2)$ (Algorithm 3.7.2).
**for** User $i \in [n]$ **do**
    $z_i \leftarrow \mathsf{Mean}^{\text{user}}_{\text{scalar}}\left(\mathcal{D}_i, [a, b], \frac{\varepsilon_0}{2}, \delta\right)$
**Return:** $\hat{x} = \frac{1}{n}\sum_{i=1}^{n} z_i$.
```

Algorithm 3.7.4 $\mathsf{Mean}^{\text{user}}_{\text{scalar}}(\mathcal{D}, [a, b], \varepsilon_0, \delta)$

```
**Inputs:** $\mathcal{D} = (x_1, \dots, x_m)$, concentration range $[a, b]$, and user-level LDP parameters $\varepsilon_0, \delta$.
**Return:** $z = \mathcal{R}\left(\prod_{[a,b]} y\right)$, where $y = \frac{1}{m}\sum_{j=1}^{m} x_j$ and $\prod_{[a,b]}$ is the projection operator onto $[a, b]$.
```

has MSE $f(\tau, \varepsilon_0) = \frac{\tau^2}{\min\{\varepsilon_0^2, \varepsilon_0\}}$. Thus, the MSE is bounded by $\mathcal{O}\left(\frac{\tau^2)}{n\min\{\varepsilon_0^2, \varepsilon_0\}}\right)$ with probability at least $1 - \beta$.

Remark 3.7.3 (User-level LDP vs user-level DP)**.** In [LSA21], the authors proposed a (central) user-level DP mean estimation algorithm that achieves estimation error $\mathcal{O}(\frac{\tau^2}{n^2\varepsilon^2})$ with probability $(1 - \beta_c)$, where $\beta_c = \min\{1, \gamma + \frac{B}{\tau}e^{-\frac{n\varepsilon}{8}}\}$ and ε is the (central) DP parameter. Although, the confidence probability $1 - \beta$ is almost same for both user-level LDP and user-level DP, it is clear that there is a gap of $\mathcal{O}(n)$ in the estimation error between the central and the local models. This is not surprising as the same gap appears in the item-level DP and LDP as well [BNO08, CSS12]. In order to amplify the privacy of the user-level LDP to match with that of the user-level DP, we can assume the existence of a trusted shuffler [EFM19, FMT22, GDD21e] or secure aggregation [KLS21] between the users and the untrusted server.

Remark 3.7.4 (Vector case)**.** We can extend our results for the vector case as follows. We follow similar steps as in the centralized Algorithm presented in [LSA21] for user-level DP mean estimation. The idea of the private mean estimation Algorithm is to observe that the means $y_1, \ldots, y_n$ are concentrated in ℓ_2-norm with radius τ. Similar to [LSA21], we first apply an encoding step to bound them in ℓ_∞-norm with radius $\mathcal{O}(\frac{\tau}{\sqrt{d}})$. This step can be obtained by applying a random rotation as in [SFK17, LSA21] or by applying Kashin's representation as in [CKM18, CKO20]. Then, we apply the scalar Algorithm 3.7.2 for each coordinate separately.

3.7.2 Proofs of The Scalar Case

In this section, we prove Theorem 3.7.1 for the scalar case. The algorithm $\mathsf{Mean}_{\text{scalar}}$ is composed of two sub-routines $\mathsf{Range}_{\text{scalar}}$ and $\mathsf{Mean}_{\text{scalar}}^{\text{user}}$. In order to show that $\mathsf{Mean}_{\text{scalar}}$ satisfies user-level ε_0-LDP, it suffices to prove that $\mathsf{Range}_{\text{scalar}}$ satisfies user-level $\varepsilon_0/2$-LDP and $\mathsf{Mean}_{\text{scalar}}^{\text{user}}$ satisfies user-level $\varepsilon_0/2$-LDP, and then the result follows by composing these two mechanisms.

- $\mathsf{Range}_{\text{scalar}}$ is user-level $\varepsilon_0/2$-LDP: We show this along with other results that will be useful to bound the error in the following lemma which is proved in Appendix B.3.

Lemma 3.7.1. $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0)$ *is user-level* ε_0*-LDP. Furthermore, if the samples* $x_j^{(i)}$ *are sub-Gaussian with proxy* σ^2*, then with probability at least* $1-\beta$*, we have*

$$y_i \in [a, b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0) \qquad \forall i \in [n] \tag{3.50}$$

where $y_i = \frac{1}{m}\sum_{j=1}^{m} x_j^{(i)}$ *is the average of local samples at the i-th user, and* $\beta = \min\left\{1, \gamma + \frac{2B}{\tau}\exp\left(-\frac{n(e^{\varepsilon_0/2}-1)^2}{200(e^{\varepsilon_0/2}+1)^2}\right)\right\}$.

This lemma shows that with probability at least $1-\beta$, the server can privately estimate an interval of length 6τ in which the averages $y_1, \ldots, y_n$ of local samples at all users lie. Thus, each user can project the average of her local samples onto this interval without hurting the estimation accuracy of the second stage. Furthermore, the sensitivity of replacing a user with another one would be $6\tau = \mathcal{O}(\frac{1}{\sqrt{m}})$ instead of $2B$. As a result, each user adds a noise as a function of τ that reduces the estimation error.

- $\mathsf{Mean}_{\text{scalar}}^{\text{user}}$ is user-level $\varepsilon_0/2$-LDP: Consider any two neighboring local datasets $\mathcal{D}_i = (x_1^{(i)}, \ldots, x_m^{(i)})$, $\mathcal{D}'_i = (x_1'^{(i)}, \ldots, x_m'^{(i)})$. Let $y_i = \frac{1}{m}\sum_{j=1}^{m} x_j^{(i)}$ denotes the average of local samples in $\mathcal{D}$; similarly define y'_i. The user-level sensitivity for computing its projection $\prod_{[a,b]} y_i$ is bounded by

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

From the assumption that the local mechanism $\mathcal{R}$ is ε_0-LDP, then the mechanism $\mathsf{Mean}_{\text{scalar}}^{\text{user}}$ satisfies user-level $\varepsilon_0 2$-LDP for given any neighboring points $y_i, y'_i \in [a,b]$.

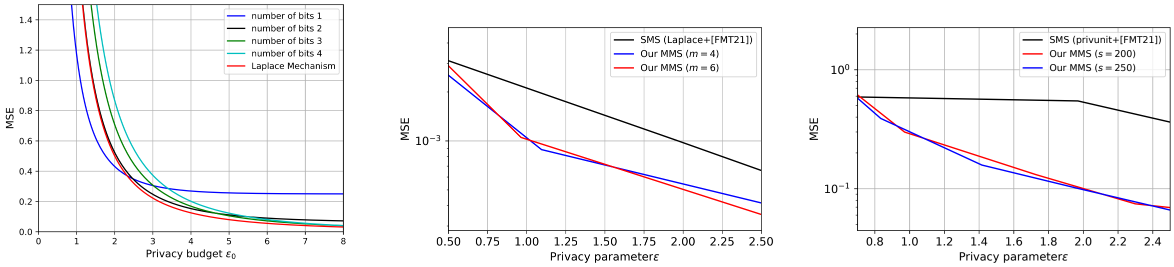
- Bounding the error of $\mathsf{Mean}_{\text{scalar}}$: Let $[a,b] \leftarrow \mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0/2)$ and $\tilde{y}_i = \Pi_{[a,b]} y_i$. Note that $(b-a) = 6\tau$. Let $\hat{x} = \frac{1}{n}\sum_{i=1}^{n} y_i$ be the estimator of the exact mean $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} y_i$, where $z_i = \mathcal{R}(\tilde{y}_i)$. Thus, we have

$$\begin{aligned} \mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\tilde{y}_i - \frac{1}{n}\sum_{i=1}^{n} z_i\right|^2\right] &= \mathbb{E}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\mathcal{R}(\tilde{y}_i) - \tilde{y}_i\right|^2\right] \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\left|\mathcal{R}(\tilde{y}_i) - \tilde{y}_i\right|^2\right] \leq \frac{f(\tau, \varepsilon_0)}{n}, \end{aligned} \tag{3.51}$$

where the last inequality is obtained from the assumption that the mechanism $\mathcal{R}$ has MSE $f(\tau, \varepsilon_0)$. From Lemma 3.7.1, we have that $y_i = \tilde{y}_i$ for all $i \in [n]$ with probability at least $1-\beta$. Thus, we get that, with probability at least $1-\beta$, the error $\mathcal{E}_1$ (defined in (3.49)) is bounded by $\mathcal{E}_1 = \mathcal{O}\left(\frac{f(\tau, \varepsilon_0)}{n}\right)$. This completes the proof of Theorem 3.7.1.

3.8 Numerical Results

In this section, we compare the performance of our proposed algorithm with the Laplace mechanism in the LDP model. Furthermore, we compare our algorithms for multi-message shuffled model with the best known algorithms for the single-message shuffled model for both scalar and vector summation.



(a) Comparison of our LDP mechanism $\mathcal{R}_{v,m,s}^{\ell\infty}$ with Laplace mechanism for $d=1$, $n=1$, and $m \in \{1,2,3,4\}$.

(b) Comparison of our MMS mechanism $\mathcal{R}_{v,m,s}^{\ell\infty}$ with SMS (Laplace+[FMT21]) for $d=1$, $n=1000$, and $m \in \{4,6\}$.

(c) Comparison of our MMS mechanism $\mathcal{R}_{v,m,s}^{\ell 2}$ with SMS (privunit+[FMT21]) for $d=300$, $n=1000$.

Figure 3.2: Evaluating the performance of the proposed private DME algorithms.

Local DP model: We start by comparing the performance of our algorithm $\mathcal{R}_{v,m,s}^{\ell\infty}$ with the performance of the Laplace mechanism [DMN06] in the local model for scalar case, i.e., $d=1$, where the elements $\mathbf{x}_i \in [-r_\infty, r_\infty]$ and $r_\infty = 0.5$. Observe that the Laplace mechanism has infinite communication bits. In Figure 3.2a, we plot the MSE of our $\mathcal{R}_{v,m,s}^{\ell\infty}$ with different communication budget $s=1$ and $m \in \{1,2,3,4\}$ for a single client $n=1$. We can observe that our mechanism achieves MSE closer to the MSE of the Laplace mechanism. Furthermore, we only need at most $m=3$ bits to achieve similar performance as Laplace mechanism.

Shuffled model: We consider two cases in the shuffler model: 1) The scalar case when $d=1$ to evaluate the performance of our $\mathcal{R}_{v,m,s}^{\ell\infty}$ mechanism in the multi-message shuffled model. 2) The vector case when $d=1000$ to evaluate the performance of our $\mathcal{R}_{v,m,s}^{\ell 2}$ mechanism in the multi-message shuffled model.

In Figure 3.2b, we plot the MSE of two different mechanisms versus the central privacy ε for fixed $\delta = 10^{-5}$. The first mechanism is single message shuffled (SMS) model obtained using Laplace mechanism with privacy amplification results in [FMT22]. Observe that Laplace mechanism is the optimal LDP mechanism for $\varepsilon_0 = \mathcal{O}(1)$ and the privacy amplification results in [FMT22] is approximately optimal for computing the (ε, δ)-DP of the shuffled model. Hence, we expect that this is the best that an SMS mechanism can achieve. The second mechanism is our multi-message shuffled (MMS) mechanism $\mathcal{R}_{v,m,s}^{\ell_\infty}$ mechanism for $d = 1$ and $m \in \{4, 6\}$. Since we have MMS, we use our RDP results of privacy amplification by shuffling in Chapter 5 which is better for composition to compute the RDP of our mechanism. Then, we transform from RDP bound to approximate (ε, δ)-DP. We choose number of clients $n = 1000$. We can see that our multi-message shuffled algorithm achieve lower MSE than the single message shuffled especially for large value of central DP parameter ε.

Similar to the scalar case, we consider two mechanisms. The first mechanism SMS is obtained by using `privunit` mechanism with the privacy amplification results in [FMT22], where `privunit` [BDF18] is asymptotically optimal LDP mechanism [AFT22]. We choose $n = 1000$ and $d = 300$. For our MMS $\mathcal{R}_{v,m,s}^{\ell_2}$, we choose $s \in \{200, 250\}$. It is clear from Figure 3.2c that our MMS mechanism has better performance compared to SMS mechanism.

3.9 Related Work

Distributed mean estimation has received considerable attention due to its broad applications in distrusted learning and statistics. We briefly review some of the main developments on this topic below.

Local differential privacy There has been significant recent progress in communication-efficient distributed mean estimation (see [SFK17, AGL17, SCJ18, BDK19] and references therein). However, their algorithms do not provide privacy guarantees. Furthermore, there are multiple works addressing differentially private mean estimation [DMN06, DR14], however, their algorithms are not communication-efficient. There has been less work in combining privacy and compression in distributed mean estimation. Agarwal et al. proposed in [ASY18] a communication-efficient and private algorithm for mean estimation. Their algorithm cp-sgd is based on a Binomial noise addition mechanism that requires $\mathcal{O}\left(d\log(d)\right)$-bits per client. In contrast, we show that we can achieve the optimal MSE with only $\mathcal{O}\left(\log(d)\right)$-bits per client in the high privacy regime. In [CKO20], Chen *et al.* established the order optimal private DME under LDP constraints for bounded ℓ_2-norm vectors. This work is is done concurrently and independently of our work in [GDD21d].

Multi-message shuffled model For single-message shuffled model, Balle *et al.* presented lower and matching upper bounds for the scalar private real summation, showing that the MSE is order $\Theta\left(n^{1/3}\right)$, where n denotes the number of clients. This was further enhanced by using multi-message shufflers in [BBG20b, GKM20]. A multi-message shuffling (MMS) mechanism based on IKOS scheme [IKO06] was proposed in [BBG20b, GKM20] for scalar summation in which each client needs to send only $\mathcal{O}(1)$ messages to the shuffler, each of size $\mathcal{O}(\log(n))$ bits. The private vector DME has received less attention in the shuffled model. In [CJM22], a MMS mechanism for vector summation is proposed which has $\mathcal{O}(d\sqrt{n})$ communication bits per client, where d is the vector dimension. In [CGK21], a MMS mechanism for vector summation in MMS model is proposed that requires $\mathcal{O}\left(d\log(n)\right)$-bits of communication per client. In this work, we establish the fundamental privacy-communication-performance trade-offs for computing *vector sum* in the MMS model. Our private vector DME results in Theorem 3.6.2 improves the privacy-communication-performance order-wise, see Table 3.1 for comparison.

User-level differential privacy There has been a lot of recent work in applying *item-level* DP to distributed mean estimation, and much less work on user-level privacy, with notable exceptions in [MAE18, LSY20, WSZ19, LSA21, GKM21a]. Our algorithms are inspired from that in [LSA21], but with an important distinction that [LSA21] only provide user-level *central* DP guarantees, whereas, our algorithms provide user-level *local* DP guarantees; in distributed learning with an untrusted server, clients need local DP guarantees. Our algorithm is based on distributed private heavy-hitter estimation, whereas the algorithm in [LSA21] is based on estimating the median-based mechanism.

CHAPTER 4

Differentially Private Federated Learning

In this chapter, we consider federated learning (FL) framework [Kai19] with communication efficiency and privacy requirements. Unique challenges to the traditional empirical risk minimization (ERM) problem in the context of FL include (i) need to provide privacy guarantees on clients' data, (ii) compress the communication between clients and the server, since clients might have low-bandwidth links, (iii) work with a dynamic client population at each round of communication between the server and the clients. We exploit the communication-efficient schemes for private mean estimation proposed in Chapter 3 to enable efficient gradient aggregation for each iteration of federated learning. To get the overall communication, privacy, and optimization performance operation point, we combine this with privacy amplification opportunities inherent to this setup. Our solution takes advantage of the inherent privacy amplification provided by client sampling and data sampling at each client (through Stochastic Gradient Descent) as well as privacy amplification via shuffling. Putting these together, we demonstrate that one can get the same privacy, optimization-performance operating point developed in recent methods that use full-precision communication, but at a much lower communication cost, *i.e.,* effectively getting communication efficiency for "free". We then propose a statistical framework that unifies several personalized federated learning algorithms as well as suggests new algorithms. We develop personalized learning methods with guarantees for user-level privacy and composition. We numerically evaluate the performance of our proposed FL algorithms, demonstrating the advantages of our proposed methods.

4.1 Introduction

Federated learning (FL) is a distributed system approach to build machine learning models from multiple clients without directly sharing the local data [MMR17, Kai19]. In standard FL algorithms, the central server sends the global model to a set of sampled clients at each round. The server aggregates the local updates (stochastic gradients) of the participated clients to update the global model towards the next round. In FL, communication becomes a bottleneck for training high dimensional model as the communication is performed over a limited-bandwidth networks [Kai19, LAS14, BWA18]. To address this challenge, there are several works for designing communication-efficient FL algorithms [AGL17, SCJ18]. Besides communication, the clients' data might contain sensitive information, and hence, each client wants to preserve privacy of her own local data. Although, the local data doesn't leave the client's device, FL algorithm cannot provide a provable privacy guarantees, where sensitive data can be reconstructed from observing the global model and/or the local updates [ZLH19, GBD20, CTW21, SSS17]. Thus, providing privacy guarantees for FL algorithms has received a considerable attention from academia as well as industry [CMS11, BST14, KMS21, GDD21a, KLS21, SFK17, ACG16]. The goal of this chapter is to design communication-efficient and private mechanisms for federated learning in the LDP and the multi-message shuffled models.

We propose a communication-efficient and private federated learning algorithm (CLDP-SGD) that enables privacy amplification using both forms of amplification: shuffling and sampling (data and clients). Note that privacy amplification by subsampling (both data and clients) happens automatically while the secure shuffling (anonymization) is performed explicitly which adds an additional layer of privacy that allows transferring the local privacy guarantees to central privacy guarantees. We analyze the convergence-privacy trade-offs of the proposed CLDP-SGD algorithm for Lipschitz convex function under several ℓ_p geometries. We prove that one can get communication efficiency "for free" by using the communication-efficient schemes for private mean estimation proposed in Chapter 3. 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 our privacy proof, of Lemma 4.3.1 in Appendix C.1, that composes sampling and shuffling techniques.

We extend our work to explore a distributed self-sampling approach initiated by the clients that does not need a selection by the shuffler. Self-sampling is desirable from a system-level perspective, where coordination is not needed in order to randomly sub-sample which clients will participate in each iteration of the stochastic gradient descent (SGD) algorithm. At each iteration of the training process, clients independently toss a biased coin. If the biased coin of a client turns a head, that client participates in the current iteration and share its model privately with the untrusted server. One of the main challenges in our self-sampling scheme is that the number of participated clients at each iteration is unknown a priori as it is random varying from iteration to iteration. We analyze the privacy of our self-sampling scheme by composing amplification by sub-sampling along with amplification by shuffling. Furthermore, we analyze the convergence rate of the SGD with client self-sampling and shuffling.

Due to the statistical heterogeneity of local data, a single global learning model may perform poorly for individual clients for some applications. This motivates the need for personalized learning achieved through collaboration, and there have been a plethora of personalized models proposed in the literature as well [FMO20, DTN20, DKM20, MMR20, AZZ21, LHB21, ZSF21, HGL20]. However, the proposed approaches appear to use very different forms and methods, and there is a lack of an underlying the fundamental statistical framework. Such a statistical framework could help develop theoretical bounds for performance, suggest new algorithms as well as perhaps give grounding to known methods. We develop a statistical framework for personalized federated learning that leads to new algorithms with provable privacy guarantees, and performance bounds. We numerically evaluate the performance of our proposed private algorithms for real data.

Organization The remainder of this chapter is organized as follows. We present the problem setup in Section 4.2. We present a communication-efficient and private CLDP-SGD algorithm for federated learning and its performance in Section 4.3. We give private federated learning algorithm with client self-sampling in Section 4.4. We present personalized federated learning algorithms under user-level privacy requirements in Section 4.5. We give numerical results evaluating the performance of our proposed schemes in Section 4.6. Some proofs are delegated to Appendix C.

4.2 Problem Formulation

We study federated learning (FL) framework.: We have a set of n clients, where each client has a local dataset $\mathcal{D}_i = \{d_{i1}, \ldots, d_{im}\}$ comprising m data points drawn from a universe $\mathcal{X}$. Let $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ denote the entire dataset and $r = mn$ denote the total number of data points in the system. The clients are connected to an untrusted server in order to solve the following empirical risk minimization (ERM) problem:

$$\min_{\theta} F\left(\theta, \mathcal{D}\right) = \frac{1}{n}\sum_{i=1}^{n} F_i(\theta, \mathcal{D}_i), \tag{4.1}$$

where $\theta \in \mathbb{R}^d$ denotes the global model. $F_i\left(\theta, \mathcal{D}_i\right) = \mathbb{E}_{d_i \sim \mathcal{D}_i}\left[F_i\left(\theta, d_i\right)\right]$ denotes the loss function of the i-th client. Our goal is to solve (4.1) 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. We consider two distributed privacy models, where the server is untrusted: (i) Local DP (LDP) model (ii) Multi-message shuffled (MMS) model; see Section 3.2 for the detailed description of these

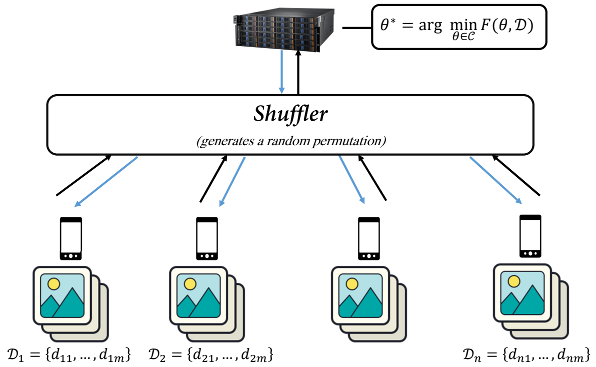

$\theta^* = \arg \min_{\theta \in \mathcal{C}} F(\theta, \mathcal{D})$
Shuffler
(generates a random permutation)
$\mathcal{D}_1 = \{d_{11}, \ldots, d_{1m}\}$
$\mathcal{D}_2 = \{d_{21}, \ldots, d_{2m}\}$
$\mathcal{D}_n = \{d_{n1}, \ldots, d_{nm}\}$

Figure 4.1: Shuffled model of differential privacy in federated learning.

two privacy models)

4.3 Shuffled Model of DP in Federated Learning

In this section, we propose CLDP-SGD, a differentially-private SGD algorithm that works with compressed updates and dynamic client population. The procedure is described in Algorithm 4.3.1. In each round of CLDP-SGD, we choose uniformly at random a set $\mathcal{U}_t$ of $k \leq n$ clients out of n clients. Each client $i \in \mathcal{U}_t$ computes the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij}\right)$ for a random subset $\mathcal{S}_{it}$ of $s \leq m$ samples. The i'th client clips the ℓ_p-norm of the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij}\right)$ for each $j \in \mathcal{S}_{it}$ and applies the LDP-compression mechanism $\mathcal{R}_p$, where $\mathcal{R}_p : \mathbb{B}_p^d \rightarrow \mathcal{Y}$ is a communication-efficient and private mechanism when inputs vector is bounded ℓ_p-norm. We can use the proposed private mechanisms in Chapter 3. After that, each client i sends the set of s private-compressed gradients $\left\{\mathcal{R}_p\left(\mathbf{g}_t\left(d_{ij}\right)\right)\right\}_{j \in \mathcal{S}_{it}}$ 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

Algorithm 4.3.1 $\mathcal{A}_{\text{cldp}}$: CLDP-SGD

```
Inputs: Datasets $\mathcal{D} = \bigcup_{i\in[n]} \mathcal{D}_i$, where $\mathcal{D}_i = \{d_{i1}, \ldots, d_{im}\}$ for $i \in [n]$, loss function $F(\theta) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} f(\theta; d_{ij})$, gradient norm bound $C$, and learning rate schedule $\{\eta_t\}$.
Initialize: $\theta_0 \in \mathcal{C}$
for $t \in [T]$ do
    Sampling of clients: A random set $\mathcal{U}_t$ of $k$ clients is chosen.
    for clients $i \in \mathcal{U}_t$ do
        Sampling of data: Client $i$ chooses uniformly at random a set $\mathcal{S}_{it}$ of $s$ samples.
        for Samples $j \in \mathcal{S}_{it}$ do
            Compute gradient: $\mathbf{g}_t(d_{ij}) \leftarrow \nabla_{\theta_t} f(\theta_t; d_{ij})$
            Clip gradient: $\tilde{\mathbf{g}}_t(d_{ij}) \leftarrow \mathbf{g}_t(d_{ij}) / \max\left\{1, \frac{\|\mathbf{g}_t(d_{ij})\|_p}{C}\right\}$[1]
            Private-compressed gradient: $\mathbf{q}_t(d_{ij}) \leftarrow \mathcal{R}_p(\tilde{\mathbf{g}}_t(d_{ij}))$
        Client $i$ sends $\{\mathbf{q}_t(d_{ij}) : j \in \mathcal{S}_{it}\}$ to the shuffler.
        Shuffling: The shuffler randomly shuffles the elements in $\{\mathbf{q}_t(d_{ij}) : i \in \mathcal{U}_t, j \in \mathcal{S}_{it}\}$ and sends them to the server.
    Aggregate: $\overline{\mathbf{g}}_t \leftarrow \frac{1}{ks}\sum_{i\in\mathcal{U}_t}\sum_{j\in\mathcal{S}_{it}} \boldsymbol{q}_t(d_{ij})$
    Gradient Descent $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}(\theta_t - \eta_t\overline{\mathbf{g}}_t)$, where $\prod_{\mathcal{C}}$ denotes the projection operator onto the set $\mathcal{C}$.
Output: The model $\theta_T$
```

vector.

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

[1] Note that gradient clipping may not preserve unbiasedness of the stochastic gradients. However, if the loss function f is L-Lipschitz (with respect to the model parameters) in the dual norm ℓ_g, where $\frac{1}{p} + \frac{1}{g} = 1, p, g \geq 1$, then the norm of the gradients (with respect to some ℓ_p-norm, for $p \geq 1$) is bounded, and hence we do not need to clip it.

the gradients at each client to prevent information leakage; and (v) shuffling. The two main technical ingredients needed for the analysis are (a) Privacy analysis of coupled sampling and shuffling (b) Communication-efficient private mean estimation.

Privacy of coupled sampling and shuffling: As explained in Section 4.1, 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 Lemma 2.1.1) 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 subsampling. 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.

Communication-efficient private mean estimation: For compressing and privatizing the gradients, we use the proposed scheme for private mean estimation in Chapter 3 to estimate the mean of a set of bounded ℓ_p-norm gradients. This privacy mechanism is composed with the sampling and shuffling to provide the overall privacy analysis. Our CLDP-SGD algorithm and the result of Theorem 4.3.1 (stated below) are given for a general local randomizer $\mathcal{R}_p$ that satisfies the following conditions: (i) The randomized mechanism $\mathcal{R}_p$ is an ε_0-LDP mechanism. (ii) The randomized mechanism $\mathcal{R}_p$ is unbiased, i.e., $\mathbb{E}\left[\mathcal{R}_p\left(\mathbf{x}\right)|\mathbf{x}\right]=\mathbf{x}$ for all $\mathbf{x}\in\mathbb{B}_p^d\left(L\right)$. (iii) The output of the randomized mechanism $\mathcal{R}_p$ can be represented using $b\in\mathbb{N}^+$ bits. (iv) The randomized $\mathcal{R}_p$ has a bounded MSE: $\sup_{\mathbf{x}\in\mathbb{B}_p^d(L)}\mathbb{E}\|\mathcal{R}_p\left(\mathbf{x}\right)-\mathbf{x}\|_2^2\leq L^2 f_p(\varepsilon_0,b)$.

We assume that the constraint set $\mathcal{C}$ is closed convex set with diameter D, where a diameter of a bounded set $\mathcal{C}\subseteq\mathbb{R}^d$ is defined as $\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{C}}\|\boldsymbol{x}-\boldsymbol{y}\|$. Furthermore, we assume that the loss function $f\left(\theta,.\right)$ is convex and L-Lipschitz continuous with respect to the ℓ_g-norm which is the dual of the ℓ_p-norm[2]. Let $r=nm$ denote the total number of data points in the

[2] For any data point $d\in\mathcal{X}$, 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 $|f(\theta_1;d)-f(\theta_2;d)|\leq L\|\theta_1-\theta_2\|_g$.

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 $q = \frac{ks}{mn}$.

Theorem 4.3.1. *Let $\theta^* = \arg\min_{\theta \in \mathcal{C}} F(\theta)$ denote the minimizer of the problem (4.1). For $s = 1$ and $q = \frac{k}{nm}$, if we run Algorithm $\mathcal{A}_{cldp}$ over T rounds, then we have*

1. ***Privacy:*** *For $\varepsilon_0 = \mathcal{O}(1)$, $\mathcal{A}_{cldp}$ is (ε, δ)-DP, where $\delta > 0$ is arbitrary, and*

$$\varepsilon = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{qT \log(2qT/\delta) \log(2/\delta)}{nm}}\right). \tag{4.2}$$

2. ***Communication:*** *Our algorithm $\mathcal{A}_{cldp}$ requires $\frac{k}{n}s \times b$ bits of communication in expectation[3] per client per iteration, where expectation is taken with respect to client sampling.*

3. ***Convergence:*** *If we run $\mathcal{A}_{cldp}$ with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = L^2\left(1 + \frac{f_p(\varepsilon_0, b)}{qmn}\right)$, then*

$$\mathbb{E}\left[F(\theta_T)\right] - F(\theta^*) \leq \mathcal{O}\left(\frac{LD \log(T) \sqrt{f_p(\varepsilon_0, b)}}{\sqrt{qmnT}}\right). \tag{4.3}$$

We prove Theorem 4.3.1 in Section 4.3.1. Observe that the privacy results in Theorem 4.3.1 is stated for $\varepsilon_0 = \mathcal{O}(1)$ using the strong composition theorem 2.1.2. In Chapter 5, we provide tighter privacy guarantees for general ε_0 by characterizing the RDP of the subsampled shuffled model.

Remark 4.3.1 (Arbitrary SGD mini-batch size s)**.** The communication and convergence results in Theorem 4.3.1 are general and hold for any $s \in [m]$; however, the privacy result is stated for $s = 1$, i.e., each client only samples a single data point in each SGD iteration. Results for any mini-batch size $s \in [m]$ are provided in Appendix C.1.

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

For bounded ℓ_2-norm, our mechanism $\mathcal{R}_{v,m,s}^{\ell_2}$ proposed in Section 3.6 with $v = \varepsilon_0$, $m = 1$, and $s = \lceil \varepsilon_0 \rceil$ satisfies ε_0-LDP with communication cost $b = \lceil \varepsilon_0 \rceil$ and $f_2(\varepsilon_0, b) = \tilde{\mathcal{O}}\left(\frac{d}{n\min\{\varepsilon_0^2, \varepsilon_0\}}\right)$, see Theorem 3.6.1. In the following, we show that our convergence results in Theorem 4.3.1 is order optimal.

Remark 4.3.2 (Recovering the Result [EFM20b, ESA])**.** In [EFM20b], each client has only one data point ($m = 1$) and all clients participate in each iteration, and gradients have bounded ℓ_2-norm. If we put $T = n/\log^2(n)$, and $q = 1$ in (4.3), we get the following privacy-accuracy trade-off, which is the same as that in [EFM20b, Theorem VI.1].

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log^2(n)\sqrt{d}}{\sqrt{n}\varepsilon_0}\right); \quad \varepsilon = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{T\log\left(T/\delta\right)\log\left(1/\delta\right)}{n}}\right)$$

We want to emphasize that the above privacy-accuracy trade-off in [EFM20b] 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 we do two types of sampling, one of clients and other of data for SGD.

Remark 4.3.3 (Optimality of CLDP-SGD for ℓ_2-norm case)**.** Suppose that our target is to achieve $\varepsilon = \mathcal{O}(1)$ and $\delta \ll 1$. Substituting $\varepsilon_0 = \varepsilon\sqrt{\frac{mn}{qT\log(2qT/\delta)\log(2/\delta)}}$, $T = mn/q$ in (4.3), we get

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) = \mathcal{O}\left(\frac{LD\log^{\frac{3}{2}}\left(\frac{mn}{\delta}\right)\sqrt{d\log\left(\frac{1}{\delta}\right)}}{mn\varepsilon}\right). \tag{4.4}$$

This matches the optimal excess risk of central differential privacy presented in [BST14]. Note that the results in [BST14] are for centralized SGD with full precision gradients, whereas, our results are for federated learning (which is a distributed setup) with compressed gradient exchange.

4.3.1 Optimization: Privacy, Communication, and Convergence Analyses

In this section, we establish the privacy, communication, and convergence guarantees of Algorithm 4.3.1 and prove Theorem 4.3.1.

4.3.1.1 Proof of Theorem 4.3.1: Privacy

Recall from Algorithm 4.3.1 that 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. This implies that the mechanism $\mathcal{A}_{cldp}$ guarantees local differential privacy ε_0 for each sample d_{ij} per epoch. 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. Recall that the input dataset at client $i \in [n]$ is denoted by $\mathcal{D}_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\} \in \mathcal{X}^m$ and $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ denotes the entire dataset. Thus, the mechanism $\mathcal{M}_t$ on any input dataset $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i \in \mathcal{X}^{nm}$ can be defined as:

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{ks} \circ \text{samp}_{n,k}\left(\mathcal{G}_1, \ldots, \mathcal{G}_n\right), \tag{4.5}$$

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

For convenience, in the rest of the proof, we suppress the auxiliary input θ_t and simply denote $\mathcal{M}_t(\theta_t; \mathcal{D})$ by $\mathcal{M}_t(\mathcal{D})$. We can do this because θ_t only affects the gradients, and the analysis in this section is for an arbitrary set of gradients.

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

Lemma 4.3.1. *Let $s = 1$ and $q = \frac{k}{nm}$. Suppose $\mathcal{R}$ is an ε_0-LDP mechanism, where $\varepsilon_0 \leq \frac{\log\left(qmn/\log\left(1/\tilde{\delta}\right)\right)}{2}$ and $\tilde{\delta} > 0$ is arbitrary. Then, for any $t \in [T]$, the mechanism $\mathcal{M}_t$ is*

$\left(\overline{\varepsilon}, \overline{\delta}\right)$*-DP, where* $\overline{\varepsilon} = \ln(1 + q(e^{\tilde{\varepsilon}} - 1)), \overline{\delta} = q\tilde{\delta}$ *with* $\tilde{\varepsilon} = \mathcal{O}\left(\min\{\varepsilon_0, 1\}e^{\varepsilon_0}\sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{qmn}}\right)$*. In particular, if* $\varepsilon_0 = \mathcal{O}\left(1\right)$*, we get* $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{mn}}\right)$*.*

We prove Lemma 4.3.1 in Appendix C.1. In the statement of Lemma 4.3.1, we are amplifying the privacy by using the subsampling as well as shuffling ideas.

Observe that the shuffler first chooses uniformly at random k clients of the available n clients. Then, each client samples her local dataset $\mathcal{D}_i$ by choosing uniformly at random $s = 1$ data points out of the available m data points. This two-steps sampling procedure is not the same as choosing uniformly at random ks data points from the entire dataset $\mathcal{D}$[4]. Therefore, we cannot directly apply the amplification by subsampling result stated in Lemma 2.1.1. Thus, we derive a new privacy proof to compute the privacy parameters of the mechanism $\mathcal{M}_t$ under non-uniform sampling. Consider two neighboring datasets $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$, $\mathcal{D}' = \mathcal{D}'_1 \bigcup \bigcup_{i=2}^{n} \mathcal{D}_i$ 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 picks the first data point or not (Please, see (C.5)). We use the bipartite graph 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.1 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 4.3.1. Now, we invoke the strong composition theorem stated in Lemma 2.1.2 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

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

[4] For example, when $s = 1$, the probability to observe two data points from the same client is zero in our sampling procedure, while observing these two data points has non-zero probability in the uniform sampling of the entire dataset $\mathcal{D}$.

where $\overline{\varepsilon}$ is from Lemma 4.3.1. We have from Lemma 2.1.2 that if $\overline{\varepsilon} = \mathcal{O}\left(\sqrt{\frac{\log(1/\delta')}{T}}\right)$, then $\varepsilon = \mathcal{O}\left(\overline{\varepsilon}\sqrt{T\log(1/\delta')}\right)$. If $\varepsilon_0 = \mathcal{O}(1)$, then we can satisfy this condition on $\overline{\varepsilon}$ by choosing $\varepsilon_0 = \mathcal{O}\left(\sqrt{\frac{n\log(1/\delta')}{qT\log(1/\tilde{\delta})}}\right)$. By substituting the bound on $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{mn}}\right)$ from Lemma 4.3.1, we have $\varepsilon = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{qT\log\left(1/\tilde{\delta}\right)\log(1/\delta')}{mn}}\right)$. By setting $\tilde{\delta} = \frac{\delta}{2qT}$ and $\delta' = \frac{\delta}{2}$, we get $\varepsilon_0 = \mathcal{O}\left(\sqrt{\frac{mn\log(2/\delta)}{qT\log(2qT/\delta)}}\right)$ and $\varepsilon = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{qT\log(2qT/\delta)\log(2/\delta)}{mn}}\right)$. This completes the proof of the privacy part of Theorem 4.3.1.

4.3.1.2 Proof of Theorem 4.3.1: Communication

The private mechanism $\mathcal{R}_p : \mathcal{X} \to \mathcal{Y}$ used in Algorithm 4.3.1 has communication cost of b bits. Let $B = 2^b$. 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 [MT20] 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 $s\left(\log(e) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits to send the s compressed gradients. Since the probability that the client is chosen at any time $t \in [T]$ is given by $\frac{k}{n}$, the expected number of bits per client in Algorithm $\mathcal{A}_{cldp}$ is given by $\frac{k}{n} \times T \times s\left(\log(e) + \log\left(\frac{s+B-1}{s}\right)\right)$ bits, where expectation is taken over the sampling of k out of n clients in all T iterations. This completes the proof of the second part of Theorem 4.3.1.

4.3.1.3 Proof of Theorem 4.3.1 : Convergence

At iteration $t \in [T]$ of Algorithm 4.3.1, the server averages the ks received compressed and privatized gradients and obtains $\overline{\mathbf{g}}_t = \frac{1}{ks}\sum_{i\in\mathcal{U}_t}\sum_{j\in\mathcal{S}_{it}} \mathbf{q}_t(d_{ij})$ 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 with respect to the random 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 4.3.2. *For any $d \in \mathcal{X}$, if the function $f\left(\theta; .\right) : \mathcal{C} \to \mathbb{R}$ is convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of ℓ_p-norm, then we have*

$$\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 \leq L^2 \left(1 + \frac{f_p(\varepsilon_0, b)}{qmn}\right). \tag{4.6}$$

where $f_p(\varepsilon_0, b)$ is the MSE of the private mechanism $\mathcal{R}_p$.

Proof. Under the conditions of the lemma, we have from [Sha12, Lemma 2.6] that $\|\nabla_\theta f\left(\theta; d\right)\| \leq L$ for all $d \in \mathcal{X}$, which implies that $\nabla_\theta F(\theta) \leq L$. Thus, we have

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 &= \|\mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 \overset{(a)}{\leq} L^2 + \mathbb{E}\|\overline{\mathbf{g}}_t - \mathbb{E}\left[\overline{\mathbf{g}}_t\right]\|_2^2 \\ &\overset{(b)}{\leq} L^2 + L^2\frac{f_p(\varepsilon_0, b)}{ks} \overset{(c)}{=} L^2 + L^2\frac{f_p(\varepsilon_0, b)}{qmn},\end{aligned}$$

where step (a) follows from the fact that $\|\nabla_{\theta_t} F\left(\theta_t\right)\| \leq L$. Step (b) follows from the fact that the private mechanism $\mathcal{R}_p$ has MSE $f_p(\varepsilon_0, b)$. Step (c) uses $q = \frac{ks}{mn}$. ■

Now, we can use standard SGD convergence results for convex functions. In particular, we use the following result from [SZ13].

Lemma 4.3.3 (SGD Convergence [SZ13])**.** *Let $F\left(\theta\right)$ be a convex function, and the set $\mathcal{C}$ has diameter D. Consider a stochastic gradient descent algorithm $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}\left(\theta_t - \eta_t \mathbf{g}_t\right)$, where $\mathbf{g}_t$ satisfies $\mathbb{E}\left[\mathbf{g}_t\right] = \nabla_{\theta_t} F\left(\theta_t\right)$ and $\mathbb{E}\|\mathbf{g}_t\|_2^2 \leq G^2$. By setting $\eta_t = \frac{D}{G\sqrt{t}}$, we get*

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq 2DG\frac{2 + \log\left(T\right)}{\sqrt{T}} = \mathcal{O}\left(DG\frac{\log\left(T\right)}{\sqrt{T}}\right). \tag{4.7}$$

As shown in Lemma 4.3.2 and above that Algorithm 4.3.1 satisfies the premise of Lemma 4.3.3. Now, using the bound on G^2 from Lemma 4.3.2, we have that the output θ_T of Algorithm 4.3.1 satisfies

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\sqrt{\left(1+\frac{f_p(\varepsilon_p,b)}{qmn}\right)}}{\sqrt{T}}\right). \tag{4.8}$$

Note that if $\sqrt{\frac{f_p(\varepsilon_0,b)}{qmn}} \leq \mathcal{O}(1)$, then we recover the convergence rate of vanilla SGD without privacy. So, the interesting case is when $\sqrt{\frac{f_p(\varepsilon_0,b)}{qmn}} \geq \Omega(1)$, which gives

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\sqrt{f_p(\varepsilon_p,b)}}{\sqrt{qmnT}}\right).$$

This completes the proof of Theorem 4.3.1.

4.4 DP Federated Learning with Client-Self Sampling

In our CLDP-SGD algorithm, we apply uniform client sampling by choosing uniformly at random a *fixed* number of clients at each iteration. Choosing a fixed number of clients at each iteration requires a selection by the shuffler. In this section, we extend our work to explore a distributed self-sampling approach initiated by the clients that does not need a selection by the shuffler. In order to obtain the new algorithm $\mathcal{A}_{dss}$ distributed self-sampling (DSS-SGD), we replace the client-sampling (Line 4 in Algorithm 4.3.1) with client-self sampling approach.

At each round $t \in [T]$ of DSS-SGD, each client independently and identically tosses a biased coin with probability q. If the biased coin of the ith client returns a head (one), then the ith client participates in the current round and shares its model privately with the untrusted server with the help of the trusted shuffler. Otherwise, the ith client does not participate in the current round. Let $\mathcal{U}_t$ denote the set of participating clients at round $t \in [T]$. We follow the same steps as in Algorithm 4.3.1, where each client $i \in \mathcal{U}_t$ computes the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij_i}\right)$ for a randomly chosen sample d_{ij_i} from its local dataset $\mathcal{D}_i$. The

i'th client clips the ℓ_p-norm of the gradient $\nabla_{\theta_t} f\left(\theta_t; d_{ij_i}\right)$ and applies the LDP-compression mechanism $\mathcal{R}_p$. After that, each client i sends the private gradient $\mathcal{R}_p\left(\mathbf{g}_t\left(d_{ij_i}\right)\right)$ to the secure shuffler that sends a random permutation of the received gradients to the server. Finally, the server takes the average of the received gradients and updates the global model.

Our DSS-SGD is different from the CLDP-SGD algorithm in the client sampling scheme. In CLDP-SGD, a fixed number of clients are chosen uniformly at random in each iteration that requires a selection by the shuffler. While, in DSS-SGD, each individual client decides to participate in each iteration depending on independent randomness generated at the client-side. Hence, the proposed self-sampling does not need the coordination with the shuffler that reflects the random availability of the clients in practical FL. This modification in the client sampling raises challenges in analyzing the central privacy of the algorithm as well as analyzing the convergence of the SGD, since the number of clients participating at each iteration is random.

Our CLDP-SGD algorithm and the result of Theorem 4.4.1 (stated below) are given for a general local randomizer $\mathcal{R}_p$ that satisfies the following conditions: (i) The randomized mechanism $\mathcal{R}_p$ is an ε_0-LDP mechanism. (ii) The randomized mechanism $\mathcal{R}_p$ is unbiased, i.e., $\mathbb{E}\left[\mathcal{R}_p\left(\mathbf{x}\right)|\mathbf{x}\right] = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{B}_p^d\left(L\right)$. (iii) The output of the randomized mechanism $\mathcal{R}_p$ can be represented using $b \in \mathbb{N}^+$ bits. (iv) The randomized $\mathcal{R}_p$ has a bounded MSE: $\sup_{\mathbf{x}\in\mathbb{B}_p^d(L)} \mathbb{E}\|\mathcal{R}_p\left(\mathbf{x}\right) - \mathbf{x}\|_2^2 \leq L^2 f_p(\varepsilon_0, b)$. For example, we can apply our proposed communication-efficient and private schemes proposed in Chapter 3.

Theorem 4.4.1. *Let the set $\mathcal{C}$ be convex with diameter D[5] and the function $f\left(\theta; .\right) : \mathcal{C} \rightarrow \mathbb{R}$ be convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of the ℓ_p-norm. Let $\theta^* = \arg\min_{\theta\in\mathcal{C}} F\left(\theta\right)$ denote the minimizer of the problem* (4.1)*. For participation probability $0 < q \leq 1$, let $\bar{q} = \frac{q}{m}$. If we run Algorithm $\mathcal{A}_{dss}$ over T iterations, then we have*

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

1. ***Privacy:*** *For $\varepsilon_0 = \mathcal{O}(1)$, $\mathcal{A}_{dss}$ is (ε, δ)-DP, where*

$$\begin{aligned} \varepsilon &= \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\bar{q}T \log(\bar{q}T/\delta') \log(1/\delta')}{mn}}\right), \\ \delta &= 2\delta' + Te^{-c'qn}, \end{aligned} \tag{4.9}$$

where $\delta' > 0$ is arbitrary, and $c' \in (0,1)$ is a constant.

2. ***Communication:*** *Our algorithm $\mathcal{A}_{dss}$ requires $q \times b$ bits of communication in expectation per client per iteration, where expectation is taken with respect to client sampling.*

3. ***Convergence:*** *If we run $\mathcal{A}_{dss}$ with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = L^2\left(1 + \frac{f_p(\varepsilon_0,b)}{\bar{q}mn}\right)$, then*

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\sqrt{f_p(\varepsilon_0,b)}}{\sqrt{T\bar{q}mn(1-e^{-qn})}} + e^{-c'qn}\right). \tag{4.10}$$

We prove Theorem 4.4.1 in Section 4.4.1

Remark 4.4.1 (Impact of self-sampling of clients)**.** In our algorithm DSS-SGD, the number of clients participating in any time slot $t \in [T]$ is a binomial random variable K_t. Note that $K_t = |\mathcal{U}_t|$. Thus, the expected number of effective iterations in which $K_t > 0$ is bounded below by $T(1 - e^{-qn})$. Thus, the output of our algorithm converges with rate $\mathcal{O}\left(1/\sqrt{T(1-e^{-qn})}\right)$ instead of $\mathcal{O}\left(1/\sqrt{T}\right)$ as in the standard SGD. Furthermore, the impact of such client sampling appears in the privacy parameter δ that has an additive term $Te^{-c'qn}$. This term does not appear if we choose uniformly at random a fixed number of clients at each time slot (see Theorem 4.3.1). However, in cross-device federated learning [Kai19], the number of participated clients at each time slot is typically in thousands, i.e., qn is equal to a few thousands. Thus, the term $Te^{-qn} \ll 1/mn$ and e^{-qn} are negligible.

Remark 4.4.2 (Optimality of DSS-SGD for ℓ_2-norm case)**.** Suppose that our target is to achieve $\varepsilon = \mathcal{O}(1)$ and $\delta \ll 1/mn$. We can apply our private mechanism $\mathcal{R}_{v,m,s}^{\ell_2}$ proposed in Section 3.6 with $v = \varepsilon_0$, $m = 1$, and $s = \lceil \varepsilon_0 \rceil$. Substituting $\varepsilon_0 = \varepsilon\sqrt{\frac{nm}{qT\log(2qT/\delta')\log(2/\delta')}}$, $T = nm/\bar{q}(1 - e^{-qn})$ in (4.10), we recover the optimal excess risk of central differential privacy presented in [BST14], except an additive term Te^{-qn} in δ of the privacy parameter.

4.4.1 Proof of Theorem 4.4.1

4.4.1.1 Privacy

Hereafter, we denote $\mathcal{R}_p$ by $\mathcal{R}$, for simplicity which is an ε_0-LDP mechanism. This implies that the mechanism $\mathcal{A}_{dss}$ guarantees local differential privacy ε_0 for each sample d_{ij} per iteration. Thus, it remains to analyze the central DP guarantee of the mechanism $\mathcal{A}_{dss}$ in each iteration and also for the entire execution. The proof follows similar steps as the proof of Theorem 4.3.1 in Section 4.3.1.1.

Fix a time slot $t \in [T]$. 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 and generates the parameter θ_{t+1} as an output. Let $K_t = |\mathcal{U}_t|$ denote the random variable corresponding to the number of participating clients in the t'th time slot. Thus, the mechanism $\mathcal{M}_t$ on input dataset $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i \in \mathcal{X}^n$ when $K_t > 0$ can be defined as:

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{K_t} \circ \text{samp}_{n,q}^{\text{iid}}\left(\mathcal{G}_1, \ldots, \mathcal{G}_n\right), \tag{4.11}$$

where $\mathcal{G}_i = \text{samp}_{m,1}^{\text{fix}}\left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{im}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [n], j \in [m]$. Here, $\text{samp}_{n,q}^{\text{iid}}$ denotes the sampling operation for choosing each of the n elements independently with probability q, $\text{samp}_{m,1}^{\text{fix}}$ denotes the sampling operation for choosing uniformly at random a single element from a set of m elements, and $\mathcal{H}_{K_t}$ denotes the shuffling operation on K_t elements, which outputs a random permutation of the K_t input elements. For convenience, in the rest of the proof, we suppress the auxiliary input θ_t and simply denote $\mathcal{M}_t(\theta_t; \mathcal{D})$ by $\mathcal{M}_t(\mathcal{D})$. We can do this because θ_t only affects the gradients, and the analysis in this part is for an arbitrary set of gradients.

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

Lemma 4.4.1. *Fix an arbitrary iteration $t \in [T]$. Let $\overline{q} = \frac{q}{m}$. Suppose $\mathcal{R}$ is an ε_0-LDP mechanism with $\varepsilon_0 = \mathcal{O}(1)$. Then, for any $\tilde{\delta} > 0$, the mechanism $\mathcal{M}_t$ is $\left(\overline{\varepsilon}, \overline{\delta}\right)$-DP, where $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{\overline{q}\log(1/\tilde{\delta})}{mn}}\right)$ and $\overline{\delta} = \overline{q}\tilde{\delta} + e^{-c'qn}$ for some constant $c' \in (0,1)$.*

We provide a proof of Lemma 4.4.1 in Appendix C.2. Note that the Algorithm $\mathcal{A}_{dss}$ 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 4.4.1. Now, we invoke the strong composition [DR14, Theorem 3.20] to obtain the privacy guarantee of the algorithm $\mathcal{A}_{dss}$. We can conclude that for any $\delta', \tilde{\delta} > 0$, $\mathcal{A}_{dss}$ is (ε, δ)-DP for

$$\varepsilon = \sqrt{2T\log\left(1/\delta'\right)}\overline{\varepsilon} + T\overline{\varepsilon}\left(e^{\overline{\varepsilon}} - 1\right)$$
$$\delta = \bar{q}T\tilde{\delta} + \delta' + Te^{-c'qn},$$

where $\overline{\varepsilon}$ is from Lemma 4.4.1. When $\varepsilon_0 = \mathcal{O}\left(1\right)$, then we get that $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{\bar{q}\log\left(1/\tilde{\delta}\right)}{mn}}\right)$ from Lemma 4.4.1. Thus, from Lemma [DR14, Theorem 3.20], we get that

$$\varepsilon = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{\bar{q}T\log\left(1/\tilde{\delta}\right)\log\left(1/\delta'\right)}{mn}}\right).$$

By setting $\tilde{\delta} = \frac{\delta'}{\bar{q}T}$, we get $\varepsilon = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{\bar{q}T\log(\bar{q}T/\delta')\log(1/\delta')}{mn}}\right)$, and $\delta = 2\delta' + Te^{-c'qn}$.

4.4.1.2 Communication

Suppose that the randomized mechanism $\mathcal{R}$ is ε_0-LDP having with communication cost b-bits. Therefore, the expected number of bits per client in Algorithm $\mathcal{A}_{dss}$ is given by $q \times b$ bits per iteration, where expectation is taken over the client sampling.

4.4.1.3 Convergence

Note that the number of clients participating $|\mathcal{U}_t|$ at any time slot $t \in [T]$ is a binomial random variable K_t. Hence, the probability that $\mathcal{U}_t$ is empty is given by $\Pr\left[K_t = 0\right] = \left(1-q\right)^n \leq e^{-qn}$. Thus, the expected number of effective iterations (when $K_t > 0$) is given by $\overline{T} = \left(1 - \left(1-q\right)^n\right) T \geq (1-e^{-qn})T$.

At iteration $t \in [T]$ of Algorithm DSS-SGD when $K_t > 0$, the server averages the K_t received compressed and privatized gradients and obtains $\overline{\mathbf{g}}_t = \frac{1}{K_t}\sum_{i\in\mathcal{U}_t}\mathbf{q}_t(d_{ij_i})$. We show that the average gradient $\overline{\mathbf{g}}_t$ is unbiased:

Claim 4.4.1. We have $\mathbb{E}[\overline{\mathbf{g}}_t] = \nabla F(\theta_t)$, where expectation is taken with respect to the random participation of clients, the sampling of data points, and the randomness of the mechanism $\mathcal{R}_p$.

We prove Claim 4.4.1 in Appendix C.3. Now we show that $\overline{\mathbf{g}}_t$ has a bounded second moment.

Lemma 4.4.2. *For any $d \in \mathcal{X}$, if the function $f\left(\theta;.\right): \mathcal{C} \to \mathbb{R}$ is convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of ℓ_p-norm, then we have*

$$\mathbb{E}_{\substack{\mathcal{U}_t\sim \text{samp}^{iid}_{n,q},\mathcal{R}_p,\\ j_i\in[m],i\in\mathcal{U}_t}}\|\overline{\mathbf{g}}_t\|_2^2 \leq L^2\left(1+\frac{f_p(\varepsilon_0,b)}{\bar{q}mn}\right) + e^{-c'qn}. \tag{4.12}$$

The proof of Lemma 4.4.2 is presented in Appendix C.4. Although the number of participating clients at each iteration (K_t) is varying from iteration to iteration, Lemma 4.4.2 shows that the second moment of the descent direction $\overline{\mathbf{g}}_t$ decreases with order $\mathcal{O}\left(1/qn\right)$, where $qn = \mathbb{E}\left[K_t\right]$. Now, we can use standard SGD convergence results for convex functions. In particular, we use the result from [SZ13], which is stated in Lemma 4.3.3. Algorithm DSS-SGD satisfies the premise of Lemma 4.3.3, where the expected number of effective iteration is given by $\overline{T} \geq (1-e^{-qn})T$. Now, using the bound on G^2 from (4.12) (and ignoring the exponentially small term $e^{-c'\bar{q}n}$), we have that the output θ_T of Algorithm DSS-SGD satisfies

$$\mathbb{E}\left[F\left(\theta_T\right)\right]-F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\sqrt{1+\frac{f_b(\varepsilon_0,b)}{\bar{q}mn}}}{\sqrt{T(1-e^{-qn})}}\right). \tag{4.13}$$

Note that if $\sqrt{\frac{f_p(\varepsilon_0,b)}{\bar{q}mn}} \leq \mathcal{O}(1)$, then we recover the convergence rate of vanilla SGD without privacy. Therefore, the interesting case is when $\sqrt{\frac{f_p(\varepsilon_0,b)}{\bar{q}mn}} \geq \Omega(1)$, which gives $\mathbb{E}\left[F\left(\theta_T\right)\right]-F\left(\theta^*\right) \leq \mathcal{O}\left(\frac{LD\log(T)\sqrt{f_b(\varepsilon_0,b)}}{\sqrt{T\bar{q}mn(1-e^{-qn})}}\right)$. This completes the proof of Theorem 4.4.1.

4.5 DP Personalized Federated Learning

In the previous sections, we consider federated learning framework, where the server learns a global model from the clients' dataset. However, due to the statistical heterogeneity of the clients' data, learning a single global model may perform poorly for individual clients. This motivates the need for personalized federated learning by learning individual models for each client through collaboration. Consider a set of n clients, where each client has a local dataset $\mathcal{D}_i=\{d_{i1},\ldots,d_{im}\}$ comprising m data points drawn from a universe $\mathcal{X}$. The goal is to learn n local models $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n$ that fits the local datasets $\{\mathcal{D}_i : i\in[n]\}$, respectively. Observe that the local dataset $\mathcal{D}_i$ is not large enough to train a local model $\boldsymbol{\theta}_i$ with a reasonable performance. Thus, we propose a Bayesian approach for personalized federated learning that motivates the collaboration between clients.

Let each data point d_{ij} consists of a pair $d_{ij}=(X_{ij},Y_{ij})$, where X_{ij} denotes the feature vector and Y_{ij} denotes the target for $i\in[n]$ and $j\in[m]$. Let $\mathbb{P}(\Gamma)$ be an unknown global population distribution[6] over $\mathbb{R}^d$ from which the local parameters $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\in\mathbb{R}^d$ are sampled i.i.d. For given features $\{X_{ij}\}$ and local model $\boldsymbol{\theta}_i$, let the targets Y_{ij}'s are generated from $(X_{ij},\boldsymbol{\theta}_i)$ using some distribution $p_{\boldsymbol{\theta}_i}\left(Y_{ij}|X_{ij}\right)$. Let $Y_i:=(Y_{i1},\ldots,Y_{im})$ and

[6] For simplicity, we will consider this unknown population distribution $\mathbb{P}$ to be parameterized by unknown (arbitrary) parameters Γ.

$X_i := (X_{i1}, \ldots, X_{im})$ for $i \in [n]$. The underlying statistical model for our setting is given by

$$p_{\{\boldsymbol{\theta}_i, Y_i\}|\{X_i\}}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n, Y_1, \ldots, Y_n | X_1, \ldots, X_n) = \prod_{i=1}^{n} p(\boldsymbol{\theta}_i) \prod_{i=1}^{n} \prod_{j=1}^{m} p_{\boldsymbol{\theta}_i}(Y_{ij}|X_{ij}). \tag{4.14}$$

Note that if we minimize the negative log likelihood of (4.14), we would get the optimal parameters:

$$\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_m := \underset{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m} -\log(p_{\boldsymbol{\theta}_i}(Y_{ij}|X_{ij})) + \sum_{i=1}^{n} -\log(p(\boldsymbol{\theta}_i)). \tag{4.15}$$

Here, $f_i(\boldsymbol{\theta}_i) := \sum_{j=1}^{m} -\log(p_{\boldsymbol{\theta}_i}(Y_{ij}|X_{ij}))$ denotes the loss function at the i-th client, which only depends on the local data $\mathcal{D}_i$, and $R(\{\boldsymbol{\theta}_i\}) := \sum_{i=1}^{n} -\log(p(\boldsymbol{\theta}_i))$ is the regularizer that depends on the (unknown) global population distribution $\mathbb{P}$ parameterized by unknown Γ. Note that when clients have little data and we have large number of clients, i.e., $m \ll n$ – the setting of federated learning, clients may not be able to learn good personalized models from their local data alone (if they do, it would lead to large loss). In order to learn better personalized models, clients may utilize other clients' data through collaboration, and the above regularizer (and estimates of the unknown prior distribution $\mathbb{P}$, through estimating its parameters Γ) dictates how the collaboration might be utilized. The above-described statistical framework (4.15) can model many different scenarios, as detailed below:

- When $\mathbb{P}(\Gamma) \equiv \mathcal{N}(\boldsymbol{\mu}, \sigma_\theta^2 \mathbb{I}_d)$ is a Gaussian for $\Gamma = \left\{\boldsymbol{\mu}, \sigma_\theta : \sigma_\theta \geq 0, \boldsymbol{\mu} \in \mathbb{R}^d\right\}$, then $R(\{\boldsymbol{\theta}_i\}) = \frac{nd}{2}\log(2\pi\sigma_\theta^2) + \sum_{i=1}^{n} \frac{\|\boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2}{2\sigma_\theta^2}$. Here, unknown $\boldsymbol{\mu}$ can be connected to the global model and $\boldsymbol{\theta}_i$'s as local models, and the alternating iterative optimization optimizes over both. This justifies the use of ℓ_2 regularizer in earlier personalized learning works [DTN20, HR20, LHB21].

- When $\mathbb{P}(\Gamma) \equiv \mathsf{Laplace}(\boldsymbol{\mu}, b)$, for $\Gamma = \{\boldsymbol{\mu}, b > 0\}$, then $R(\{\boldsymbol{\theta}_i\}) = n\log(2b) + \sum_{i=1}^{n} \frac{\|\boldsymbol{\theta}_i - \boldsymbol{\mu}\|_1}{b}$.

In Section 4.5.1, we propose a personalized federated learning algorithm under user-level DP constraints that exploits our proposed Bayesian approach with knowledge distillation (KD) regularizer. See Definition 3.7.1 in Section 3.7 and the remarks for more details about the user-level DP.

4.5.1 DP-AdaPeD: Adaptive Personalization via Distillation

It has been empirically observed that the knowledge distillation (KD) regularizer (between local and global models) results in better performance than the ℓ_2 regularizer [OSD21]. In fact, using our framework, we can define, for the first time, a certain prior distribution that gives the KD regularizer. We use the following loss function at the i-th client:

$$f_i(\boldsymbol{\theta}_i) + \frac{1}{2}\log(2\psi) + \frac{f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i, \boldsymbol{\mu})}{2\psi}, \tag{4.16}$$

where $\boldsymbol{\mu}$ denotes the global model, $\boldsymbol{\theta}_i$ denotes the personalized model at client i, and ψ can be viewed as controlling heterogeneity. The goal for each client is to minimize its local loss function, so individual components cannot be too large. For the second term, this implies that ψ cannot be unbounded. For the third term, if $f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i, \boldsymbol{\mu})$ is large, then ψ will also increase (implying that the local parameters are too deviated from the global parameter), hence, it is better to emphasize local training loss to make the first term small. If $f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i, \boldsymbol{\mu})$ is small, then ψ will also decrease (implying that the local parameters are close to the global parameter), so it is better to collaborate and learn better personalized models. Such adaptive weighting quantifies the uncertainty in population distribution during training, balances the learning accordingly, and improves the empirical performance over non-adaptive methods.

To optimize (4.16) under user-level DP, we propose an alternating minimization approach, which we call DP-AdaPeD; see Algorithm 0. Besides the personalized model $\boldsymbol{\theta}_i^t$, each client i keeps local copies of the global model $\boldsymbol{\mu}_i^t$ and of the dissimilarity term ψ_i^t. Note that client i communicates $\boldsymbol{\mu}_i^t, \psi_i^t$ (which are updated by accessing the dataset for computing the gradients $\boldsymbol{h}_i^t, k_i^t$) to the server. Therefore, to privatize $\boldsymbol{\mu}_i^t, \psi_i^t$, client i adds appropriate noise to $\boldsymbol{h}_i^k, k_i^t$, where $\sigma_{q_1}, \sigma_{q_2} > 0, C_1, C_2$ in Lines 13 and 15 depend on the desired privacy level. At synchronization times, the server aggregates them to obtain global versions of these $\boldsymbol{\mu}^t, \psi^t$. In this way, the local training of $\boldsymbol{\theta}_i^t$ also incorporates knowledge from other clients' data through $\boldsymbol{\mu}_i^t$. In the end, clients have learned their personalized models $\{\boldsymbol{\theta}_i^T\}_{i=1}^m$. The theorem below (proved in Appendix C.5) states the Rényi Differential Privacy (RDP) guarantees of

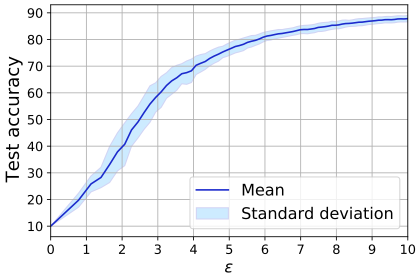


Figure 4.2: Privacy-Utility trade-offs on the MNIST dataset with ℓ_∞-norm clipping.

our DP-AdaPeDAlgorithm.

Theorem 4.5.1. *After T iterations,* DP-AdaPeD *satisfies $(\alpha, \varepsilon(\alpha))$-RDP for $\alpha > 1$, where $\varepsilon(\alpha) = \left(\frac{K}{n}\right)^2 6\frac{T}{\tau}\alpha\left(\frac{C_1^2}{K\sigma_{q_1}^2} + \frac{C_2^2}{K\sigma_{q_2}^2}\right)$, where $\frac{K}{n}$ denotes the sampling ratio of clients at each global iteration.*

We bound the RDP, as it gives better privacy composition than using the strong composition. We can also convert our results to user-level (ε, δ)-DP by using the standard conversion from RDP to approximate DP in Lemma 2.1.3. In Section 4.6, we numerically evaluate the performance of our DP-AdaPeDalgorithm.

4.6 Numerical Results

In this section, we numerically evaluate the performance of our proposed differentially private federated learning algorithms.

Shuffled model of differential privacy in federated learning: 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 [EFM20b, PTS20] and described in Table 4.1. This

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 4.1: Model Architecture for MNIST

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., $n = 60,000$ and $m = 1$. We present our results for ℓ_∞-norm clipping using our DME algorithm $\mathcal{R}_{v,m,s}^{\ell_\infty}$ proposed in Section 3.5 with parameters $v = \varepsilon_0$, $m = 1$, and $s = 1$. 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 $\mathcal{R}_{v,m,s}^{\ell_\infty}$ presented in Algorithm 3.5.1 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 $v = \varepsilon_0 = 2$ and $\delta = 10^{-5}$, while the centralized privacy parameter ε is computed numerically from Theorem 4.3.1.

Figure 4.2 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 $\varepsilon = 5$ and an accuracy 87.9% (± 1) for total privacy $\varepsilon = 10$. Furthermore, observe that our proposed CLDP-SGD algorithm preserves a local privacy of $\varepsilon_0 = 2$ per sample per epoch. In addition, the private mechanism 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× in comparison with the full precision gradient.

Table 4.2: Test Accuracy (in %) vs. ε on MNIST without client sampling for `DP-AdaPeD`.

Method	$\epsilon = 3.35$	$\epsilon = 13.16$	$\epsilon = 27.30$
DP-FedAvg	11.73 ± 0.85	29.91 ± 1.28	55.79 ± 0.29
`DP-AdaPeD` (Ours)	93.32 ± 1.18	98.51 ± 0.90	99.01 ± 0.65

In [PTS20], the authors achieve a test accuracy of 98% on MNIST with central privacy parameters $\varepsilon = 3$ and $\delta = 10^{-5}$ using a DP centralized algorithm by adding Gaussian noise to the aggregated gradients in each iteration. However, [PTS20] do not offer any local differential privacy guarantees, which can be thought of as $\varepsilon_0 = \infty$. Although, Theorem 4.3.1 and Remark 4.3.3 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 [PTS20]. The privacy parameters of our algorithm can be improved by analyzing the Rényi differential privacy of the shuffled model (see Chapter 5).

DP Personalized Federated Learning: We consider image classification on MNIST, FEMNIST [CDW18]; and train a CNN, similar to the one considered in [MMR17], that has 2 convolutional and 3 fully connected layers. We set $n = 66$ for FEMNIST and $n = 50$ for MNIST. For FEMNIST, we use a subset of 198 writers so that each client has access to data from 3 authors, which results in a natural type of data heterogeneity due to writing styles of authors. On MNIST, we introduce pathological heterogeneity by letting each client sample data from 3 randomly selected classes only. We set $\tau = 10$ and vary the batch size so that each epoch consists of 60 iterations.

In Figure 4.3 and Table 4.2, we observe performance of `DP-AdaPeD` under different ε values. `DP-AdaPeD` outperforms DP-FedAvg because personalized models do not need to be privatized by DP mechanism, whereas the global model needs to be in DP-FedAvg. Our

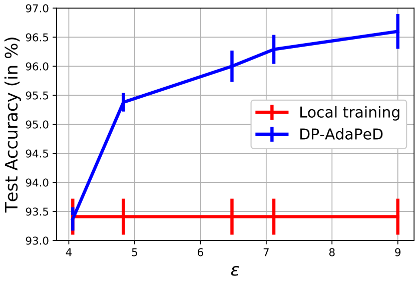


Figure 4.3: Test Accuracy (in %) vs. ε on FEMNIST with client sampling ratio of 0.33 for `DP-AdaPeD`.

experiments provide user-level privacy that is appropriate in FL.

4.7 Related Work

The federated learning (FL) paradigm has had huge recent success both in industry and academia [MMR17, Kai19], as it enables to leverage data available in dispersed devices for learning while maintaining data privacy. Below we give a brief description of related work

DP Federated Learning There has been a lot of work on privacy in the context of FL (see [Kai19] and references therein). In [CMS11], Chaudhuri et al. 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. In [BST14], Bassily et al. 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 [ACG16], the authors have generalized the private SGD algorithm proposed in [BST14] for non-convex optimization framework. In addition, the authors have proposed a new analysis technique (moment accounting) to improve on the strong composition theorems to compute the central differential privacy guarantee for iterative algorithms. However, the mentioned works [CMS11, BST14, ACG16] assume that there exists a trusted server that collects the clients' data. This motivates other works to design a private 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 [War65, DWJ13, BDF18]). However, it is well understood that LDP does not give good performance guarantees as it requires significant local randomization to give privacy guarantees [DWJ13, KLN11, KBR16]. The two most related papers to our work are [EFM20b, ASY18] which we describe below.

In [EFM20b], Erlingsson et al. 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 [BBG19d]. 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 [EFM20b], 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 each 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. 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.

In [ASY18], Agarwal et al. proposed a communication-efficient algorithm for learning models with differential privacy. They proposed cp-SGD, a communication efficient algorithm, where clients need to send $O\left(d\log(n)\right)$ bits of communication per client 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. 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.

Client sampling procedures: In [BKM20], the authors have proposed a novel sampling scheme called *random check-in*, in which each client independently chooses which time slot to participate in the training process. However, their sampling scheme is different from client-self sampling proposed in Section 4.4 in the following sense: (i) We consider multiple data samples at each client, whereas, in their work they assume that each client has a single sample. This provides an additional layer of sampling the local datasets at clients that amplifies the central privacy of the SGD. Furthermore, this creates non-uniform sampling of data points, because clients either do not participate or they participate with a mini-batch gradient of a certain size. (ii) Our self-sampling scheme allows flexibility to the clients to participate in more than one iteration. In contrast, in [BKM20] each client participates only in one time slot of the training process. These differences also lead to distinct technical approaches to proving privacy and the trade-offs.

Private personalized learning: There has been a lot of work in privacy for FL when the goal is to learn a *single* global model; though there are fewer papers that address user-level privacy [MAE18, LSY20, WSZ19, LSA21, GKM21a]. There has been more recent work on applying these ideas to learn personalized models [JRS21, HGL20, LKC20]. These are for specific algorithms/models, *e.g.,* [JRS21] focuses on the common representation model for linear regression described earlier or on item-level privacy [HGL20, LKC20]. We believe that `DP-AdaPeD` proposed in this chapter is among the first user-level private personalized learning algorithms with user-level DP privacy guarantees, applicable to general deep learning.

Algorithm 4.4.1 DP Adaptive Personalization via Distillation (`DP-AdaPeD`)

```
Parameters: local variances $\{\psi_i^0\}$, personalized models $\{\boldsymbol{\theta}_i^0\}$, local copies of the global model $\{\boldsymbol{\mu}_i^0\}$, learning rates $\eta_1, \eta_2, \eta_3$, synchronization gap $\tau$, and privacy variances $\sigma_{q_1}$, $\sigma_{q_2}$.
for $t = 0$ to $T - 1$ do
    if $\tau$ divides $t$ then
        On Server do:
        Choose a subset $\mathcal{K}^t \subseteq [n]$ of $K$ clients
        Broadcast $\boldsymbol{\mu}^t$ and $\psi^t$
        On Clients $i \in \mathcal{K}^t$ (in parallel) do:
        Receive $\boldsymbol{\mu}^t, \psi^t$; set $\boldsymbol{\mu}_i^t = \boldsymbol{\mu}^t$, $\psi_i^t = \psi^t$
    On Clients $i \in \mathcal{K}^t$ (in parallel) do:
    Compute $\boldsymbol{g}_i^t := \nabla_{\boldsymbol{\theta}_i^t} f_i(\boldsymbol{\theta}_i^t) + \frac{\nabla_{\boldsymbol{\theta}_i^t} f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i^t,\boldsymbol{\mu}_i^t)}{2\psi_i^t}$
    Update: $\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_i^t - \eta_1 \boldsymbol{g}_i^t$
    Compute $\boldsymbol{h}_i^t := \nabla_{\boldsymbol{\mu}_i^t} f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i^{t+1},\boldsymbol{\mu}_i^t)/2\psi_i^t$.
    Update: $\boldsymbol{\mu}_i^{t+1} = \boldsymbol{\mu}_i^t - \eta_2 \left(\frac{\boldsymbol{h}_i^t}{\max\{\|\boldsymbol{h}_i^t\|/C_1,1\}} + \boldsymbol{\nu}_1\right)$, where $\boldsymbol{\nu}_1 \sim \mathcal{N}(0, \sigma_{q_1}^2 \mathbb{I}_d)$.
    Compute $k_i^t := \frac{1}{2\psi_i^t} - f_i^{\mathsf{KD}}(\boldsymbol{\theta}_i^{t+1},\boldsymbol{\mu}_i^{t+1})/2(\psi_i^t)^2$.
    Update: $\psi_i^{t+1} = \psi_i^t - \eta_3 \left(\frac{k_i^t}{\max\{|k_i^t|/C_2,1\}} + \nu_2\right)$, where $\nu_2 \sim \mathcal{N}(0, \sigma_{q_2}^2)$.
    if $\tau$ divides $t + 1$ then
        Clients send $\boldsymbol{\mu}_i^t$ and $\psi_i^t$ to Server
        Server receives $\{\boldsymbol{\mu}_i^t\}_{i \in \mathcal{K}^t}$ and $\{\psi_i^t\}_{i \in \mathcal{K}^t}$

        Server computes $\boldsymbol{\mu}^{t+1} = \frac{1}{K}\sum_{i \in \mathcal{K}^t} \boldsymbol{\mu}_i^t$ and $\psi^{t+1} = \frac{1}{K}\sum_{i \in \mathcal{K}^t} \psi_i^t$
Output: Personalized models $(\boldsymbol{\theta}_i^T)_{i=1}^m$
```

CHAPTER 5

Rényi Differential Privacy of the Shuffled Model

In this chapter, we characterize the Rényi differential privacy (RDP) of the shuffled model by proposing upper and lower bounds for general LDP mechanisms. RDP is a useful privacy notion that enabled a much tighter composition for interactive mechanisms. Furthermore, we characterize the RDP of the subsampled shuffled model that combines the privacy amplification via shuffling and amplification by subsampling. To achieve these results, we propose a novel analysis technique by reducing any general neighboring datasets to a special case datasets that can be analyzed in a closed form solution.

5.1 Introduction

Shuffled model is a privacy framework using anonymization [BEM17,EFM19,CSU19,BBG19d], where each client sends her (randomized) report to a secure shuffler that randomly permutes all the received reports before forwarding them to the server. This model enables significantly better privacy-utility performance by amplifying LDP through this mechanism.

In federated learning, there are repeated interactions (*e.g.,* through distributed gradient descent), and hence, one needs privacy composition [BST14] to compute the overall privacy budget. Clearly, from an optimization viewpoint, we might need to run these interactions longer for better models, but these also result in privacy leakage. Though the privacy leakage can be quantified using advanced composition theorems for DP (*e.g.,* [DRV10,KOV15]), these might be loose. To address this, Abadi *et al.* [ACG16] developed a “moments accountant” framework, which enabled a much tighter composition. This is enabled by providing the composition privacy guarantee in terms of rényi differential privacy (RDP) [Mir17], and then mapping it back to the DP guarantee [MTZ19]. It is known that RDP provides a significant saving in the total privacy budget in comparison with using the strong composition theorems [DRV10, KOV15]. Therefore, analyzing the RDP of the shuffle model could have several applications such as private statistics using interactive schemes for heavy hitters, mean estimation, federated learning, and distributed differentially private stochastic gradient descent (DP-SGD). Thus, the central question studied in this chapter is RDP guarantees for general discrete local randomizers in the *shuffled* privacy model. Our results could be adapted to enhance the privacy guarantees of the federated learning algorithms in the shuffled model presented in Chapter 4.

The principal result in this chapter is the *first* direct RDP guarantee for general discrete local randomization mechanisms in the shuffle privacy model. In particular, given an *arbitrary discrete* local mechanism with ε_0-LDP guarantee, we provide an RDP guarantee for the shuffle model, as a function of ε_0 and the number of users n. This can be seen as a privacy amplification result for amplifying pure LDP guarantee to RDP guarantee via shuffling. In contrast, the existing amplification by shuffling results [EFM19, BBG19d, FMT22] amplify pure LDP guarantee to approximate DP guarantee.

In order to obtain our upper bound on the RDP of the shuffled model, we develop new analysis techniques which could be of independent interest. In particular, we develop a novel RDP analysis for neighboring datasets with a special structure, in which one of the datasets has all the data points to be the same. We first observe that the output distribution of the shuffling mechanism is the multinomial distribution. Using this observation, then we show that the ratio of the distributions of the mechanism on special structure neighboring datasets is a sub-Gaussian random variable (r.v.), and we can write the Rényi divergence of the shuffle mechanism in terms of the moments of this r.v. Bounding the moments of this r.v. then gives an upper bound on the RDP for the special neighboring datasets. A key technical result is then to relate the RDP of general neighboring datasets to those with special structure. To do so, a crucial observation is to write the output distribution of the local randomizer on the i'th client's data point as a mixture distribution, where the number of clients sampling from the same distribution is a Binomial random variable. Therefore, if we restrict the dataset to these clients only, the resulting datasets will have the special structure. Finally, in order to be able to reduce the problem to the special case, we remove the effect of the clients that do not sample from the same distribution without affecting the Rényi divergence.

Another technique for amplifying the privacy is subsampled mechanism, where we first take a random subsample of the dataset, and then apply a known randomized mechanism on the subsampled data points. This subsampled mechanism enables another privacy amplification opportunity, which, in several cases, is shown to yield a privacy advantage proportional to the subsampling rate (see [KLN11, Ull17]). We analyze the RDP of subsampled mechanisms in the shuffled framework for any discrete LDP mechanism by bounding the ternary $|\chi|^{\lambda}$-DP [WBK19] of the shuffled model using our previous technique. Our new bound saves a factor of 2.5× better than combining RDP of the shuffled model with the sub-sampling result in [WBK19].

The shuffled model of privacy has been of significant recent interest [EFM19, GGK19, BBG19c, GPV19, BBG19b, CSU19, BBG19d, BBG20b]. However, all the existing works in literature [EFM19, BBG19d, FMT22] only characterize the approximate DP of the shuffled model. To the best of our knowledge, there is no bound on RDP of the shuffle model in the literature except for the one mentioned briefly in a remark in [EFM19, Remark 1], which is obtained by the standard conversion results from DP to RDP. However, this bound is loose and not useful for conversion to approximate DP as well as for composition. Recently, authors in [FMT23] has proposed a reduction technique for the shuffled model that enables getting tighter results for the RDP of the shuffled model. The works [MTZ19, WBK19, ZW19] have studied the RDP of subsampled mechanisms *without shuffling*. They demonstrated that this

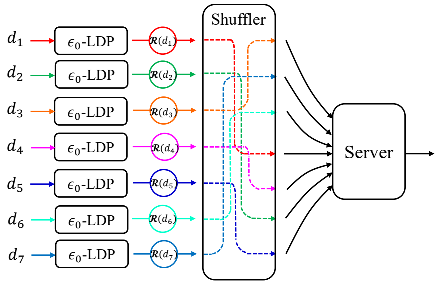


Figure 5.1: Shuffled model: clients apply the local randomizer $\mathcal{R}$ on their data points and send them to a secure shuffler that randomly permutes clients' messages before passing them to the server.

provides a tighter bound on the total privacy loss than the bound that can be obtained using the standard strong composition theorems. The RDP analysis of subsampled mechanisms in the shuffled privacy framework has not been studied before. One naive approach is to plug in the RDP analysis of shuffle model [GDD21e] into the results of [WBK19]; however, our direct analysis of subsampled mechanisms yields better results in several interesting regimes.

5.2 Problem Formulation

Let $\mathcal{D} = (d_1, \ldots, d_n)$ be a dataset consisting of n data points, where d_i is a data point at the i'th client that takes values from a set $\mathcal{X}$. Let $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ be a local randomizer that satisfies the following two properties:

1. $\mathcal{R}$ is an ε_0-LDP mechanism (see Definition 2.2.1).

2. The range of $\mathcal{R}$ is a discrete set, i.e., the output of $\mathcal{R}$ takes values in a discrete set $[B] = \{1, \ldots, B\}$ for some $B \in \mathbb{N} := \{1, 2, 3, \ldots\}$. Here, $[B]$ could be the whole of $\mathbb{N}$.

Client i applies $\mathcal{R}$ on d_i (each client uses independent randomness for computing $\mathcal{R}(d_i)$) and sends $\mathcal{R}(d_i)$ to the shuffler, who shuffles the received n inputs and outputs the result; see Figure 5.1. To formalize this, let $\mathcal{H}_n : \mathcal{Y}^n \to \mathcal{Y}^n$ denote the shuffling operation that takes n

inputs and outputs their uniformly random permutation. We define the shuffling mechanism as

$$\mathcal{M}\left(\mathcal{D}\right) := \mathcal{H}_n\left(\mathcal{R}\left(d_1\right), \ldots, \mathcal{R}\left(d_n\right)\right). \tag{5.1}$$

Our goal is to characterize the Rényi differential privacy of $\mathcal{M}$. Since the output of $\mathcal{M}$ is a random permutation of the n outputs of $\mathcal{R}$, the server cannot associate the n messages to the clients; and the only information it can use from the messages is the histogram, i.e., the number of messages that give any particular output in $[B]$. We define a set $\mathcal{A}_B^n$ as follows

$$\mathcal{A}_B^n = \left\{\boldsymbol{h} = (h_1, \ldots, h_B) : \sum_{j=1}^{B} h_j = n\right\}, \tag{5.2}$$

to denote the set of all possible histograms of the output of the shuffler with n inputs. Therefore, we can assume, without loss of generality (w.l.o.g.), that the output of $\mathcal{M}$ is a distribution over $\mathcal{A}_B^n$ for input dataset $\mathcal{D}$ of n data points.

We also characterize the RDP of the subsampled shuffled model that defined as follows. First subsample $k \leq n$ clients of the n clients (without replacement), where $\gamma = \frac{k}{n}$ denotes the sampling parameter. Each client i out of the k selected clients applies $\mathcal{R}$ on d_i and sends $\mathcal{R}(d_i)$ to the shuffler that randomly permutes the received k messages and outputs the result. We formally define the subsampled shuffled mechanism as

$$\mathcal{M}_s\left(\mathcal{D}\right) := \mathcal{H}_k \circ \text{samp}_{n,k}\left(\mathcal{R}\left(d_1\right), \ldots, \mathcal{R}\left(d_n\right)\right), \tag{5.3}$$

where $\mathcal{H}_k$ denotes the shuffling operation on k elements and $\text{samp}_{n,k}$ denotes the sampling operation for choosing a random subset of k elements from a set of n elements. A succinct summary of the notation used throughout the paper is given in Table 5.1.

5.3 RDP of the Shuffled Model

This section is dedicated to presenting upper and lower bounds on the RDP of the shuffled model.

Symbol	Description
$[B]$	$\{1, 2, \ldots, B\}$ for any $B \in \mathbb{N}$
ε_0	LDP parameter (see Definition 2.2.1)
(ε, δ)	Approximate DP parameters (see Definition 2.1.2)
$(\alpha, \varepsilon(\alpha))$	RDP parameters (see Definition 2.1.3)
$\mathcal{R} : \mathcal{X} \to [B]$	A discrete ε_0-LDP mechanism at clients for mapping their data points to elements in $[B]$
$\boldsymbol{p} = (p_1, \ldots, p_B)$	The output distribution of $\mathcal{R}$ when the data point is d
$\boldsymbol{p}' = (p'_1, \ldots, p'_B)$	The output distribution of $\mathcal{R}$ when the data point is d'
$\boldsymbol{p}_i = (p_{i1}, \ldots, p_{iB})$	The output distribution of $\mathcal{R}$ when the data point is d_i for $i \in [n]$
$\boldsymbol{p}'_n = (p'_{n1}, \ldots, p'_{nB})$	The output distribution of $\mathcal{R}$ when the data point is d'_n
$\mathcal{P}$	A collection of n distributions $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$
$\mathcal{P}_{-i}$	A collection of $(n-1)$ distributions $\mathcal{P} \setminus \{\boldsymbol{p}_i\}$
$\mathcal{P}_{\mathcal{C}}$, where $\mathcal{C} \subseteq [n-1]$	A collection of n distributions, where clients in the set $\mathcal{C}$ map according to $\boldsymbol{p}'_n$, clients in the set $[n-1] \setminus \mathcal{C}$ map according to $\tilde{\boldsymbol{p}}_i$ (see (5.19)), and client n maps according to $\boldsymbol{p}_n$ (see (5.20)-(5.22))
$\mathcal{A}_B^n$	A set of all possible histograms with B bins and n elements (see (5.2))
$\boldsymbol{h}$	$\boldsymbol{h} = (h_1, \ldots, h_B)$ with $\sum_{i=1}^{B} h_i = n$ is an element of $\mathcal{A}_B^n$
$\mathcal{M}(\mathcal{D})$	The shuffle mechanism $\mathcal{M}$ on the dataset $\mathcal{D} \in \mathcal{X}^n$; $\mathcal{M}(\mathcal{D})$ is a distribution over $\mathcal{A}_B^n$ (see (5.1))
$F(\mathcal{P})$	Distribution over $\mathcal{A}_B^n$ when client i maps its data point according to the distribution $\boldsymbol{p}_i$ (see (5.18))

Table 5.1: Notation used throughout Chaper 5

Theorem 5.3.1 (Upper Bound 1)**.** *For any $n \in \mathbb{N}$, $\varepsilon_0 \geq 0$,and any integer $\alpha \geq 2$, the RDP of the shuffle model is upper-bounded by*

$$\begin{aligned}\varepsilon(\alpha) \leq & \frac{1}{\alpha-1} \log \Bigg(1 + \binom{\alpha}{2} \frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{n} e^{\varepsilon_0}} \\ & + \sum_{i=3}^{\alpha} \binom{\alpha}{i} i\Gamma\left(i/2\right) \left(\frac{\left(e^{2\varepsilon_0}-1\right)^2}{2e^{2\varepsilon_0}\overline{n}}\right)^{i/2} + e^{\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}}\Bigg),\end{aligned} \tag{5.4}$$

where $\overline{n} = \lfloor \frac{n-1}{2e^{\varepsilon_0}} \rfloor + 1$ and $\Gamma\left(z\right) = \int_0^\infty x^{z-1}e^{-x}dx$ is the Gamma function.

We give a complete proof of Theorem 5.3.1 in Section 5.3.1.1. When n, ε_0, α satisfy a certain condition, we can simplify the bound in (5.4) to the following:

Corollary 5.3.1 (Simplified Upper Bound 1)**.** For any $n \in \mathbb{N}$, $\varepsilon_0 \geq 0$, and any integer $\alpha \geq 2$ that satisfy $\alpha^4 e^{5\varepsilon_0} < \frac{n}{9}$, we can simplify the bound in (5.4) to the following:

$$\varepsilon(\alpha) \leq \frac{1}{\alpha-1} \log \left(1 + \binom{\alpha}{2} \frac{4\left(e^{\varepsilon_0}-1\right)^2}{n}\right). \tag{5.5}$$

We prove Corollary 5.3.1 in Appendix D.1. Note that the upper bounds in Theorem 5.3.1 and Corollary 5.3.1 hold for any ε_0-LDP mechanism.

Remark 5.3.1. Note that any α, ε_0, n that satisfy $\alpha^4 e^{5\varepsilon_0} < \frac{n}{9}$ lead to the bound in (5.5). For example, we can take $\varepsilon_0 = c \ln n$ and $\alpha < \frac{n^{(1-5c)/4}}{2}$ for any $c < \frac{1}{5}$, and it will satisfy the condition. In particular, taking $\varepsilon_0 = \frac{1}{25} \ln n$ and $\alpha < \frac{n^{1/5}}{2}$ will also give the bound in (5.5).

Remark 5.3.2 (Generalization to real orders α)**.** Theorem 5.3.1 provides an upper bound on the RDP of the shuffle model for only integer orders $\alpha \geq 2$. However, the result can be generalized to real orders α using convexity of the function $(\alpha-1)\,\varepsilon\left(\alpha\right)$ as follows. From [EH14, Corollary 2], the function $(\alpha-1)\,D_\alpha\left(\mathbf{P}||\mathbf{Q}\right)$ is convex in α for any given two distributions $\mathbf{P}$ and $\mathbf{Q}$. Thus, for any real order $\alpha > 1$, we can bound the RDP of the shuffle model by

$$\varepsilon\left(\alpha\right) \leq \frac{a \cdot \left(\lfloor \alpha \rfloor - 1\right) \cdot \varepsilon\left(\lfloor \alpha \rfloor\right) + (1-a) \cdot \left(\lceil \alpha \rceil - 1\right) \cdot \varepsilon\left(\lceil \alpha \rceil\right)}{\alpha - 1}, \tag{5.6}$$

where $a = \lceil\alpha\rceil - \alpha$, since $\alpha = a\lfloor\alpha\rfloor + (1-a)\lceil\alpha\rceil$ for any real α. Here, $\lfloor\alpha\rfloor$ and $\lceil\alpha\rceil$ respectively denote the largest integer smaller than or equal to α and the smallest integer bigger than or equal to α.

In the following theorem, we also present another bound on RDP that readily holds for all $\alpha \geq 1$.

Theorem 5.3.2 (Upper Bound 2)**.** *For any $n \in \mathbb{N}$, $\varepsilon_0 \geq 0$, and any $\alpha \geq 1$ (including the non-integral α), the RDP of the shuffle model is upper-bounded by*

$$\varepsilon(\alpha) \leq \frac{1}{\alpha - 1} \log\left(e^{\alpha^2 \frac{(e^{\varepsilon_0}-1)^2}{\overline{n}}} + e^{\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}} \right), \tag{5.7}$$

where $\overline{n} = \lfloor \frac{n-1}{2e^{\varepsilon_0}} \rfloor + 1$.

We prove Theorem 5.3.2 in Section 5.3.1.2.

Remark 5.3.3 (Improved Upper Bounds – Saving a Factor of 2)**.** The exponential term $e^{\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}}$ in both the upper bounds stated in (5.4) and (5.7) comes from the Chernoff bound, where we naively choose the factor $\gamma = 1/2$ instead of optimizing it; see the proof of Theorem 5.3.1 in Section 5.3.1.1. If we instead had optimized γ and chosen it to be, for example, $\gamma = \sqrt{\frac{2\varepsilon_0 e^{\varepsilon_0}}{\sqrt{n}\log(n)}}$ (which goes to 0 when, say, $\varepsilon_0 \leq \frac{1}{4}\log(n)$), we would have asymptotically saved a multiplicative factor of 2 in the leading term in both upper bounds, because in this case we have $\overline{n} = \lfloor (1-\gamma)\frac{n-1}{e^{\varepsilon_0}} \rfloor + 1 \to \lfloor \frac{n-1}{e^{\varepsilon_0}} \rfloor + 1$ as $n \to \infty$. We chose to evaluate our bound with $\gamma = 1/2$ because of two reasons: first, it gives a simpler expression to compute; and second, the evaluated bound does not give good results (as compared to the ones with $\gamma = 1/2$) for the parameter ranges of interest.

Remark 5.3.4 (Difference in Upper Bounds)**.** Since the quadratic term in α inside the log in (5.7) has an extra multiplicative factor of e^{ε_0} in comparison with the corresponding term in (5.4), our first upper bound presented in Theorem 5.3.1 is better than our second upper bound presented in Theorem 5.3.2 for all parameter ranges of interest; see also Figure 5.2 in

Section 5.5. However, the expression in (5.7) is much cleaner to state as well as to compute as compared to that in (5.4). As we will see later, the techniques required to prove both upper bounds are different.

Remark 5.3.5 (Potentially Better Upper Bounds for Specific Mechanisms)**.** Since both our upper bounds are worse-case bounds that hold for *all* ε_0-LDP mechanisms, it is possible that for specific mechanisms, we may be able to exploit their structure for potentially better bounds. See Remark 5.3.8 on this just after (5.33).

The upper bounds on the RDP of the shuffle model presented in (5.4) and (5.7) are general and hold for any discrete ε_0-LDP mechanism. Furthermore, these bounds are in closed form expressions that can be easily implemented. To the best of our knowledge, there is no bound on RDP of the shuffle model in literature except for the one given in [EFM19, Remark 1], which we provide below[1] in (5.8). For the LDP parameter ε_0 and number of clients n, they showed that for any $\alpha > 1$, the shuffle mechanism $\mathcal{M}$ is $(\alpha, \varepsilon(\alpha))$-RDP, where

$$\varepsilon\left(\alpha\right) = \alpha \frac{2e^{4\varepsilon_0}\left(e^{\varepsilon_0}-1\right)^2}{n}. \tag{5.8}$$

In Section 5.5, we evaluate numerically the performance of both our bounds (from Theorems 5.3.1 and 5.3.2) against the above bound in (5.8). We demonstrate that both our bounds outperform the above bound in all cases; and in particular, the gap is significant when $\varepsilon_0 > 1$ – note that the bound in [EFM19] is worse than our simplified bound given in Corollary 5.3.1 by a multiplicative factor of $e^{4\varepsilon_0}$.

Theorem 5.3.3 (Lower Bound)**.** *For any $n \in \mathbb{N}$, $\varepsilon_0 \geq 0$, and any integer $\alpha \geq 2$, the RDP of the shuffle model is lower-bounded by:*

$$\begin{aligned}\varepsilon\left(\alpha\right) \geq & \frac{1}{\alpha-1}\log\Big(1+\binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{ne^{\varepsilon_0}} \\ & +\sum_{i=3}^{\alpha}\binom{\alpha}{i}\left(\frac{\left(e^{2\varepsilon_0}-1\right)}{ne^{\varepsilon_0}}\right)^i \mathbb{E}\left[\left(k-\frac{n}{e^{\varepsilon_0}+1}\right)^i\right]\Big),\end{aligned} \tag{5.9}$$

[1] As mentioned in Section 5.1, this was obtained by the standard conversion results from DP to RDP, which could be loose.

where expectation is taken w.r.t. the binomial random variable $k \sim Bin(n,p)$ with parameter $p = \frac{1}{e^{\varepsilon_0}+1}$.

We give a complete proof of Theorem 5.3.3 in Section 5.3.4. When i is an even integer, then the expectation term in (5.9) is positive. When $i \geq 3$ is an odd integer, then using the convexity of function $f(x) = x^i$, it follows from the Jensen's inequality (i.e., $\mathbb{E}f(X) \geq f(\mathbb{E}X)$) and $\mathbb{E}[k] = \frac{n}{e^{\varepsilon_0}+1}$, that $\mathbb{E}\left[\left(k - \frac{n}{e^{\varepsilon_0}+1}\right)^i\right] \geq \left(\mathbb{E}\left[k - \frac{n}{e^{\varepsilon_0}+1}\right]\right)^i = 0$. Using these observations, we can safely ignore the summation term from (5.9) and obtain the following simplified lower bound.

Corollary 5.3.2 (Simplified Lower Bound)**.** For any $n \in \mathbb{N}$, $\varepsilon_0 \geq 0$, and integer $\alpha \geq 2$, the RDP of the shuffle model is lower-bounded by:

$$\varepsilon(\alpha) \geq \frac{1}{\alpha - 1} \log\left(1 + \binom{\alpha}{2} \frac{\left(e^{\varepsilon_0} - 1\right)^2}{ne^{\varepsilon_0}}\right). \tag{5.10}$$

Remark 5.3.6 (Upper and Lower Bound Proofs)**.** Both our upper bounds stated in Theorems 5.3.1 and 5.3.2 hold for any ε_0-LDP mechanism. In other words, they are the worst case privacy bounds, in the sense that there is no ε_0-LDP mechanism for which the associated shuffle model gives a higher RDP parameter than those stated in (5.4) and (5.7). Therefore, the lower bound that we derive should serve as the lower bound on the RDP privacy parameter of the mechanism that achieves the largest privacy bound (i.e., worst privacy).

We prove our lower bound result (stated in Theorem 5.3.3) by showing that a specific mechanism (in particular, the binary Randomized response (RR)) on a specific pair of neighboring datasets yields the RDP privacy parameter stated in the right hand side (RHS) of (5.9). This implies that RDP privacy bound (which is the supremum over all neighboring datasets) of binary RR for the shuffle model is at least the bound stated in (5.9), which in turn implies that the lower bound (which is the tightest bound for any ε_0-LDP mechanism) is also at least that.

Remark 5.3.7 (Gap in Upper and Lower Bounds)**.** When comparing our simplified upper and lower bounds from Corollaries 5.3.1 and 5.3.2, respectively, we observe that when $\alpha^4 e^{5\varepsilon_0} < \frac{n}{9}$, our upper and lower bounds differ by a multiplicative factor of $4e^{\varepsilon_0}$. In our generic upper bound (5.4), note that when n is large, only the term corresponding to α^2 matters, and with our improved upper bound (which saves a factor of 2 in that term asymptotically – see Remark 5.3.3), the upper and lower bounds are away by the factor of e^{ε_0}, which tends to 1 as $\varepsilon_0 \to 0$. Thus, in the regime of large n and small ε_0, our upper and lower bounds coincide. Without any constraints on n, ε_0, we believe that our lower bound is tight. Closing this gap by showing a tighter upper bound is an interesting and important open problem.

5.3.1 Proofs of The Upper Bounds

In this section, we will prove our upper bounds stated in Theorems 5.3.1 and 5.3.2 in Sections 5.3.1.1 and 5.3.1.2, respectively.

5.3.1.1 Proof of Theorem 5.3.1

The proof has two main steps. In the first step, we reduce the problem of deriving RDP for arbitrary neighboring datasets to the problem of deriving RDP for specific neighboring datasets, $\mathcal{D}, \mathcal{D}'$, where all elements in $\mathcal{D}$ are the same and $\mathcal{D}'$ differs from $\mathcal{D}$ in one entry. In the second step, we derive RDP for the special neighboring datasets. The specific neighboring datasets to which we reduce our general problem have the following form:

$$\begin{aligned}\mathcal{D}^{m}_{\text{same}} =\Big\{&(\mathcal{D}_m, \mathcal{D}'_m) : \mathcal{D}_m = (d, \ldots, d, d) \in \mathcal{X}^m, \\ &\mathcal{D}'_m = (d, \ldots, d, d') \in \mathcal{X}^m, \text{ where } d, d' \in \mathcal{X}\Big\}.\end{aligned} \tag{5.11}$$

Consider arbitrary neighboring datasets $\mathcal{D} = (d_1, \ldots, d_n) \in \mathcal{X}^n$ and $\mathcal{D}' = (d_1, \ldots, d_{n-1}, d'_n) \in \mathcal{X}^n$. For any $m \in \{0, \ldots, n-1\}$, define new neighboring datasets $\mathcal{D}^{(n)}_{m+1} = (d'_n, \ldots, d'_n, d_n) \in \mathcal{X}^{m+1}$ and $\mathcal{D}'^{(n)}_{m+1} = (d'_n, \ldots, d'_n, d'_n) \in \mathcal{X}^{m+1}$, each having $(m+1)$ elements. Observe that $\left(\mathcal{D}'^{(n)}_{m+1}, \mathcal{D}^{(n)}_{m+1}\right) \in \mathcal{D}^{m+1}_{\text{same}}$. The first step of our proof is summarized in the following theorem.

Theorem 5.3.4 (Reduction to the Special Case)**.** *Let* $q = \frac{1}{e^{\varepsilon_0}}$ *and* $m \sim \mathrm{Bin}\,(n-1, q)$ *be a* *binomial random variable. We have:*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}')}\left[\left(\frac{\mathcal{M}(\mathcal{D})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}')(\boldsymbol{h})}\right)^{\alpha}\right]\\ &\quad\leq \mathbb{E}_{m\sim\text{Bin}(n-1,q)}\left[\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})}\left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})}\right)^{\alpha}\right]\right].\end{aligned} \tag{5.12}$$

We provide a complete proof of Theorem 5.3.4 in Section 5.3.2. Since E_m is precisely what is required to bound the RDP for the specific neighboring datasets, we have reduced the problem of computing RDP for arbitrary neighboring datasets to the problem of computing RDP for specific neighboring datasets. The second step of the proof bounds E_m, which follows from the result below that holds for any $m \in \mathbb{N}$.

Theorem 5.3.5 (RDP for the Special Case)**.** *Let $m \in \mathbb{N}$ be arbitrary. For any integer $\alpha \geq 2$, we have*

$$\begin{aligned}&\sup_{(\mathcal{D}_m,\mathcal{D}'_m)\in\mathcal{D}^m_{\text{same}}} \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})}\right)^{\alpha}\right]\\ &\quad\leq 1+\binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{me^{\varepsilon_0}}+\sum_{i=3}^{\alpha}\binom{\alpha}{i}i\Gamma(i/2)\left(\frac{\left(e^{2\varepsilon_0}-1\right)^2}{2me^{2\varepsilon_0}}\right)^{i/2}.\end{aligned} \tag{5.13}$$

We give a complete proof of Theorem 5.3.5 in Section 5.3.3. We show in Appendix D.2.1 that E_m is a non-increasing function of m. Using this and concentration properties of the Binomial r.v., we get:

$$\mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}')(\boldsymbol{h})}\right)^{\alpha}\right] \leq e^{\epsilon_0\alpha}e^{-\frac{q(n-1)\gamma^2}{2}} + E_{(1-\gamma)q(n-1)}, \tag{5.14}$$

where $\gamma > 0$ is arbitrary, and expectation is taken w.r.t. $\boldsymbol{h} \sim \mathcal{M}(\mathcal{D}')$. Note that we have already bounded E_m for all m in Theorem 5.3.5. By setting $\gamma = \frac{1}{2}$ and $\overline{n} = \lfloor (1-\gamma)q(n-1)\rfloor + 1 = \lfloor \frac{n-1}{2e^{\epsilon_0}}\rfloor + 1$, we get from Theorem 5.3.5, that:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}')}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}\right)\left(\boldsymbol{h}\right)}{\mathcal{M}\left(\mathcal{D}'\right)\left(\boldsymbol{h}\right)}\right)^{\alpha}\right] \leq E_{\overline{n}-1} + e^{\epsilon_0\alpha-\frac{n-1}{8e^{\epsilon_0}}}\\ &\leq 1+\binom{\alpha}{2}\frac{\left(e^{\epsilon_0}-1\right)^2}{\overline{n}e^{\epsilon_0}}+\sum_{i=3}^{\alpha}\binom{\alpha}{i}i\Gamma\left(i/2\right)\left(\frac{\left(e^{2\epsilon_0}-1\right)^2}{2\overline{n}e^{2\epsilon_0}}\right)^{i/2}+e^{\epsilon_0\alpha-\frac{n-1}{8e^{\epsilon_0}}}.\end{aligned} \tag{5.15}$$

Since the above bound holds for arbitrary pairs of neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, this completes the proof of Theorem 5.3.1.

5.3.1.2 Proof of Theorem 5.3.2

The proof of Theorem 5.3.2 follows the same steps as that of the proof of Theorem 5.3.1, except for the following change. Instead of using Theorem 5.3.5 for bounding the RDP for specific neighboring datasets, we will use the following theorem.

Theorem 5.3.6. *Let $m \in \mathbb{N}$ be arbitrary. For any $\alpha \geq 2$ (including the non-integral α) and any $(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}$, we have*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}'_m\right)(\boldsymbol{h})}{\mathcal{M}\left(\mathcal{D}_m\right)(\boldsymbol{h})}\right)^{\alpha}\right] \leq \exp\left(\alpha^2\frac{\left(e^{\epsilon_0}-1\right)^2}{m}\right). \tag{5.16}$$

We prove Theorem 5.3.6 in Appendix D.2.2. Note that Theorem 5.3.6 implies that $E_{m-1} \leq \exp\left(\alpha^2\frac{(e^{\epsilon_0}-1)^2}{m}\right)$ holds for every integer $m \geq 2$. Substituting this in (5.15) (by putting $m = \overline{n} = \lfloor\frac{n-1}{2e^{\epsilon_0}}\rfloor + 1$), we get

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}')}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}\right)(\boldsymbol{h})}{\mathcal{M}\left(\mathcal{D}'\right)(\boldsymbol{h})}\right)^{\alpha}\right] \leq e^{\alpha^2\frac{(e^{\epsilon_0}-1)^2}{\overline{n}}} + e^{\epsilon_0\alpha-\frac{n-1}{8e^{\epsilon_0}}}.$$

This proves Theorem 5.3.2.

5.3.2 Proof of the Reduction to the Special Case

In this section, we prove Theorem 5.3.4 by reducing the problem of computing RDP for the arbitrary pairs of neighboring datasets to the problem of computing RDP for the neighboring datasets with the special structure.

Recall that the LDP mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ has a discrete range $\mathcal{Y} = [B]$ for some $B \in \mathbb{N}$. Let $\boldsymbol{p}_i := (p_{i1}, \ldots, p_{iB})$ and $\boldsymbol{p}'_n := (p'_{n1}, \ldots, p'_{nB})$ denote the probability distributions over $\mathcal{Y}$ when the input to $\mathcal{R}$ is d_i and d'_n, respectively, where $p_{ij} = \Pr[\mathcal{R}(d_i) = j]$ and $p'_{nj} = \Pr[\mathcal{R}(d'_n) = j]$ for all $j \in [B]$ and $i \in [n]$. Let $\mathcal{P} = \{\boldsymbol{p}_i : i \in [n]\}$ and $\mathcal{P}' = \{\boldsymbol{p}_i : i \in [n-1]\}\bigcup\{\boldsymbol{p}'_n\}$. For $i \in [n-1]$, let $\mathcal{P}_{-i} = \mathcal{P} \setminus \{\boldsymbol{p}_i\}$, $\mathcal{P}'_{-i} = \mathcal{P}' \setminus \{\boldsymbol{p}_i\}$, and also $\mathcal{P}_{-n} = \mathcal{P} \setminus \{\boldsymbol{p}_n\}$, $\mathcal{P}'_{-n} = \mathcal{P}' \setminus \{\boldsymbol{p}'_n\}$. Here, $\mathcal{P}, \mathcal{P}'$ correspond to the datasets $\mathcal{D} = \{d_1, \ldots, d_n\}, \mathcal{D}' = \{d_1, \ldots, d_{n-1}, d'_n\}$, respectively, and for any $i \in [n]$, $\mathcal{P}_{-i}$ and $\mathcal{P}'_{-i}$ correspond to the datasets $\mathcal{D}_{-i} = \{d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_n\}$ and $\mathcal{D}'_{-i} = \{d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_{n-1}, d'_n\}$, respectively.

For any collection $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ of n distributions, we define $F(\mathcal{P})$ to be the distribution over $\mathcal{A}_B^n$ (which is the set of histograms on B bins with n elements as defined in (5.2)) that is induced when every client i (independent to the other clients) samples an element from $[B]$ accordingly to the probability distribution $\boldsymbol{p}_i$. Formally, for any $\boldsymbol{h} \in \mathcal{A}_B^n$, define

$$\begin{aligned} \mathcal{U}_{\boldsymbol{h}} := \Big\{ &(\mathcal{U}_1, \ldots, \mathcal{U}_B) : \mathcal{U}_1, \ldots, \mathcal{U}_B \subseteq [n] \\ &\text{s.t. } \bigcup_{j=1}^{B} \mathcal{U}_j = [n] \text{ and } |\mathcal{U}_j| = h_j, \forall j \in [B] \Big\}. \end{aligned} \tag{5.17}$$

Note that for each $(\mathcal{U}_1, \ldots, \mathcal{U}_B) \in \mathcal{U}_{\boldsymbol{h}}$, $\mathcal{U}_j$ for $j = 1, \ldots, B$ denotes the identities of the clients that map to the j'th element in $[B]$, where $\mathcal{U}_j$'s are disjoint for all $j \in [B]$. Note also that $|\mathcal{U}_{\boldsymbol{h}}| = \binom{n}{\boldsymbol{h}} = \frac{n!}{h_1! h_2! \ldots h_B!}$. It is easy to verify that for any $\boldsymbol{h} \in \mathcal{A}_B^n$, $F(\mathcal{P})(\boldsymbol{h})$ is equal to

$$F(\mathcal{P})(\boldsymbol{h}) = \sum_{(\mathcal{U}_1, \ldots, \mathcal{U}_B) \in \mathcal{U}_{\boldsymbol{h}}} \prod_{j=1}^{B} \prod_{i \in \mathcal{U}_j} p_{ij} \tag{5.18}$$

Similarly, we can define $F(\mathcal{P}'), F(\mathcal{P}_{-i}), F(\mathcal{P}'_{-i})$. Note that $F(\mathcal{P})$ and $F(\mathcal{P}')$ are distributions over $\mathcal{A}_B^n$, whereas, $F(\mathcal{P}_{-i})$ and $F(\mathcal{P}'_{-i})$ are distributions over $\mathcal{A}_B^{n-1}$. It is easy to see that $F(\mathcal{P}) = \mathcal{M}(\mathcal{D})$ and $F(\mathcal{P}') = \mathcal{M}(\mathcal{D}')$. Similarly, $F(\mathcal{P}_{-i}) = \mathcal{M}(\mathcal{D}_{-i})$ and $F(\mathcal{P}'_{-i}) = \mathcal{M}(\mathcal{D}'_{-i})$. Now we are ready to prove Theorem 5.3.4.

Since $\mathcal{R}$ is an ε_0-LDP mechanism, we have

$$e^{-\varepsilon_0} \leq \frac{p_{ij}}{p'_{nj}} \leq e^{\varepsilon_0}, \qquad \forall j \in [B] \,, i \in [n].$$

A crucial observation is that any distribution $\boldsymbol{p}_i$ can be written as the following mixture distribution:

$$\boldsymbol{p}_i = q \boldsymbol{p}'_n + (1-q)\, \tilde{\boldsymbol{p}}_i, \tag{5.19}$$

where $q = \frac{1}{e^{\varepsilon_0}}$. The distribution $\tilde{\boldsymbol{p}}_i = [\tilde{p}_{i1}, \ldots, \tilde{p}_{iB}]$ is given by $\tilde{p}_{ij} = \frac{p_{ij} - q p'_{nj}}{1-q}$, where it is easy to verify that $\tilde{p}_{ij} \geq 0$ and $\sum_{j=1}^{B} \tilde{p}_{ij} = 1$. This idea of writing the distribution of the output of an LDP mechanism as a mixture distribution is inspired from [BBG19d, FMT22]. However, we create different mixtures and use them in a distinct way to reduce the Renyi divergence calculation to those distributions with a certain neighborhood structure using Lemma 5.3.3.

Now we show that since each $\boldsymbol{p}_i = q\boldsymbol{p}'_n + (1-q)\tilde{\boldsymbol{p}}_i$ is a mixture distribution, we can write $F(\mathcal{P})$ and $F(\mathcal{P}')$ as certain convex combinations. Before stating the result, we need some notation.

For any $\mathcal{C} \subseteq [n-1]$, define two sets $\mathcal{P}_{\mathcal{C}}, \mathcal{P}'_{\mathcal{C}}$, having n distributions each, as follows:

$$\mathcal{P}_{\mathcal{C}} = \{\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_{n-1}\} \bigcup \{\boldsymbol{p}_n\}, \tag{5.20}$$

$$\mathcal{P}'_{\mathcal{C}} = \{\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_{n-1}\} \bigcup \{\boldsymbol{p}'_n\}, \tag{5.21}$$

where, for every $i \in [n-1]$, $\hat{\boldsymbol{p}}_i$ is defined as follows:

$$\hat{\boldsymbol{p}}_i = \begin{cases} \boldsymbol{p}'_n & \text{if } i \in \mathcal{C}, \\ \tilde{\boldsymbol{p}}_i & \text{if } i \in [n-1] \setminus \mathcal{C}. \end{cases} \tag{5.22}$$

Note that $\mathcal{P}_{\mathcal{C}}$ and $\mathcal{P}'_{\mathcal{C}}$ differ only in one distribution, where $\mathcal{P}_{\mathcal{C}}$ contains $\boldsymbol{p}_n$ whereas $\mathcal{P}'_{\mathcal{C}}$ contains $\boldsymbol{p}'_n$. In words, if clients map their data points according to the distributions in either $\mathcal{P}_{\mathcal{C}}$ or $\mathcal{P}'_{\mathcal{C}}$ for any $\mathcal{C} \subseteq [n-1]$, then for all clients $i \in \mathcal{C}$, the i'th client maps its data point according to $\boldsymbol{p}'_n$ (which is the distribution of $\mathcal{R}$ on input d'_n), and for all clients $i \in [n-1] \setminus \mathcal{C}$, the i'th client maps its data point according to $\tilde{\boldsymbol{p}}_i$. The last client maps its data point according to $\boldsymbol{p}_n$ or $\boldsymbol{p}'_n$ depending on whether the set is $\mathcal{P}_{\mathcal{C}}$ or $\mathcal{P}'_{\mathcal{C}}$.

In the following lemma, we show that $F(\mathcal{P})$ and $F(\mathcal{P}')$ can be written as convex combinations of $\{F(\mathcal{P}_{\mathcal{C}}) : \mathcal{C} \subseteq [n-1]\}$ and $\{F(\mathcal{P}'_{\mathcal{C}}) : \mathcal{C} \subseteq [n-1]\}$, respectively, where for any $\mathcal{C} \subseteq [n-1]$, both $F(\mathcal{P}_{\mathcal{C}})$ and $F(\mathcal{P}'_{\mathcal{C}})$ can be computed analogously as in (5.18).

Lemma 5.3.1 (Mixture Interpretation)**.** *$F(\mathcal{P})$ and $F(\mathcal{P}')$ can be written as the following convex combinations:*

$$F(\mathcal{P}) = \sum_{\mathcal{C} \subseteq [n-1]} q^{|\mathcal{C}|}(1-q)^{n-|\mathcal{C}|-1} F(\mathcal{P}_{\mathcal{C}}), \tag{5.23}$$

$$F(\mathcal{P}') = \sum_{\mathcal{C}\subseteq[n-1]} q^{|\mathcal{C}|}(1-q)^{n-|\mathcal{C}|-1}F(\mathcal{P}'_{\mathcal{C}}), \tag{5.24}$$

where $\mathcal{P}_{\mathcal{C}}, \mathcal{P}'_{\mathcal{C}}$ *are defined in* (5.20)-(5.22)*.*

We prove Lemma 5.3.1 in Appendix D.3.1. Now, using Lemma 5.3.1, in the following lemma we show that the Rényi divergence between $F(\mathcal{P})$ and $F(\mathcal{P}')$ can be upper-bounded by a convex combination of the Rényi divergence between $F(\mathcal{P}_{\mathcal{C}})$ and $F(\mathcal{P}'_{\mathcal{C}})$ for $\mathcal{C} \subseteq [n-1]$.

Lemma 5.3.2 (Joint Convexity)**.** *For any* $\alpha > 1$*, the function* $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\alpha}\right]$ *is jointly convex in* $(F(\mathcal{P}), F(\mathcal{P}'))$*, i.e.,*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F\left(\mathcal{P}\right)\left(\boldsymbol{h}\right)}{F\left(\mathcal{P}'\right)\left(\boldsymbol{h}\right)}\right)^{\alpha}\right]\\ &\leq \sum_{\mathcal{C}\subseteq[n-1]} q^{|\mathcal{C}|}\left(1-q\right)^{n-|\mathcal{C}|-1}\mathbb{E}_{\boldsymbol{h}\sim F\left(\mathcal{P}'_{\mathcal{C}}\right)}\left[\left(\frac{F\left(\mathcal{P}_{\mathcal{C}}\right)\left(\boldsymbol{h}\right)}{F\left(\mathcal{P}'_{\mathcal{C}}\right)\left(\boldsymbol{h}\right)}\right)^{\alpha}\right].\end{aligned} \tag{5.25}$$

We prove Lemma 5.3.2 in Appendix D.3.2. For any $\mathcal{C} \subseteq [n-1]$, let $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} = \{\tilde{\boldsymbol{p}}_i : i \in [n-1]\setminus\mathcal{C}\}$. With this notation, note that $\mathcal{P}_{\mathcal{C}} \setminus \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} = \{\boldsymbol{p}'_n, \ldots, \boldsymbol{p}'_n\}\bigcup\{\boldsymbol{p}_n\}$ and $\mathcal{P}'_{\mathcal{C}} \setminus \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} = \{\boldsymbol{p}'_n, \ldots, \boldsymbol{p}'_n\}\bigcup\{\boldsymbol{p}'_n\}$ is a pair of specific neighboring distributions, each containing $|\mathcal{C}|+1$ distributions. In other words, if we define $\mathcal{D}^{(n)}_{|\mathcal{C}|+1} = (d'_n, \ldots, d'_n, d_n)$ and $\mathcal{D}'^{(n)}_{|\mathcal{C}|+1} = (d'_n, \ldots, d'_n, d'_n)$, each having $(|\mathcal{C}|+1)$ data points, then the mechanisms $\mathcal{M}(\mathcal{D}^{(n)}_{|\mathcal{C}|+1})$ and $\mathcal{M}(\mathcal{D}'^{(n)}_{|\mathcal{C}|+1})$ will have distributions $F(\mathcal{P}_{\mathcal{C}} \setminus \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}})$ and $F(\mathcal{P}'_{\mathcal{C}} \setminus \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}})$, respectively.

Now, since $(\mathcal{D}'^{(n)}_{|\mathcal{C}|+1}, \mathcal{D}^{(n)}_{|\mathcal{C}|+1}) \in \mathcal{D}^{|\mathcal{C}|+1}_{\text{same}}$, if we remove the effect of distributions in $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}}$ in the RHS of (5.25), we would be able to bound the RHS of (5.25) using the RDP for the special neighboring datasets in $\mathcal{D}^{|\mathcal{C}|+1}_{\text{same}}$. This is precisely what we will do in the following lemma and the subsequent corollary, where we will eliminate the distributions in $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}}$ in the RHS (5.25).

The following lemma holds for arbitrary pairs $(\mathcal{P}, \mathcal{P}')$ of neighboring distributions $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ and $\mathcal{P}' = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}, \boldsymbol{p}'_n\}$, where we show that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\alpha}\right]$ does not decrease when we eliminate a distribution $\boldsymbol{p}_i$ (i.e., remove the data point d_i from the datasets) for any $i \in [n-1]$. We need this general statement as it will be required in the proof of Theorem 5.3.1 later.

Lemma 5.3.3 (Monotonicity)**.** *For any $i \in [n-1]$, we have*

$$\mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{F\left(\mathcal{P}\right)\left(\boldsymbol{h}\right)}{F\left(\mathcal{P}'\right)\left(\boldsymbol{h}\right)} \right)^{\alpha} \right] \leq \mathbb{E}_{\boldsymbol{h} \sim F\left(\mathcal{P}'_{-i}\right)} \left[\left(\frac{F\left(\mathcal{P}_{-i}\right)\left(\boldsymbol{h}\right)}{F\left(\mathcal{P}'_{-i}\right)\left(\boldsymbol{h}\right)} \right)^{\alpha} \right], \tag{5.26}$$

where, for $i \in [n-1]$, $\mathcal{P}_{-i} = \mathcal{P} \setminus \{\boldsymbol{p}_i\}$ and $\mathcal{P}'_{-i} = \mathcal{P}' \setminus \{\boldsymbol{p}_i\}$. Note that in the left hand side (LHS) of (5.26)*, $F(\mathcal{P}), F(\mathcal{P}')$ are distributions over $\mathcal{A}_B^n$, whereas, in the RHS, $F(\mathcal{P}_{-i}), F(\mathcal{P}'_{-i})$ for any $i \in [n-1]$ are distributions over $\mathcal{A}_B^{n-1}$.*

We prove Lemma 5.3.3 in Appendix D.3.3. Note that Lemma 5.3.3 is a general statement that holds for arbitrary pairs $(\mathcal{P}, \mathcal{P}')$ of neighboring distributions. For our purpose, we apply Lemma 5.3.3 with $(\mathcal{P}_{\mathcal{C}}, \mathcal{P}'_{\mathcal{C}})$ for any $\mathcal{C} \subseteq [n-1]$ and then eliminate the distributions in $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}}$ one by one. The result is stated in the following corollary.

Corollary 5.3.3. Consider any $m \in \{0, 1, \ldots, n-1\}$. Let $\mathcal{D}_{m+1}^{(n)} = (d'_n, \ldots, d'_n, d_n)$ and $\mathcal{D}'^{(n)}_{m+1} = (d'_n, \ldots, d'_n)$. Then, for any $\mathcal{C} \in \binom{[n-1]}{m}$ (i.e., $\mathcal{C} \subseteq [n-1]$ such that $|\mathcal{C}| = m$), we have

$$\mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}'_{\mathcal{C}})} \left[\left(\frac{F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h})}{F(\mathcal{P}'_{\mathcal{C}})(\boldsymbol{h})} \right)^{\alpha} \right] \leq \mathbb{E}_{\boldsymbol{h} \sim \mathcal{M}(\mathcal{D}'^{(n)}_{m+1})} \left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})} \right)^{\alpha} \right]. \tag{5.27}$$

We prove Corollary 5.3.3 in Appendix D.3.4. Substituting from (5.27) into (5.25) and noting that for every $\boldsymbol{h} \in \mathcal{A}_B^n$, $F(\mathcal{P})(\boldsymbol{h})$ and $F(\mathcal{P}')(\boldsymbol{h})$ are distributionally equal to $\mathcal{M}(\mathcal{D})(\boldsymbol{h})$ and $\mathcal{M}(\mathcal{D}')(\boldsymbol{h})$, respectively, we get

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{h} \sim \mathcal{M}(\mathcal{D}')} \left[\left(\frac{\mathcal{M}\left(\mathcal{D}\right)\left(\boldsymbol{h}\right)}{\mathcal{M}\left(\mathcal{D}'\right)\left(\boldsymbol{h}\right)} \right)^{\alpha} \right] \\
&\overset{\text{(a)}}{\leq} \sum_{m=0}^{n-1} \sum_{\mathcal{C} \in \binom{[n-1]}{m}} q^{m} \left(1-q\right)^{n-m-1} \mathbb{E}_{\boldsymbol{h} \sim F\left(\mathcal{P}'_{\mathcal{C}}\right)} \left[\left(\frac{F\left(\mathcal{P}_{\mathcal{C}}\right)\left(\boldsymbol{h}\right)}{F\left(\mathcal{P}'_{\mathcal{C}}\right)\left(\boldsymbol{h}\right)} \right)^{\alpha} \right] \\
&\overset{\text{(b)}}{\leq} \sum_{m=0}^{n-1} \sum_{\mathcal{C} \in \binom{[n-1]}{m}} q^{m} \left(1-q\right)^{n-m-1} \mathbb{E}_{\boldsymbol{h} \sim \mathcal{M}(\mathcal{D}'^{(n)}_{m+1})} \left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})} \right)^{\alpha} \right]
\end{aligned}$$

$$\begin{aligned}&\overset{\text{(c)}}{=} \sum_{m=0}^{n-1} \binom{n-1}{m} q^m (1-q)^{n-m-1} \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})} \left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})} \right)^{\alpha} \right] \\ &= \mathbb{E}_{m\sim\text{Bin}(n-1,q)} \left[\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})} \left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})} \right)^{\alpha} \right] \right].\end{aligned}$$

The inequality (a) is the same as (5.25), just writing it differently. In (b) we used (5.27) and in (c) we used the fact that number of m-sized subsets of $[n-1]$ is equal to $\binom{n-1}{m}$. This completes the proof of Theorem 5.3.4.

5.3.3 Proof of RDP for the Special Form

Fix an arbitrary $m \in \mathbb{N}$ and consider any pair of neighboring datasets $(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}$. Let $\mathcal{D}_m = (d, \ldots, d) \in \mathcal{X}^m$ and $\mathcal{D}'_m = (d, \ldots, d, d') \in \mathcal{X}^m$. Let $\boldsymbol{p} = (p_1, \ldots, p_B)$ and $\boldsymbol{p}' = (p'_1, \ldots, p'_B)$ be the probability distributions of the discrete ε_0-LDP mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y} = [B]$ when its inputs are d and d', respectively, where $p_j = \Pr[\mathcal{R}(d) = j]$ and $p'_j = \Pr[\mathcal{R}(d') = j]$ for all $j \in [B]$. Since $\mathcal{R}$ is ε_0-LDP, we have

$$e^{-\varepsilon_0} \leq \frac{p_j}{p'_j} \leq e^{\varepsilon_0}, \qquad \forall j \in [B]. \tag{5.28}$$

Since $\mathcal{M}$ is a shuffle mechanism, it induces a distribution on $\mathcal{A}^m_B$ for any input dataset. So, for any $\boldsymbol{h} \in \mathcal{A}^m_B$, $\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})$ and $\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})$ are equal to the probabilities of seeing $\boldsymbol{h}$ when the inputs to $\mathcal{M}$ are $\mathcal{D}_m$ and $\mathcal{D}'_m$, respectively. Thus, for a given histogram $\boldsymbol{h} = (h_1, \ldots, h_B) \in \mathcal{A}^m_B$ with m elements and B bins, we have

$$\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h}) = MN(m, \boldsymbol{p}, \boldsymbol{h}) = \binom{m}{\boldsymbol{h}} \prod_{j=1}^{B} p_j^{h_j}, \tag{5.29}$$

where $MN(m, \boldsymbol{p}, \boldsymbol{h})$ denotes the Multinomial distribution with $\binom{m}{\boldsymbol{h}} = \frac{m!}{h_1! \cdots h_B!}$. Note that (5.29) can be obtained as a special case of the general distribution in (5.18) by putting $\boldsymbol{p}_j = \boldsymbol{p}$ for each client j.

For $\mathcal{M}(\mathcal{D}'_m)$, note that the last client (independent of the other clients) maps its input data point d' to the j'th bin with probability p'_j, and the remaining $(m-1)$ clients' mappings induce a distribution on $\mathcal{A}_B^{m-1}$. Thus, $\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})$ for any $\boldsymbol{h} \in \mathcal{A}_B^m$ can be written as

$$\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h}) = \sum_{j=1}^{B} p'_j MN\left(m-1, \boldsymbol{p}, \widetilde{\boldsymbol{h}}_j\right), \tag{5.30}$$

where $\widetilde{\boldsymbol{h}}_j = (h_1, \ldots, h_{j-1}, h_j - 1, h_{j+1}, \ldots, h_B) \in \mathcal{A}_B^{m-1}$. We implicitly assume that if $h_j = 0$ for some $j \in [B]$, then $MN\left(m-1, \boldsymbol{p}, \widetilde{\boldsymbol{h}}_j\right) = 0$ as one of the elements is negative. Note that similar to (5.29), (5.30) can also be obtained from (5.18) as a special case. Using the polynomial expansion $(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i$ (with $x = \frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} - 1$ in the following), we have:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)} &\left[\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})}\right)^{\alpha}\right] \\ &= \sum_{i=0}^{\alpha} \binom{\alpha}{i} \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)} \left[\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)^{i}\right].\end{aligned} \tag{5.31}$$

Let $X : \mathcal{A}_B^m \to \mathbb{R}$ be a random variable associated with the distribution $\mathcal{M}(\mathcal{D}_m)$ on $\mathcal{A}_B^m$, and for any $\boldsymbol{h} \in \mathcal{A}_B^m$, define $X(\boldsymbol{h}) := m\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)$. Substituting this in (5.31) gives:

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)} \left[\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})}\right)^{\alpha}\right] = 1 + \sum_{i=1}^{\alpha} \binom{\alpha}{i} \frac{\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[(X(\boldsymbol{h}))^i\right]}{m^i}. \tag{5.32}$$

The RHS of (5.32) is in terms of the moments of X, which we bound in the following lemma. Before that, first we simplify the expression for $X(\boldsymbol{h})$ by computing the ratio $\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})}$ for any $\boldsymbol{h} \in \mathcal{A}_B^m$:

$$\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} = \sum_{j=1}^{B} p'_j \frac{MN(m-1, \boldsymbol{p}, \widetilde{\boldsymbol{h}}_j)}{MN(m, \boldsymbol{p}, \boldsymbol{h})} = \sum_{j=1}^{B} \frac{p'_j}{p_j} \frac{h_j}{m}. \tag{5.33}$$

Thus, we get $X(\boldsymbol{h}) = m\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right) = \left(\sum_{j=1}^{B} \frac{p'_j}{p_j} h_j\right) - m$.

Remark 5.3.8. As mentioned in Remark 5.3.5, we could tighten our upper bounds for specific mechanisms. As shown in (5.32) above, the Rényi divergence of a mechanism between two neighboring datasets can be written in terms of the moments of a r.v. X, which is defined as the ratio of distributions of the mechanism on these two neighboring datasets. However, since our goal is to bound RDP for all ε_0-LDP mechanisms, we prove the worse-case bound on the moments of X that holds for all mechanisms; see (5.35) in Lemma 5.3.4 for bound on the $i \geq 3$'rd moments of X and (5.39) in Lemma 5.3.5 for bound on the variance of X.

Lemma 5.3.4. *The random variable X has the following properties:*

1. *X has zero mean, i.e., $\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[X(\boldsymbol{h})\right]=0$.*
2. *The variance of X is equal to*
$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[X(\boldsymbol{h})^2\right]=m\left(\sum_{j=1}^{B}\frac{p_j'^2}{p_j}-1\right). \tag{5.34}$$
3. *For $i \geq 3$, the i'th moment of X is bounded by*
$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[(X(\boldsymbol{h}))^i\right]\leq i\Gamma\left(i/2\right)\left(2m\nu^2\right)^{i/2}, \tag{5.35}$$
where $\nu^2=\frac{\left(e^{\varepsilon_0}-e^{-\varepsilon_0}\right)^2}{4}$ and $\Gamma\left(z\right)=\int_0^\infty x^{z-1}e^{-x}dx$ is the Gamma function.

A proof of Lemma 5.3.4 is presented in Appendix D.4.1. Substituting the bounds from Lemma 5.3.4 into (5.32), we get

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}_m'\right)(\boldsymbol{h})}{\mathcal{M}\left(\mathcal{D}_m\right)(\boldsymbol{h})}\right)^{\alpha}\right]&\leq 1+\binom{\alpha}{2}\frac{1}{m}\left(\sum_{j=1}^{B}\frac{p_j'^2}{p_j}-1\right)\\&+\sum_{i=3}^{\alpha}\binom{\alpha}{i}i\Gamma\left(i/2\right)\left(\frac{\left(e^{\varepsilon_0}-e^{-\varepsilon_0}\right)^2}{2m}\right)^{i/2}\end{aligned} \tag{5.36}$$

Note that $p_1,\ldots,p_m,p_1',\ldots,p_m'$ are defined for the fixed pair of datasets $(\mathcal{D}_m,\mathcal{D}_m')\in\mathcal{D}_{\text{same}}^m$ that we started with. So, the term containing $\left(\sum_{j=1}^{B}\frac{p_j'^2}{p_j}-1\right)$ in the RHS of (5.36) depends on $(\mathcal{D}_m,\mathcal{D}_m')$, and that is the only term in (5.36) that depends on $(\mathcal{D}_m,\mathcal{D}_m')$. Since Theorem 5.3.5 requires us to bound (5.36) for any pair of neighboring datasets $(\mathcal{D}_m,\mathcal{D}_m')\in\mathcal{D}_{\text{same}}^m$, so, in order to prove Theorem 5.3.5, we need to compute $\sup_{(\mathcal{D}_m,\mathcal{D}_m')\in\mathcal{D}_{\text{same}}^m}\left(\sum_{j=1}^{B}\frac{p_j'^2}{p_j}-1\right)$. We bound this in the following.

Define a set $\mathcal{T}_{\varepsilon_0}$ consisting of all pairs of B-dimensional probability vectors satisfying the ε_0-LDP constraints as follows:

$$\mathcal{T}_{\varepsilon_0}=\Big\{(\boldsymbol{p},\boldsymbol{p}')\in\mathbb{R}^B\times\mathbb{R}^B:p_j,p_j'\geq 0,\forall j\in[B],\sum_{j=1}^{B}p_j=\sum_{j=1}^{B}p_j'=1,$$

$$\text{and } e^{-\varepsilon_0} \leq \frac{p'_j}{p_j} \leq e^{\varepsilon_0}, \forall j \in [B]\Big\}. \tag{5.37}$$

Note that $\mathcal{T}_{\varepsilon_0}$ contains *all* pairs of the output probability distributions $(\boldsymbol{p}, \boldsymbol{p}')$ of *all* ε_0-LDP mechanisms $\mathcal{R}$ on *all* neighboring data points $d, d' \in \mathcal{X}$. Since any $(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}$ generates a pair of probability distributions $(\boldsymbol{p}, \boldsymbol{p}') \in \mathcal{T}_{\varepsilon_0}$ (because $\mathcal{D}_m = (d, \ldots, d)$ and $\mathcal{D}'_m = (d, \ldots, d, d')$ together contain only two distinct data points d, d'), we have

$$\sup_{(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}} \left(\sum_{j=1}^{B} \frac{p'^2_j}{p_j} - 1 \right) \leq \sup_{(\boldsymbol{p}, \boldsymbol{p}') \in \mathcal{T}_{\varepsilon_0}} \left(\sum_{j=1}^{B} \frac{p'^2_j}{p_j} - 1 \right). \tag{5.38}$$

In the following lemma, we bounds the RHS of (5.38).

Lemma 5.3.5. *We have the following bound:*

$$\sup_{(\boldsymbol{p}, \boldsymbol{p}') \in \mathcal{T}_{\varepsilon_0}} \left(\sum_{j=1}^{B} \frac{p'^2_j}{p_j} - 1 \right) = \frac{\left(e^{\varepsilon_0} - 1\right)^2}{e^{\varepsilon_0}}. \tag{5.39}$$

We prove Lemma 5.3.5 in Appendix D.4.2. Taking supremum over $(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}$ in (5.36) and then using (5.38) and (5.39), we get the bound in Theorem 5.3.5.

5.3.4 Lower Bound

In this section, we provide a proof of Theorem 5.3.3. Consider the binary case, where each data point d can take a value from $\mathcal{X} = \{0, 1\}$. Let the local randomizer $\mathcal{R}$ be the binary randomized response (2RR) mechanism, where $\Pr\left[\mathcal{R}\left(d\right) = d\right] = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}$ for $d \in \mathcal{X}$. It is easy to verify that $\mathcal{R}$ is an ε_0-LDP mechanism. For simplicity, let $p = \frac{1}{e^{\varepsilon_0}+1}$. Consider two neighboring datasets $\mathcal{D}$, $\mathcal{D}' \in \{0, 1\}^n$, where $\mathcal{D} = (0, \ldots, 0, 0)$ and $\mathcal{D}' = (0, \ldots, 0, 1)$. Let $k \in \{0, \ldots, n\}$ denote the number of ones in the output of the shuffler. Since the output of the shuffle mechanism $\mathcal{M}$ can be thought of as the distribution of the number of ones in the output, we have that $k \sim \mathcal{M}(\mathcal{D})$ is distributed as a Binomial random variable $\text{Bin}(n, p)$. Thus, we have

$$\mathcal{M}(\mathcal{D})(k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

$$\mathcal{M}(\mathcal{D}')(k) = (1-p)\binom{n-1}{k-1}p^{k-1}(1-p)^{n-k} \\ + p\binom{n-1}{k}p^{k}(1-p)^{n-k-1}.$$

It will be useful to compute $\frac{\mathcal{M}(\mathcal{D})(k)}{\mathcal{M}(\mathcal{D}')(k)} - 1$ for the calculations later.

$$\begin{aligned}\frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)} - 1 &= \frac{k}{n}\frac{(1-p)}{p} + \frac{(n-k)}{n}\frac{p}{(1-p)} - 1 \\ &= \frac{k}{n}e^{\varepsilon_0} + \frac{(n-k)}{n}e^{-\varepsilon_0} - 1 \\ &= \frac{k}{n}\left(e^{\varepsilon_0} - e^{-\varepsilon_0}\right) + e^{-\varepsilon_0} - 1 \\ &= \frac{k}{n}\left(\frac{e^{2\varepsilon_0}-1}{e^{\varepsilon_0}}\right) - \left(\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}}\right) \\ &= \left(\frac{e^{2\varepsilon_0}-1}{ne^{\varepsilon_0}}\right)\left(k - \frac{n}{e^{\varepsilon_0}+1}\right) \qquad (5.40)\end{aligned}$$

Thus, we have that

$$\begin{aligned}&\mathbb{E}_{k\sim\mathcal{M}(\mathcal{D})}\left[\left(\frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)}\right)^{\alpha}\right] = \mathbb{E}\left[\left(1 + \frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)} - 1\right)^{\alpha}\right] \\ &\overset{\text{(a)}}{=} 1 + \sum_{i=1}^{\alpha}\binom{\alpha}{i}\mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)} - 1\right)^{i}\right] \\ &\overset{\text{(b)}}{=} 1 + \sum_{i=2}^{\alpha}\binom{\alpha}{i}\mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)} - 1\right)^{i}\right] \\ &= 1 + \sum_{i=2}^{\alpha}\binom{\alpha}{i}\left(\frac{(e^{2\varepsilon_0}-1)}{ne^{\varepsilon_0}}\right)^{i}\mathbb{E}\left[\left(k - \frac{n}{e^{\varepsilon_0}+1}\right)^{i}\right] \qquad \text{(from (5.40))} \\ &\overset{\text{(c)}}{=} 1 + \binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{ne^{\varepsilon_0}} + \sum_{i=3}^{\alpha}\binom{\alpha}{i}\left(\frac{(e^{2\varepsilon_0}-1)}{ne^{\varepsilon_0}}\right)^{i}\mathbb{E}\left[\left(k - \frac{n}{e^{\varepsilon_0}+1}\right)^{i}\right].\end{aligned}$$

Here, step (a) from the polynomial expansion $(1+x)^n = \sum_{k=0}^{n}\binom{n}{k}x^k$, step (b) follows because the term corresponding to $i = 1$ is zero (i.e., $\mathbb{E}_{k\sim\mathcal{M}(\mathcal{D})}\left[\left(\frac{\mathcal{M}(\mathcal{D}')(k)}{\mathcal{M}(\mathcal{D})(k)} - 1\right)\right] = 0$), and step (c) from the from the fact that $\mathbb{E}_{k\sim\mathcal{M}(\mathcal{D})}\left[\left(k - \frac{n}{e^{\varepsilon_0}+1}\right)^2\right] = np(1-p) = \frac{ne^{\varepsilon_0}}{(e^{\varepsilon_0}+1)^2}$, which is equal to the variance of the Binomial random variable. In view of Remark 5.3.6, this completes the proof of Theorem 5.3.3.

5.4 RDP of the Subsampled Shuffled Model

In this section, we characterize the RDP of the subsampled shuffled mechanism by presenting an upper bound in Theorem 5.4.1 and a lower bound in Theorem 5.4.2. We then present the privacy-convergence trade-offs of the CLDP-SGD Algorithm 4.3.1 in Theorem 5.4.3.

Theorem 5.4.1 (Upper Bound). *For any $n \in \mathbb{N}$, $k \leq n$, $\varepsilon_0 \geq 0$, and any integer $\alpha \geq 2$, the RDP of the subsampled shuffle mechanism $\mathcal{M}$ (defined in* (5.1)*) is upper-bounded by*

$$\varepsilon(\alpha) \leq \frac{1}{\alpha-1}\log\left(1+4\binom{\alpha}{2}\gamma^2\frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{k}e^{\varepsilon_0}}+\sum_{j=3}^{\alpha}\binom{\alpha}{j}\gamma^j j\Gamma\left(j/2\right)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{\overline{k}e^{2\varepsilon_0}}\right)^{j/2}+\Upsilon\right),$$

where $\overline{k} = \lfloor\frac{k-1}{2e^{\varepsilon_0}}\rfloor + 1$, $\gamma = \frac{k}{n}$, and $\Gamma\left(z\right) = \int_0^\infty x^{z-1}e^{-x}dx$ is the Gamma function. The term Υ is given by $\Upsilon = \left(\left(1+\gamma\frac{e^{2\varepsilon_0}-1}{e^{\varepsilon_0}}\right)^\alpha - 1 - \alpha\gamma\frac{e^{2\varepsilon_0}-1}{e^{\varepsilon_0}}\right)e^{-\frac{k-1}{8e^{\varepsilon_0}}}$.

Theorem 5.4.2 (Lower Bound). *For any $n \in \mathbb{N}$, $k \leq n$, $\varepsilon_0 \geq 0$, and any integer $\alpha \geq 2$, the RDP of the subsampled shuffle mechanism $\mathcal{M}$ (defined in* (5.1)*) is lower-bounded by*

$$\varepsilon\left(\alpha\right) \geq \frac{1}{\alpha-1}\log\left(1+\binom{\alpha}{2}\gamma^2\frac{\left(e^{\varepsilon_0}-1\right)^2}{ke^{\varepsilon_0}}+\sum_{j=3}^{\alpha}\binom{\alpha}{j}\gamma^j\left(\frac{\left(e^{2\varepsilon_0}-1\right)}{ke^{\varepsilon_0}}\right)^j\mathbb{E}\left(m-\frac{k}{e^{\varepsilon_0}+1}\right)^j\right),$$

where expectation is taken w.r.t. the binomial r.v. $m \sim Bin(k,p)$ with parameter $p = \frac{1}{e^{\varepsilon_0}+1}$.

The proof of Theorem 5.4.1 is presented in Section 5.4.1. The proof of Theorem 5.4.2 can be obtained by following the same steps as the proof of Theorem 5.3.3 presented in Section 5.3. Our CLDP-SGD Algorithm 4.3.1 and its privacy-convergence trade-offs (stated in Theorem 5.4.3 below) are given for a general local randomizer $\mathcal{R}_p$ (whose inputs comes from an ℓ_p-ball for any $p \in [1,\infty]$) that satisfies the following conditions: (i) The randomized mechanism $\mathcal{R}_p$ is an ε_0-LDP mechanism. (ii) The randomized mechanism $\mathcal{R}_p$ is unbiased, i.e., $\mathbb{E}\left[\mathcal{R}_p\left(\mathbf{x}\right)|\mathbf{x}\right] = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{B}_p(a)$, where a is the radius of the ball $\mathbb{B}_p$. (iii) The output of the randomized mechanism $\mathcal{R}_p$ can be represented using $b \in \mathbb{N}^+$ bits. (iv) The randomized $\mathcal{R}_p$ has a bounded MSE: $\sup_{\mathbf{x}\in\mathbb{B}_p(a)}\mathbb{E}\|\mathcal{R}_p\left(\mathbf{x}\right)-\mathbf{x}\|_2^2 \leq L^2 f_p^2(\varepsilon_0, b)$, where $f_p^2(\varepsilon_0, b)$ is a function from $\mathbb{R}^+ \times \mathbb{N}^+$ to $\mathbb{R}^+$.

In Chapter 3, we proposed unbiased ε_0-LDP mechanisms $\mathcal{R}_p$ for several values of norms $p \in [1, \infty]$ that require $b = \mathcal{O}(\log(d))$ bits of communication in the high privacy regimes and satisfy the above conditions. The privacy-convergence trade-off of our CLDP-SGD algorithm is given below.

Theorem 5.4.3 (Privacy-Convergence tradeoffs)**.** *Let the set $\mathcal{C}$ be convex with diameter D and the function $f(\theta; .) : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ be convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of the ℓ_p-norm. Let $\theta^* = \arg\min_{\theta \in \mathcal{C}} F(\theta)$ denote the minimizer of the problem* (4.1)*. For $\gamma = \frac{k}{n}$, if we run Algorithm $\mathcal{A}_{\text{cldp}}$ over T iterations, then we have*

1. ***Privacy:** $\mathcal{A}_{\text{cldp}}$ is (ε, δ)-DP, where $\delta > 0$ is arbitrary and ε is given by*

$$\varepsilon = \min_{\alpha} \left(T\varepsilon(\alpha) + \frac{\log(1/\delta) + (\alpha - 1)\log(1 - 1/\alpha) - \log(\alpha)}{\alpha - 1} \right), \tag{5.41}$$

where $\varepsilon(\alpha)$ is the RDP of the subsampled shuffle mechanism given in Theorem 5.4.1.

2. ***Communication:** Our algorithm $\mathcal{A}_{cldp}$ requires $\frac{k}{n} \times b$ bits of communication in expectation[2] per client per iteration, where expectation is taken with respect to client sampling.*

3. ***Convergence:** If we run $\mathcal{A}_{cldp}$ with learning rate schedule $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = L^2\left(1 + \frac{f_p(\varepsilon_0, b)}{\gamma m n}\right)$, then*

$$\mathbb{E}[F(\theta_T)] - F(\theta^*) \leq \mathcal{O}\left(\frac{DG\log(T)}{\sqrt{T}}\right). \tag{5.42}$$

The proof of Theorem 5.4.3 is as follows: Note that $\mathcal{A}_{\text{cldp}}$ is an iterative algorithm, where in each iteration we use the subsampled shuffle mechanism as defined in (5.3), for which we have computed the RDP guarantees in Theorem 5.4.1. Now, for the privacy analysis of $\mathcal{A}_{\text{cldp}}$, we use the adaptive composition theorem from [Mir17, Proposition 1] and then use the RDP to DP conversion given in Lemma 2.1.3. For the convergence analysis, we use a

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

standard non-private SGD convergence result and compute the required parameters for that (see similar proof in Section 4.3.1.3).

Remark 5.4.1. Note that our convergence bound is affected by the MSE of the ε_0-LDP mechanism $\mathcal{R}_p$. For example, when f is L-Lipschitz continuous w.r.t. the ℓ_2-norm, we can use the LDP mechanism $\mathcal{R}_{v,m,s}^{\ell_2}$ proposed in Section 3.6 with parameters $v = \varepsilon_0$, $m = 1$, and $s = \lceil \varepsilon_0 \rceil$ that has MSE $f_2(\varepsilon_0, b) = \tilde{\mathcal{O}}\left(\frac{d}{n\min\{\varepsilon_0^2, \varepsilon_0\}}\right)$. When f is L-Lipschitz continuous w.r.t. the ℓ_1-norm or ℓ_∞-norm, we can use the LDP mechanisms $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ or $\mathcal{R}_{\varepsilon_0,1,\varepsilon_0}^{\ell_\infty}$, respectively, proposed in Chapter 3. By plugging these variances $f_p(\varepsilon_0, b)$ (for $p = 1, 2, \infty$) into Theorem 5.4.3, we get the convergence rate of the L-Lipschitz continuous loss function w.r.t. the ℓ_p-norm (for $p = \infty, 2, 1$).

Remark 5.4.2. The privacy parameter in (5.41) is not in a closed form expression and could be obtained by solving an optimization problem. However, we numerically compute it for several interesting regimes of parameters in our numerical experiments; see Section 5.5 for more details.

5.4.1 Proof of Theorem 5.4.1: Upper Bound

Recall from (5.3), for any dataset $\mathcal{D}_n = (d_1, \ldots, d_n) \in \mathcal{X}^n$ containing n data points, the subsampled-shuffle mechanism is defined as $\mathcal{M}_s(\mathcal{D}) := \mathcal{H}_k \circ \text{samp}_k^n\left(\mathcal{R}(d_1), \ldots, \mathcal{R}(d_n)\right)$. The proof of Theorem 5.4.1 consists of two steps. First, we bound the ternary-$|\chi|^\lambda$-DP of the shuffle mechanism $\mathcal{M}_{sh}$, which is the main technical contribution in this proof. Then, using this, we bound the RDP of the subsampled shuffle mechanism $\mathcal{M}$.

Theorem 5.4.4 (ζ-ternary-$|\chi|^\lambda$-DP of the shuffle mechanism $\mathcal{M}_{sh}$)**.** *For any integer $k \geq 2$, $\varepsilon_0 > 0$, and all $\lambda \geq 2$, the ζ-ternary-$|\chi|^\lambda$-DP of the shuffle mechanism $\mathcal{M}_{sh}$ is bounded by:*

$$\zeta(\lambda)^\lambda \leq \begin{cases} 4\frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{k}e^{\varepsilon_0}} + \left(e^{\varepsilon_0} - e^{-\varepsilon_0}\right)^\lambda e^{-\frac{k-1}{8e^{\varepsilon_0}}} & \textit{if } \lambda = 2, \\ \lambda\Gamma(\lambda/2)\left(\frac{2\left(e^{2\varepsilon_0}-1\right)^2}{\overline{k}e^{2\varepsilon_0}}\right)^{\lambda/2} + \left(e^{\varepsilon_0} - e^{-\varepsilon_0}\right)^\lambda e^{-\frac{k-1}{8e^{\varepsilon_0}}} & \textit{otherwise}, \end{cases} \tag{5.43}$$

where $\overline{k} = \lfloor \frac{k-1}{2e^{\varepsilon_0}} \rfloor + 1$ *and* $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ *is the Gamma function.*

Theorem 5.4.4 is one of the core technical results of this paper, and we prove it in Section 5.4.2. It was shown in [WBK19, Proposition 16] that if a mechanism obeys ζ-ternary-$|\chi|^\lambda$-DP, then its subsampled version (with subsampling parameter γ) will obey $\gamma\zeta$-ternary-$|\chi|^\lambda$-DP. Using that result, the authors then bounded the RDP of the subsampled mechanism in [WBK19, Eq. (9)]. Adapting that result to our setting, we have the following lemma.

Lemma 5.4.1 (From ζ-ternary-$|\chi|^\lambda$-DP to subsampled RDP)**.** *Suppose the shuffle mechanism* $\mathcal{M}_{sh}$ *obeys* ζ*-ternary-*$|\chi|^\lambda$*-DP. For any* $\alpha \geq 2, k \leq n$*, RDP of the subsampled shuffle mechanism* $\mathcal{M}$ *(with subsampling parameter* $\gamma = k/n$*) is bounded by:* $\varepsilon(\alpha) \leq \frac{1}{\alpha-1} \log\big(1 + \sum_{\lambda=2}^{\alpha} \binom{\alpha}{\lambda} \gamma^\lambda \zeta(\lambda)^\lambda\big)$.

Lemma 5.4.1 can be seen as a corollary to [WBK19, Proposition 16 and Eq. (9)]. Substituting the bound on $\zeta(\lambda)$ from Theorem 5.4.4 into Lemma 5.4.1 together with some algebraic manipulation gives proves Theorem 5.4.1.

5.4.2 Proof of Theorem 5.4.4: Ternary $|\chi|^\alpha$-DP of the Shuffle Model

The proof follows the same two steps as our results in Section 5.3. In the first step, we reduce the problem of deriving ternary divergence for arbitrary neighboring datasets to the problem of deriving the ternary divergence for specific neighboring datasets, $\mathcal{D} \sim \mathcal{D}' \sim \mathcal{D}''$, where all elements in $\mathcal{D}$ are the same and $\mathcal{D}', \mathcal{D}''$ differ from $\mathcal{D}$ in one entry. In the second step, we derive the ternary divergence for the special neighboring datasets.

Consider arbitrary neighboring datasets $\mathcal{D} = (d_1, \ldots, d_{k-1}, d_k)$, $\mathcal{D}' = (d_1, \ldots, d_{k-1}, d_k')$, and $\mathcal{D}'' = (d_1, \ldots, d_{k-1}, d_k'')$, each having k elements. For any $m \in \{0, \ldots, k-1\}$, we define new neighboring datasets $\mathcal{D}_{m+1}^{(k)} = (d_k'', \ldots, d_k'', d_k)$, $\mathcal{D}_{m+1}'^{(k)} = (d_k'', \ldots, d_k'', d_k')$, and $\mathcal{D}_{m+1}''^{(k)} = (d_k'', \ldots, d_k'')$, each having $m+1$ elements. Observe that $(\mathcal{D}_{m+1}''^{(k)}, \mathcal{D}_{m+1}'^{(k)}, \mathcal{D}_{m+1}^{(k)}) \in \mathcal{D}_{\text{same}}^m$. The first step of the proof is given in the following theorem.

Theorem 5.4.5 (Reduction to the Special Case)**.** *Let $q = \frac{1}{e^{\varepsilon_0}}$. We have:*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}'')}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D})(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}')(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}'')(\boldsymbol{h})}\right|^{\alpha}\right]$$
$$\leq \mathbb{E}_{m\sim\text{Bin}(k-1,q)}\left[\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m+1})(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{m+1})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}\right|^{\alpha}\right]\right]. \tag{5.44}$$

We know (by Chernoff bound) that the binomial r.v. is concentrated around its mean, which implies that the terms in the RHS of (5.44) that correspond to $m < (1-\tau)q(k-1)$ (we will take $\tau = 1/2$) will contribute in a negligible amount. Then we show that $E_m := \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m+1})(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{m+1})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}\right|^{\alpha}\right]$ is a non-increasing function of m. These observation together imply that the RHS in (5.12) is approximately equal to $E_{(1-\tau)q(k-1)}$. Since E_m is precisely what is required to bound the ternary DP for the specific neighboring datasets, we have reduced the problem of computing the ternary DP for arbitrary neighboring datasets to the problem of computing ternary DP for specific neighboring datasets. The second step of the proof bounds $E_{(1-\tau)q(n-1)}$, which follows from the result below that holds for any $m \in \mathbb{N}$.

Theorem 5.4.6 ($|\chi|^{\alpha}$-DP for special case)**.** *For any $m \in \mathbb{N}$, integer $\alpha \geq 2$, and $(\mathcal{D}''_m, \mathcal{D}'_m, \mathcal{D}_m) \in \mathcal{D}^m_{same}$,*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})}\right|^{\alpha}\right] \leq \begin{cases} 4\frac{(e^{\varepsilon_0}-1)^2}{me^{\varepsilon_0}} & \textit{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{2(e^{2\varepsilon_0}-1)^2}{me^{2\varepsilon_0}}\right)^{\alpha/2} & \textit{otherwise.} \end{cases}$$

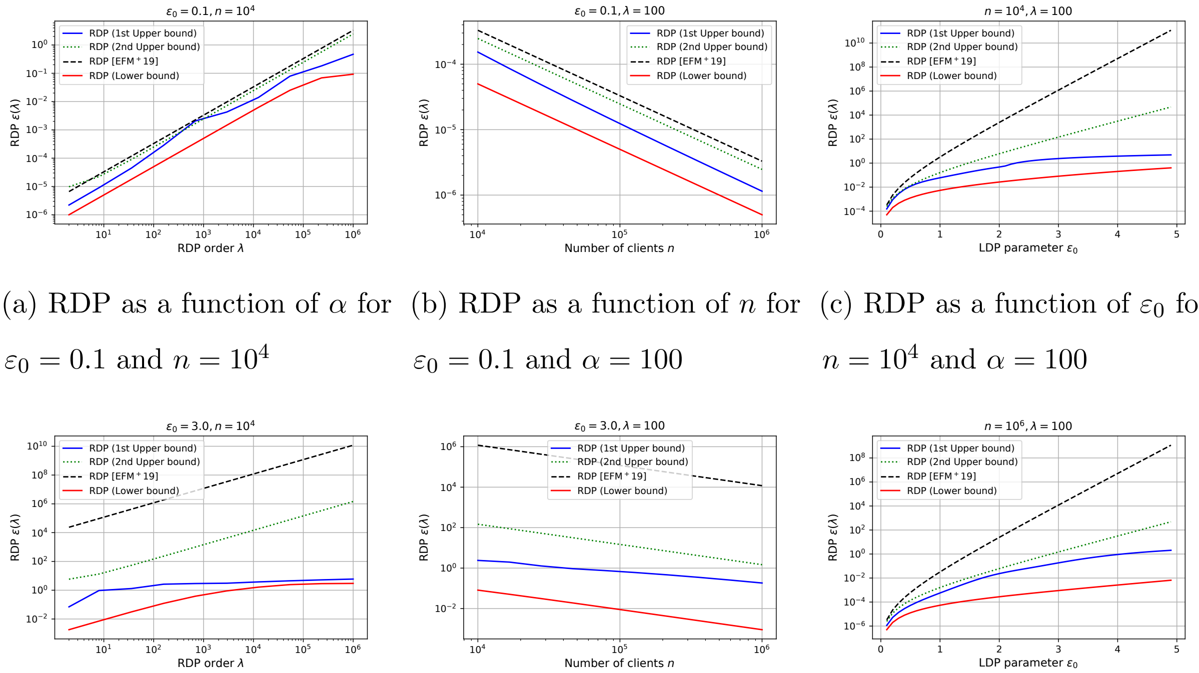
Proof of Theorem 5.4.5. Let $\boldsymbol{p}_i, i \in [k], \boldsymbol{p}'_k, \boldsymbol{p}''_k$ denote the distributions of $\mathcal{R}$ when the input data point is d_i, d'_k, d''_k, respectively. The main idea of the proof is the observation that each $\boldsymbol{p}_i$ can be written as a mixture distribution $\boldsymbol{p}_i = \frac{1}{e^{\varepsilon_0}}\boldsymbol{p}''_k + \left(1 - \frac{1}{e^{\varepsilon_0}}\right)\tilde{\boldsymbol{p}}_i$, where $\tilde{\boldsymbol{p}}_i$ is defined in terms of $\boldsymbol{p}_i, \boldsymbol{p}''_k$. So, instead of client $i \in [k-1]$ mapping its data point d_i according to $\boldsymbol{p}_i$, we can view it as the client i maps d_i according to $\boldsymbol{p}''_k$ with probability (w.p.) ${}^1/_{e^{\varepsilon_0}}$ and according to $\tilde{\boldsymbol{p}}_i$ w.p. $(1 - {}^1/_{e^{\varepsilon_0}})$. As a result, the number of clients that sample from the distribution $\boldsymbol{p}''_k$ follows a binomial distribution $\text{Bin}(k-1, {}^1/_{e^{\varepsilon_0}})$. This allows us to write the distribution of $\mathcal{M}_{sh}$ when clients map their data points according to $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k, \boldsymbol{p}'_k, \boldsymbol{p}''_k$

as a convex combination of the distribution of $\mathcal{M}$ when clients map their data points according to $\tilde{\boldsymbol{p}}_1, \ldots, \tilde{\boldsymbol{p}}_{k-1}, \boldsymbol{p}_k, \boldsymbol{p}'_k, \boldsymbol{p}''_k$. Then using a joint convexity argument, we write the ternary divergence between the original triple of distributions of $\mathcal{M}_{sh}$ in terms of the same convex combination of the ternary divergence between the resulting triples of distributions of $\mathcal{M}_{sh}$. Using a monotonicity argument, we can remove the effect of clients that do not sample from the distribution $\boldsymbol{p}''_k$ without decreasing the ternary divergence. By this chain of arguments, we have reduced the problem to the one involving the computation of ternary divergence only for the special form of neighboring datasets (as in Theorem 5.4.6), which proves Theorem 5.4.5. ■

Proof of Theorem 5.4.6. Consider $(\mathcal{D}''_m, \mathcal{D}'_m, \mathcal{D}_m) \in \mathcal{D}^m_{\text{same}}$ as in the statement of Theorem 5.4.6. First, we observe that for any $\alpha \geq 1$ and any three distributions p, q, r over the same domain, we can write $\mathbb{E}_r\left[\left|\frac{p-q}{r}\right|^\alpha\right] \leq 2^{\alpha-1}\left(\mathbb{E}_r\left[\left|\frac{p}{r}-1\right|^\alpha\right] + \mathbb{E}_r\left[\left|\frac{q}{r}-1\right|^\alpha\right]\right)$. This is a straight-forward application of the standard inequality $|x+y|^\alpha \leq 2^{\alpha-1}(|x|^\alpha + |y|^\alpha)$ which holds for all $x, y \in \mathbb{R}$ and $\alpha \geq 1$. Now, by taking $p = \mathcal{M}_{sh}(\mathcal{D}'_m)$, $q = \mathcal{M}_{sh}(\mathcal{D}''_m)$, and $r = \mathcal{M}_{sh}(\mathcal{D}_m)$, we reduce the problem of computing the ternary $|\chi|^\alpha$-divergence (which we need to bound) to the problem of computing the Pearson-Vajda divergence [WBK19], which we can write in terms of the α-th absolute moment of the r.v. $X : \mathcal{A}^m_B \to \mathbb{R}$, defined as $X(\boldsymbol{h}) := \left(\frac{\mathcal{M}_{sh}(\mathcal{D}')(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)$ for all $\boldsymbol{h} \in \mathcal{A}^m_B$ (where $\mathcal{D}' \in \{\mathcal{D}'_m, \mathcal{D}''_m\}$) and distributed according to $X(\boldsymbol{h}) \sim \mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})$. In Section 5.3, we have bounded the absolute moments of the r.v. $X(\boldsymbol{h})$ by showing that $X(\boldsymbol{h})$ is sub-Gaussian r.v. and using standard concentration results. ■

5.5 Numerical Results

In this section, we present numerical experiments to show the performance of our bounds on the RDP of the shuffle model and its usage for getting approximate DP and composition results.



(a) RDP as a function of α for $\varepsilon_0 = 0.1$ and $n = 10^4$

(b) RDP as a function of n for $\varepsilon_0 = 0.1$ and $\alpha = 100$

(c) RDP as a function of ε_0 for $n = 10^4$ and $\alpha = 100$

(d) RDP as a function of α for $\varepsilon_0 = 3$ and $n = 10^4$

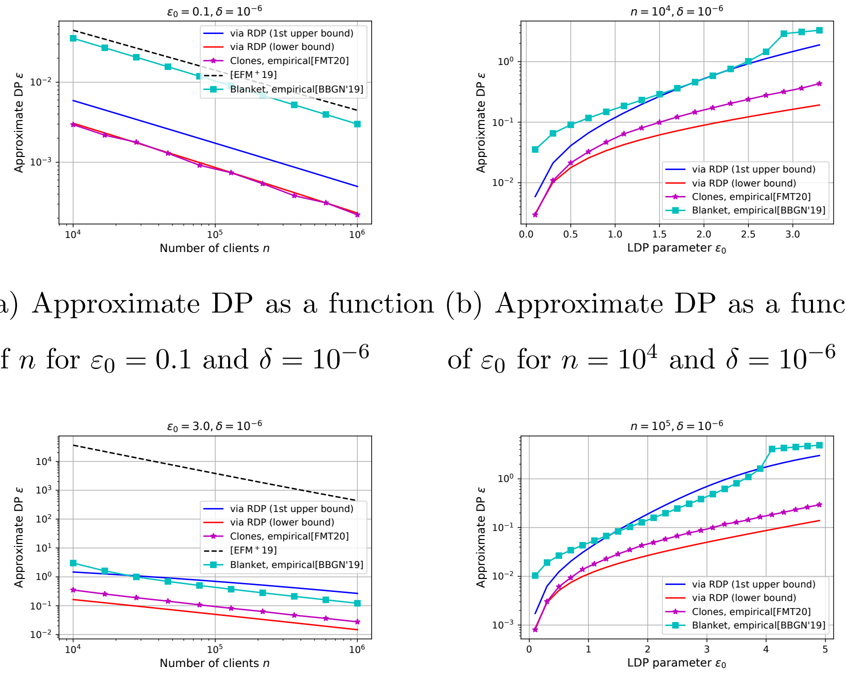
(e) RDP as a function of n for $\varepsilon_0 = 3$ and $\alpha = 100$

(f) RDP as a function of ε_0 for $n = 10^6$ and $\alpha = 100$

Figure 5.2: Comparison of several bounds on the RDP of the shuffle model.

RDP of the shuffle model: In Figure 5.2, we plot several bounds on the RDP of the shuffle model in different regimes. In particular, we compare between the first upper bound on the RDP given in Theorem 5.3.1, the second upper bound on the RDP given in Theorem 5.3.2, the lower bound on the RDP given in Theorem 5.3.3, and the upper bound on the RDP given in [EFM19, Remark 1] and stated in (5.8).[3] It is clear that our first upper bound (5.4) gives a tighter bound on the RDP in comparison with the second bound (5.7) and the upper bound given in [EFM19]. Furthermore, the first upper bound is close to the lower bound for small values of the LDP parameter ε_0 and for high orders α. In addition, the gap between our proposed bound in Theorem 5.3.1 and the bound given in [EFM19] increases as the LDP parameter ε_0 increases. We also observe that the curves of the lower and upper bounds on the

[3] The results in [FMT22] are for approximate DP (not for RDP), that is why we did not compare with them in Figure 5.2.



(a) Approximate DP as a function of n for $\varepsilon_0 = 0.1$ and $\delta = 10^{-6}$

(b) Approximate DP as a function of ε_0 for $n = 10^4$ and $\delta = 10^{-6}$

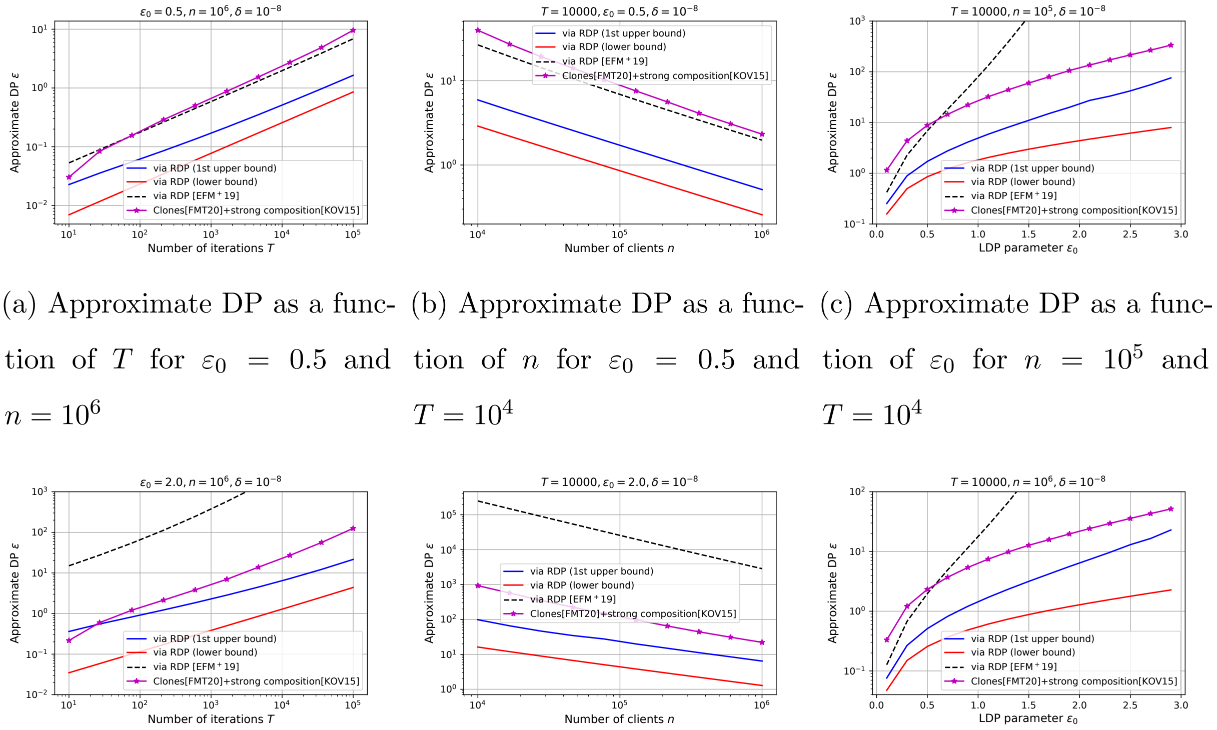
(c) Approximate DP as a function of n for $\varepsilon_0 = 3$ and $\delta = 10^{-6}$

(d) Approximate DP as a function of ε_0 for $n = 10^5$ and $\delta = 10^{-6}$

Figure 5.3: Comparison of several bounds on the Approximate (ε, δ)-DP of the shuffle model for $\delta = 10^{-6}$.

RDP of the shuffle model saturate close to ε_0 when the order α approaches to infinity. This indicates that the pure DP of the shuffle model is bounded below by ε_0, an observation made in literature [EFM19, BC20]. As can be seen in Figures 5.2d and 5.2e, the RDP obtained by standard approximate DP to RDP conversion in [EFM19, Remark 1], can be several orders of magnitude loose in comparison to our analysis.

Approximate DP of the shuffle model: Analyzing RDP of the shuffle model provides a bound on the approximate DP of the shuffle model from the relation between the RDP and approximate DP as shown in Lemma 2.1.3. In Figure 5.3, we plot several bounds on the approximate (ε, δ)-DP of the shuffle model for fixed $\delta = 10^{-6}$. In Figures 5.3d and 5.3b, we do not plot the results given in [EFM19], since their bounds are quite loose and are far from the plotted range when $\varepsilon_0 > 1$. We can see that our analysis of the RDP of the shuffle model provides a tighter bound on the approximate DP of the shuffle model in comparison with the



(a) Approximate DP as a function of T for $\varepsilon_0 = 0.5$ and $n = 10^6$

(b) Approximate DP as a function of n for $\varepsilon_0 = 0.5$ and $T = 10^4$

(c) Approximate DP as a function of ε_0 for $n = 10^5$ and $T = 10^4$

(d) Approximate DP as a function of T for $\varepsilon_0 = 2$ and $n = 10^6$

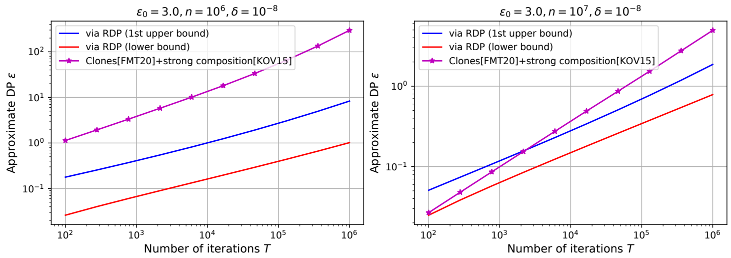
(e) Approximate DP as a function of n for $\varepsilon_0 = 2$ and $T = 10^4$

(f) Approximate DP as a function of ε_0 for $n = 10^6$ and $T = 10^4$

Figure 5.4: Comparison of several bounds on the Approximate (ε, δ)-DP for composition of a sequence of shuffle models for $\delta = 10^{-8}$.

bound given in [BBG19d] in some regimes. However, our RDP analysis performs worse than the best known bound given in [FMT22], when used without composition. This might be due to the gap between our upper and lower bound on the RDP of the shuffle model as the lower bound provides better performance than the bound given in [FMT22] for all values of LDP parameter ε_0. Note that the main use case for converting our RDP analysis to approximate DP is after composition rather than in the single-shot conversion illustrated in Figure 5.3.

Composition of a sequence of shuffle models: We now numerically evaluate the privacy parameters of the approximate (ε, δ)-DP for a composition of T mechanisms $(\mathcal{M}_1, \ldots, \mathcal{M}_T)$,



(a) Approximate DP as a function of T for $\varepsilon_0 = 3$, $\gamma = 0.001$ and $n = 10^6$.

(b) Approximate DP as a function of n for $\varepsilon_0 = 3$, $\gamma = 0.001$ and $n = 10^7$.

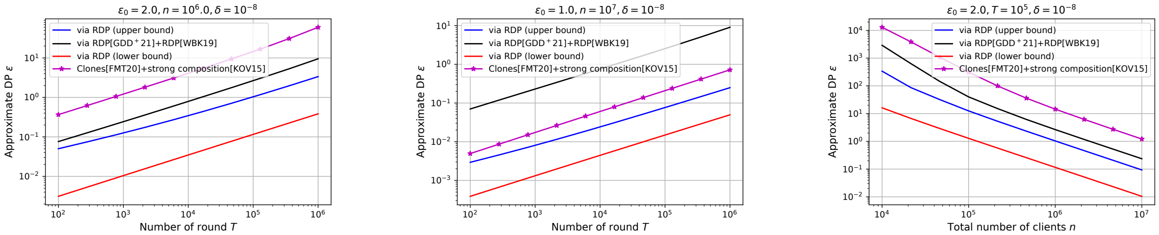
Figure 5.5: Comparison of several bounds on the Approximate (ε, δ)-DP for composition a sequence of shuffle models with Poisson sub-sampling for $\delta = 10^{-8}$ and $\gamma = 0.001$.

where $\mathcal{M}_t$ is a shuffle mechanism for all $t \in [T]$. In Figure 5.4, we plot three different bounds on the overall privacy parameter ε for fixed $\delta = 10^{-8}$ for a composition of T identical shuffle models. The first bound on the overall privacy parameter ε is obtained as a function of δ and the number of iterations T by optimizing over the RDP order α using our upper bound on the RDP of the shuffle model given in Theorem 5.3.1. The second bound is obtained by optimizing over the RDP order α using the upper bound on the RDP of the shuffle model given in [EFM19]. The third bound is obtained by first computing the privacy parameters $(\tilde{\varepsilon}, \tilde{\delta})$ of the shuffle model given in [FMT22]. Then, we use the strong composition theorem given in [KOV15] to obtain the overall privacy loss ε. We observe that there is a significant saving in the overall privacy parameter ε-DP using our bound on RDP in comparison with using the bound on DP [FMT22] with the strong composition theorem [KOV15]. For example, we save a factor of $8\times$ in computing the overall privacy parameter ε for number of iterations $T = 10^5$, LDP parameter $\varepsilon_0 = 0.5$, and number of clients $n = 10^6$. We observe that the bound given in [FMT22] with the strong composition theorem [KOV15] behaves better for small number of iterations $T < 10$ and large LDP parameter $\varepsilon_0 = 2$. However, the typical number of iterations T in the standard SGD algorithm is usually larger. Therefore, this demonstrates the significance of our RDP analysis for composition in the regimes of interest.

Privacy amplification by shuffling and Poisson sub-sampling: In the Differentially Private Stochastic Gradient Descent (DP-SGD), shuffling and sampling the dataset at each iteration are important tools to provide a strong privacy guarantee [GDD21d, EFM20a]. In these frameworks, the further advantage of sampling with shuffling[4] can be analyzed by standard combination of approximate DP with Poisson subsampling [LQS12]. The resulting approximate DP along with the strong composition theorem given in [KOV15] gives the overall privacy loss ε. An alternate path we use is to combine our RDP analysis with sampling of RDP mechanisms using [WBK19, ZW19]. This enables us to get an RDP guarantee with sampling, which we can then compose using properties of RDP. We can use the conversion from RDP to approximate DP to obtain a bound on the overall privacy loss of multiple iterations. In Figure 5.5, we compare our results of amplifying the RDP of the shuffle model by Poisson sub-sampling to the strong composition [KOV15] after getting the approximate DP of the shuffle model given in [FMT22] with Poisson sub-sampling given in [LQS12]. We observe that we save a factor of 11× by using our RDP bound for $n = 10^6$ and $\gamma = 0.001$. However, we can see that the gap between our (lower/upper) bounds and the strong composition decreases when $n = 10^7$. This could be due to the simplistic combination of our analysis with the RDP subsampling of [ZW19].

Composition of a sequence of subsampled shuffled models: In Figure 5.4, we plot several bounds on the approximate (ε, δ)-DP for a composition of T mechanisms $(\mathcal{M}_1, \ldots, \mathcal{M}_T)$, where $\mathcal{M}_t$ is a subsampled shuffled mechanism for $t \in [T]$. In all our experiments reported in Figure 5.6, we fix $\delta = 10^{-8}$. We observe that our new bound on the RDP of the subsampled shuffled mechanism achieves a significant saving in total privacy ε compared to the state-of-the-art. For example, we save a factor of 14× compared to the bound on DP [FMT22] with strong composition theorem [KOV15] and 2.5× compared to the bound on the RDP given in [GDD21e] with subsampled RDP [WBK19] in computing the overall privacy parameter

[4] In this framework we assume that the sampling and shuffling is done by a secure mechanism which is separated from the server, *i.e.,* the server does not know which clients are participating.



(a) Approx. DP as a function of T for $\varepsilon_0 = 2$, $\gamma = 0.001$, $n = 10^6$

(b) Approx. DP as a function of T for $\varepsilon_0 = 1$, $\gamma = 0.001$, $n = 10^7$

(c) Approx. DP as a function of n for $\varepsilon_0 = 2$, $\gamma n = 10^3$, $T = 10^5$

Figure 5.6: Comparison of several bounds on the Approximate (ε, δ)-DP for composition of a sequence of subsampled shuffle mechanisms for $\delta = 10^{-8}$.

ε for number of iterations $T = 10^5$, subsampling parameter $\gamma = 0.001$, LDP parameter $\varepsilon_0 = 2$, and number of clients $n = 10^6$. We observe in Figure 5.4b that the bound given in [FMT22] with the strong composition theorem [KOV15] behaves better than the bound on the RDP [GDD21e] with subsampled RDP bound [WBK19] when the number of subsampled clients per iteration is equal to $k = \gamma n = 10^4$; however, our bound beats both of them. In Figure 5.6c, we fix the number of subsampled clients per iteration to be $k = \gamma n = 10^3$, and hence, the subsampling parameter γ varies with n.

Distributed private learning: We numerically evaluate the proposed privacy-learning performance on training machine learning models. 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 [EFM20a, PTS20] and described in Table 4.1. This model has $d = 13,170$ parameters and achieves an accuracy of 99% for non-private, uncompressed vanilla SGD. We assume that we have $n = 60,000$ clients, where each client has one sample. At each step of the CLDP-SGD Algorithm, we choose uniformly at random 10,000 clients, where each client clips the ℓ_∞-norm of the gradient with clipping parameter $C = 1/100$ and applies the $\mathcal{R}_\infty$ ε_0-LDP mechanism proposed in Chapter 3 with $\varepsilon_0 = 1.5$. We run the CLDP-SGD Algorithm with $\delta = 10^{-5}$ for 200 epochs, with learning rate $\eta = 0.3$ for

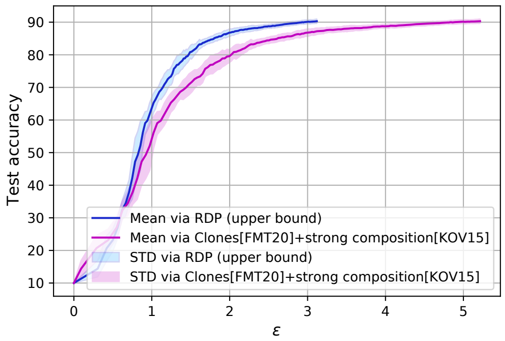


Figure 5.7: Privacy-Utility trade-offs on the MNIST dataset with ℓ_∞-norm clipping.

the first 70 epochs, and then decrease it to 0.18 in the remaining epochs.

Figure 5.7 plots the mean and the standard deviation of privacy-accuracy trade-offs averaged over 10 runs. For our privacy analysis, the total privacy budget is computed by optimizing over RDP order α using our upper bound given in Theorem 5.3.1. For privacy analysis of [FMT22], we first compute the privacy amplification by shuffling numerically given in [FMT22]; then we compute its privacy obtained when amplified via subsampling [Ull17]; and finally we use the strong composition theorem [KOV15] to obtain the central privacy parameter ε. We observe that we achieve an accuracy of 80%(±1.8) with a total privacy budget of $\varepsilon = 1.4$ using our new privacy analysis, whereas, [FMT22] achieves an accuracy of only 70.7%(±2.1) with the same privacy budget of $\varepsilon = 1.4$ using the standard composition theorems. Furthermore, we can see that we achieves accuracy 90%(±0.5) with total privacy budget $\varepsilon = 2.91$ using our new privacy analysis, whereas, [FMT22] (together with the standard strong composition theorem) achieves the same accuracy with a total privacy budget of $\varepsilon = 4.82$.

CHAPTER 6

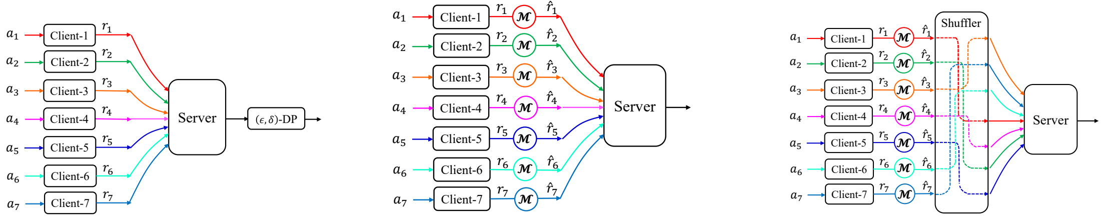
Differentially Private Stochastic Linear Bandits

In this chapter, we study stochastic linear bandits under privacy constraints. Stochastic linear bandits offer a sequential decision framework where a learner interacts with an environment over rounds, and decides what is the optimal (from a potentially infinite set) action to play so as to achieve the best possible reward. In particular, at each round, the learner may take into account all past rewards and actions to decide the next action to play, and in return receive a new reward. This model has been widely adopted both in theory but also in a number of applications, including recommendation systems, health, online education, and resource allocation [MGP15, BRC17, RYW18, BR19]. Motivated by the fact that many of these applications are privacy-sensitive, we explore what is the performance in terms of regret we can achieve, if we are constrained to use a privacy-preserving stochastic linear bandit algorithm.

6.1 Introduction

We aim to design algorithms that preserve the privacy of the rewards, from an adversary that can observe all actions that the learner plays. For example, the central learner may make restaurant recommendations to mobile devices, may regulate the operation of on-body sensors in senior living communities, may decide what educational exercises to provide to students, or what jobs to allocate to workers. The actions the clients play - what restaurant is visited, which sensor is activated, what is the exercise solved, what is the job performed - may be naturally visible especially in public environments. What we care to protect are the rewards, that may capture private information, such as personal preferences in recommendation systems, health indices in online health, performance in online education, and income gained in resource allocation. Our goal is to design algorithms that preserve the privacy of the rewards, while still (almost) achieve the same regret as the traditional algorithms that do not take privacy into consideration.

We do so for three different setups, depicted in Figure 6.1. In the **central DP model**, the learner is a trusted server. The adversary observes the decisions of the trusted server. The server employs a DP mechanism on aggregates of the reward realizations she collects, to ensure that the actions do not reveal information on the rewards. We design an algorithm that guarantees ε-DP and achieves regret that matches existing lower bounds. In particular, over T rounds, it achieves regret $R_T = O\left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon}\right)$ w.h.p., which is optimal within a $\log T$ factor: a lower bound of $O(\sqrt{T})$ is proven in [RT10] for non-private linear bandits, while a lower bound of $O(\frac{\log T}{\varepsilon})$ is shown in [SS18] for ε-DP linear bandits. Note that for $\varepsilon \approx 1$ (perhaps the most common case), the dominant term $O(\sqrt{T \log T})$ matches the regret of the best known algorithms for the non-private case (eg., LinUCB [APS11, RT10]), and hence, we get privacy for free. In the **local DP model**, the learner is an untrusted server, where the adversary (including the learner) can access the individual private rewards of the clients. The clients provide privatized rewards to the server, who then uses this noisy input to decide her next actions. We design an algorithm that guarantees ε_0-LDP and achieves regret $R_T = \mathcal{O}\left(\sqrt{T \log(T)}/\varepsilon_0\right)$ w.h.p. In the **shuffled model**, the learner is still an untrusted server, but now a trusted node, that can act as a relay in the communication between the clients and the server, serves as a shuffler, and can randomly permute the privatized rewards before making them available to the server. A shuffler offers a privacy-amplification mechanism that has recently become popular in the literature, as it is easy to implement, and may enable better privacy-regret performance [CSU19, EFM19, BBG19d, FMT22, GDD21e].



(a) Central model.

(b) Local model.

(c) Shuffled model.

Figure 6.1: DP stochastic linear bandits: (a) Central DP model, (b) Local DP model, and (c) Shuffled DP model.

We leverage the help of a trusted shuffler to ensure both that the output of each client satisfies ε_0-LDP and that the output of the secure shuffler satisfies ε-DP requirements. Our algorithm achieves regret $R_T = \mathcal{O}\left(\sqrt{T\log(T)} + \frac{\log(T)}{\varepsilon}\right)$ w.h.p. that matches the regret of the best non-private algorithms, same as the central model. Furthermore, our algorithm outperforms the best known algorithm for private (contextual) linear bandits in [GCP22, CZ22] that use shuffling. Our results are summarized in Table 6.1, where we also provide known results in the literature.

The rest of the chapter is organized as follows. We present the problem formulation in Section 6.2. We design and analyze privacy-preserving linear bandit algorithms and analyze their privacy-regret tradeoffs for the central model in Section 6.3, for the local model in Section 6.4 and for the shuffled model in Section 6.5. We provide numerical results in Section 6.6. Some proofs are deferred to Appendix E

6.2 Problem Formulation

Stochastic Linear Bandits: In stochastic linear bandits a learner interacts with clients over T rounds by taking a sequence of decisions and receiving rewards. In particular, at each round $t \in [T]$, the learner plays an action a_t from a set $\mathcal{A} \subset \mathbb{R}^d$ and receives a reward $r_t \in \mathbb{R}$. The reward r_t is a noisy linear function of the action, i.e., $r_t = \langle \theta_*, a_t \rangle + \eta_t$, where $\langle . \rangle$ denotes

Algorithm	Regret Bound	Context	Privacy Model	
			Central DP	Local DP
Central DP [SS18]	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon}\right)$	Adversarial	(ε, δ)	N/A
LDP [ZCH20]	$\tilde{\mathcal{O}}\left(\frac{T^{3/4}}{\varepsilon_0}\right)$	Adversarial	$(\varepsilon = \varepsilon_0, \delta)$	(ε_0, δ)
LDP+shuffling [GCP22]	$\tilde{\mathcal{O}}\left(\frac{T^{2/3}}{\varepsilon^{1/3}}\right)$	Adversarial	(ε, δ)	$\big(\varepsilon_0 = \varepsilon^{2/3}T^{1/6}, \delta\big)$
LDP [HLW21]	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon_0}\right)$	Stochastic	$(\varepsilon = \varepsilon_0, \delta)$	(ε_0, δ)
Central DP (Theorem 6.3.1)	$\tilde{\mathcal{O}}\left(\sqrt{T} + \frac{1}{\varepsilon}\right)$	Free	$(\varepsilon, 0)$	N/A
LDP (Theorem 6.4.1)	$\tilde{\mathcal{O}}\left(\frac{\sqrt{T}}{\varepsilon_0}\right)$	Free	$(\varepsilon = \varepsilon_0, 0)$	$(\varepsilon_0, 0)$
LDP+shuffling(Theorem 6.5.1)	$\tilde{\mathcal{O}}\left(\sqrt{T} + \frac{1}{\varepsilon}\right)$	Free	(ε, δ)	$\big(\varepsilon_0 = \varepsilon T^{1/4}, 0\big)$

Table 6.1: Upper part: known results. Lower part: our results. The $\tilde{\mathcal{O}}$ notation hides the dependencies on the dimension d, privacy parameter δ and log factors.

inner product, η_t is an independent zero-mean noise and $\theta_* \in \mathbb{R}^d$ is an unknown parameter vector. The goal of the learner is to minimize the total regret over the T rounds, which is calculated as:

$$R_T = T \max_{a \in \mathcal{A}} \langle \theta_*, a \rangle - \sum_{t=1}^{T} \langle \theta_*, a_t \rangle. \tag{6.1}$$

The regret captures the difference between the reward for the optimal action and the rewards for the actions chosen by the learner. The basic approach in all algorithms is to play actions that enable the learner to learn θ_* well enough to identify a (near) optimal action. The best known algorithms (for example, LinUCB [APS11, RT10]) achieve a regret of order $O(\sqrt{T \log T})$, which is the best we can hope for (matches existing lower bounds [RT10]).

Contextual Linear Bandits: In contextual bandits, the learner observes the context of the client at time t, c_t, plays an action $a_t \in \mathcal{A}$, and receives a reward $r_t = \langle \theta_\star, \phi(a_t, c_t) \rangle + \eta_t$, where ϕ is a known feature map and η_t is noise. In this case the regret R_T is defined as $R_T = \sum_{t=1}^{T} \max_{a \in \mathcal{A}} \langle \theta_*, \phi(a, c_t) \rangle - \langle \theta_*, \phi(a_t, c_t) \rangle$. Equivalently, contextual linear bandits can be seen as linear bandits with action set that changes over time $\mathcal{A}_t = \{\phi(a, c_t) | a \in \mathcal{A}\}$.

We make the following standard assumptions (see, e.g., [APS11, SS18]).

Assumption 6.2.1. We consider stochastic linear bandits with:

1. Sub-gaussian noise: $\mathbb{E}[\eta_{t+1}|\mathcal{F}_t] = 0$ and $\mathbb{E}[\exp(\lambda\eta_{t+1})|\mathcal{F}_t] \leq \exp(\frac{\lambda^2}{2}) \forall \lambda \in \mathbb{R}$, where $\mathcal{F}_t = \sigma(a_1, r_1, ..., a_t, r_t)$ is the σ-field summarizing the information available before round t.
2. Bounded actions, unknown parameter, and rewards: $\|a\|_2 \leq 1 \ \forall a \in \mathcal{A}$, $\|\theta_*\|_2 \leq 1$ and $|r_t| \leq 1$.

Privacy Goal and Measures: Our goal is to achieve the minimum possible regret in (6.1) while preserving privacy of the rewards $\{r_t\}_{t \in [T]}$. To measure privacy, we use the popular central and local differential privacy definitions that we provide for completeness next. For simplicity, we assume that a different client plays each action (e.g., visits a recommended restaurant).

Differential Privacy (DP). We say that two sequences of rewards $\mathcal{R} = (r_1, \ldots, r_T)$ and $\mathcal{R}' = (r'_1, \ldots, r'_T)$ are neighboring if they differ in a single reward, i.e., there is a round $t \in [T]$ such that $r_t \neq r'_t$, but $r_j = r'_j$ for all $j \neq t$. To preserve privacy, we use a randomized mechanism $\mathcal{M}$ designed for stochastic linear bandits, that observes rewards and outputs publicly observable actions.

Definition 6.2.1. (Central DP [DMN06, DR14]): A randomized mechanism $\mathcal{M}$ for stochastic linear bandits is said to be (ε, δ) Differentially Private ((ε, δ)-DP) if for any two neighboring sequences of rewards $\mathcal{R} = (r_1, \ldots, r_T)$ and $\mathcal{R}' = (r'_1, \ldots, r'_T)$, and any subset of output actions $\mathcal{O} \subset \mathcal{A}^T$, $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(\mathcal{R}) \in \mathcal{O}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{R}') \in \mathcal{O}] + \delta. \tag{6.2}$$

When $\delta = 0$, we say that the mechanism $\mathcal{M}$ is pure differentially private (ε-DP). The DP mechanisms maintain that the distribution on the output of the mechanism does not significantly change when replacing a single client with reward r_t with another client with reward r'_t. Thus, the adversary observing the output of the DP mechanism does not infer the clients rewards.

Local Differential Privacy (LDP). If the central learner is untrusted, we need a local private mechanism $\mathcal{M}$ whose output is all the information available to the central learner. We denote the range of the output of the local mechanism by $\mathcal{Z}$.

Definition 6.2.2. (LDP [KLN11]) A randomized mechanism $\mathcal{M} : [-1, 1] \to \mathcal{Z}$ is said to be $(\varepsilon_0, \delta_0)$ Local Differentially Private ($(\varepsilon_0, \delta_0)$-LDP) if for any rewards r_t and r'_t, and any subset of outputs $\mathcal{O} \subset \mathcal{Z}$, the algorithm $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(r_t) \in \mathcal{O}] \leq e^{\varepsilon_0} \Pr[\mathcal{M}(r'_t) \in \mathcal{O}] + \delta_0. \tag{6.3}$$

Similar to the DP definition, we say that $\mathcal{M}$ is pure locally differentially private (ε_0-LDP) when $\delta_0 = 0$. Observe that the input of the LDP mechanism is a single reward, and hence, each client preserves privacy of her observed reward r_t, even if the adversary knows what is the action she plays and observes a function of her reward.

In contextual linear bandits, the context c_t and the reward r_t are considered sensitive information about the client. Hence, the goal of private contextual bandits is to keep both the context and the reward private. Unfortunately, a linear regret bound is unavoidable in contextual bandits under DP constraints [SS18]. Therefore, Shariff et al. in [SS18] have presented the notion of joint differential privacy (JDP) for contextual bandits. For any two sequences $\mathcal{S} = \{(\mathcal{A}_1, r_1), (\mathcal{A}_2, r_2), \ldots, (\mathcal{A}_T, r_T)\}$ and $\mathcal{S}' = \{(\mathcal{A}'_1, r'_1), (\mathcal{A}'_2, r'_2), \ldots, (\mathcal{A}'_T, r'_T)\}$, we say that $\mathcal{S}$ and $\mathcal{S}'$ are t-neighbors if it holds that $(\mathcal{A}_j, r_j) = (\mathcal{A}'_j, r'_j)$ for all $j \neq t$.

Definition 6.2.3. (JDP [SS18]) A randomized algorithm $\mathcal{M}$ for the contextual bandit problem is (ε, δ)-jointly differentially private (JDP) under continual observation if for any t

and any t-neighboring sequences $\mathcal{S}$ and $\mathcal{S}'$, and any subset $\mathcal{S}_{>t} \subset \mathcal{A}_{t+1} \times \cdots \times \mathcal{A}_T$, it holds that:

$$\Pr[\mathcal{M}(\mathcal{S}) \in \mathcal{S}_{>t}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{S}') \in \mathcal{S}_{>t}] + \delta. \tag{6.4}$$

Thus, changing the pair (c_t, r_t) of a single client cannot have a significant impact on determining future actions.

System Model: We consider three different models for private stochastic linear bandits. In all three cases, our setup is that of a learner, who asks clients to play publicly observable actions, and collects the resulting rewards (see Figure 6.1). The models differ on whether the learner is a trusted or untrusted server, and whether a shuffler is available or not. A shuffler simply performs a random permutation on its input.

1) Central DP model: The learner is a **trusted server** who can collect the clients' rewards and take actions. Thus, the trusted server can apply a DP mechanism (see Definition 6.2.1) to preserve the privacy of the collected rewards against any adversary observing the actions of the clients.

2) LDP model: The learner is an **untrusted server**. Hence, each client needs to privatize her own reward by applying an LDP mechanism (see Definition 6.2.2) before sending it to the untrusted server. The server takes decisions on next actions using the collected privatized rewards.

3) Shuffled model: Similar to the LDP model, the learner is an **untrusted server**. However, we consider that there exists a **trusted shuffler** that collects the LDP responses of the clients and randomly permutes them before passing them to the server, see Figure 6.1.

6.3 Stochastic Linear Bandits with central DP

In this section we consider the case where the learner is a trusted server. We present an algorithm that offers ε-DP (see Definition 6.2.1) for stochastic linear bandits, with no regret penalty: we achieve the same order regret performance as the best algorithms that operate under no privacy considerations.

Algorithm 6.3.1 ε-DP algorithm for stochastic linear bandits: central model

```
Input: set of actions $\mathcal{A}$, time horizon $T$, and privacy parameter $\varepsilon$.
Let $\mathcal{A}_1$ be a $\zeta$-net for $\mathcal{A}$ as in Lemma 6.3.1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ do
    $\gamma_i \leftarrow \sqrt{\frac{4d}{q^i} \log\left(4|\mathcal{A}_i|T^2\right)} + \frac{2Bd^2 + 2d\log\left(4|\mathcal{A}_i|T^2\right)}{\varepsilon q^i}$.
    For $\mathcal{A}_i \subseteq \mathbf{R}^m$, $m \leq d$, let $\mathcal{C}_i$ be a core set of size at most $Bm$ as in Lemma 6.3.2 and $\pi_i$ the associated distribution.
    Pull each action $a \in \mathcal{C}_i$, $n_{ia} = \lceil \pi_i(a) q^i \rceil$ times to get rewards $r_{ia}^{(1)}, ..., r_{ia}^{(n_{ia})}$.
    $\bar{r}_{ia} \leftarrow \sum_{k=1}^{n_{ia}} r_{ia}^{(k)}$, $\hat{r}_{ia} \leftarrow \bar{r}_{ia} + z_{ia}$ $\forall a \in \mathcal{C}_i$, where $z_{ia}$ is an independent noise that follows $\mathsf{Lap}(\frac{1}{\varepsilon})$.
    $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$, $\hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
    $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.
```

Main Idea: Our algorithm follows the structure of elimination algorithms: it runs in batches, where we maintain a "good set of actions" $\mathcal{A}_i$, in each batch i that almost surely contain the optimal one, and gradually eliminate sub-optimal actions, shrinking the sets $\mathcal{A}_i$ as i increases. As is fairly standard in elimination algorithms, in our case as well, during batch i, the learner plays actions in $\mathcal{A}_i$, calculates an updated estimate $\hat{\theta}_i$ of the unknown parameter vector θ_*, and eliminates from $\mathcal{A}_i$ actions if their estimated reward is $2\gamma_i$ from the estimated reward of the arm that appears to be best, where γ_i is the confidence of the reward estimates.

We note that our adversary observes actions generated through the estimate of θ_i. Since, the $\hat{\theta}_i$ is generated from the private rewards, all functions of $\hat{\theta}_i$ (including estimate of next actions) is ε-DP from post-processing [DR14]. Our new observation on how to achieve this is as follows. **If by playing a smaller number of distinct actions we are able to identify the optimal action, we need to overall add a smaller amount of noise to guarantee privacy than if we play a larger number of distinct actions.** Indeed, if an action a is played for n_a times, the learner, to estimate θ_*, only needs to use the sum of these n_a rewards. To offer ε-DP we can perturb this sum by adding independent Laplacian noise ($\text{Lap}(\frac{1}{\varepsilon})$); clearly, the smaller the number of distinct actions we play, the smaller the overall amount of noise we need to add. Thus our algorithm, at each batch iteration i, plays actions from a carefully selected subset of $\mathcal{A}_i$, of cardinality as small as possible. The technical question we address is, starting from a continuous action space $\mathcal{A}$, how to select at each batch iteration a small cardinality subset that maintains the ability to identify the optimal action.

We next describe the steps in implementing this idea. Recall that our actions come from a set $\mathcal{A} \subseteq \mathbb{R}^d$, and we assume they are bounded, namely, $\|a\|_2 \leq 1,\ \forall a \in \mathcal{A}$ (see Assumptions 6.2.1 in Section 6.2).

1. Our first step is to **reduce the continuous action space to a discrete action space problem**. To do so, we finely discretize $\mathcal{A}$ to create what we call a ζ-net, a discrete set of actions $\mathcal{N}_\zeta \subseteq \mathcal{A}$ such that distances are approximately preserved. Namely, for any $a \in \mathcal{A}$, there is some $a' \in \mathcal{N}_\zeta$ with $\|a' - a\|_2 \leq \zeta$. Lemma 6.3.1, proved in [Ver18, Cor. 4.2.13], states that we can always find such a discrete set with cardinality at most $(\frac{3}{\zeta})^d + d$. As a result, all the "good sets" $\mathcal{A}_i$ will also be discrete.

Lemma 6.3.1. *(ζ-net for $\mathcal{A}$ [Ver18]) For any set $\mathcal{A} \subseteq \{x \in \mathbb{R}^d | \|x\|_2 \leq 1\}$ that spans $\mathbb{R}^d$, there is a set $\mathcal{N}_\zeta \subseteq \mathcal{A}$ (zeta-net) with cardinality at most $(\frac{3}{\zeta})^d + d$ such that $\mathcal{N}_\zeta$ spans $\mathbb{R}^d$, and for any $a \in \mathcal{A}$, there is some $a' \in \mathcal{N}_\zeta$ with $\|a' - a\|_2 \leq \zeta$.*

2. We introduce the use of a **core set** $\mathcal{C}_i$, a subset of the actions of the set of "good actions" $\mathcal{A}_i$. During batch i, **the learner only plays actions in $\mathcal{C}_i$, each with some probability** $\pi_i(a)$. Lemma 6.3.2, proved in [LS20, Ch.21], states that if $\mathcal{A}_i$ spans some space $\mathbf{R}^k$, we can find a core set of size at most Bk (with B a constant) and an associated probability distribution π, so that, playing actions only from C_i enables to calculate a good estimate of $\langle a, \theta_* \rangle$ for each $a \in \mathcal{A}_i$.

Lemma 6.3.2. *(Core set for $\mathcal{A}$ [LS20]) For any finite set of actions $\mathcal{A} \subset \{x \in \mathbf{R}^d | \|x\|_2 \leq 1\}$ that spans $\mathbb{R}^d$, there is a constant B, a subset $\mathcal{C}$ and a distribution π on $\mathcal{C}$, that can be computed in polynomial time, such that $|\mathcal{C}| \leq Bd$, $\mathcal{C}$ spans $\mathbb{R}^d$, and for any $a \in \mathcal{A}$*

$$a^\top \left(\sum_{\alpha \in \mathcal{C}} \pi(\alpha) \alpha \alpha^\top \right)^{-1} a \leq 2d. \tag{6.5}$$

3. To preserve the privacy of rewards, we **perturb the sum rewards of each action by adding Laplace noise**. Adding noise affects the confidence of the reward estimates γ (step 5 in Algorithm 6.3.1 shows that γ increases as ε decreases), and thus delays the elimination of bad actions and increases the regret by an additive term of $\tilde{O}(\frac{1}{\varepsilon})$. Replacing a possibly large set $\mathcal{A}_i$ with the smaller core set $\mathcal{C}_i$ effectively decreases the cumulative noise affecting the estimate of $\theta_\star$. The computation of $\mathcal{C}, \pi$ can be formulated as a convex optimization problem with many efficient approximation algorithms available [FW56, LS20].

Algorithm Pseudo-Code: Algorithm 6.3.1, starts by initializing the good action set $\mathcal{A}_1$ to be an $\frac{1}{T}$-net of $\mathcal{A}$ according to Lemma 6.3.1. Then, the algorithm operates in batches that grow exponentially in length, where the length of batch i is approximately q^i and $q = (2T)^{1/\log T}$[1]. In each batch i, we construct the core set C_i and the associated distribution π_i as per Lemma 6.3.2. Each action in $\mathcal{C}_i$ is pulled $n_{ia} = \lceil \pi(a) q^i \rceil$ times, where the length of batch i is $n_i = \sum_{a \in \mathcal{C}_i} n_{ia}$. To preserve privacy, the sum of the rewards of each action is perturbed with Lap$(1/\varepsilon)$ noise. The learner uses these privatized sum rewards to compute the estimate of θ_*, $\hat{\theta}_i$. At the end of batch i, the learner eliminates from $\mathcal{A}_i$ the actions with estimated mean reward, $\langle a, \hat{\theta}_i \rangle$, that fail to be within $2\gamma_i$ from the action that appears to be best, where γ_i is our confidence in the mean estimates. After the iteration $i = \log T - 1$ is completed, the learner simply plays the action that appears to be best.

[1] We note that $e \leq q \leq e^2$.

Algorithm 6.3.2 ε_0-LDP algorithm for stochastic linear bandits: local model

```
Input: set of actions $\mathcal{A}$, time horizon $T$, and privacy parameter $\varepsilon_0$.
Let $\mathcal{A}_1$ be a $\zeta$-net for $\mathcal{A}$ as in Lemma 6.3.1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ do
    Client side:
        Receive action $a$ from the server. Play action $a$ and receive a reward $r$.
        Send $\hat{r} = r + \mathsf{Lap}(\frac{1}{\varepsilon_0})$.
    Server side:
        Let $\mathcal{C}_i$ be a core set for $\mathcal{A}_i$ as in Lemma 6.3.2 with distribution $\pi_i$, and $n_{ia} = \lceil \pi_i(a) q^i \rceil$.
        Send each action $a \in \mathcal{C}_i$ to a set of $n_{ia}$ clients to get rewards $\hat{r}_{ia}^{(1)}, ..., \hat{r}_{ia}^{(n_{ia})}$.
        $n_i \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia}$.
        $\gamma_i \leftarrow \sqrt{\log(4|\mathcal{A}_i|T^2)}(\sqrt{\frac{4d}{q^i}} + \frac{2d\sqrt{n_i}}{q^i \varepsilon_0})$.
        $\hat{r}_{ia} \leftarrow \sum_{k=1}^{n_j} \hat{r}_{ia}^{(1)} \ \forall a \in \mathcal{C}_i$.
        $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$, $\hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
        $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}_\rangle} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$.
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.
```

Algorithm Performance: We next prove that Algorithm 6.3.1 is ε-DP and provide a bound on its regret.

Theorem 6.3.1. *Algorithm 6.3.1 is ε-differentially private. Moreover, it achieves a regret*

$$R_T \leq C \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon} \right), \tag{6.6}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε, T.

Proof Outline. The privacy result follows from the Laplace mechanism [DR14]. To bound the regret, we first argue that with probability at least $1 - \frac{1}{T}$, and for all i and all $a \in \mathcal{A}_i$, we have that $|\langle a, \hat{\theta}_i \rangle - \langle a, \theta_\star \rangle| \leq \gamma_i$. Conditioned on this event, an action with gap

Δ_a is eliminated when, or before, $\gamma_i < \Delta_a/2$. Hence, all actions in batch i have a gap that is at most $4\gamma_i$. The regret bound follows by summing $4\gamma_i n_i$ for all batches. The complete proof is provided in Appendix E.1. $\square$

Remark 6.3.1. We note that the high probability bound in Theorem 6.3.1 implies a bound in expectation

$$\mathbb{E}[R_T] \leq C\left(\sqrt{T\log T} + \frac{\log^2 T}{\varepsilon}\right). \tag{6.7}$$

The regret is trivially $O(T)$ and the failure probability is $\frac{1}{T}$, which overall contributes $O(1)$ to $\mathbb{E}[R_T]$.

Remark 6.3.2. The regret in Theorem 6.3.1 is optimal up to $\log T$ factor; a lower bound of $O(\sqrt{T})$ is proven in [RT10] for the non-private case, while a lower bound of $\frac{\log T}{\varepsilon}$ is shown in [SS18] for private case.

Remark 6.3.3. We observe that the privacy parameter ε is typically ≈ 1. In this case, the dominating term in (6.6) is $O(\sqrt{T\log T})$ which matches the regret of the best-known algorithm for the non-private case (see LinUCB in [RT10, APS11]), and hence, we get privacy for free.

6.3.1 Stochastic Contextual Bandits with Central DP

In this section, we extend our results to the contextual linear bandits with known context distribution. In the following, we focus on the stochastic context setting where the context c_t is generated from a distribution $\mathcal{P}$ independently from other iterations. We assume that the distribution $\mathcal{P}$ is known to the learner[2]. The main idea is to use the reduction proposed in [HYF23] to represent the contextual linear bandits with known context distribution as a stochastic linear bandits problem, and then, we apply our DP algorithm for stochastic linear bandits.

[2] The knowledge of the distribution $\mathcal{P}$ can be practical in multiple cases, e.g., known age, and gender distribution. The extension to unknown context distribution is a future direction of our work.

First, we briefly review the reduction for the case of known context distribution and refer the reader to [HYF23] for a detailed description. The basic idea in [HYF23] is to establish a linear bandit action for each possible parameter vector θ of the contextual bandit instance.

This is achieved through the use of the function $g : \mathbb{R}^d \to \mathbb{R}^d$, which computes the expected best action under the context distribution $\mathcal{P}$ with respect to the parameter θ: $g(\theta) = \mathbb{E}_{c_t \sim \mathcal{P}}[\arg\max_{a \in \mathcal{A}} \langle \phi(a, c_t), \theta \rangle]$. As stated in [HYF23, Theorem 1], when $a_t = \arg\max_{a \in \mathcal{A}} \langle \phi(a, c_t), \theta_t \rangle$ for some $\theta_t \in \mathbb{R}^d$, then the reward generated by the contextual bandit instance can be expressed as $r_t = \langle g(\theta_t), \theta_\star \rangle + \eta_t'$, where η_t' is noise with zero mean conditioned on the history. Consequently, the reward can be viewed as generated by pulling action $g(\theta_t)$ in a linear bandit instance with an action set $\mathcal{X} = \{g(\theta) | \theta \in \Theta\}$. Moreover, the same theorem demonstrates that if a linear bandit algorithm is employed to choose $g(\theta_t) \in \mathcal{X}$ at round t and thus play action $a_t = \arg\max_{a \in \mathcal{A}} \langle \phi(a, c_t), \theta_t \rangle$, then $|R_T - R_T^L| = \tilde{O}(\sqrt{T})$ with high probability, where $R_T^L = \sum_{t=1}^T \sup_{\theta \in \Theta} \langle g(\theta) - g(\theta_t), \theta_\star \rangle$ is the regret of the algorithm on the linear bandit instance.

As a result, if the context distribution is known, then the function g is known to the learner as well as the users. Thus, we can construct a contextual bandits algorithm under joint differential privacy (JDP) constraints to privatize the contexts and rewards using our Algorithm 6.3.1 as follows. We apply our Algorithm 6.3.1 with action set $\mathcal{A} \triangleq \mathcal{X} \triangleq \{g(\theta) : \theta \in \Theta\}$. When a client receives an action $x_t \triangleq g(\theta_t)$ (from linear bandits), the client chooses an actual action a_t by solving $a_t = \arg\max_{a \in \mathcal{A}} \langle \phi(a, c_t), \theta_t \rangle$, where $\theta_t = g^{-1}(x_t)$ with ties broken arbitrarily. The client observes a reward r_t and sends it to the learner. Following Algorithm 6.3.1, at the end of the batch, the learner privatizes the aggregated rewards and updates the action set $\mathcal{X}_{i+1}$ to the next batch, see Steps $8 - 10$ in Algorithm 6.3.1.

Corollary 6.3.1. There exists an $(\varepsilon, 0)$-JDP algorithm for stochastic contextual bandits with know context distribution with bounded regret:

$$R_T \leq C \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon} \right), \tag{6.8}$$

with probability at least $1 - \frac{2}{T}$, where C is a constant that does not depend on ε, T.

Proof. The results are obtained by applying the algorithm explained above which is a combination of the reduction from [HYF23] and our Algorithm 6.3.1. Observe that at any iteration $t \in [T]$, all the past history of context-reward pairs $\{(c_{t'}, r_{t'}) : t' < t\}$ are encoded in the returned reward set $\{r_{t'} : t' < t\}$. Furthermore, the past sequence rewards are $(\varepsilon, 0)$-DP from Theorem 6.3.1, where the learner uses only these private rewards to estimate the unknown parameter $\theta_\star$ and decides the new action of the next iteration. Thus, the presented algorithm is $(\varepsilon, 0)$-JDP.

The regret of our algorithm of stochastic linear bandits is bounded by $C' \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon}\right)$ from Theorem 6.3.1 with probability at least $1 - \frac{1}{T}$. Furthermore, from [HYF23, Theorem 1], the difference between the regrets of the linear and contextual bandits instances $|R_T - R_T^L| = \tilde{O}(\sqrt{T})$ with probability at least $1 - 1/T$. By the triangle inequality and the union bound, it follows that the regret of the algorithm is bounded by $C \left(\sqrt{T \log T} + \frac{\log^2 T}{\varepsilon}\right)$ with probability at least $1 - 2/T$. This completes the proof of Corollary 6.3.1. ■

Remark 6.3.4. In this section, we showed that our Algorithm 6.3.1 for DP stochastic linear bandits can be extended to give a JDP algorithm for contextual bandits with known distribution. A similar argument can be applied to the local DP model and the shuffled model in the next sections.

6.4 Stochastic Linear Bandits with LDP

In this section, the learner is an untrusted server, and thus we design a linear bandit algorithm (Algorithm 6.3.2) that operates under LDP constraints.

Main Idea: As in Algorithm 6.3.1, we here also utilize a core set of actions; the difference is that, since the server is untrusted, each client privatizes her own reward before providing it to the server. Our algorithm offers an alternative approach to [HLW21] that achieves

Algorithm 6.4.1 DP algorithm for stochastic linear bandits: shuffled model

```
Input: actions $\mathcal{A}$, horizon $T$, privacy parameters $(\varepsilon, \delta)$.
Let $\mathcal{A}_1$ be a $\zeta$-net for $\mathcal{A}$ as in Lemma 6.3.1, with $\zeta = \frac{1}{T}$.
$q \leftarrow (2T)^{1/\log T}$.
for $i = 1 : \log(T) - 1$ do
    Client side:
        Receive action $a$ and the value $n_i$ from shuffler.
        Play action $a$ and receive a reward $r$.
        $\varepsilon_0^{(i)} \leftarrow f_{n_i,\delta}^{-1}(\varepsilon)$.
        Send $\hat{r} = r + \mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$ to the shuffler.
    Shuffler:
        Send action $a_{\pi(j)}$ and $n_i$ to client $j,\ j = [n_i]$, where $\pi$ is a random permutation of $[n_i]$.
        Receive the action-reward pairs $\{(a_j, \hat{r}_{ia_j})\}_{j=1}^{n_i}$, and send them to the server.
    Server side:
        Let $\mathcal{C}_i$ be a core set for $\mathcal{A}_i$ as in Lemma 6.3.2 with distribution $\pi_i$.
        Let $n_{ia} = \lceil \pi_i(a) q^i \rceil, n_i \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia}$.
        Let $\mathcal{A}_{\mathcal{C}_i} = \cup_{a \in \mathcal{C}_i} \{a\}_{l=1}^{n_{ia}}$ be a set of $n_i$ actions where action $a \in \mathcal{C}_i$ is repeated $n_{ia}$ times.
        Let $a_1, ..., a_{n_i}$ be an enumeration of $\mathcal{A}_{\mathcal{C}_i}$. Send them to the shuffler
        Receive the action-reward pairs from the shuffler.
        $\gamma_i \leftarrow \sqrt{\log(4|\mathcal{A}_i|T^2)}(\sqrt{\frac{4d}{q^i}} + \frac{2d\sqrt{n_i}}{q^i \varepsilon_0^{(i)}})$.
        $\hat{r}_{ia} \leftarrow \sum_{k=1}^{n_j} \hat{r}_{ia}^{(1)}\ \forall a \in \mathcal{C}_i$.
        $V \leftarrow \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top,\ \hat{\theta}_i \leftarrow V^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$.
        $\mathcal{A}_{i+1} \leftarrow \{a \in \mathcal{A}_i | \langle a, \hat{\theta}_i \rangle \geq \max_{\alpha \in \mathcal{A}} \langle \alpha, \hat{\theta}_i \rangle - 2\gamma_i\}$.
Play action $\arg\max_{\alpha \in \mathcal{A}_{\log(T)-1}} \langle \alpha, \hat{\theta}_{\log(T)-1} \rangle$ for the remaining time.
```

the same regret, while using operation in batches, which may in some applications be more implementation-friendly (e.g., multi-stage clinical trials and online marketing with high response rates) [PRC16, EKM21], and also forms a foundation for the Algorithm 6.4.1 we discuss in the next section.

Algorithm Pseudocode: Algorithm 6.3.2 operates like Algorithm 6.3.1, except for the addition of $\mathsf{Lap}(1/\varepsilon_0)$ noise for each reward individually as opposed to adding $\mathsf{Lap}(1/\varepsilon)$ to the sum of the rewards of each arm in the central model. The value of γ_i is adjusted to account for this change. **Algorithm Performance.** The following Theorem 6.4.1 presents the privacy-regret tradeoffs of the LDP stochastic bandits Algorithm 6.3.2. The proof is deferred to Appendix E.3 and follows the same main steps as the proof of Theorem 6.3.1, but with the modified values of γ_i.

Theorem 6.4.1. *Algorithm 6.3.2 is ε_0-LDP. Moreover, it achieves a regret*

$$R_T \leq C(1 + \frac{1}{\varepsilon_0}) \left(\sqrt{T \log T}\right), \tag{6.9}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε_0 and T.

Remark 6.4.1. When $\varepsilon_0 > 1$, the regret R_T would be $\mathcal{O}\left(\sqrt{T}\log(T)\right)$ that matches the non-private case. However, the constants of the regret convergence are larger than that of the non-private case.

Remark 6.4.2. (Comparison to the central (ε, δ)-DP model.) Observe that when $\varepsilon_0 < 1$, the dominating term in the regret is $R_T = \mathcal{O}\left(\frac{T\log(T)}{\varepsilon_0}\right)$. In other words, we obtain the regret of the non-private case divided by the LDP parameter ε_0. In contrast, the central DP parameter ε appears as an additive term in the regret of the central model. This difference is because, in the local model noise is added on every reward, while in the central model directly on the reward aggregates; thus the noise variance of the aggregate rewards and the confidence parameter γ_i increases in the local model. In the high privacy regimes; for example, assume that $\varepsilon_0 = \mathcal{O}\left(\frac{1}{T^\alpha}\right)$ for some $0 < \alpha \leq \frac{1}{2}$, we get a regret R_T of order $\mathcal{O}\left(T^{\frac{1}{2}+\alpha}\right)$ that becomes linear function of T as $\varepsilon_0 \to \frac{1}{\sqrt{T}}$.

6.5 Stochastic Linear Bandits in the Shuffled Model

In this section, we consider the case of an untrusted server and a trusted shuffler. We propose Algorithm 6.4.1 that (almost) achieves the same regret as the best non-private algorithms.

Main idea: To use shuffling, we need to use an algorithm that operates over batches of actions, so as to be able to shuffle them. The use of a core set is critical to enable a selection of actions that lead to a good estimate for $\theta\star$. For example, if the original set $\mathcal{A}$ contains a large number of actions along one direction in the space, but only a few actions along other directions, then pulling each action in $\mathcal{A}$ once will not result in a good estimate of $\theta_\star$. Use of the core set and the associated distribution π will balance such assymetries and enable to exploration multiple directions of the space a sufficient number of times to acquire a good estimate of $\theta_\star$. Accordingly, we follow the same approach as in Algorithm 6.3.2 with two changes: we use a shuffler (in a manner tailored to bandits) to realize privacy amplification gains, and we adjust the amount of Laplace noise we add in each batch, depending on the batch size.

We use the trusted shuffler as follows. The actions to be played in the ith batch are shuffled by the trusted shuffler at the beginning of the batch. The shuffler asks clients to play actions in the shuffled order. Then, at the end of the batch, the shuffler reverses the shuffling operation, associates every action with its observed LDP reward, and conveys it to the untrusted learner.[3]

We adjust the amount of added Laplace noise per batch as follows. To offer privacy

[3] The server cannot directly observe which action is played by which client, for instance due to geographical separation.

guarantees, we want to add noise to the rewards so that the output of the shuffler is (ε, δ)-DP for each batch $i \in [\log(T)]$. This implies that the entire algorithm will be (ε, δ)-DP since we assume that each client contributes to only one of the batches. The privacy amplification of the shuffling depends on the size of the batch (see e.g. [FMT22, Theorem 1]); thus the larger the batch size, the less noise needs to be added to the rewards of the clients. To ensure that the output of batch i is (ε, δ)-DP, it is sufficient to add to each reward noise $\mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$, where $\varepsilon_0^{(i)} \leftarrow f_{n_i,\delta}^{-1}(\varepsilon)$, and n_i is the size of batch i. The function $f_{n,\delta} : \mathbb{R}^+ \to \mathbb{R}^+$ captures privacy amplification via shuffling [FMT22] and is defined as follows

$$f_{n,\delta}(\varepsilon_0) = \log\left(1 + \frac{e^{\varepsilon_0} - 1}{e^{\varepsilon_0} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0}\log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_0}}{n}\right)\right). \tag{6.10}$$

Since the noise added to the rewards varies for each batch i, we modify the confidence bounds, γ_i, to reflect this. The pseudo-code is provided in Algorithm 6.4.1.

Algorithm Performance: The following theorem proves that Algorithm 6.4.1 is (ε, δ)-DP and provides an upper bound on its regret that matches the information theoretic lower bound for $\varepsilon = \tilde{O}(\frac{1}{\sqrt{T}})$.

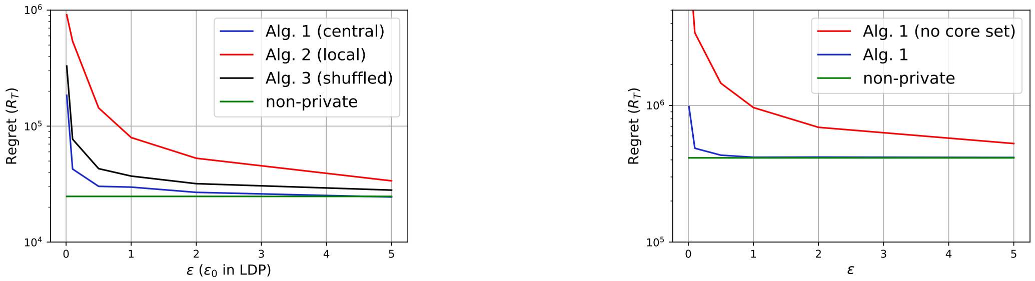
Theorem 6.5.1. *Algorithm 6.4.1 is (ε, δ)-differentially private. Moreover, for $\varepsilon = O(\sqrt{\frac{\log(1/\delta)}{T}})$ it achieves a regret*

$$R_T \leq C\left(\sqrt{T\log T} + \frac{\sqrt{\log(1/\delta)}\log^{3/2} T}{\varepsilon}\right), \tag{6.11}$$

with probability at least $1 - \frac{1}{T}$, where C is a constant that does not depend on ε and T.

Proof Outline: The proof of Theorem 6.5.1 is deferred to Appendix E.4. The privacy guarantee is proved by reducing the scheme to one that shuffles the rewards but does not shuffle the corresponding actions and using results from [FMT22]. The regret analysis follows similar ideas as in Theorem 6.3.1 and Theorem 6.4.1.

Remark 6.5.1. Algorithm 6.4.1 almost achieves the same order regret as the best non-private algorithms. Indeed, Theorem 6.5.1 proves that Algorithm 6.4.1 achieves a regret that matches



(a) Stochastic linear bandits: $K = 10, T = 10^6$. (b) Effect of core set size, $K = 1000, T = 10^7$.

Figure 6.2: Regret-privacy trade-offs for DP stochastic linear bandits algorithms.

the regret of the central DP Algorithm 6.3.1 for the high privacy regimes $\varepsilon = O(\sqrt{\log(1/\delta)/T})$. For the low privacy regime $\varepsilon > 1$, the shuffling does not offer privacy gains, $\varepsilon_0^{(i)} \approx \varepsilon$ for all $i \in [\log(T)]$ and the regret of Algorithm 6.4.1 is similar to the regret of Algorithm 6.3.2 of the local DP model. However, for the low privacy regime the local DP model also achieves the same regret as non-private algorithms up to constant factors (see Remark 6.4.1). Hence in both cases, Algorithm 6.4.1 achieves the same order regret as Algorithm 6.3.1 which almost matches the regret of non-private algorithms.

Remark 6.5.2. Algorithm 6.4.1 improved regret performance over Algorithm 6.3.2 is thanks to the smaller amount of noise added to rewards. In particular, the noise added in Step 9 of Algorithm 6.4.1 has variance $\frac{2}{\varepsilon_0^{(i)2}} \approx \frac{2}{n_i \varepsilon^2}$ for small ε.

6.6 Numerical Results

We here present indicative results on the performance of our proposed DP stochastic linear bandits algorithms.

Data Generation: We generate synthetic data generated as follows. The set of actions $\mathcal{A}$ contains K actions, where each action $a \in \mathcal{A}$ is a $d = 2$-dimensional vector. The actions

$a \in \mathcal{A}$ and the optimal parameter θ_* are generated uniformly at random from the unit sphere $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : ||x||_2 = 1\}$. A similar method is considered in [HLW21]. Figure 6.2 plots the total regret R_T over a horizon $T = 10^6$ as a function of the privacy budget (ε or ε_0 in the case of LDP mechanisms).

Comparison of central DP, local DP, and shuffled DP models: In Figure 6.2, the set of actions $\mathcal{A}$ contains $K = 10$ actions. Figure 6.2 shows that the regret achieved by all three algorithms, Algorithm 6.3.1 (central model), Algorithm 6.3.2 (local model), and Algorithm 6.4.1 (shuffled model) converges to the regret of non-private stochastic linear bandit algorithms [LS20, Ch. 22] as $\varepsilon \to \infty$ ($\varepsilon_0 \to \infty$), albeit at different rates. As predicted from the theoretical analysis, Algorithms 6.3.1 (central) and 6.4.1 (shuffled) offer privacy (almost) for free, closely following the non-private regret. Furthermore, the central Algorithm 6.3.1 is close to the non-private case and significantly outperforms the LDP Algorithm 6.3.2. We observe that the shuffled model has a performance close to the central algorithm and outperforms the regret of the LDP Algorithm 6.3.2.

Usefulness of Core Set: In Figure 6.2b, we explore potential benefits on the performance of Algorithm 6.3.1 that use of the core set can offer. We consider $K = 1000$ and $T = 10^7$, and plot the regret of Algorithm 6.3.1 for two cases: (i) when we use a core set of size 2-3 actions, similar to the dimension of our space (labeled as Alg. 1), and (ii) when no core set is used, and instead the good set of actions of the batched algorithm is the whole action set (labeled as Alg. 1 no-core-set). We find that, as expected from our theoretical analysis, using a core set enables to achieve performance very close to that of a non-private batched algorithm that adds no noise. In contrast, using (and adding noise to) the entire action space significantly degrades the performance.

6.7 Related Work

Differential Privacy (DP) algorithms have been proposed for the generic multi-armed bandits (MAB) problems [SS19, RZL20, TKM21], yet these algorithms would not work well for linear bandits, as linear bandits allow for an infinite set of actions while generic MAB have a regret that increases with the number of actions. Closer to ours is work on DP for contextual linear bandits [SS18, ZCH20, HLW21, GCP22]; indeed, linear bandits can be viewed as (a special case of) contextual linear bandit setup with a single context. The work in [SS18] considers contextual linear bandits with DP and shows that linear regret is unavoidable. Instead, the work considers a weaker notion of privacy, JDP (joint differential privacy), in a centralized setting and propose an algorithm that achieves a regret of $\tilde{O}(\sqrt{T}/\varepsilon)$. This does not match the best known lower bound for the centralized setting of $\Omega(\sqrt{T}+\log(T)/\varepsilon)$ [SS18]. Our work consider the stronger DP notion and achieves the lower bound of $\Omega(\sqrt{T}+\log(T)/\varepsilon)$ up to logarithmic factors for the special case of stochastic linear bandits. Recent work shows that contextual linear bandits can be reduced to stochastic linear bandits if the context distribution is known [HYF22], which is the case for many application [HYF22]. This implies direct generalizations of our algorithms to contextual linear bandits with DP and known context distribution without affecting the regret bounds. The work in [ZCH20] considers contextual linear bandits with LDP, where the contexts can be adversarial. The work proposes an algorithm that achieves a regret of $\tilde{O}(T^{3/4}/\varepsilon_0)$ and conjectures that the regret is optimal up to a logarithmic factor. The authors in [HLW21] consider a special case, where the contexts are generated from a distribution, and propose a method that achieves a regret of $\tilde{O}(\sqrt{T}/\varepsilon_0)$ under certain assumptions on the context distribution. Our algorithm for the local model achieves the same regret order using an alternative method. The works in [GCP22, CZ22] consider contextual linear bandits in the shuffled model where the best-known algorithm achieves a regret of $\tilde{O}(T^{3/5})$. Our proposed algorithms achieve a regret of $\tilde{O}(\sqrt{T}+1/\varepsilon)$, matching the information-theoretic lower bound in [SS18], for stochastic linear bandits in the shuffled model. A summary of the best results for DP contextual linear bandits and our results is presented in Table 6.1.

We mention two works in the literature studying the DP stochastic linear bandits problem, which are close to our work. The work in [LZJ22] proposed DP mechanisms for stochastic linear bandits using a similar approach to the batched algorithm. The main difference between their schemes and our proposed schemes is that the work in [LZJ22] focuses on designing communication-efficient schemes for DP stochastic linear bandits. The work in [HZZ22], which was published concurrently to our work [HGF22], primarily focuses on deriving lower bounds for differentially private contextual bandits in the central DP model, matching our upper bound in the central case and thereby showing the optimality of our scheme. Moreover, our contributions go well beyond the central DP model to include local DP and shuffled DP models as well.

CHAPTER 7

Successive Refinement of Privacy

An underlying assumption in the body of work on differential privacy has long been that an unlimited amount of randomness is available for use by any privacy mechanism. Under this assumption, the vast majority of the literature has focused on achieving better privacy-utility trade-offs – see, for example, [DR14, SC13] for surveys. In this chapter, we ask: how much randomness do we need to achieve a desired level of privacy and utility, and study privacy-utility-randomness trade-offs instead. Answering this question both contributes to our theoretical understanding, and also could support specific emerging applications that we discuss later.

We consider local differential privacy (LDP) that has recently seen use in industrial applications, [EPK14, RAPPOR], [App17]. Here, an untrusted analyst acquires already-privatized pieces of information from a number of users, and aggregates them into a statistic or a machine learning model. Concretely, there are n users who observe i.i.d. inputs $X_1, X_2, \ldots, X_n$ (user i observes X_i) from a finite alphabet $\mathcal{X}$ of size k, where each X_i is distributed according to a probability distribution $\mathbf{p}$. Each user has a certain amount of randomness, measured in Shannon entropy, to randomize her input, that she then publicly shares. Our general setup also includes d analysts who would like to use the users' public outputs to estimate $\mathbf{p}$, each at a different level of privacy $\varepsilon_1, \ldots, \varepsilon_d$, where smaller ε means higher privacy. Each analyst may or may not share some common randomness with the users. We call this general setup *successive refinement of privacy*, in which each user shares a public output with highest privacy level. Then, each analyst uses a shared random key to partially undo the randomization of the public output to get less privacy and higher utility.

This general formulation includes several interesting special cases, for which we study the trade-offs between privacy, utility, and randomness. These are:

(i) There is a single analyst ($d = 1$), who shares no randomness with the users and estimates $\mathbf{p}$ with privacy level ε. This setting directly generalizes the classical setup of LDP to the case of limited randomness.

(ii) There are two analysts ($d = 2$), who observe the same public outputs from the users; the first analyst who shares common randomness with the users has permission to perfectly recover the original inputs (i.e., privacy level $\varepsilon_1 \to \infty$), while the second analyst who shares no randomness with the users estimates $\mathbf{p}$ with privacy level ε_2. This setting is an adaptation of the classical *perfect secrecy* setup of Shannon [Sha49] to the differential privacy world. In Shannon's setup, Alice (users) wants to send a secret to Bob (the first analyst), which must remain perfectly private from Eve (the second analyst); whereas, in our setting, instead of complete independence, we only want that the secret remains hidden from Eve in the sense of differential privacy. We call this setup *private-recoverability*.

(iii) There are $d > 1$ analysts, who share some common randomness with the users. Analyst i would like to estimate $\mathbf{p}$ with privacy level ε_i, where $\varepsilon_1 > \ldots > \varepsilon_d$.[1]

7.1 Introduction

In general, designing private mechanisms with a small amount of randomness can be translated into communication efficiency and/or storage efficiency. For instance, when there are multiple privacy levels, each user needs to send additional information to some analysts, that is a function of the randomness used in the mechanism. Hence, using a smaller amount

[1] We can assume, without loss of generality, that $\varepsilon_j > \varepsilon_{j+1}, \forall j \in [d-1]$; otherwise, we can group the equal ε_j's together and the corresponding analysts can use the same privatized data that the users share with them.

of randomness implies delivering a smaller number of bits to each analyst.

The private-recoverability setup ($d = 2$) can be useful in applications such as census surveys, [Dwo19], that collect large amounts of data and are prohibitively expensive to repeat. Using our approach, we can store the randomized data on a public database (second analyst) without compromising the privacy of individuals; we can also give to the first analyst (e.g., the government, who may wish to exactly calculate the population count, or verify the validity of census results) a secret key, that can be used to "de-randomize" the publicly stored data and perfectly reconstruct the user inputs. An alternative approach would be to store the data twice (once randomized in a public database and once in a secure government database), which would incur an additional storage cost, as also shown in Section 7.7. Another alternative would be to use a cryptographic scheme to encode the user inputs; in this case, the resulting outputs may not allow public use in an efficient manner.[2]

The multi-level privacy $d > 1$ illustrates a new technical capability of hierarchical access to the raw data that might inspire and support a variety of applications. For example, given data collected from a fleet of autonomous cars, we could imagine different privacy access levels provided to the car manufacturer itself, to police departments, to applications interested in online traffic regulation, to applications interested in long-term traffic predictions or road planning. Essentially, this capability enables providing the desired utility needed for each application while maintaining the maximum possible amount of privacy. This chapter has three folds:

- For the single analyst case ($d = 1$), we characterize the trade-off between randomness and utility for a fixed privacy level ε, by proving an information-theoretic lower bound and a matching upper bound for a minimax private estimation problem.
- For private-recoverability ($d = 2$), we derive an information-theoretic lower bound on

[2] In principle, we could use homomorphic encryption that allows to compute a function on the encrypted data without decrypting it explicitly; however, such encryption schemes are computationally inefficient and expensive to deploy.

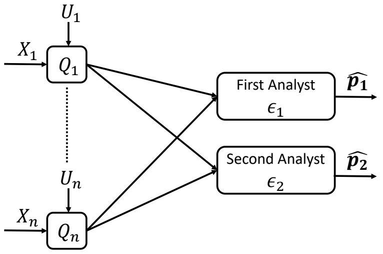


Figure 7.1: We have n users, each observing a sample X_i. A private randomization mechanism Q_i is applied to X_i using a random key U_i. Two analysts want to estimate $\mathbf{p}$. Each analyst requires a different privacy level.

the minimum randomness required to achieve it, and prove that the Hadamard scheme proposed in [ASZ19] is order optimal. We also show that we cannot reuse random keys over time while preserving privacy of each user. Hence, to preserve privacy of T samples, any ε-DP mechanism has to use an amount of randomness equal to T times the amount of randomness used for a single data sample. We also extend this result to estimating *heavy hitters*.

- In the multi-level privacy ($d > 1$) setting, a trivial scheme is to use the $d = 1$ scheme multiple times, separately for each analyst. We propose instead a non-trivial scheme that uses a smaller amount of randomness with no sacrifice in utility. Our scheme publicly announces the users' outputs, and allows each analyst to remove an appropriate amount of (shared) randomness with the help of an associated key. This approach enables efficient hierarchical access to the data (for example, when analysts have different levels of authorized access).

Overall, our investigation into privacy-utility-randomness trade-offs for LDP yields (optimal) privacy mechanisms that use randomness more economically. These include new guarantees for existing schemes such as the Hadamard mechanism, as well as new multi-user and multi-level mechanisms that allow for hierarchically private data access.

The rest of the chapter is organized as follows. We present the problem formulation in

Section 7.2. We study single-level privacy under randomness constraints in Section 7.3. We propose an efficient algorithm for multi-level privacy in Section 7.4. We study the private recoverability problem in Section 7.5. We provide numerical results in Section 7.7. Some proofs are deferred to Appendix E.

7.2 Problem Formulation

Notation: We use $[k]$ to define the set $\{1,\ldots,k\}$ of integers. We use uppercase letters X, Y, etc., to denote random variables, and lowercase letter x, y, etc., to denote their realizations. For any two distributions $\mathbf{p}$ and $\mathbf{q}$ supported over a set $\mathcal{X}$, let $\|\mathbf{p}-\mathbf{q}\|_{\text{TV}} = \sup_{\mathcal{A}\subseteq\mathcal{X}} |\mathbf{p}(\mathcal{A})-\mathbf{q}(\mathcal{A})|$ be the total variation distance between $\mathbf{p}$ and $\mathbf{q}$. We use $\oplus$ to define the XOR operation. For $p \in [0,1]$, we use $H_2(p)$ to denote the binary entropy function defined by $H_2(p) = -p\log(p) - (1-p)\log(1-p)$, and $H(X)$ to denote the entropy of the random variable X. Also, we use $H(\mathbf{p})$ to denote the entropy of a random variable X drawn from a distribution $\mathbf{p}$.

7.2.1 Local Differential Privacy (LDP)

Let $\mathcal{X} \triangleq \{1,\ldots,k\}$ be an input alphabet and $\mathcal{Y} \triangleq \{1,\ldots,m\}$ be an output alphabet, of sizes $|\mathcal{X}| = k$ and $|\mathcal{Y}| = m$, respectively, that are not required to be the same. A private randomization mechanism Q is a conditional distribution that takes an input $X \in \mathcal{X}$ and generates a privatized output $Y \in \mathcal{Y}$. Q is said to satisfy the ε-local differential privacy (ε-LDP) [DWJ13], if for every pair of inputs $x, x' \in \mathcal{X}$, we have

$$\sup_{y\in\mathcal{Y}} \frac{Q(y|x)}{Q(y|x')} \leq \exp(\varepsilon), \tag{7.1}$$

where $Q(y|x) = \Pr[Y = y|X = x]$ and ε captures the privacy level. For small values of ε, the adversary cannot infer whether the input was $X = x$ or $X = x'$. Hence, a smaller privacy level ε implies higher privacy.

7.2.2 Randomness in LDP Mechanisms

A private mechanism Q with input $X \in \mathcal{X}$ and output $Y \in \mathcal{Y}$ is said to satisfy (ε, R)-LDP, if for every pair of inputs $x, x' \in \mathcal{X}$, we have

$$\begin{aligned} &\sup_{y \in \mathcal{Y}} \frac{Q\left(y|x\right)}{Q\left(y|x'\right)} \leq \exp\left(\varepsilon\right), \text{ and} \\ &H\left(Y|X=x\right) \leq R \quad \forall x \in \mathcal{X}, \end{aligned} \tag{7.2}$$

where $H\left(Y|X=x\right) = \sum_{y \in \mathcal{Y}} Q\left(y|x\right) \log\left(\frac{1}{Q(y|x)}\right)$ denotes the entropy of the random output Y conditioned on the input $X = x$. Note that an (ε, R)-LDP mechanism is an ε-LDP mechanism that requires an amount of randomness less than or equal to R-bits to be designed.

Suppose that a random key U with $H\left(U\right) \leq R$ is used to design an (ε, R)-LDP mechanism Q. We consider U to be a random variable that takes values from a discrete set $\mathcal{U} = \{u_1, \ldots, u_l\}$ according to a distribution $\mathbf{q} = [q_1, \ldots, q_l]$, where $q_u = \Pr\left[U = u\right]$ for $u \in \mathcal{U}$. We assume that $\mathcal{U}$ is a discrete set, since we focus on finite randomness. Let $\mathcal{U}_{yx} \subset \mathcal{U}$ be a subset of key values such that input $X = x$ is mapped to $Y = y$ when $u \in \mathcal{U}_{yx}$. The private mechanism Q can be represented as

$$Q\left(y|x\right) = \sum_{u \in \mathcal{U}_{yx}} q_u. \tag{7.3}$$

Note that the output Y is a function of (X, U). Therefore, we have $\mathcal{U}_{y'x} \bigcap \mathcal{U}_{yx} = \phi$ for $y' \neq y$, since there is only one output for each input. In addition, if we want (7.3) to satisfy the privacy condition (7.1), we also have[3] $\bigcup_{y \in \mathcal{Y}} \mathcal{U}_{yx} = \mathcal{U}$ for each $x \in \mathcal{X}$. We will leverage this representation of randomness in LDP mechanisms to design multi-level privacy mechanisms. Figure 7.2 shows an example of designing a private mechanism with binary inputs $\mathcal{X} = \{0, 1\}$, binary random keys $\mathcal{U} = \{0, 1\}$, and binary outputs $\mathcal{Y} = \{0, 1\}$. In this example, we can represent the output of the mechanism as a function of (X, U) by $Y = X \oplus U$, where $\oplus$ denotes the XOR operation. If the random key U is drawn from a distribution $\mathbf{q} = \left[\frac{e^{\varepsilon}}{e^{\varepsilon}+1}, \frac{1}{e^{\varepsilon}+1}\right]$, then it is easy to show that the mechanism is ε-LDP.

[3] Otherwise we can distinguish inputs causing $\varepsilon \to \infty$.

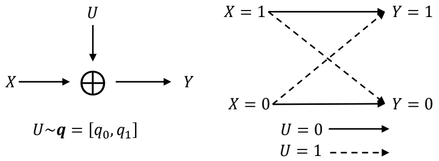


Figure 7.2: An example of designing an ε-LDP mechanism using a private key: (left) representing the output Y of the mechanism Q as a function of the input X and the private key U, (right) representing the mechanism Q as a probabilistic mapping from the input X to the output Y depending on the private key U.

7.2.3 Problem Formulation

We consider n users who observe i.i.d. inputs $X_1, X_2, \ldots, X_n$ (user i observes input X_i), drawn from an unknown discrete distribution $\mathbf{p} \in \Delta_k$, where $\Delta_k = \left\{\mathbf{p} \in \mathbb{R}^k | \sum_{j=1}^{k} p_j = 1, p_j \geq 0,\ \forall j \in [k]\right\}$ denotes the probability simplex over $\mathcal{X}$. The i'th user has a random key U_i with $H(U_i) \leq R$; we assume that $U^n = [U_1, \ldots, U_n]$ are independent random variables, unless otherwise stated. The i'th user generates (and publicly shares) an output Y_i, using an (ε, R)-LDP mechanism Q_i and her random key U_i. The output Y_i has a marginal distribution given by

$$\mathbf{M}_i\left(y|\mathbf{p}\right) = \sum_{x\in\mathcal{X}} Q_i\left(y|x\right) p_x \qquad \forall y \in \mathcal{Y}_i, \tag{7.4}$$

where $\mathcal{X}$ and $\mathcal{Y}_i$ are the input and output alphabets. We also have d analysts who want to use the users' public outputs $Y^n = [Y_1, \ldots, Y_n]$ to estimate $\mathbf{p}$, each at a different level of privacy $\varepsilon_1 > \ldots > \varepsilon_d$. The system model is shown in Figure 7.1.

Risk Minimization: For simplicity of exposition, consider for now a single analyst, and let $\hat{\mathbf{p}} = [\hat{p}_1, \cdots, \hat{p}_k]$ denote the analyst's estimator (this is a function $\hat{\mathbf{p}} : Y^n \to \mathbb{R}^k$ that maps the outputs Y^n to a distribution in the simplex Δ_k)[4]. For given private mechanisms

[4] Observe that it is sufficient to consider a deterministic estimator $\hat{\mathbf{p}}$, since for any randomized estimator, there exists a deterministic estimator that dominates the performance of the randomized one.

$Q^n = [Q_1, \dots, Q_n]$, the estimator $\hat{\mathbf{p}}$ is obtained by solving the problem

$$r^{\ell}_{\varepsilon,R,n,k}\left(Q^n\right) = \inf_{\hat{\mathbf{p}}} \sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}\right)\right], \tag{7.5}$$

where $r^{\ell}_{\varepsilon,R,n,k}$ is the minimax risk, the expectation is taken over the randomness in the outputs $Y^n = [Y_1, \dots, Y_n]$ with $Y_i \sim \mathbf{M}_i$, and $\ell : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+$ is a loss function that measures the distance between two distributions in Δ_k. Unless otherwise stated, we adopt as loss function the 1-norm, namely $\ell = \ell_1$ and the squared 2-norm, namely $\ell = \ell_2^2$. Our task is to design private mechanisms $Q_1, \dots, Q_n$ that minimize the minimax risk estimation, namely,

$$\begin{aligned} r^{\ell}_{\varepsilon,R,n,k} &= \inf_{\{Q_i\in\mathcal{Q}_{(\varepsilon,R)}\}} r^{\ell}_{\varepsilon,R,n,k}\left(Q^n\right) \\ &= \inf_{\{Q_i\in\mathcal{Q}_{(\varepsilon,R)}\}} \inf_{\hat{\mathbf{p}}} \sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}\right)\right], \end{aligned} \tag{7.6}$$

where $\mathcal{Q}_{(\varepsilon,R)}$ denotes the set of mechanisms that satisfy (ε, R)-LDP. Observe that when $R \to \infty$, the problem (7.6) is reduced to the standard LDP distribution estimation studied previously in [DWJ13, KBR16, YB18, ASZ19]. The difference in the formulation in (7.6) is the randomness constraint.

LDP heavy hitter estimation: In heavy hitter estimation, the input samples $X^n = [X_1, \dots, X_n]$ do not have an associated distribution. Furthermore, the analyst is interested in estimating the frequency of each element $x \in \mathcal{X}$ with the infinity norm being the loss function (i.e., $\ell = \ell_\infty$). Frequency of each element $x \in \mathcal{X}$ is defined by $f\left(x\right) = \frac{\sum_{i=1}^{n} \mathbb{1}(X_i = x)}{n}$. We then want to calculate

$$\begin{aligned} & r^{\ell_\infty}_{hh,\varepsilon,R,n,k} = \\ & \inf_{\{Q_i\in\mathcal{Q}_{(\varepsilon,R)}\}} \inf_{\hat{\mathbf{p}}} \sup_{X^n\in\mathcal{X}^n} \mathbb{E}\left[\max_{x\in\mathcal{X}} \left|\hat{p}_x\left(Y^n\right) - f\left(x\right)\right|\right], \end{aligned}$$

where the expectation is taken over the randomness in the outputs $Y^n = [Y_1, \dots, Y_n]$ and $\hat{\mathbf{p}}$ denotes the estimator of the analyst. Note, again, that in this case we do not make any distributional assumptions on $X_1, \dots, X_n$.

Multi-level privacy: Consider now the general case of d analysts each operating at a different level of privacy $\varepsilon_1 > \ldots > \varepsilon_d$. All analysts observe the users' public outputs

Y^n; additionally, analyst j may also observe some side information on the user randomness. The question we ask is: what is the minimum amount of randomness U per user required to maintain the privacy of each user while achieving the minimum risk estimation for each analyst?

Sequence of distribution (or heavy hitter) estimation: We assume that each user i has a random key U_i to preserve the privacy of a sequence of T independent samples $X_i^{(1)},\ldots,X_i^{(T)}$, where the t'th samples for $t \in [T]$ at all users are drawn i.i.d. from an unknown distribution $\mathbf{p}^{(t)}$.[5] At time t, the i'th user generates an output $Y_i^{(t)}$ that may be a function of the random key U_i and all input samples $\{X_i^{(m)}\}_{m=1}^{t}$. Each of the d analysts uses the outputs $Y_i^{(t)}, i \in [n], t \in [T]$ to estimate T distributions $\mathbf{p}^{(1)},\ldots,\mathbf{p}^{(T)}$ (or estimate the heavy hitters).

A private mechanism Q with a sequence of inputs $X^T = \left(X^{(1)},\ldots,X^{(T)}\right)$ and a sequence of outputs $Y^T = \left(Y^{(1)},\ldots,Y^{(T)}\right)$ is said to satisfy ε-DP, if for every neighboring databases $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^T$, we have

$$\sup_{\mathbf{y}\in\mathcal{Y}^T} \frac{Q\left(\mathbf{y}|\mathbf{x}\right)}{Q\left(\mathbf{y}|\mathbf{x}'\right)} \leq \exp\left(\varepsilon\right), \tag{7.7}$$

where $Q\left(\mathbf{y}|\mathbf{x}\right) = \Pr\left[Y^T = \mathbf{y}|X^T = \mathbf{x}\right]$; and we say that two databases, $\mathbf{x} = \left(x^{(1)},\ldots,x^{(T)}\right)$ and $\mathbf{x}' = \left(x'^{(1)},\ldots,x'^{(T)}\right) \in \mathcal{X}^T$ are neighboring, if there exists an index $t \in [T]$, such that $x^{(t)} \neq x'^{(t)}$ and $x^{(l)} = x'^{(l)}$ for $l \neq t$. Observe that when $T = 1$, the definition of ε-DP in (7.7) coincides with the definition of ε-LDP in (7.1). We are interested in the question: Is there a private mechanism that uses a smaller amount of randomness than T times the amount of randomness used for a single data sample? In other words, can we perhaps reuse the randomness over time while preserving privacy?

[5] As mentioned earlier, for heavy hitter estimation, the samples $X_i^{(1)},\ldots,X_i^{(T)}$ do not have an associated distribution.

7.3 Single-level Privacy

We here study the fundamental trade-off between randomness and utility for a fixed privacy level ε. In the following theorem, we derive a lower bound on the minimax risk estimation $r^{\ell_2^2}_{\varepsilon,R,n,k}$ and $r^{\ell_1}_{\varepsilon,R,n,k}$ defined in (7.6).

Theorem 7.3.1. *For every $\varepsilon, R \geq 0$ and $k, n \in \mathbb{N}$, the minimax risk under ℓ_2-norm loss is bounded by*

$$r^{\ell_2^2}_{\varepsilon,R,n,k} \geq \tau = \begin{cases} \frac{k(e^{\varepsilon}+1)^2}{16ne^{\varepsilon}(e^{\varepsilon}-1)^2} & \textit{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right), \\ \frac{ke^{\varepsilon}}{16np_R^2(e^{\varepsilon}-1)^2} & \textit{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right), \end{cases} \tag{7.8}$$

where $p_R \leq 0.5$ is the inverse of the binary entropy function $p_R = H_2^{-1}(R)$. The minimax risk under 1-norm loss is bounded by $r^{\ell_1}_{\varepsilon,R,n,k} \geq \sqrt{k\tau/8}$.

The main contribution in our proof (see Section 7.3.1) is a formulation of a non-convex optimization problem to bound the minimax risk under privacy and randomness constraints, and obtaining a tight bound on its solution for every value of privacy level ε and randomness R.

Remark 7.3.1. In [YB18], the authors derive the following lower bound on the minimax risk estimation without randomness constraints ($R \to \infty$)

$$r^{\ell_2^2}_{\varepsilon,\infty,n,k} \geq \begin{cases} \frac{k(e^{\varepsilon}+1)^2}{512n(e^{\varepsilon}-1)^2} & \text{for } e^{\varepsilon} < 3, \\ \frac{k}{64n(e^{\varepsilon}-1)} & \text{for } e^{\varepsilon} \geq 3. \end{cases} \tag{7.9}$$

For $\varepsilon = \mathcal{O}(1)$ and $R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right)$ (which includes $R \to \infty$ as well), our lower bound from Theorem 7.3.1 gives $r^{\ell_2^2}_{\varepsilon,R,n,k} = \Omega\left(\frac{k}{n\varepsilon^2}\right)$, which coincides with (7.9). However, our lower bound is tighter for all values of $\varepsilon \in [0, \infty)$ with smaller constant factors.

We next show that there exists an achievable scheme for all values of $\varepsilon, R \geq 0$ that matches (up to a constant factor) the lower bound given in Theorem 7.3.1 for $\varepsilon = \mathcal{O}(1)$ and $R \geq 0$.

Theorem 7.3.2. *For any $\varepsilon, R \geq 0$, there exists (ε, R)-LDP mechanisms $Q_1, \ldots, Q_n$ and an estimator $\hat{\mathbf{p}}$ such that the error $\mathcal{E} := \sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\|\hat{\mathbf{p}}\left(Y^n\right) - \mathbf{p}\|_2^2\right]$ is bounded by*

$$\mathcal{E} \leq \eta = \begin{cases} \frac{2k(e^{\varepsilon}+1)^2}{n(e^{\varepsilon}-1)^2} & \textit{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right), \\ \frac{2ke^{2\varepsilon}}{np_R^2(e^{\varepsilon}-1)^2} & \textit{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right). \end{cases} \tag{7.10}$$

The error under ℓ_1-norm loss is bounded by $\sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\|\hat{\mathbf{p}}\left(Y^n\right) - \mathbf{p}\|_1\right] \leq \sqrt{k\eta}$.

We prove Theorem 7.3.2 constructively in Section 7.3.2, by adapting the Hadamard response scheme given in [AS19] to our setting of limited randomness. Note that the value $H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right)$ in both the lower and the upper bounds is an exact threshold for randomness that determines the value of the minimax risk. Furthermore, we can see that the multiplicative gap between the lower bound presented in Theorem 7.3.1 and the achievable scheme in Theorem 7.3.2 is exactly $32e^{\varepsilon}$ for all randomness regimes. Theorems 7.3.1 and 7.3.2 together imply the following characterization for $r_{\varepsilon,R,n,k}^{\ell_2^2}$ and $r_{\varepsilon,R,n,k}^{\ell_1}$, for the case when $\varepsilon = \mathcal{O}(1)$:

Corollary 7.3.1. For $\varepsilon = \mathcal{O}\left(1\right)$ and $R \geq 0$, we have

$$r_{\varepsilon,R,n,k}^{\ell_2^2} = \begin{cases} \Theta\left(\frac{k}{n\varepsilon^2}\right) & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right), \\ \Theta\left(\frac{k}{np_R^2\varepsilon^2}\right) & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right), \end{cases} \tag{7.11}$$

and $r_{\varepsilon,R,n,k}^{\ell_1} = \sqrt{kr_{\varepsilon,R,n,k}^{\ell_2^2}}$.

We next provide a comparison between well-known mechanisms from randomness perspective. Table 7.1 describe the amount of randomness required to implement different ε-LDP mechanisms: RAPPOR [EPK14], Randomized Response (RR) [War65], Hadamard Response (HR) [ASZ19], and Binary Hadamard (BH) [AS19].

Observe that all private mechanisms are order optimal in the high privacy regime except for the RR scheme. However, only the BH scheme uses the smallest amount of randomness $R = H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right)$ per user, while the other mechanisms require a larger amount of randomness. Table 7.1 considers only the regime of randomness $R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right)$, since the privacy-utility

	RAPPOR	RR	HR	BH
Randomness per user (R in bits)	$kH_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$	$\log\left(k-1+e^\varepsilon\right)-$ $\frac{\varepsilon e^\varepsilon}{k-1+e^\varepsilon}$	$\leq \log\left(2k\frac{3e^\varepsilon-1}{e^\varepsilon}\right)-\frac{\varepsilon e^\varepsilon}{3e^\varepsilon-1}$	$H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$
Minimax risk ($r^{\ell_2^2}_{\varepsilon,R,n,k}$)	$\mathcal{O}\left(\frac{k}{n\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{k^2}{n\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{k}{n\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{k}{n\varepsilon^2}\right)$

Table 7.1: Randomness requirement to implement each private mechanism and its corresponding minimax risk under ℓ_2^2 loss function for $\varepsilon = \mathcal{O}\left(1\right)$.

trade-off when the amount of randomness $R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$ has not been studied before. Corollary 7.3.1 characterizes the privacy-utility trade-offs for all regions of randomness R.

Remark 7.3.2. Observe that when $R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$, there exists a trade-off between R and $r^{\ell_2^2}_{\varepsilon,R,n,k}$ – as R increases, $r^{\ell_2^2}_{\varepsilon,R,n,k}$ decreases proportionally to $1/p_R^2$. However, when $R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$, the minimax risk is not affected by R. Hence, $R = H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$ is a critical point that defines the minimum amount of randomness required for each user to generate an ε-LDP mechanism, while achieving the optimal utility at the analyst.

Remark 7.3.3. Corollary 7.3.1 also characterizes the number of users n (sample complexity) required to estimate the distribution $\mathbf{p}$ with estimation error at most α for given privacy level ε and randomness R bits per user is (where k is the input alphabet size):

$$n = \begin{cases} \Theta\left(\frac{k}{\alpha\varepsilon^2}\right) & \text{if } R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right), \\ \Theta\left(\frac{k}{\alpha p_R^2\varepsilon^2}\right) & \text{if } R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right). \end{cases} \tag{7.12}$$

A remark analogous to Remark 7.3.2 also holds here.

7.3.1 Lower Bound on The Minimax Risk Using the Assouad's Method

Now we prove the lower bound on the minimax risk given in Theorem 7.3.1 (see page 150). We first follow similar steps as in [DJW18, YB18] to reduce the minimax problem into multiple binary testing problems using Assouad's method. We note that [DJW18, YB18] do not consider a randomness constraint. Hence, we formulate an optimization problem to obtain a lower bound on the minimax risk estimation with a randomness constraint. Finding a tight bound on the solution of this problem is the main step in our proof. We also provide an alternative proof of Theorem 7.3.1 by using Fisher information, which leads to a tight bound for $\ell = \ell_2^2$ with smaller constant factors (see Appendix F.1).

Let $|\mathcal{X}| = k$ be the input alphabet size. Let $\{\mathbf{p}^\nu\}$ be a set of distributions parameterized by $\nu = \left(\nu_1, \ldots, \nu_{k/2}\right) \in \mathcal{V} = \{-1, 1\}^{k/2}$. The distribution $\mathbf{p}^\nu = (p_1^\nu, \ldots, p_k^\nu)$ is given by:

$$p_j^\nu = \begin{cases} \frac{1}{k} + \delta\nu_j & \text{if } j \in \{1, \ldots, k/2\} \\ \frac{1}{k} - \delta\nu_{j-k/2} & \text{if } j \in \{k/2+1, \ldots, k\} \end{cases}, \tag{7.13}$$

where $0 \leq \delta \leq 1/k$ is a parameter that will be chosen later. Let $Y^n = [Y_1, \ldots, Y_n]$ and $\mathcal{Y}^n = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n$. Following [DJW18], for any loss function $\ell\left(\hat{\mathbf{p}}, \mathbf{p}\right) = \sum_{j=1}^k \phi\left(\hat{p}_j - p_j\right)$, where $\phi : \mathbb{R} \to \mathbb{R}_+$ is a symmetric function, we have[6]

$$\begin{aligned} \ell\left(\hat{\mathbf{p}}\left(y^n\right), \mathbf{p}^\nu\right) &= \sum_{j=1}^k \phi\left(\hat{p}_j\left(y^n\right) - p_j^\nu\right) \\ &\geq \phi\left(\delta\right) \sum_{j=1}^{k/2} \mathbb{1}\left(\operatorname{sgn}\left(\hat{p}_j\left(y^n\right) - \frac{1}{k}\right) \neq \nu_j\right), \end{aligned} \tag{7.14}$$

where $\operatorname{sgn}\left(x\right) = 1$ if $x \geq 0$ and $\operatorname{sgn}\left(x\right) = 0$ otherwise. Suppose that user i chooses a private mechanism $Q_i \in \mathcal{Q}_{(\varepsilon,R)}$ that generates an output $Y_i \in \mathcal{Y}_i$. Let $\mathbf{M}_i^\nu$ be the output distribution on $\mathcal{Y}_i$ for an input distribution $\mathbf{p}^\nu$ on $\mathcal{X}$ defined by

$$\mathbf{M}_i^\nu\left(y\right) = \sum_{j=1}^k Q_i\left(y|X_i = j\right) p_j^\nu. \tag{7.15}$$

Let $\mathbf{M}_{+j}^n$ and $\mathbf{M}_{-j}^n$ denote the marginal distribution on $\mathcal{Y}^n$ conditioned on $\nu_j = +1$ and $\nu_j = -1$, respectively, where

$$\mathbf{M}_{+j}^n\left(y^n\right) = \frac{1}{|\mathcal{V}|} \sum_{\nu:\nu_j=+1} \prod_{i=1}^n \mathbf{M}_i^\nu\left(y_i\right)$$

[6] Observe that for loss function $\ell = \ell_2^2$, we have $\phi\left(x\right) = x^2$, and for loss function $\ell = \ell_1$, we have $\phi\left(x\right) = |x|$.

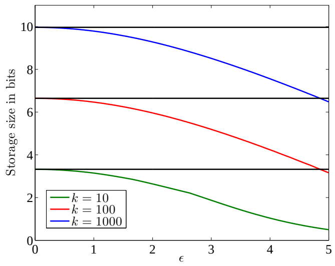


Figure 7.3: Comparison between storage required for X and a random key U, for input alphabet sizes $k \in \{10, 100, 1000\}$. The black lines represent $\log(k)$.

$$\mathbf{M}_{-j}^{n}\left(y^{n}\right) = \frac{1}{|\mathcal{V}|} \sum_{\nu:\nu_j=-1} \prod_{i=1}^{n} \mathbf{M}_{i}^{\nu}\left(y_i\right).$$

Thus, the minimax risk can be bounded using the following lemma whose proof is presented in Appendix F.1.1.

Lemma 7.3.1. *For the family of distributions* $\left\{\mathbf{p}^{\nu} : \nu \in \mathcal{V} = \{-1, 1\}^{k/2}\right\}$, *and a loss function* $\ell\left(\hat{\mathbf{p}}, \mathbf{p}\right) = \sum_{j=1}^{k} \phi\left(\hat{p}_j - p_j\right)$ *defined above, we have*

$$\begin{aligned} r_{\varepsilon,R,n,k}^{\ell} \geq \phi\left(\delta\right) \frac{k}{2} \Big(1 & \\ & - \sqrt{\frac{n}{2} \sup_{j\in[k/2]} \sup_{i\in[n]} \sup_{\nu:\nu_j=1} \sup_{Q_i \in \mathcal{Q}_{(\varepsilon,R)}} D_{KL}\left(\mathbf{M}_{i}^{\nu} || \mathbf{M}_{i}^{\nu-2e_j}\right)} \Big) \end{aligned} \tag{7.16}$$

Fix arbitrary $i \in [n]$, $j \in [k/2]$ and $\nu \in \mathcal{V}$. We have

$$\begin{aligned} & D_{\text{KL}}\left(\mathbf{M}_{i}^{\nu} || \mathbf{M}_{i}^{\nu-2e_j}\right) \\ & \quad \overset{(a)}{\leq} D_{\text{KL}}\left(\mathbf{M}_{i}^{\nu} || \mathbf{M}_{i}^{\nu-2e_j}\right) + D_{\text{KL}}\left(\mathbf{M}_{i}^{\nu-2e_j} || \mathbf{M}_{i}^{\nu}\right) \\ & \quad = \sum_{y\in\mathcal{Y}_i} \left(\mathbf{M}_{i}^{\nu}\left(y\right) - \mathbf{M}_{i}^{\nu-2e_j}\left(y\right)\right) \log\left(\frac{\mathbf{M}_{i}^{\nu}\left(y\right)}{\mathbf{M}_{i}^{\nu-2e_j}\left(y\right)}\right) \\ & \quad \overset{(b)}{\leq} \sum_{y\in\mathcal{Y}_i} \frac{\left(\mathbf{M}_{i}^{\nu}\left(y\right) - \mathbf{M}_{i}^{\nu-2e_j}\left(y\right)\right)^2}{\mathbf{M}_{i}^{\nu-2e_j}\left(y\right)} \end{aligned}$$

$$\overset{(c)}{=} \sum_{y\in\mathcal{Y}_i} \delta^2 \frac{\left(Q_i\left(y|j\right) - Q_i\left(y|j+k/2\right)\right)^2}{\sum_{j'=1}^{k} Q_i\left(y|j'\right) p_{j'}^{\nu-2e_j}}$$

$$\overset{(d)}{\leq} 2\delta^2 e^{\varepsilon} \sum_{y\in\mathcal{Y}_i} \frac{\left(Q_i\left(y|j\right) - Q_i\left(y|j+k/2\right)\right)^2}{Q_i\left(y|j\right) + Q_i\left(y|j+k/2\right)} \ , \tag{7.17}$$

where step (a) follows from the fact that $D_{\text{KL}}\left(.||.\right)$ is not negative. Step (b) follows from the inequality $\log\left(x\right) \leq x - 1$. Step (c) follows from the definition of $\mathbf{M}_i^{\nu}$ in (7.15). Step (d) follows from bounding the denominator as follows:

$$\begin{aligned}\sum_{j'=1}^{k} & Q_i\left(y|j'\right) p_{j'}^{\nu-2e_j} \\ & \geq e^{-\varepsilon} \frac{Q_i\left(y|j\right) + Q_i\left(y|j+k/2\right)}{2} \sum_{j'=1}^{k} p_{j'}^{\nu-2e_j} \qquad (7.18) \\ & = e^{-\varepsilon} \frac{Q_i\left(y|j\right) + Q_i\left(y|j+k/2\right)}{2},\end{aligned}$$

where we use the fact that $Q_i\left(y|j'\right) \geq e^{-\varepsilon} Q_i\left(y|j\right)$ and $Q_i\left(y|j'\right) \geq e^{-\varepsilon} Q_i\left(y|j+k/2\right), \ \forall j' \in [k]$.

Lemma 7.3.2. *For any randomized mechanism $Q \in \mathcal{Q}_{(\varepsilon,R)}$ that generates an output $Y \in \mathcal{Y}$, we have*

$$\begin{aligned}\sup_{Q\in\mathcal{Q}_{(\varepsilon,R)}} & \sum_{y\in\mathcal{Y}} \frac{\left(Q\left(y|j\right) - Q\left(y|j+k/2\right)\right)^2}{Q\left(y|j\right) + Q\left(y|j+k/2\right)} \\ & \leq \begin{cases} 2\frac{\left(e^{\varepsilon}-1\right)^2}{\left(e^{\varepsilon}+1\right)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2\frac{p_R^2\left(e^{\varepsilon}-1\right)^2}{e^{2\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases}\end{aligned} \tag{7.19}$$

This lemma presents an upper bound on equation (7.17) as a function of the randomness R for any private mechanism $Q \in \mathcal{Q}_{(\varepsilon,R)}$. To prove this lemma, we first show that the optimization problem (7.19) is non-convex due to the randomness constraint. We then prove that the maximum value of this function (7.19) is obtained when the output of the mechanism $Q \in \mathcal{Q}_{(\varepsilon,R)}$ is binary. Then, we obtain a tight bound numerically for the binary output.

Proof of Lemma 7.3.2. Without loss of generality assume that $\mathcal{Y} = \{y_1, \ldots, y_m\}$ with $|\mathcal{Y}| = m$. For ease of notation, we write $Q\left(y_l|j\right) = q_{l,j}$ and $Q\left(y_l|j+k/2\right) = q_{l,j+k/2}$. The problem (7.19) can be formulated as follows

$$\textbf{P1:} \qquad \max_{\{q_{l,j},q_{l,j+k/2}\}_{l=1}^{m}} \sum_{l=1}^{m} \frac{\left(q_{l,j}-q_{l,j+k/2}\right)^2}{q_{l,j}+q_{l,j+k/2}} \tag{7.20}$$

$$\begin{aligned} \text{s.t.} \quad & H\left([q_{1,j},\dots,q_{m,j}]\right) \leq R, \\ & H\left(\left[q_{1,j+k/2},\dots,q_{m,j+k/2}\right]\right) \leq R \qquad (7.21) \\ & e^{-\varepsilon} \leq \frac{q_{l,j}}{q_{l,j+k/2}} \leq e^{\varepsilon}, \quad \forall l \in [m] \\ & q_{l,j} \geq 0, \qquad q_{l,j+k/2} \geq 0, \qquad\qquad \forall l \in [m] \\ & \sum_{l=1}^{m} q_{l,j} = 1, \quad \sum_{l=1}^{m} q_{l,j+k/2} = 1 \end{aligned}$$

Note that the objective function (7.20) is jointly convex in both $\{q_{l,j}\}_{l=1}^{m}$ and $\{q_{l,j+k/2}\}_{l=1}^{m}$. However, the optimization problem **P1** is non-convex due to two reasons. First, we maximize a convex function, and second the entropy constraints (7.21) are sub-level sets of a concave function and are non-convex constraints. However, we can solve the optimization problem **P1** by exploiting the results of Lemma 7.3.3 below.

Lemma 7.3.3. *The optimal solution of the non-convex optimization problem **P1** is obtained when the output size is $m = 2$.*

The proof of Lemma 7.3.3 is presented in Appendix F.2. Since the output alphabet is binary, we can efficiently plot the feasible region of **P1** for $m = 2$ as depicted in Figure 7.4. Since we maximize a convex function, the optimal solution is at the boundary of the feasible set. Furthermore, the objective function (7.20) is symmetric on $q_{1,j},\ q_{1,j+k/2}$ for $m = 2$. As a result, the optimal solution is given by.

$$\begin{aligned} q_{1,j}^{*} &= \begin{cases} \frac{e^{\varepsilon}}{e^{\varepsilon}+1} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ p_R & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \\ q_{1,j+k/2}^{*} &= \begin{cases} \frac{1}{e^{\varepsilon}+1} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ \frac{p_R}{e^{\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases}, \end{aligned} \tag{7.22}$$

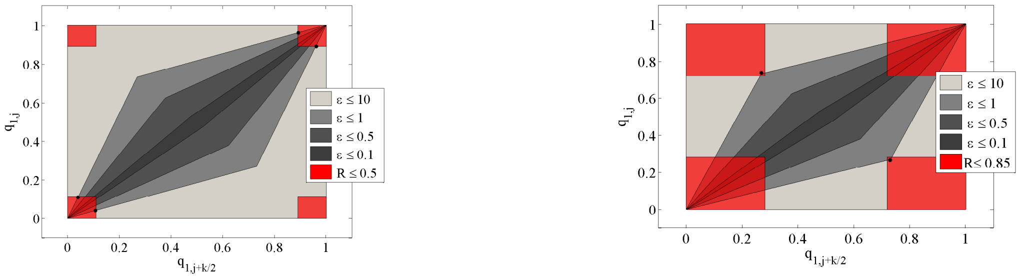


Figure 7.4: The feasible region of the optimization problem $P1$ for $m = 2$.

where $q^*_{2,j} = 1 - q^*_{1,j}$, and $q^*_{2,j+k/2} = 1 - q^*_{1,j+k/2}$. Substituting from (7.22) into the objective function (7.20), we get

$$\sum_{l=1}^{m} \frac{\left(q_{l,j} - q_{l,j+k/2}\right)^2}{q_{l,j} + q_{l,j+k/2}} \leq \begin{cases} 2\frac{(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2\frac{p_R^2(e^{\varepsilon}-1)^2}{e^{2\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \tag{7.23}$$

Hence, the proof is completed for Lemma 7.3.2. ■

Using the bound from Lemma 7.3.2 in (7.17) and taking supremum over all $Q_i \in \mathcal{Q}_{\varepsilon,R}$, we get

$$\begin{aligned} &\sup_{Q_i \in \mathcal{Q}_{(\varepsilon,R)}} D_{\text{KL}}\left(\mathbf{M}_i^{\nu} || \mathbf{M}_i^{\nu-2e_j}\right) \\ &\leq 2\delta^2 e^{\varepsilon} \sup_{Q_i \in \mathcal{Q}_{(\varepsilon,R)}} \sum_{y \in \mathcal{Y}_i} \frac{\left(Q_i\left(y|j\right) - Q_i\left(y|j+k/2\right)\right)^2}{Q_i\left(y|j\right) + Q_i\left(y|j+k/2\right)} \\ &= 2\delta^2 e^{\varepsilon} \begin{cases} 2\frac{(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2\frac{p_R^2(e^{\varepsilon}-1)^2}{e^{2\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \end{aligned} \tag{7.24}$$

Substituting from (7.24) into (7.16), we get

$$\begin{aligned} &r^{\ell}_{\varepsilon,R,n,k} \\ &\geq \begin{cases} \phi\left(\delta\right)\frac{k}{2}\left(1 - \sqrt{2\delta^2 n e^{\varepsilon}\frac{(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2}}\right) & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ \phi\left(\delta\right)\frac{k}{2}\left(1 - \sqrt{2\delta^2 n \frac{p_R^2(e^{\varepsilon}-1)^2}{e^{\varepsilon}}}\right) & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \end{aligned} \tag{7.25}$$

By setting $\delta^2 = \frac{(e^\varepsilon+1)^2}{8ne^\varepsilon(e^\varepsilon-1)^2}$ if $R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$ and $\delta^2 = \frac{e^\varepsilon}{8np_R^2(e^\varepsilon-1)^2}$ if $R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right)$, we get

$$r^{\ell}_{\varepsilon,R,n,k} \geq \begin{cases} \phi\left(\sqrt{\frac{(e^\varepsilon+1)^2}{8ne^\varepsilon(e^\varepsilon-1)^2}}\right)\frac{k}{4} & \text{if } R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right) \\ \phi\left(\sqrt{\frac{e^\varepsilon}{8np_R^2(e^\varepsilon-1)^2}}\right)\frac{k}{4} & \text{if } R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon+1}\right) \end{cases} \tag{7.26}$$

For the loss function $\ell = \ell_2^2$, we set $\phi(x) = x^2$ and for $\ell = \ell_1$, we set $\phi(x) = |x|$. This completes the proof of Theorem 7.3.1 with a slightly worse constant of 32 instead of 16 in the denominator. We provide a different proof of Theorem 7.3.1 in Appendix F.1 using Fisher information that gives the exact bound as stated in Theorem 7.3.1.

7.3.2 Upper Bound on The Minimax Risk Using the Hadamard Response

In this section, we prove Theorem 7.3.2 (see page 151) by proposing a private mechanism by adapting the Hadamard response given in [AS19], where each user answers to a yes-no question such that the probability of telling the truth depends on the amount of randomness R. Each user $i \in [n]$ has a binary output $Y_i \in \{0,1\}$. The (ε, R)-LDP mechanism of the i-th user is defined by

$$Q(Y_i = 1|X) = \begin{cases} q & \text{if } X \in B_i \\ \frac{q}{e^\varepsilon} & \text{if } X \notin B_i \end{cases} \tag{7.27}$$

where $B_i \subset [k]$ is a subset of inputs, and q is a probability value that will be determined later such that $H_2(q) \leq R$. Let $K = 2^{\lceil \log(k) \rceil}$ denote the smallest power of 2 larger than k, and H_K be the $K \times K$ Hadamard matrix. In the following, we assume an extended distribution $\overline{\mathbf{p}}$ over the set $\mathcal{X} = [K]$ with $|\mathcal{X}| = K$ that is obtained by zero-padding the original distribution $\mathbf{p}$ with $(K-k)$ zeros, i.e., $\overline{\mathbf{p}} = [\overline{p}_1, \ldots, \overline{p}_K] = [p_1, \ldots, p_k, 0, \ldots, 0]$. For $j \in [K]$, let B^j be a set of row indices that have 1 in the j-th column of the Hadamard matrix H_K. For example, when $K = 4$, the Hadamard matrix is given by

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{7.28}$$

Hence, $B^1 = \{1,2,3,4\}$, $B^2 = \{1,3\}$, $B^3 = \{1,2\}$, and $B^4 = \{1,4\}$. We divide the users into K sets $(\mathcal{US}_1, \ldots, \mathcal{US}_K)$, where each set contains n/K users. For each user $i \in \mathcal{US}_j$, we set $B_i = B^j$. Let $p\left(B^j\right) = \Pr\left[X \in B^j\right] = \sum_{x \in B^j} \overline{p}_x$, and $s_j = \Pr\left[Y_i = 1\right]$ for $i \in \mathcal{U}_j$. Then, we can easily see that

$$\begin{aligned} s_j &= p\left(B^j\right) q + \left(1 - p\left(B^j\right)\right) \frac{q}{e^\varepsilon} \\ &= p\left(B^j\right) q \left(\frac{e^\varepsilon - 1}{e^\varepsilon}\right) + \frac{q}{e^\varepsilon} \end{aligned} \tag{7.29}$$

Let $\hat{s}_j = \frac{1}{|\mathcal{US}_j|} \sum_{i \in \mathcal{US}_j} \mathbb{1}\left\{Y_i = 1\right\}$ denote the estimate of s_j. Then, we can estimate $p\left(B^j\right)$ as $\hat{p}\left(B^j\right) = \frac{e^\varepsilon}{q(e^\varepsilon - 1)}\left(\hat{s}_j - \frac{q}{e^\varepsilon}\right)$. Observe that the relation between the distribution $\overline{\mathbf{p}}$ and $\mathbf{p}\left(B\right) = \left[p\left(B^1\right), \ldots, p\left(B^K\right)\right]$ is given by [AS19, Eq. 13]

$$\mathbf{p}\left(B\right) = \frac{H_K \overline{\mathbf{p}} + \mathbf{1}_K}{2}, \tag{7.30}$$

where $\mathbf{1}_K$ denotes a vector of K ones. Hence, we can estimate the distribution $\overline{\mathbf{p}}$ as

$$\hat{\overline{\mathbf{p}}} = H_K^{-1}\left(2\hat{\mathbf{p}}\left(B\right) - \mathbf{1}_K\right) = \frac{1}{K} H_K \left(2\hat{\mathbf{p}}\left(B\right) - \mathbf{1}_K\right). \tag{7.31}$$

Lemma 7.3.4. *For arbitrary* $\mathbf{p} \in \Delta_k$*, we have*

$$\mathbb{E}\left[\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2\right] \leq \frac{2ke^{2\varepsilon}}{nq^2\left(e^\varepsilon - 1\right)^2}. \tag{7.32}$$

The proof is exactly the same as the proof in [AS19, Theorem 5]. By setting $q = \frac{e^\varepsilon}{e^\varepsilon + 1}$ if $R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon + 1}\right)$ and $q = p_R$ if $R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon + 1}\right)$, we get

$$r_{\varepsilon,R,n,k}^{\ell_2^2} \leq \begin{cases} \frac{2k(e^\varepsilon + 1)^2}{n(e^\varepsilon - 1)^2} & \text{if } R \geq H_2\left(\frac{e^\varepsilon}{e^\varepsilon + 1}\right), \\ \frac{2ke^{2\varepsilon}}{np_R^2(e^\varepsilon - 1)^2} & \text{if } R < H_2\left(\frac{e^\varepsilon}{e^\varepsilon + 1}\right). \end{cases} \tag{7.33}$$

The difference in our mechanism is that we design the private mechanism (7.27) for all values of randomness R. This completes the proof of Theorem 7.3.2.

7.4 Multi-level Privacy

Here, we study the case of d different analysts, with privacy levels $\varepsilon_1 > \cdots > \varepsilon_d$, and $\varepsilon_j = \mathcal{O}(1)$ for $j \in [d]$ (See Section 7.1 for the motivation of this setup). A trivial scheme is to use the $d = 1$ scheme multiple times, separately for each analyst: each user $i \in [n]$ generates d samples $\left(Y_i^1, \ldots, Y_i^d\right)$ from its input sample X_i. The jth sample Y_i^j is delivered privately to the jth analyst. Note that the jth sample must be generated from an ε_j-LDP. It then follows from Corollary 7.3.1 that the minimum risk for the jth analyst is given by $r^{\ell_2^2}_{\varepsilon_j,\infty,n,k} = \Theta\left(\frac{k}{n\varepsilon_j^2}\right)$, which requires each user to have $R_j \geq H_2\left(\frac{e^{\varepsilon_j}}{e^{\varepsilon_j}+1}\right)$ bits of randomness, and results in a total amount of randomness

$$R_{\text{total}}^{\text{trivial}} = \sum_{j=1}^{d} H_2\left(\frac{e^{\varepsilon_j}}{e^{\varepsilon_j}+1}\right).$$

We propose a new solution for this problem, in which each user generates a single output that is publicly accessible by all analysts; each analyst is given a part of the random key that was used to privatize the data, and leverages this key to reduce the perturbation of the public output. The next theorem is proved in Section 7.4.

Theorem 7.4.1. *There exists a private mechanism using a total amount of randomness given by $R_{total}^{proposed} = \sum_{j=1}^{d} H_2\left(q_j\right)$, such that the jth analyst achieves the minimum risk estimation $r^{\ell_2^2}_{\varepsilon_j,\infty,n,k} = \Theta\left(\frac{k}{n\varepsilon_j^2}\right)$, while preserving privacy of each user with privacy level ε_j for $j \in [d]$. Here, for every $j \in [d]$, q_j is defined as follows (where $z_j = \frac{1}{e^{\varepsilon_j}+1}$):*

$$q_j = \begin{cases} z_j & \text{if } j = 1, \\ \frac{z_j - z_{j-1}}{1-2z_{j-1}} & \text{if } j > 1. \end{cases} \tag{7.34}$$

Our results demonstrate that the proposed scheme in Theorem 7.4.1 achieves exactly the same privacy and minimax risk as the trivial scheme, but with a much lower amount of randomness (See Eqn. (7.40) in Section 7.4 for more details) as we elaborate in the next remark. The reason is that by doing the hierarchical randomization, each analyst can recover the same output as in the trivial scheme using the shared private key. Hence, we get the same privacy and minimax risk as the trivial scheme.

Remark 7.4.1. Note that $z_j > z_{j-1}$ as $\varepsilon_{j-1} > \varepsilon_j$. Moreover, we also have $z_j = 1/\left(e^{\varepsilon_j}+1\right) < 0.5$ for all $j \in [d]$. As a result, we can show that for $j > 1$, we have

$$q_j = \frac{z_j - z_{j-1}}{1-2z_{j-1}} = z_j - \frac{z_{j-1}\left(1-2z_j\right)}{1-2z_{j-1}} < z_j. \tag{7.35}$$

Hence, we get that $H_2\left(q_j\right) < H_2\left(z_j\right)$ holds for all $j > 1$. Therefore, our proposed scheme uses a strictly smaller amount of randomness than the trivial scheme.

7.4.1 Proof of Theorem 7.4.1

This section proves Theorem 7.4.1 by establishing a new technique using a smaller amount of randomness than the trivial scheme mentioned before while achieving the minimum risk estimation for each analyst. Our proposed mechanism for multi-level privacy (where $\varepsilon_1 > \ldots > \varepsilon_d$) is a cascading mechanism, where in each step, we add a random key to the output of the previous step (see Figure 7.5, for example). The common output of the mechanism is accessible by all analysts. However, each analyst would have a different privacy level depending on the amount of randomness shared with it. Thus, each analyst uses the shared random key to partially undo the randomization of the common output to get less privacy and higher utility. Let $z_j = \frac{1}{e^{\varepsilon_j}+1}$ for $j \in [d]$. For $i \in [n]$, let $\{U_i^1, \ldots, U_i^d\}$ be a set of d Bernoulli random variables, where U_i^j has a parameter $q_j = \Pr\left[U_i^j = 1\right]$ given by

$$q_j = \begin{cases} z_j & \text{if } j = 1, \\ \frac{z_j - z_{j-1}}{1-2z_{j-1}} & \text{if } j > 1. \end{cases} \tag{7.36}$$

We first use the Hadamard response proposed in [AS19] for getting the first step of our mechanism (see Section 7.3.2 for more details). Let H_K be the $K \times K$ Hadamard matrix. Let B^l be a set of the row indices that have 1 in the l-th column of Hadamard matrix H_K for $l \in [K]$. We divide the users into K sets $(\mathcal{US}_1, \ldots, \mathcal{US}_K)$, where each set contains n/K

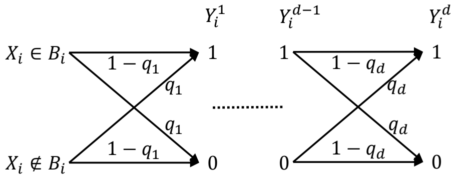


Figure 7.5: Multiple privacy levels mechanism.

users. We assign a set $B_i = B^l$ representing a subset of inputs for each user $i \in \mathcal{US}_l$. Then, user i generates a virtual output $Y_i^1 \in \{0,1\}$ as follows

$$Y_i^1 = \begin{cases} 1 & \text{if } \left(X_i \in B_i \;\&\; U_i^1 = 0\right) \text{ or } \left(X \notin B_i \;\&\; U_i^1 = 1\right), \\ 0 & \text{otherwise.} \end{cases} \tag{7.37}$$

Observe that the representation of Y_i^1 in (7.37) is exactly the same as in (7.27) by setting $q = \Pr\left[U_i^1 = 0\right] = \frac{e^\varepsilon}{e^\varepsilon+1}$. We represent Y_i^1 with this form to explicitly show the random keys used to design the Hadamard scheme presented in Section 7.3.2. Let Y_i^j be the virtual output generated by user i for the jth analyst, which is given by

$$Y_i^j = Y_i^1 \oplus U_i^2 \oplus \ldots \oplus U_i^j, \tag{7.38}$$

where $\oplus$ denotes the bitwise XOR. Hence, we add randomization to the first step of the Hadamard scheme. User i transmits the output Y_i^d to all analysts. The private scheme is shown in Figure 7.5.

Lemma 7.4.1. *The jth output of user i satisfies ε_j-LDP, i.e.,*

$$\sup_{y_i^j \in \{0,1\}} \sup_{x_i, x_i' \in \mathcal{X}} \frac{Pr\left[Y_i^j = y_i^j | X_i = x_i\right]}{Pr\left[Y_i^j = y_i^j | X_i = x_i'\right]} \leq e^{\varepsilon_j} \tag{7.39}$$

We prove Lemma 7.4.1 in Appendix F.4. Note that each analyst has access to the public outputs $\{Y_1^d, \ldots, Y_n^d\}$ which is ε_d-LDP. Additionally, user i sends a random key $L_i^j = U_i^d \oplus \ldots \oplus U_i^{j+1}$ to the jth analyst. Using the random keys $\{L_1^j, \ldots, L_n^j\}$, the jth analyst can construct the private outputs $\{Y_1^j, \ldots, Y_n^j\}$ which are ε_j-LDP, where $Y_i^j = Y_i^d \oplus L_i^j$. Observe that the privatized output Y_i^j has a conditional distribution given by

$$Q_i\left(Y_i^j|X_i\right) = \begin{cases} \frac{e^{\varepsilon_j}}{e^{\varepsilon_j}+1} & \text{if } X_i \in B_i \\ \frac{1}{e^{\varepsilon_j}+1} & \text{if } X_i \notin B_i \end{cases} \tag{7.40}$$

which coincides with the private mechanism given in (7.27) with $q = \frac{e^{\varepsilon_j}}{e^{\varepsilon_j}+1}$. Thus, the jth analyst can recover the same output as in the trivial scheme using the shared private key. Hence, we get the same privacy and minimax risk as the trivial scheme. From Lemma 7.3.4, for privacy level $\varepsilon_j = \mathcal{O}(1)$, we get that

$$r_{\varepsilon,R,n,k}^{\ell_2^2,j} = \mathcal{O}\left(\frac{k}{n\varepsilon_j^2}\right), \tag{7.41}$$

for analyst j, which coincides with the lower bound stated in Corollary 7.3.1. Observe that the total amount of randomness per user in the proposed mechanism is given by

$$R_{\text{total}}^{\text{proposed}} = \sum_{j=1}^{d} H\left(U^j\right) = \sum_{j=1}^{d} H_2\left(q_j\right) \leq R_{\text{total}}^{\text{trivial}}, \tag{7.42}$$

where q_j is defined in (7.36). Note that the last inequality is strict for $d > 1$. This completes the proof of Theorem 7.4.1.

7.5 Private-Recoverability

We here consider a legitimate analyst with permission to access the data $\{X_i\}_{i=1}^n$, i.e., $\varepsilon_1 \to \infty$, and an untrusted analyst with privacy level $\varepsilon_2 < \infty$. The ith user uses a random private key U_i and her mechanism Q_i to generate an output Y_i that is publicly accessible by both analysts.

Definition 7.5.1 (**LDP-Rec mechanisms**). We say that a private mechanism Q is ε-LDP-Rec, if it is an ε-LDP mechanism and it is possible to recover the input X from output Y and the key U.

We derive necessary and sufficient conditions on the random keys $\{U_i\}$ and the mechanisms $\{Q_i\}$, such that the legitimate analyst can recover X_i from observing U_i and Y_i, while preserving privacy level ε_2 against the untrusted analyst who does not have access to the keys.

We first consider a simplified setting as shown in Figure 7.6. Alice (an arbitrary user [7]) has a sample $X \in \mathcal{X}$. Alice wants to send her sample X to Bob (the legitimate analyst) while keeping her sample X private against Eve (the untrusted analyst) with differential privacy level ε. Eve has access to the message between Alice and Bob. However, Alice has a random key U shared with Bob that Eve does not have access to. Let Y be the output of the private mechanism Q used by Alice. The following theorem (which we prove in Section 7.5.1) provides necessary and sufficient conditions on the random key U and the privatized output Y to generate an ε-LDP-Rec mechanism.

Remark 7.5.1. Observe that in the simplified model in Figure 7.6, we do not impose any assumptions on the input X. Furthermore, we do not impose any assumptions about the task for Eve. Hence, our model and results in Theorem 7.5.1 are applicable to any task for Eve including distribution estimation, heavy hitter estimation, or learning from sample X.

Theorem 7.5.1. *Let Q be an ε-LDP-Rec mechanism that uses a random key $U \in \mathcal{U}$ and an input $X \in \mathcal{X}$ to produce a privatized output $Y \in \mathcal{Y}$. The following conditions are necessary and sufficient to allow recovery of X from (U, Y):*
(1) $|\mathcal{U}| \geq |\mathcal{Y}| \geq |\mathcal{X}|$.
(2) The entropy of the random key must satisfy $H(U) \geq H\left(U_{\min}^{s^*}\right)$, where $s^* = \arg\min_{s \in \{\lceil l \rceil, \lfloor l \rfloor\}} H\left(U_{\min}^{s}\right)$ for $l = k\frac{e^{\varepsilon}(\varepsilon-1)+1}{(e^{\varepsilon}-1)^2}$ and $U_{\min}^{s}$ is a random variable with support size equal to $|\mathcal{X}| = k$ and has the following distribution:

$$\mathbf{q}_{\min}^{s} = [1/t, \ldots, 1/t, e^{\varepsilon}/t, \ldots, e^{\varepsilon}/t],$$

[7] Since the input samples $X_1, \ldots, X_n$ are i.i.d., and the random keys $U_1, \ldots, U_n$ are independent random variables, it is sufficient to study the private-recoverable mechanism for any single user.

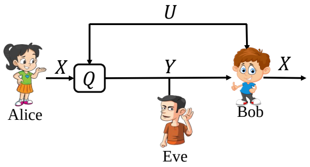


Figure 7.6: Private-Recoverability: Alice has data X. An ε-LDP-Rec mechanism Q is applied to X using a random key U to generate output Y. Bob is capable to recover X from Y and U. Eve only observes Y.

where $t = (se^{\varepsilon} + k - s)$, the first $k - s$ terms are equal to $1/t$ and the remaining s terms are equal to e^{ε}/t.

We now discuss the effect of ε on the structure of optimal distribution $\mathbf{q}_{\min}^{s^*}$ for $U_{\min}^{s^*}$: **(i)** When $\varepsilon \gg \log(k)$, the optimal $s^* = 1$, and the corresponding $\mathbf{q}_{\min}^{1}$ has its first $k-1$ terms equal to $1/(e^{\varepsilon} + k - 1)$ and the last term equal to $e^{\varepsilon}/(e^{\varepsilon} + k - 1)$. This distribution is equivalent to the one used in the Randomized Response (RR) model proposed in [War65]. **(ii)** When $\varepsilon \to 0$, the optimal s^* is around $k/2$, and the corresponding $\mathbf{q}_{\min}^{k/2}$ has its first $k/2$ terms equal to $2/k(e^{\varepsilon} + 1)$ and the remaining $k/2$ terms equal to $2e^{\varepsilon}/k(e^{\varepsilon} + 1)$. **(iii)** When $\varepsilon = 0$, the distribution $q_{\min}^{s}$ becomes uniform (irrespective of the value of s). Thus, when ε decreases, the distribution $\mathbf{q}_{\min}^{s}$ approaches to the uniform distribution. On the other hand, when ε increases, the distribution $\mathbf{q}_{\min}^{s}$ becomes skewed. It turns out that the minimum randomness required to generate an ε-LDP-Rec mechanism for input recoverability is a non-increasing function of ε. In other words, more privacy requires more randomness.

Remark 7.5.2. Consider the cryptosystem introduced by Shannon in [Sha49], where Alice wants to send a secure message X to Bob using a shared random key U. Let Y be the encrypted message sent to Bob. Eve eavesdrops the channel between Alice and Bob and observes Y. This cryptosystem achieves *perfect secrecy* if and only if $I\left(X;Y\right) = 0$. Shannon showed that perfect secrecy requires $H\left(U\right) \geq H\left(X\right)$. Since the distribution of X is not known to any node (Alice, Bob, and Eve), this implies $H\left(U\right) \geq \max_{p_X \in \Delta_k} H(X) = \log k$. We can easily verify that the ε-LDP-Rec mechanism satisfies a cryptosystem with secrecy measure

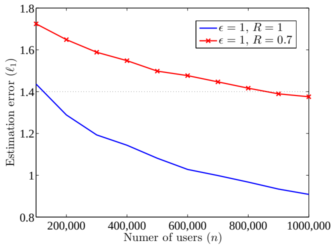


Figure 7.7: ℓ_1-estimation error for input alphabet size $k = 1000$, privacy level $\varepsilon = 1$, and $\mathbf{p} = \text{Geo}\,(0.8)$.

$\max_{\mathbf{p}\in\Delta_k} I\left(X;Y\right) \leq \varepsilon$. Hence, a perfect secrecy system with unknown input distribution is a 0-LDP-Rec mechanism, which is a special case of our problem. Moreover, the ε-LDP-Rec mechanism with data recovery is a cryptosystem leaking an amount of information measured by $\max_{\mathbf{p}\in\Delta_k} I\left(X;Y\right) \leq \varepsilon$.

Observe that Theorem 7.5.1 does not provide performance guarantees for Eve, it only guarantees privacy for Alice with respect to Eve, and recoverability for Bob. Hence, we can ask the question: Does there exist an ε-LDP-Rec mechanism using the smallest amount of randomness and guaranteeing the smallest error for distribution estimation or heavy hitter estimation for Eve (the untrusted analyst)? In the following theorem (which we prove in Section 7.5.2), we show that such a mechanism exists.

Theorem 7.5.2. *The Hadamard Response mechanism from [ASZ19] satisfies private-recoverability, and is utility-wise order-optimal for distribution estimation and heavy hitter estimation while using an order-optimal amount of randomness.*

7.5.1 Proof of Theorem 7.5.1

This section proves the necessary and sufficient conditions on the random key U and the privatized output Y to design an ε-LDP-Rec mechanism. We first prove that $|\mathcal{Y}| \geq |\mathcal{X}|$ is necessary to recover X from Y and U. We then prove that each input $x \in \mathcal{X}$ should be mapped with non-zero probability to every output $y \in \mathcal{Y}$; hence, we get $|\mathcal{U}| \geq |\mathcal{Y}|$, since each input $x \in \mathcal{X}$ can be mapped with non-zero probability to at most $|\mathcal{U}|$ outputs. The main part of our proof is bounding the randomness of the key U in the second condition. We first prove in Lemma 7.5.2 that for any ε-LDP-Rec mechanism designed using a random key of size greater than the input size, there exists another ε-LDP-Rec mechanism designed using a random key of size equal to the input size with the same or smaller amount of randomness. Thus, we can assume that $|\mathcal{U}| = |\mathcal{X}|$ and minimize the entropy of the random key U over all possible distributions and under the ε-LDP constraint. Since entropy is a concave function of the distribution, we get a non-convex problem. However, we can obtain an exact solution for the problem due to the structure of the privacy constraints that form a closed polytope. For the sufficiency part, we prove in Lemma 7.5.1 that we can construct an ε-LDP-Rec mechanism using the random key $U_{\min}^{s^*}$ defined in Theorem 7.5.1 that satisfies the two necessary conditions.

Before we proceed into the proof of Theorem 7.5.1, we first present the following two lemmas whose proofs are given in Appendix F.5 and Appendix F.5.1, respectively.

Lemma 7.5.1. *For given a random key $U \in \mathcal{U}$ with size $|\mathcal{U}| = k$ having a distribution $\mathbf{q} = [q_1, \ldots, q_k]$ such that $\frac{q_{\max}}{q_{\min}} \leq e^{\varepsilon}$, where $q_{\max} = \max_{j \in [k]} q_j$ and $q_{\min} = \min_{j \in [k]} q_j$, there exists an ε-LDP-Rec mechanism with input $X \in [k]$ and an output $Y \in [k]$ designed using U.*

This lemma shows that we can design an ε-LDP mechanism with output size equal to the input size if we have a random key with size equal the input size and having a distribution such that $\frac{q_{\max}}{q_{\min}} \leq e^{\varepsilon}$.

Lemma 7.5.2. *Suppose that an ε-LDP-Rec mechanism with an input $X \in [k]$ and an output $Y \in \mathcal{Y}$ is designed using a random key $U \in \mathcal{U}$ with size $|\mathcal{U}| = m > k$. Then there exists an ε-LDP-Rec mechanism with an input $X \in [k]$ and an output $Y \in [k]$ designed using a random key $U' \in [k]$ such that $H(U) \geq H(U')$.*

Now, we are ready to prove Theorem 7.5.1. We prove the first necessary condition of

Theorem 7.5.1 in two parts: We can show $|\mathcal{Y}| \geq |\mathcal{X}|$ using the recoverability constraint and $|\mathcal{U}| \geq |\mathcal{Y}|$ using the privacy constraint. We prove these in Appendix F.6.

From Lemma 7.5.2 and the first necessary condition, we see that the ε-LDP-Rec mechanism with the smallest amount of randomness is obtained when $|\mathcal{U}| = |\mathcal{Y}| = |\mathcal{X}| = k$. Hence, we restrict our attention to this case only. Let $U \in [k]$ be a random key having a distribution $\mathbf{q} = [q_1, \ldots, q_k]$. Without loss of generality, we assume that $q_1 \leq q_2 \leq \ldots \leq q_k$. Before we prove the necessity of the second condition, we claim that $q_k/q_1 \leq e^{\varepsilon}$. We prove this using both privacy and recoverability constraints in Appendix F.6.

Now, we are ready to prove the necessity of the second condition. Our objective is to find the minimum entropy of the random key U with size $|\mathcal{U}| = k$ such that the private mechanism is ε-LDP and the sample X can be recovered from observing Y and the random key U. The problem can be formulated as follows

$$\min_{\mathbf{q}=[q_1,\ldots,q_k]} H(U) = -\sum_{j=1}^{k} q_j \log(q_j) \tag{7.43}$$

$$s.t., \ 1 \leq \frac{q_j}{q_1} \leq e^{\varepsilon} \ \forall j \in [k] \tag{7.44}$$

$$\sum_{j=1}^{k} q_j = 1, \ \ q_j \geq 0 \ \forall j \in [k] \tag{7.45}$$

where the constraint (7.44) is obtained from the claim proved above. Observe that the constraints (7.44)-(7.45) form a closed polytope. Furthermore, the objective function (7.43) is a concave function on $\mathbf{q}$. Since we *minimize* a concave function over a polytope, the global optimum point is one of the vertices of the polytope [Ros83]. Since we have a single equality constraint, a vertex has to satisfy at least $k-1$ inequality constraints with equality. Observe that none of the inequalities in (7.45) can be satisfied with equality, otherwise the privacy constraints in (7.44) would be violated. Thus, the optimal vertex is of the form

$$\mathbf{q} = \left[\underbrace{q_1, \ldots, q_1}_{k-s \text{ terms}}, \underbrace{e^{\varepsilon} q_1, \ldots, e^{\varepsilon} q_1}_{s \text{ terms}}\right]$$

such that s of inequalities from $\frac{q_j}{q_1} \leq e^{\varepsilon}$ are satisfied with equality and $(k-s-1)$ of inequalities from $1 \leq \frac{q_j}{q_1}$ are satisfied with equality, where s is a variable to be optimized. Hence, the optimal distribution has the form

$$\mathbf{q}^s = \left[\underbrace{q_s, \ldots, q_s}_{k-s \text{ terms}}, \underbrace{e^{\varepsilon} q_s, \ldots, e^{\varepsilon} q_s}_{s \text{ terms}}\right], \tag{7.46}$$

where $q_s = \frac{1}{se^{\varepsilon}+k-s}$, and s is an integer parameter chosen to minimize the entropy as follows

$$\begin{aligned} s^* = & \arg\min_{s \in [k]} \sum_{j=1}^{k} q_j^s \log\left(\frac{1}{q_j^s}\right) \\ = & \arg\min_{s\in[k]} \ \log\left(s\left(e^{\varepsilon}-1\right)+k\right) - \frac{s\varepsilon e^{\varepsilon}}{s\left(e^{\varepsilon}-1\right)+k} \\ = & \arg\min_{s\in[k]} \ \log\left(s\left(e^{\varepsilon}-1\right)+k\right) \\ & + \frac{\varepsilon e^{\varepsilon} k}{\left(e^{\varepsilon}-1\right)\left(s\left(e^{\varepsilon}-1\right)+k\right)} - \frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon}-1}. \end{aligned} \tag{7.47}$$

In order to solve the optimization problem (7.47), we relax the problem by assuming s is a real number taking values in $[0,k]$. The optimization problem in (7.47) is non-convex in for general values of ε and k. Thus, we get all local minima by setting the derivative to zero along with the boundary points $s \in \{0,k\}$. Then we check all these critical points to obtain the global minimum point. However, we can see that at the boundary points $s \in \{0,k\}$, the objective function is equal to $\log(k)$ which is the maximum entropy for any random variable with support size k. Hence, the optimal solution is one of the local minimums. We can verify that the objective function has only one local minimum point by setting the derivative with respect to s to zero. Thus, we get

$$\tilde{s} = k\frac{e^{\varepsilon}\left(\varepsilon-1\right)+1}{\left(e^{\varepsilon}-1\right)^2}, \tag{7.48}$$

where $\tilde{s}$ denotes the local minimum point. Since (7.47) is a continuous function in the real variable s, the optimal discrete point s^* is within the local minimum $\tilde{s}$. Hence, we get the closest integer to the real value in (7.48). As a result, we get

$$H\left(U\right) \geq H\left(U_{\min}^{s^*}\right),$$

where $s^* = \arg \min_{s \in \{\lceil l \rceil, \lfloor l \rfloor\}} H\left(U_{\min}^s\right)$ for $l = k\frac{e^\varepsilon(\varepsilon-1)+1}{(e^\varepsilon-1)^2}$, and $U_{\min}^s$ is a random variable having a distribution $\mathbf{q}^{s^*}$ given in (7.46). Hence, the proof of the necessary part is completed.

The sufficiency part is straightforward: Note that the random key $U_{\min}^{s^*}$ defined in Theorem 7.5.1 satisfies the necessary conditions, and Lemma 7.5.1, we can construct an ε-LDP-Rec mechanism using the random key $U_{\min}^{s^*}$. Thus, these conditions are sufficient.

7.5.2 Proof of Theorem 7.5.2

In this section, we show that the Hadamard response (HR) scheme proposed in [ASZ19] is, in fact, an ε-LDP-Rec mechanism, where it is possible to recover the input X from the output Y and randomness U. Furthermore, we show that it is order optimal from a randomness perspective[8].

We briefly describe the HR mechanism, and then analyze its performance. We refer to [ASZ19] for more details. The HR mechanism is parameterized by two parameters: K denotes the support size of the private mechanism output ($\mathcal{Y} = [K]$), and $s \leq K$ is a positive integer. For each $x \in \mathcal{X}$, let $\mathcal{C}_x \subseteq [K]$ be a subset of outputs of size $|\mathcal{C}_x| = s$. The private mechanism for HR is defined by

$$Q\left(y|X\right) = \begin{cases} \frac{e^\varepsilon}{se^\varepsilon+K-s} & \text{if } y \in \mathcal{C}_x \\ \frac{1}{se^\varepsilon+K-s} & \text{if } y \notin \mathcal{C}_x \end{cases} \tag{7.49}$$

We can easily show that this is a symmetric mechanism, i.e., it can be represented using a private key U of size $|K|$ that is independent of the mechanism input X. Furthermore the distribution of the private key U is given by

$$\mathbf{q}^{\text{HR}} = \left[\underbrace{q, \ldots, q}_{K-s \text{ terms}}, \underbrace{e^\varepsilon q, \ldots, e^\varepsilon q}_{s \text{ terms}}\right],$$

[8] We mention that the Hadamard mechanism in [ASZ19] is symmetric with non-binary outputs, while the Hadamard response in [AS19] has only binary outputs.

where $q = \frac{1}{se^{\varepsilon}+K-s}$. It remains to choose K, s, and $\{\mathcal{C}_x\}_{x\in\mathcal{X}}$ for fixed ε and input size $|\mathcal{X}| = k$. In [ASZ19, Section 5], the authors proposed $K = B \times b$ and $s = b/2$, where $B = 2^{\lceil \log_2(\min\{e^{\varepsilon},2k\})\rceil-1}$, and $b = 2^{\lceil \log_2\left(\frac{k}{B}+1\right)\rceil}$. Furthermore, each set $\mathcal{C}_x$ is a subset of rows indices of the Hadamard matrix. These parameters are chosen such that s is close to $\max\{\frac{k}{e^{\varepsilon}},1\}$, and K is approximately the smallest power of 2 greater than k. The reason behind using values that are powers of 2 is to exploit the structure of the Hadamard matrix. In [ASZ19, Theorem 7], the authors proved that the minimax risk of HR for ℓ_2^2 loss function is given by

$$r_{\varepsilon,n,k}^{\ell_2^2} \leq \begin{cases} \mathcal{O}\left(\frac{k}{n\varepsilon^2}\right) & \text{for } \varepsilon < 1 \\ \mathcal{O}\left(\frac{k}{ne^{\varepsilon}}\right) & \text{for } 1 \leq \varepsilon \leq \log(k) \\ \mathcal{O}\left(\frac{1}{n}\right) & \text{for } \varepsilon > \log(k) \end{cases} \tag{7.50}$$

which is order optimal for all privacy levels. In addition, the authors in [AS19] have shown that the HR scheme is order optimal for heavy hitter estimation in the high privacy regime ($\varepsilon = \mathcal{O}(1)$). In the following, we analyze the performance of HR with respect to the randomness of the private mechanism. Observe that for fixed ε and k, the parameters K, B, and b of HR is bounded by $\frac{\min\{e^{\varepsilon},2k\}}{2} \leq B \leq \min\{e^{\varepsilon},2k\}$, $\frac{k}{\min\{e^{\varepsilon},2k\}} \leq b \leq \frac{4k}{\min\{e^{\varepsilon},2k\}}$, and $k \leq K \leq 4k$. Hence, the entropy of the private key used to generate the HR private mechanism is bounded by

$$\begin{aligned} H^{\text{HR}}(U) &= \log\left(\frac{b}{2}e^{\varepsilon} + K - \frac{b}{2}\right) - \frac{\varepsilon e^{\varepsilon}\frac{b}{2}}{\frac{b}{2}e^{\varepsilon} + K - \frac{b}{2}} \\ &\leq \log\left(\frac{2k}{\min\{e^{\varepsilon},2k\}}\left(e^{\varepsilon}-1\right) + 4k\right) \\ &\qquad - \frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon} - 1 + 2\min\{e^{\varepsilon},2k\}} \\ &= \begin{cases} \log\left(2k\frac{3e^{\varepsilon}-1}{e^{\varepsilon}}\right) - \frac{\varepsilon e^{\varepsilon}}{3e^{\varepsilon}-1} & \text{if } \varepsilon \leq \log(k)+1, \\ \log\left(e^{\varepsilon}+4k-1\right) - \frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon}+4k-1} & \text{if } \varepsilon > \log(k)+1. \end{cases} \end{aligned} \tag{7.51}$$

The minimum entropy of the private key to generate an ε-LDP-Rec mechanism is bounded by (Theorem 7.5.1)

$$
\begin{aligned}
H^{\min}\left(U\right) &= \log\left(s^{*}e^{\varepsilon}+k-s^{*}\right)-\frac{\varepsilon e^{\varepsilon}s^{*}}{s^{*}e^{\varepsilon}+k-s^{*}} \\
&\geq \begin{cases} \log\left(k\left(\frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon}-1}\right)\right)-\frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon}+\frac{(e^{\varepsilon}-1)^2}{e^{\varepsilon}(\varepsilon-1)+1}-1} & \text{if } \varepsilon\leq\log\left(k\right), \\ \log\left(e^{\varepsilon}+k-1\right)-\frac{\varepsilon e^{\varepsilon}}{e^{\varepsilon}+k-1} & \text{if } \varepsilon>\log\left(k\right). \end{cases}
\end{aligned}
\tag{7.52}
$$

From (7.51) and (7.52), we can verify that HR is randomness-order-optimal for all privacy levels ε.

7.6 Sequence of Distribution (or Heavy Hitter) Estimation

We again start from the setting in Figure 7.6, but with the modification that Alice (an arbitrary user) wants to send to Bob (a legitimate analyst) T independent samples $X^T = \left(X^{(1)},\ldots,X^{(T)}\right)$, where $X^{(t)}\in\mathcal{X}$, while keeping them private against Eve (an untrusted analyst) with differential privacy level ε. Eve has access to the sequence of outputs $Y^T = \left(Y^{(1)},\ldots,Y^{(T)}\right)$ that Alice produces, but not to the random key U that Alice and Bob share. Note that each output $Y^{(t)}$ might be a function of all input samples $X_1^t = \left(X^{(1)},\ldots,X^{(t)}\right)$ and the key U. Furthermore, the output $Y^{(t)}$ can take values from a set $\mathcal{Y}^{(t)}$ that is not required to be the same as $\mathcal{Y}^{(t')}$ for $t\neq t'$. Let $\mathcal{Y}^T=\mathcal{Y}^{(1)}\times\cdots\times\mathcal{Y}^{(T)}$. The following theorem is proved in Section 7.6.1.

We can define ε-DP-Rec mechanisms in the same way as we defined ε-LDP-Rec mechanisms in Definition 7.5.1: A mechanism Q is ε-DP-Rec, if it satisfies (7.7), and allows the recovery of input X from the output Y and the key U.

Theorem 7.6.1. *Let Q be an ε-DP-Rec mechanism that uses a random key $U\in\mathcal{U}$ and an input database $X^T\in\mathcal{X}^T$ to create an output $Y^T\in\mathcal{Y}^T$. The following conditions are necessary and sufficient to allow recovery of the input X^T from (U,Y^T).*

(1) $|\mathcal{U}|\geq|\mathcal{Y}^T|\geq|\mathcal{X}^T|$.

(2) The entropy of the random key must satisfy $H\left(U\right)\geq T\min_{s^{*}\in\{\lceil l\rceil,\lfloor l\rfloor\}}H\left(U_{\min}^{s^{*}}\right)$, where $U_{\min}^{s}$ is

the same random variable with support size $|\mathcal{X}| = k$, as defined in Theorem 7.5.1.

Theorem 7.6.1 shows that the minimum amount of randomness required to preserve privacy of T samples is equal to T times the amount of randomness required to preserve privacy of a single sample. That is, for ε-DP-Rec, it is optimal to use an ε-LDP-Rec mechanism T times.

Remark 7.6.1. Observe that Theorem 7.6.1 is applicable in a n-user setting (by setting $T = n$), where user i has a single sample $X^{(i)}$, and all users have access to a shared random key U. So we have that shared randomness among users does not help in reducing the overall required amount of randomness.

7.6.1 Proof of Theorem 7.6.1

In this section, we prove Theorem 7.6.1. The main idea of our proof is as follows. The first condition is obtained in a similar manner as in the proof of Theorem 7.5.1. For the second condition, we relate the minimum amount of randomness required to preserve privacy of T samples to the minimum amount of randomness required to preserve privacy of $T-1$ samples. In particular, we prove that $H\left(U\right) \geq H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right)$, where $H\left(U_{\min,t}\right)$ is the minimum amount of randomness of a key when we have a database of t input samples.

Definition 7.6.1. Let $U \in \mathcal{U}$ be a random key drawn from a discrete distribution $\mathbf{q} = [q_1, \cdots, q_{k^T}]$ with a support size $|\mathcal{U}| = k^T$, where $q_u = \Pr\left[U = u\right]$. We say that the distribution $\mathbf{q}$ satisfies ε-DP, if there exists a bijective function $f : \mathcal{X}^T \to \left[1 : k^T\right]$ from the dataset $\mathcal{X}^T$ to integers $\left[1 : k^T\right]$, such that for every neighboring databases $\mathbf{x}, \mathbf{x}' \in [k]^T$, we have

$$\frac{q_{f(\mathbf{x})}}{q_{f(\mathbf{x}')}} \leq e^{\varepsilon}. \tag{7.53}$$

We begin our proof with the following lemma which is a generalized version of Lemma 7.5.1. We prove it in Appendix F.7.

Lemma 7.6.1. *Consider an input database $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(T)}\right) \in [k]^T$, and a random key $U \in \mathcal{U} = \{u_1, \cdots, u_{k^T}\}$ distributed according to an ε-DP distribution $\mathbf{q} = [q_1, \cdots, q_{k^T}]$.*

Then, there exists an ε-DP-Rec mechanism $Q : [k]^T \to [k]^T$ that uses U to create an output $Y^T \in [k]^T$, such that we can recover the input database X^T from (U, Y^T).

We can prove the first necessary condition of Theorem 7.6.1 (which is to show $|\mathcal{U}| \geq |\mathcal{Y}^T| \geq |\mathcal{X}^T|$) in the same way as we proved that for Theorem 7.5.1. For completeness, we provide a proof of it in Appendix F.7. Now we prove the necessity of the second condition. Consider an arbitrary ε-DP-Rec mechanism Q with output $Y^T \in \mathcal{Y}^T$ using a random key $U \in \mathcal{U}$, where $|\mathcal{Y}^T| = m \geq k^T$ and $|\mathcal{U}| = l \geq m$. Let $U \sim \mathbf{q}$, where $\mathbf{q} = [q_1, \ldots, q_l]$ such that $q_u = \Pr[U = u]$ for $u \in \mathcal{U}$. Let $\mathcal{U}_{\mathbf{yx}} \subset \mathcal{U}$ be a subset of key values such that the input $X^T = \mathbf{x}$ is mapped to $Y^T = \mathbf{y}$ when $U \in \mathcal{U}_{\mathbf{yx}}$. Thus, the private mechanism Q can be represented as

$$Q(\mathbf{y}|\mathbf{x}) = \sum_{u \in \mathcal{U}_{\mathbf{yx}}} q_u. \tag{7.54}$$

Observe that $\sum_{\mathbf{y} \in \mathcal{Y}^T} Q(\mathbf{y}|\mathbf{x}) = 1$, since $Q(\mathbf{y}|\mathbf{x})$ is a conditional distribution for any given $\mathbf{x} \in [k]^T$. Since Q is an ε-DP-Rec mechanism, it follows from the recoverability constraint that each input $\mathbf{x}$ is mapped to $\mathbf{y}$ using a different set of key values ($\mathcal{U}_{\mathbf{yx}} \bigcap \mathcal{U}_{\mathbf{yx}'} = \phi$). Thus, for each $\mathbf{y} \in \mathcal{Y}^T$, we have $s_{\mathbf{y}} = \sum_{\mathbf{x} \in [k]^T} Q(\mathbf{y}|\mathbf{x}) \leq 1$. Furthermore, we get $\sum_{\mathbf{y} \in \mathcal{Y}^T} \sum_{\mathbf{x} \in [k]^T} Q(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}^T} s_{\mathbf{y}} = k^T$.

We sort the k^T databases in $\mathcal{X}^T$ in lexicographic order by arranging them in increasing order of $x^{(1)}$. Then, we arrange the databases that have the same $x^{(1)}$ in increasing order of $x^{(2)}$ and so on. For example, database $\mathbf{x} = \left(x^{(1)}, \ldots, x^{(i)}, x^{(i+1)}, \ldots, x^{(T)}\right)$ will appear before the database $\tilde{\mathbf{x}} = \left(x^{(1)}, \ldots, x^{(i)}, \tilde{x}^{(i+1)}, \ldots, \tilde{x}^{(T)}\right)$ when $x^{(i+1)} < \tilde{x}^{(i+1)}$. Furthermore, we denote $\mathbf{x}_i$ as the ith database in the lexicographic order for $i \in [k]^T$. Observe that $s_{\mathbf{y}} = \sum_{\mathbf{x} \in [k]^T} Q(\mathbf{y}|\mathbf{x})$ for given $\mathbf{y} \in \mathcal{Y}^T$. Thus, the probabilities $\mathbf{P}^{\mathbf{y}} = \left[P_1^{\mathbf{y}}, \ldots, P_{k^T}^{\mathbf{y}}\right]$ construct a valid distribution with support size k^T, where $P_j^{\mathbf{y}} = \frac{Q(\mathbf{y}|\mathbf{x}_j)}{s_{\mathbf{y}}}$ for $j \in [k]^T$. Furthermore, for every neighboring databases $\mathbf{x}, \mathbf{x}' \in [k]^T$, we have

$$\frac{Q(\mathbf{y}|\mathbf{x})/s_{\mathbf{y}}}{Q(\mathbf{y}|\mathbf{x}')/s_{\mathbf{y}}} = \frac{Q(\mathbf{y}|\mathbf{x})}{Q(\mathbf{y}|\mathbf{x}')} \overset{(a)}{\leq} e^{\varepsilon}, \tag{7.55}$$

where step (a) follows from the fact that Q is an ε-DP-Rec mechanism. Hence, the distribution $\mathbf{P}^{\mathbf{y}}$ is ε-DP distribution. The proof of the following lemma is presented in Appendix F.8.

Lemma 7.6.2. *For every output* $\mathbf{y} \in \mathcal{Y}^T$*, we have* $H\left(\mathbf{P}^{\mathbf{y}}\right) \geq H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right),$ *where* $H\left(U_{\min,t}\right)$ *denotes the minimum randomness of a private key when we have a database of* t *samples for* $t \in \{1, \ldots, T\}$.

Using Lemma 7.6.2, we can prove Theorem 7.6.1 as follows.

$$\begin{aligned}
H\left(U\right) &= \frac{1}{k^T} \sum_{\mathbf{x}\in[k]^T} H\left(U\right) \overset{(a)}{\geq} \frac{1}{k^T} \sum_{\mathbf{x}\in[k]^T} H\left(Y^T|X^T = \mathbf{x}\right) \\
&= \frac{1}{k^T} \sum_{\mathbf{x}\in[k]^T} \sum_{\mathbf{y}\in\mathcal{Y}^T} -Q\left(\mathbf{y}|\mathbf{x}\right) \log\left(Q\left(\mathbf{y}|\mathbf{x}\right)\right) \\
&= \frac{1}{k^T} \sum_{\mathbf{y}\in\mathcal{Y}^T} \Bigg[s_{\mathbf{y}} \left(\sum_{\mathbf{x}\in[k]^T} -\frac{Q\left(\mathbf{y}|\mathbf{x}\right)}{s_{\mathbf{y}}} \log\left(\frac{Q\left(\mathbf{y}|\mathbf{x}\right)}{s_{\mathbf{y}}}\right)\right) \qquad (7.56) \\
&\qquad\qquad - s_{\mathbf{y}} \log\left(s_{\mathbf{y}}\right) \Bigg] \\
&= \frac{1}{k^T} \sum_{\mathbf{y}\in\mathcal{Y}^T} \left[s_{\mathbf{y}} H\left(\mathbf{P}^{\mathbf{y}}\right) - s_{\mathbf{y}} \log\left(s_{\mathbf{y}}\right)\right] \\
&\overset{(b)}{\geq} \frac{1}{k^T} \sum_{\mathbf{y}\in\mathcal{Y}^T} \Big[s_{\mathbf{y}} \left(H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right)\right) \qquad (7.57) \\
&\qquad\qquad - s_{\mathbf{y}} \log\left(s_{\mathbf{y}}\right) \Big] \\
&\overset{(c)}{\geq} H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right), \qquad (7.58)
\end{aligned}$$

where step (a) follows from the fact that $Q\left(\mathbf{y}|\mathbf{x}\right)$ is a function of U. Step (b) follows from Lemma 7.6.2. The inequality (c) follows from solving the problem

$$\begin{aligned}
&\min_{\{s_{\mathbf{y}}\}} \sum_{\mathbf{y}\in\mathcal{Y}^T} s_{\mathbf{y}} \left[H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right)\right] - s_{\mathbf{y}} \log\left(s_{\mathbf{y}}\right) \\
&\textit{s.t.} \sum_{\mathbf{y}\in\mathcal{Y}^T} s_{\mathbf{y}} = k^T \quad \text{and } 0 \leq s_{\mathbf{y}} \leq 1,\ \forall\ \mathbf{y} \in \mathcal{Y}^T
\end{aligned} \qquad (7.59)$$

Note that $f\left(x\right) = -x \log\left(x\right)$ is a concave function on $0 \leq x \leq 1$. Therefore, the objective function in (7.59) is concave in $\{s_{\mathbf{y}}\}$. The minimum value of a concave function is one of the vertices which is obtained when all the inequalities are satisfied by equalities. By setting k^T

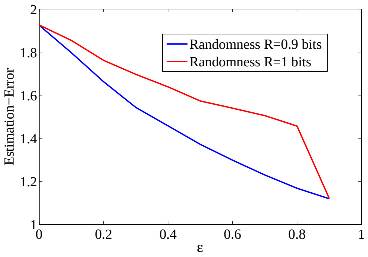


Figure 7.8: Estimation error for input alphabet size $k = 1000$, number of users $n = 500000$, and $\mathbf{p} = \text{Geo}\,(0.8)$.

of the $s_{\mathbf{y}}$'s to be one and setting the remaining $|\mathcal{Y}^T| - k^T$ of $s_{\mathbf{y}}$'s to be zero, the objective value in (7.59) becomes k_T, which gives inequality (c).

Now, from (7.58), we conclude that $H\,(U) \geq TH\,(U_{\min,1})$, where $H\,(U_{\min,1})$ is the minimum amount of randomness required to design an ε-LDP-Rec mechanism given in Theorem 7.5.1. This completes the proof of Theorem 7.6.1.

7.7 Numerical Results

In this section, we numerically validate our theoretical results through simulation.

Single-level privacy: In this part, we investigate the performance of the estimator presented in Theorem 7.3.2 for a single-level privacy. Each point is obtained by averaging over 20 runs. In Figure 7.7, we plot the estimation error for the $\ell = \ell_1$ loss function ($\|\mathbf{p} - \hat{p}\,(Y^n)\,\|_1$) for estimating a discrete distribution $\mathbf{p} \in \Delta_k$. The input size is $k = 1000$, the number of users is $n \in [10^5 : 10^6]$, and the privacy level is $\varepsilon = 1$ for two values of randomness $R \in \{0.7, 1\}$ bits per user. The input samples are drawn from a Geometric distribution with parameter $q = 0.8$ (Geo (0.8)), in which $p_i = Cq^{i-1}\,(1 - q)$ for $i \in [k]$, where C is a normalization term. Figure 7.7 shows that the number of users required to achieve a certain estimation error increases as the amount of randomness per user decreases. For instance, to achieve an ℓ_1-error equal to 1.4, we need $n \approx 150,000$ users if $R = 1$ bits per user, while we need $n \approx 850,000$

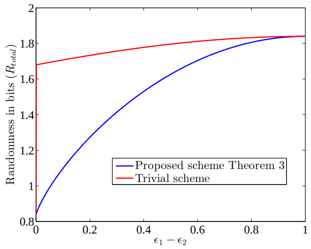


Figure 7.9: Comparison between our privacy scheme proposed in Theorem 7.4.1 and the trivial scheme for two privacy levels $\varepsilon_1 = 1$ and $\varepsilon_2 = [0.01 : 1]$.

users if $R = 0.7$ bits per user.

Figure 7.8 depicts the ℓ_1 estimation error as a function of the privacy level ε for input size $k = 1000$ and number of users $n = 500000$ for two different values of randomness $R \in \{1, 0.6\}$ bits per user. As we discussed in Theorem 1, for each privacy level ε, there is a critical point of randomness $R = H\left(e^{\varepsilon}/\left(e^{\varepsilon}+1\right)\right)$. When each user has $R < H\left(e^{\varepsilon}/\left(e^{\varepsilon}+1\right)\right)$ bits of randomness, then the ℓ_1 estimation loss increases as the randomness R decreases. While when each user has $R \geq H\left(e^{\varepsilon}/\left(e^{\varepsilon}+1\right)\right)$ bits of randomness, the estimation error is not affected by the amount of randomness R. In Figure 7.8, we find that the ℓ_1 error depends on the randomness R for all $\varepsilon < 0.8$, since we have $R = 0.9 < H\left(e^{\varepsilon}/\left(e^{\varepsilon}+1\right)\right)$ for all $\varepsilon < 0.8$.

Multi-level privacy: Figure 7.9 and Figure 7.10 compare our proposed scheme in Theorem 7.4.1 with the trivial scheme with respect to the total amount of randomness used. In the trivial scheme, each user generates d different privatized samples, one for each analyst. In Figure 7.9 we consider two privacy levels $\varepsilon_1 = 1$ and $\varepsilon_2 \leq \varepsilon_1$. We find that when $\varepsilon_1 - \varepsilon_2$ is small, then the trivial scheme requires approximately twice the total amount of randomness used in our scheme. However, when $\varepsilon_1 - \varepsilon_2$ is large, then our scheme and the trivial scheme use similar amounts of randomness. In Figure 7.10, we consider $d \in [1 : 10]$, $\varepsilon_1 = 2$ and $\varepsilon_j = \varepsilon_1 - 0.1j$, for $j \in \{2, \ldots, d\}$. We find that the gap between the amount of randomness used in our scheme and the trivial scheme increases with d.

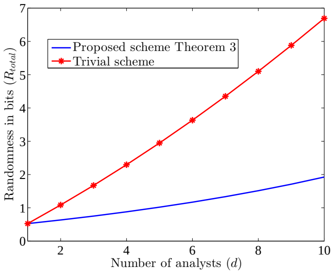


Figure 7.10: Comparison between our privacy scheme proposed in Theorem 7.4.1 and the trivial scheme for d privacy levels $\varepsilon_1 = 2$ and $\varepsilon_j = \varepsilon - 0.1j$ for $j \in [2 : d]$.

Private-recoverability: Observe that each user needs $\log(k)$ bits to store her input sample $X \in [k]$, since she does not know the distribution $X \sim \mathbf{p}$. In private-recoverability, we can recover X from observing Y and U; hence, we only need to store U. Figure 4.2 plots the number of bits required to store U (see Theorem 7.5.1) as a function of the privacy level ε and different values of input size $k \in \{10, 100, 1000\}$. The black lines represent the $\log(k)$ bits required to store X (an additional secure copy). Note that the amount of bits needed to store U is strictly smaller than $\log(k)$ for $\varepsilon > 0$, and decreases as the privacy level ε increases. Observe that the gain in Fig 4.2 is per user. Hence, the total amount of saving in storage would be considerable when the number of users is large and $\varepsilon > 0$. For example, when $\varepsilon = 5$, alphabet size $k = 2, 4, 10$, we get gain in efficiency $\frac{\log(k) - H(U)}{\log(k)}$ of 94.2%, 91.4%, and 85% respectively.

7.8 Related Work

To the best of our knowledge, the role of limited randomness has not been previously explored either in the context of local or global differential privacy.[9] In this work, we consider local

[9] Except for a notable exception of [DLM12], which showed that imperfect source of randomness allows efficient protocols with global differential privacy. This is different from our problem, where our goal is to quantify the amount of randomness required (measured in terms of Shannon entropy) in local differential privacy and give privacy-utility-randomness trade-offs.

differential privacy in the context of distribution estimation and heavy hitter estimation for reasons of simplicity.

Popular local differentially private mechanisms for distribution estimation include RAPPOR [EPK14], randomized response (RR) [War65]), subset selection (SS) [YB18, WHW16], and the Hadamard response (HR) [ASZ19]. The randomized response mechanism is known to be order optimal in the low privacy regime, and the RAPPOR scheme in the high privacy regimes [KBR16, KOV14]. Subset selection and the Hadamard mechanisms are order optimal in utility for all privacy regimes; additionally, the Hadamard mechanism has the advantage of communication and computational efficiency for all privacy regimes [ASZ19]. We build on this extensive literature, and show that the Hadamard mechanism is also near-optimal in terms of the amount of randomness used.

Heavy hitter estimation under local differential privacy has been studied in [BS15, QYY16, HKR12, BNS17, BNS18], again with unrestricted randomness. Our work adds to this line of work by showing that the Hadamard mechanism is capable of achieving order-optimal accuracy for heavy hitter estimation while using an order-optimal amount of randomness.

Local differential privacy in a multi-user setting where the users and the server may have some shared randomness has also been looked at in prior work – see [BS15, AS19, ACF18] among others. These works however investigate other orthogonal aspects of such multi-user protocols. Local differentially private mechanisms with bounded communication have also been studied by [AS19]; in their setup, multiple agents transmit their data in a locally private manner to an aggregator, and communication is measured by the number of bits transmitted by each user. They consider both private and public coin mechanisms, and show that the Hadamard mechanism is near optimal in terms of communication for both distribution and heavy-hitter estimation; however, unlike ours, their mechanisms do not impose any randomness constraints.

Our results in the multiple analyst setting are also related to privacy amplification by stochastic postprocessing [BBG19a] – which analyzes the privacy risk achieved by applying a (stochastic) post-processing mechanism to the output of a differentially private algorithm. While these methods might also be used to provide multi-level privacy to multiple analysts, our work is different from [BBG19a] in the following aspect. First, their privacy amplification methodology does not apply to pure DP and applies instead to approximate DP, while our work focuses on pure DP. Second, the work in [BBG19a] does not include a randomness constraint, and finally, a closer look at their mechanism reveals that it does not use the optimal amount of randomness.

Finally, a line of work on locally differentially private estimation considers the case when the inputs comprise of i.i.d. samples from the same distribution. [DJW18, DR19] derive lower and upper bounds for estimation under LDP in this setting – their work considers that all users observe i.i.d. samples from the same distribution, and the goal for each user is to preserve privacy of its raw sample. Our work is also different from this setting in that we focus on designing private mechanisms with finite randomness.

CHAPTER 8

Conclusion and Future Directions

In this thesis, we have studied communication-privacy-utility trade-offs for different distributed systems: distributed mean estimation, federated learning, stochastic linear bandits, and discrete distribution estimation. Furthermore, we characterize the Rényi differential privacy of the shuffled mode. This chapter is dedicated to discussion and future directions.

8.0.1 Distributed Mean Estimation

We studied the problem of distributed mean estimation under privacy and communication constraints in both the local privacy model and the multi-message shuffled model. We proposed communication-efficient and private algorithms for estimating the mean of bounded ℓ_p-norm vectors for $p \in [1, \infty]$. Furthermore, we proposed information-theoretic lower bounds on the mean squared error (MSE) for bounded ℓ_1-norm and ℓ_2-norm vectors in the local privacy models. We showed that our proposed algorithms achieve order optimal communication-privacy-utility trade-offs. We also studied distributed mean estimation under user-level local differential privacy (LDP), where each client has multiple vectors drawn i.i.d. from sub-Gaussian distribution.

In this line of work, there are multiple interesting open questions. First, our achievable algorithms are order-optimal in the sense of the minimax approach that considers the worst-case datasets. However, there is a few work studying instance-optimal algorithms for private mean estimation [MSU22, HLY21]. It is an open question to design achievable schemes that adapt automatically to the inputs in the local privacy model and the shuffled model.

Second, we proposed a user-level private mean estimation algorithm under the assumption that all clients' data are drawn i.i.d. from an unknown sub-Gaussian distribution. It is worth investigating user-level private algorithms for the heterogeneous case when each subset of clients' data is generated i.i.d. from different sub-Gaussian distributions.

8.0.2 Differentially Private Federated Learning

We proposed a communication-efficient and private optimization algorithm to solve the empirical risk minimization (ERM) in the federated learning framework in the shuffled model. We analyzed privacy-convergence trade-offs of our proposed algorithm showing that it matches the convergence of central DP algorithms. We extended the sampling scheme of our private federated learning algorithm to client-self sampling, where each client decides to contribute at each round by tossing a biased coin. Furthermore, we proposed a user-level DP algorithm for personalized federated learning based on the Bayesian approach with KL divergence regularization.

Differentially private algorithms for federated learning converge linearly with the model dimension [BST14], where most of the DP algorithms are based on privatizing the gradient of the loss function. This convergence rate becomes a bottleneck of privately training large and complex models. One of the interesting questions is to design private learning algorithms with convergence rates almost independent of the model dimension by avoiding the worst-case lower bound in [BST14].

8.0.3 Rényi Differential Privacy of The Shuffled Model

We characterized the Rényi differential privacy (RDP) of the shuffled model for a general ε_0-LDP mechanism by proposing a closed-form upper and lower bounds. Furthermore, we characterized the RDP of the subsampled shuffled model that combines privacy amplification via shuffling and privacy amplification by subsampling. To achieve these results, we proposed a novel analysis technique by reducing any general neighboring datasets to special case neighboring datasets that can be analyzed in a closed-form solution.

Our analysis for privacy amplification via shuffling is dedicated to pure LDP mechanisms. An open question for extending our work is how to get an overall RDP guarantee if we are given local RDP guarantees instead of pure LDP guarantees. This extension is non-trivial and it appears in many applications including federated learning with local Gaussian mechanism.

8.0.4 Differentially Private Stochastic Linear Bandits

We proposed differential privacy algorithms for stochastic linear bandits in the central privacy model, the local privacy model, and the shuffled model. Our algorithms are based on privatizing batched algorithms for stochastic linear bandits. We show that the regret of our proposed algorithms almost matches the regret of non-private stochastic bandits algorithms, and hence, we get privacy for free. Furthermore, we extend our proposed algorithms for stochastic linear bandits with known context distribution under joint differential privacy constraints.

In this line of work, there are multiple interesting open questions. First, we extend our algorithms for contextual linear bandits with adversarial context. The best-known achievable differentially private regret is order $\mathcal{O}\left(\frac{\sqrt{T}}{\varepsilon}\right)$ in [SS18] for central DP model. For the local DP model, the best known regret for contextual bandits is $\mathcal{O}\left(\frac{T^{3/4}}{\varepsilon0}\right)$ in [ZCH20]. It is still an open question whether these regret bounds are tight or we can achieve a similar regret as ours in the adversarial context.

8.0.5 Privacy-Utility-Randomness Trade-offs

We study successive refinement of privacy by providing multiple privacy levels when analysts have different levels of authorized access. Furthermore, we answered a fundamental question in LDP about how much randomness do we need to achieve a desired level of privacy and utility. We characterized the trade-off between randomness and utility for a fixed privacy level ε_0, by proving an information-theoretic lower bound and a matching upper bound for a minimax private estimation problem. In addition, we proposed a non-trivial scheme for providing multi-level privacy that uses a smaller amount of randomness with no sacrifice in utility.

In our analysis, we focus on studying privacy-utility-randomness for discrete distribution estimation and frequency estimation. An interesting open question is to study the fundamental privacy-utility-randomness for other estimation problems, e.g., mean estimation or real scalars.

APPENDIX A

Omitted Details From Chapter 2

A.1 Proof of Lemma 2.1.5

The proof is obtained from Lemma 2.1.3, where the ε is bounded by:

$$\varepsilon \leq \min_{\alpha} \rho\alpha + \frac{\log(1/\delta)}{\alpha - 1} + \log\left(1 - \frac{1}{\alpha}\right), \tag{A.1}$$

for given $\delta \in (0, 1)$. By setting $\alpha = 1 + \sqrt{\frac{\log(1/\delta)}{\rho}}$, we get that:

$$\begin{aligned}
\varepsilon &\leq \rho + 2\sqrt{\rho \log(1/\delta)} \\
&\leq \rho \log(1/\delta) + 2\sqrt{\rho \log(1/\delta)} \\
&\leq 3 \max\left\{\rho \log(1/\delta), \sqrt{\rho \log(1/\delta)}\right\}.
\end{aligned} \tag{A.2}$$

This completes the proof of Lemma 2.1.5.

A.2 Proof of Theorem 2.4.1

First, we show that the output of Algorithm 2.2.1 is unbiased estimate of b. Let y be the output of the *2RR* Algorithm 2.2.1. Then, we have

$$\begin{aligned}
\mathbb{E}\left[y\right] &= \frac{b-p}{1-2p}(1-p) + \frac{1-b-p}{1-2p}p \\
&= b\left(\frac{1-2p}{1-2p}\right) - \frac{p(1-p)}{1-2p} + \frac{p(1-p)}{1-2p} \\
&= b.
\end{aligned} \tag{A.3}$$

Hence, the Algorithm 2.2.1 is an unbiased estimate of the input b. Furthermore, the MSE of the *2RR* is bounded by:

$$\begin{aligned}
\mathsf{MSE}^{2RR} &= \mathbb{E}\left[\|y-b\|^2\right] = \mathbb{E}\left[y^2\right] - b^2 \\
&= \frac{1}{(1-2p)^2}\left[(b-p)^2(1-p) + (1-b-p)^2 p\right] - b^2 \\
&= \frac{1}{(1-2p)^2}\left[b^2 - 4p(1-p)b + p(1-p)\right] - b^2 \\
&= \frac{1}{(1-2p)^2}\left[b^2 - 4p(1-p)b + p(1-p)\right] - b^2 \\
&= \frac{1}{(1-2p)^2}\left[b^2(4p(1-p)) - 4p(1-p)b + p(1-p)\right] \\
&= \frac{p(1-p)}{(1-2p)^2}.
\end{aligned} \tag{A.4}$$

The LDP guarantees of the *2RR* is obtained from the fact that $e^{-\varepsilon_0} \leq 1 \leq \frac{1-p}{p} \leq e^{\varepsilon_0}$ for any $p \in (0, 1/2]$. Furthermore, we can prove that the *2RR* satisfies $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon\left(\alpha\right)$ is given by:

$$\varepsilon\left(\alpha\right) = \frac{1}{\alpha-1}\log\left(p^{\alpha}(1-p)^{1-\alpha} + p^{1-\alpha}(1-p)^{\alpha}\right), \tag{A.5}$$

where this bound is obtained from the definition of the RDP and also given in [Mir17]. This completes the proof of Theorem 2.4.1.

A.3 Proof of Lemma 2.4.1

From Theorem 2.4.1, the *2RR* mechanism with parameter $p < 1/2$ is ε_0-LDP, where $\varepsilon_0 = \log\left(\frac{1-p}{p}\right)$. Hence, it is sufficient to prove that $\varepsilon_0 = \log\left(\frac{1-p}{p}\right) \leq v$ when choosing $p = \frac{1}{2}\left(1 - \sqrt{\frac{v^2}{v^2+4}}\right)$ for any $v \geq 0$.

Observe that $1 - p = \frac{1}{2}\left(1 + \sqrt{\frac{v^2}{v^2+4}}\right)$ when $p = \frac{1}{2}\left(1 - \sqrt{\frac{v^2}{v^2+4}}\right)$. Let $f(v) = v - \log\left(\frac{\sqrt{v^2+4}+v}{\sqrt{v^2+4}-v}\right)$. We have that

$$\begin{aligned}\frac{\partial f}{\partial v} &= 1 - \frac{\sqrt{v^2+4}-v}{\sqrt{v^2+4}+v}\frac{8}{\left(\sqrt{v^2+4}-v\right)^2\sqrt{v^2+4}} \\ &= 1 - \frac{8}{(v^2+4-v^2)\sqrt{v^2+4}} \\ &= 1 - \frac{2}{\sqrt{v^2+4}} \\ &\geq 0 \qquad \forall\; v \geq 0.\end{aligned} \tag{A.6}$$

Hence the function $f(v)$ is a non-decreasing function for all $v \geq 0$. As a result $f(v) \geq f(0) = 0$ for all $v \geq 0$. Thus, we have $v \geq \log\left(\frac{1-p}{p}\right)$ for all $v \geq 0$. This completes the proof of Lemma 2.4.1.

APPENDIX B

Omitted Details From Chapter 3

B.1 Proof of Lemma 3.6.1

The proof is straightforward from the proof of Duchi and Rogers [DR19, Corollary 3]. In their setting, $\mathcal{P}_{p,d}^{\text{Bern}}$ is supported on $\{0,1\}^d$, and they proved a lower bound of $\Omega\left(\min\left\{1, \frac{d}{n\min\{\varepsilon_0,\varepsilon_0^2\}}\right\}\right)$. In our setting, since $\mathcal{P}_{p,d}^{\text{Bern}}$ is supported on $\left\{0, \frac{1}{d^{1/p}}\right\}^d$, we can simply scale the elements in the support of $\mathcal{P}_{p,d}^{\text{Bern}}$ by a factor of $1/d^{1/p}$, which will also scale the mean $\boldsymbol{\mu_q}$ by the same factor. Note that the best estimator $\widehat{\boldsymbol{x}}$ will be equal to the scaled version of the best estimator from [DR19, Corollary 3] with the same value $1/d^{1/p}$. This proves Lemma 3.6.1.

B.2 Proof of Lemma 3.4.1

We show the properties one-by-one below.

1. Observe that the output of the mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$ can be represented using the index $j \in [d]$ and one bit of the sign of $\{\pm a\mathbf{H}_d\left(j\right)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)\}$. 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 $\mathcal{R}_{\varepsilon_0}^{\ell_1}$

is ε_0-LDP. Let $\mathcal{Z} = \left\{ \pm a\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) : j = 1, 2, \ldots, d\right\}$ denote all possible $2d$ outputs of the mechanism $\mathcal{R}_{\varepsilon_0}^{\ell_1}$. We get

$$\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{B}_1^d(a)} \sup_{\boldsymbol{z}\in\mathcal{Z}} \frac{\Pr[\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x}) = \boldsymbol{z}]}{\Pr[\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x}') = \boldsymbol{z}]} \leq \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathbb{B}_1^d(r_1)} \frac{\frac{1}{d}\sum_{j=1}^d \left(\frac{1}{2} + \frac{\sqrt{d}|\boldsymbol{y}[j]|}{2r_1}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\right)}{\frac{1}{d}\sum_{j=1}^d \left(\frac{1}{2} - \frac{\sqrt{d}|\boldsymbol{y}'[j]|}{2r_1}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\right)}$$

$$= \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathbb{B}_1^d(r_1)} \frac{\frac{1}{d}\sum_{j=1}^d \left(r_1(e^{\varepsilon_0}+1) + \sqrt{d}|\boldsymbol{y}[j]|(e^{\varepsilon_0}-1)\right)}{\frac{1}{d}\sum_{j=1}^d \left(r_1(e^{\varepsilon_0}+1) - \sqrt{d}|\boldsymbol{y}'[j]|(e^{\varepsilon_0}-1)\right)}$$

$$\overset{\text{(a)}}{\leq} \frac{2r_1e^{\varepsilon_0}}{2r_1} = e^{\varepsilon_0},$$

where (a) uses the fact that for every $j \in [d]$, we have $|\boldsymbol{y}[j]| \leq {}^{r_1}/_{\sqrt{d}}$ and $|\boldsymbol{y}'[j]| \leq {}^{r_1}/_{\sqrt{d}}$.

2. Fix an arbitrary $\boldsymbol{x} \in \mathbb{B}_1^d(r_1)$.

$$\text{Unbiasedness:} \quad \mathbb{E}\left[\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x})\right] = \frac{1}{d}\sum_{j=1}^d r_1\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)\left(\frac{\sqrt{d}\boldsymbol{y}[j]}{r_1}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\right)$$

$$= \frac{1}{d}\sum_{j=1}^d \mathbf{H}_d(j)\sqrt{d}\boldsymbol{y}[j] \overset{\text{(b)}}{=} \frac{1}{d}\sum_{j=1}^d \mathbf{H}_d(j)\mathbf{H}_d^T(j)\boldsymbol{x} \overset{\text{(c)}}{=} \boldsymbol{x}$$

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

$$\text{Bounded variance:} \quad \mathbb{E}\|\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq \mathbb{E}\|\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x})\|^2 = \mathbb{E}[\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x})^T\mathcal{R}_{\varepsilon_0}^{\ell_1}(\boldsymbol{x})]$$

$$= \frac{1}{d}\sum_{j=1}^d a^2\mathbf{H}_d(j)^T\mathbf{H}_d(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)^2$$

$$= r_1^2 d\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)^2$$

(Since $\mathbf{H}_d(j)^T\mathbf{H}_d(j) = d, \forall j \in [d]$)

This completes the proof of Lemma 3.4.1.

B.3 Proof of Lemma 3.7.1

In order to prove that $\mathsf{Range}_{\text{scalar}}(\mathcal{D}, \tau, \varepsilon_0)$ is user-level ε_0-LDP, it suffices to show that $\mathsf{Range}_{\text{scalar}}^{\text{user}}(\mathcal{D}, \tau, \varepsilon_0)$ is user-level ε_0-LDP. Consider an arbitrary user $i \in [n]$ and two local datasets $\mathcal{D}_i = (x_1^{(i)},\ldots,x_m^{(i)})$, $\mathcal{D}'_i = (x_1^{'(i)},\ldots,x_m^{'(i)})$. Let $\mathcal{Z} = \{\pm\mathbf{H}_k(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right) : j \in \{1,\cdots,k\}\}$ denote all possible outputs of the mechanism R_{range}. Thus, we get

$$\begin{aligned}
\sup_{\mathcal{D}_i,\mathcal{D}'_i\in[-B,B]^m}\sup_{z\in\mathcal{Z}}\frac{\Pr\left[\mathsf{Range}_{\text{scalar}}\left(\mathcal{D}_i\right)=z\right]}{\Pr\left[\mathsf{Range}_{\text{scalar}}\left(\mathcal{D}'_i\right)=z\right]} &\leq \sup_{\mathcal{D}_i,\mathcal{D}'_i\in[-B,B]^m}\frac{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{2}+\frac{\sqrt{k}|\mathbf{m}_i(j)|}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{2}-\frac{\sqrt{k}|\mathbf{m}'_i(j)|}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}} \\
&\overset{\text{(a)}}{\leq} \frac{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{2}+\frac{1}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{2}-\frac{1}{2}\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}} \\
&\leq e^{\varepsilon_0}
\end{aligned}\tag{B.1}$$

where the step (a) is obtained from the fact that $\mathbf{m}_i(j), m'_i(j) \in \{\pm\frac{1}{\sqrt{k}},\}$. Thus, the private range mechanism $\mathsf{Range}_{\text{scalar}}^{\text{user}}$ is user level $(\varepsilon_0, 0)$-LDP.

Now, suppose that $\{x_j^{(i)}\}$ are σ^2 sub-Gaussian. Thus, $y^n = (y_1,\ldots,y_n)$ are (τ,γ)-concentrated, where $y_i = \frac{1}{m}\sum_{j=1}^{m}x_j^{(i)}$ and $\tau = \sigma\sqrt{\frac{\log(2n/\gamma)}{m}}$ (e.g., see [RH15, Theorem 1.14]). We show that with probability $1-\beta$, we have $y_i \in [a,b]$ for all $i \in [n]$, where $[a,b] \leftarrow \mathsf{Range}_{\text{scalar}}\left(\mathcal{D},\tau,\varepsilon_0,\delta\right)$. Condition on the event that $y^n = (y_1,\ldots,y_n)$ are concentrated with radius τ. Hence, there exists $y_0 \in [-B,B]$ such that $|y_i - y_0| \leq \tau$ for all $i \in [n]$. In Algorithm 3.7.1, we split the interval $[-B,B]$ into $T = \frac{B}{\tau}$ interval each with width 2τ, where $\mathcal{T}$ denotes the set of middle points of intervals. For each $i \in [n]$, let $\nu_i = \arg\min_{a\in\mathcal{T}}|y_i - a|$ be the closest bin in $\mathcal{T}$ to the exact value y_i. We define $f(a) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(\nu_i = a\right)$ as the fraction (frequency) of elements in y^n that are close to the bin a for each bin $a \in \mathcal{T}$. Observe that when y^n are concentrated with radius τ, we expect that $f(a) = 0$ for all $a \in \mathcal{T}$ except two adjacent bins.

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

$$\begin{aligned}
\mathbb{E}\left[\mathbf{z}_i\right] &= \frac{1}{d}\sum_{j=1}^{k}\mathbf{H}_k(j)\left(\frac{e^{\varepsilon_0}+1}{e^{\varepsilon_0}-1}\right)\left[\sqrt{k}\mathbf{m}(j)\frac{e^{\varepsilon_0}-1}{e^{\varepsilon_0}+1}\right] \\
&= \frac{1}{d}\sum_{j=1}^{k}\mathbf{H}_k(j)\sqrt{k}\mathbf{m}_i(j) \\
&\overset{\text{(a)}}{=} \frac{1}{d}\sum_{j=1}^{k}\mathbf{H}_k(j)\mathbf{H}_k^T(j)e_{\nu_i} \overset{\text{(b)}}{=} e_{\nu_i},
\end{aligned}\tag{B.2}$$

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

Observe that $\overline{\mathbf{z}}(j)$ is a sum of i.i.d. Bernoulli random variables for $j \in [k]$. Thus, $\overline{\mathbf{z}}(j)$ is a sub-Gaussian with proxy $\frac{4\left(e^{\frac{\varepsilon}{0}^2}+1\right)^2}{n\left(e^{\frac{\varepsilon}{0}^2}-1\right)^2}$ and $\mathbb{E}[\overline{\mathbf{z}}(j)] = f(a_j)$. Hence, from [RH15, Theorem 1.14], we get that

$$\Pr[\max_{j\in[k]} |\overline{\mathbf{z}}(j) - f(a_j)| > t] \leq 2k \exp\left(-\frac{t^2 n \left(e^{\frac{\varepsilon}{0}^2} - 1\right)^2}{8\left(e^{\frac{\varepsilon}{0}^2} + 1\right)^2}\right) \tag{B.3}$$

By setting $t = \frac{1}{5}$, with probability at least $1 - 2k \exp\left(-\frac{n\left(e^{\frac{\varepsilon}{0}^2}-1\right)^2}{200\left(e^{\frac{\varepsilon}{0}^2}+1\right)^2}\right)$, we get

$$\max_{j\in[k]} |\overline{\mathbf{z}}(j) - f(a_j)| \leq \frac{1}{5}. \tag{B.4}$$

With probability $1 - \gamma$, since there are only two adjacent bins of non-zero frequencies, one of them has a frequency $f(a) \geq \frac{1}{2}$. Let $a_{\max}$ be the bin that has the maximum estimated frequency. Conditioned on the event (B.4), the $a_{\max}$ will be equal one of these two non-zero bins that has non-zero frequencies. This can be seen as follows: Let $j_1, j_2 \in [k]$ be such that $f(a_{j_1}), f(a_{j_2}) > 0$ and we know that one of them, say, j_1, has $f(a_{j_1}) \geq \frac{1}{2}$. Since $a_{\max} = \arg\max_{j\in[k]} \overline{z}(j)$, by (B.4), we have $\overline{z}(j_1), \overline{z}(j_2) \in [\frac{3}{10}, \frac{7}{10}]$ and $\overline{z}(j_l) < \frac{1}{5}, \forall l \in [k] \setminus \{j_1, j_2\}$. Hence, $a_{\max} \in \{j_1, j_2\}$.

This implies that each y_i lies within 3τ of $a_{\max}$. Thus, from union bound we conclude that $y_i \in [a_{\max} - 3\tau, a_{\max} + 3\tau]$ for all $i \in [n]$ with probability at least $1 - \beta$. This completes the proof of Lemma 3.7.1.

APPENDIX C

Omitted Details From Chapter 4

C.1 Proof of Lemma 4.3.1

Recall that the input dataset at client $i \in [n]$ is denoted by $\mathcal{D}_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\} \in \mathcal{X}^m$ and $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ denotes the entire dataset. Recall from (4.5) that the mechanism $\mathcal{M}_t$ on input dataset $\mathcal{D}$ can be defined as:

$$\mathcal{M}_t(\mathcal{D}) = \mathcal{H}_{ks} \circ \text{samp}_{n,k}\left(\mathcal{G}_1, \ldots, \mathcal{G}_n\right), \tag{C.1}$$

where $\mathcal{G}_i = \text{samp}_{m,s}\left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{im}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [m], j \in [m]$. 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 [BBG19d, Corollary 5.3.1] (also see Lemma 2.3.1), the mechanism $\mathcal{Z}$ is $(\tilde{\varepsilon}, \tilde{\delta})$-DP, where $\tilde{\delta} > 0$ is arbitrary, and, if $\varepsilon_0 \leq \frac{\log\left(ks/\log\left(1/\tilde{\delta}\right)\right)}{2}$, then

$$\tilde{\varepsilon} = \mathcal{O}\left(\min\{\varepsilon_0, 1\} e^{\varepsilon_0} \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{ks}}\right). \tag{C.2}$$

Furthermore, when $\varepsilon_0 = \mathcal{O}(1)$, we get $\tilde{\varepsilon} = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{ks}}\right)$.

Let $\mathcal{T} \subseteq \{1, \ldots, n\}$ denote the identities of the k clients chosen at iteration t, and for $i \in \mathcal{T}$, let $\mathcal{T}_i \subseteq \{1, \ldots, m\}$ denote the identities of the s data points chosen at client i at iteration

t.[1] For any $\mathcal{T} \in \binom{[n]}{k}$ and $\mathcal{T}_i \in \binom{[m]}{s}, i \in \mathcal{T}$, define $\overline{\mathcal{T}} = (\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T})$, $\mathcal{D}^{\mathcal{T}_i} = \{d_j : j \in \mathcal{T}_i\}$ 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, [Ull17]) – which is for uniform sampling – to compute the privacy parameters of the mechanism $\mathcal{M}_t$, where sampling is non-uniform. Define a dataset $\mathcal{D}' = (\mathcal{D}'_1) \bigcup (\cup_{i=2}^{n} \mathcal{D}_i) \in \mathcal{X}^{(mn)}$, where $\mathcal{D}'_1 = \{d'_{11}, d_{12}, \ldots, d_{1m}\}$ 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 $(\overline{\varepsilon}, \overline{\delta})$-DP, we need show that for an arbitrary subset $\mathcal{S}$ of the range of $\mathcal{M}_t$, we have

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] \leq e^{\overline{\varepsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + \overline{\delta} \tag{C.3}$$

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] \leq e^{\overline{\varepsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] + \overline{\delta} \tag{C.4}$$

Note that both (C.3) and (C.4) are symmetric, so it suffices to prove only one of them. We prove (C.3) below.

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

$$\begin{aligned}
A_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1\right] \\
A'_{11} &= \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \in \mathcal{T}_1\right] \\
A_{10} &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T} \text{ and } 1 \notin \mathcal{T}_1\right] \\
A_0 &= \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right] = \Pr\left[\mathcal{Z}(\mathcal{D}'^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right]
\end{aligned} \tag{C.5}$$

[1] 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.

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

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

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

$$A_{11} \leq e^{\tilde{\varepsilon}} A'_{11} + \tilde{\delta} \tag{C.6}$$

$$A_{11} \leq e^{\tilde{\varepsilon}} A_{10} + \tilde{\delta} \tag{C.7}$$

Here (C.6) is straightforward, but proving (C.7) requires a combinatorial argument, which we give at the end of this proof. We prove (C.3) separately for two cases, first when $s = 1$ and other when $s > 1$; k is arbitrary in both cases.

C.1.1 For $s = 1$ and arbitrary $k \in [n]$

Since the mechanism $\mathcal{Z}$ is $(\tilde{\varepsilon}, \tilde{\delta})$-DP, in addition to (C.6)-(C.7), since $s = 1$, we also have the following inequality:

$$A_{11} \leq e^{\tilde{\varepsilon}} A_0 + \tilde{\delta} \tag{C.8}$$

Similar to (C.7), proving (C.8) requires a combinatorial argument, which we will give at the end of this proof. Note that (C.8) only holds for $s = 1$ and may not hold for arbitrary s. Inequalities (C.6)-(C.8) together imply $A_{11} \leq e^{\tilde{\varepsilon}} \min\{A'_{11}, A_{10}, A_0\} + \tilde{\delta}$. Now we prove (C.3) for $\overline{\varepsilon} = \ln(1 + q(e^{\tilde{\varepsilon}} - 1)$ and $\overline{\delta} = q\tilde{\delta}$. Note that when $s = 1$, we have $q_1 = \frac{k}{n}$, $q_2 = \frac{1}{m}$, and $q = \frac{k}{mn}$.

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

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

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 $\tilde{\varepsilon}$ from (C.2) and using $ks = qmn$, we get that for $\varepsilon_0 = \mathcal{O}\left(1\right)$, we have $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{q\log\left(1/\tilde{\delta}\right)}{mn}}\right)$.

C.1.2 For $s > 1$ and arbitrary $k \in [m]$

Note that (C.6)-(C.7) together imply $A_{11} \leq e^{\tilde{\varepsilon}}\min\{A'_{11}, A_{10}\} + \tilde{\delta}$. Now we prove (C.3) for $\overline{\varepsilon} = \ln(1 + q_2(e^{\tilde{\varepsilon}} - 1))$ and $\overline{\delta} = q\tilde{\delta}$.

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

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 $\tilde{\varepsilon}$ from (C.2) and using $ks = qmn$, we get that for $\varepsilon_0 = \mathcal{O}(1)$, we have $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{q_2 \log\left(1/\tilde{\delta}\right)}{q_1 mn}}\right)$. 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 4.3.1 is complete, except for that we have to prove (C.7) and (C.8). Before proving (C.7) and (C.8), we state an important remark about the privacy amplification in both the cases.

Remark C.1.1. Note that when $s = 1$ and $\varepsilon_0 = \mathcal{O}(1)$, we have $\overline{\varepsilon} = \ln(1 + q(e^{\tilde{\varepsilon}} - 1)) = \mathcal{O}(q\tilde{\varepsilon})$. So we get a privacy amplification by a factor of $q = \frac{ks}{mn}$ – 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 $\varepsilon_0 = \mathcal{O}(1)$, we have $\overline{\varepsilon} = \ln(1 + q_2(e^{\tilde{\varepsilon}} - 1)) = \mathcal{O}(q_2\tilde{\varepsilon})$, which, unlike the case of $s = 1$, only gives the privacy amplification by a factor of $q_2 = \frac{s}{m}$ – 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 $k = n$ and any $s \in [m]$ (which implies $q_1 = 1$ and $q = q_2$), we have $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0\sqrt{\frac{q_2 \log\left(1/\tilde{\delta}\right)}{mn}}\right)$, which gives the desired amplification when we select all the clients in each iteration.

Proof of (C.7)**.** First note that the number of subsets $\mathcal{T}_1 \subset [m]$ such that $|\mathcal{T}_1| = s, 1 \in \mathcal{T}_1$ is equal to $\binom{m-1}{s-1}$ and the number of subsets $\mathcal{T}_1 \subset [m]$ such that $|\mathcal{T}_1| = s, 1 \notin \mathcal{T}_1$ is equal to $\binom{m-1}{s}$. It is easy to verify that $(m-s)\binom{m-1}{s-1} = s\binom{m-1}{s}$.

Consider the following bipartite graph $G = (V_1 \cup V_2, E)$, where the left vertex set V_1 has $\binom{m-1}{s-1}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [m]$ such that $|\mathcal{T}_1| = s, 1 \in \mathcal{T}_1$, the right vertex set V_2 has $\binom{m-1}{s}$ vertices, one for each configuration of $\mathcal{T}_1 \subset [m]$ 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 $(m-s)$ neighbors in V_2 – the neighbors of $\mathcal{T}_1 \in V_1$ will be $\{(\mathcal{T}_1 \setminus \{1\}) \cup \{i\} : i \in [n] \setminus \mathcal{T}_1\} \subset V_2$. Similarly, each vertex of V_2 has s neighbors in V_1 – the neighbors of $\mathcal{T}_1 \in V_2$ will be $\{(\mathcal{T}_1 \setminus \{i\}) \cup \{1\} : i \in \mathcal{T}_1\} \subset V_1$.

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

$$\begin{aligned} &\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T}, \mathcal{T}_1 = \boldsymbol{u}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\}\right] \\ &\quad \leq e^{\tilde{\varepsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \in \mathcal{T}, \mathcal{T}_1 = \boldsymbol{v}, \mathcal{T}_i, i \in \mathcal{T} \setminus \{1\}\right] + \tilde{\delta}. \end{aligned} \tag{C.9}$$

Now we are ready to prove (C.7).

$$\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{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_1 \in \binom{[m]}{s} : 1 \in \mathcal{T}_1 \\ \mathcal{T}_i \in \binom{[m]}{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}_n] \\ &\overset{\text{(a)}}{=} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[m]}{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}] \\ &\times \sum_{\mathcal{T}_1 \in \binom{[m]}{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}_n] \\ &= \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[m]}{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}] \\ &\times \frac{1}{(m-s)\binom{m-1}{s-1}} \sum_{\mathcal{T}_1 \in \binom{[m]}{s} : 1 \in \mathcal{T}_1} (m-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n] \\ &= \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[m]}{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}] \\ &\times \frac{1}{s\binom{m-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[m]}{s} : 1 \in \mathcal{T}_1} (m-s) \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n] \\ &\overset{\text{(b)}}{\leq} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in \binom{[m]}{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}] \\ &\times \frac{1}{s\binom{m-1}{s}} \sum_{\mathcal{T}_1 \in \binom{[m]}{s} : 1 \notin \mathcal{T}_1} s\left(e^{\tilde{\varepsilon}} \Pr[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_1, \ldots, \mathcal{T}_n] + \tilde{\delta}\right) \end{aligned}$$

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

Here, (a) and (c) follow from the fact that clients sample the data points independent of each other, and (b) follows from (C.9) together with the fact that there are $(m-s)\binom{m-1}{s-1}=s\binom{m-1}{s}$ edges in the bipartite graph $G=(V_1\cup V_2,E)$, where degree of vertices in V_1 is $(m-s)$ and degree of vertices in V_2 is s.

Proof of (C.8)**.** First note that the number of subsets $\mathcal{T}\in[m]$ such that $|\mathcal{T}|=k, 1\in\mathcal{T}$ is equal to $\binom{n-1}{k-1}$ and the number of subsets $\mathcal{T}\subset[m]$ such that $|\mathcal{T}|=k, 1\notin\mathcal{T}$ is equal to $\binom{n-1}{k}$. It is easy to verify that $(n-k)\binom{n-1}{k-1}=k\binom{n-1}{k}$.

Consider the following bipartite graph $G=(V_1\cup V_2,E)$, where the left vertex set V_1 has $\binom{n-1}{k-1}m^{k-1}$ vertices, one for each configuration of $(\mathcal{T},\mathcal{T}_i:i\in\mathcal{T})$ such that $\mathcal{T}\subset[n]$, $|\mathcal{T}|=k, 1\in\mathcal{T}$ and $\mathcal{T}_1=1$, the right vertex set V_2 has $\binom{n-1}{k}m^k$ vertices, one for each configuration of $(\mathcal{T},\mathcal{T}_i:i\in\mathcal{T})$ such that $\mathcal{T}\subset[n]$, $|\mathcal{T}|=k, 1\notin\mathcal{T}$, 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 $m(n-k)$ 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 $\mathcal{T}\in\binom{[n]}{k}$ with $1\in\mathcal{T}$ and $\mathcal{T}_i\in[n], i\in\mathcal{T}$ such that $\boldsymbol{u}=(\mathcal{T},\mathcal{T}_i:i\in\mathcal{T})$ and $\mathcal{T}'\in\binom{[n]}{k}$ with $1\notin\mathcal{T}'$ and $\mathcal{T}_i'\in[n], i\in\mathcal{T}'$ 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 $(\tilde{\varepsilon}, \tilde{\delta})$-DP, we have

$$\Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | \mathcal{T}, \mathcal{T}_i, i \in \mathcal{T}\right] \leq e^{\tilde{\varepsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}'}}) \in \mathcal{S} | \mathcal{T}', \mathcal{T}_i', i \in \mathcal{T}'\right] + \tilde{\delta}. \tag{C.10}$$

Now we are ready to prove (C.8).

$$\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{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [n] \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{n-1}{k-1} m^{k-1}} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [n] \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}{(n-k)\binom{n-1}{k-1} m^{k}} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [n] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} m(n-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{n-1}{k} m^{k}} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \in \mathcal{T} \\ \mathcal{T}_i \in [n] \text{ for } i \in \mathcal{T} : \mathcal{T}_1 = 1}} m(n-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{n-1}{k} m^{k}} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [n] \text{ for } i \in \mathcal{T}}} k \left(e^{\varepsilon} \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{n-1}{k} m^{k}} \sum_{\substack{\mathcal{T} \in \binom{[n]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [n] \text{ for } i \in \mathcal{T}}} \left(e^{\varepsilon} \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{[n]}{k} : 1 \notin \mathcal{T} \\ \mathcal{T}_i \in [m] \text{ for } i \in \mathcal{T}}} \Pr[\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T} | 1 \notin \mathcal{T}] \left(e^{\varepsilon} \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{\varepsilon}} \Pr\left[\mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}}) \in \mathcal{S} | 1 \notin \mathcal{T}\right] + \tilde{\delta} \\
&= e^{\tilde{\varepsilon}} A_0 + \tilde{\delta}
\end{aligned}$$

Here, (a) uses $(n-k)\binom{n-1}{k-1} = k\binom{n-1}{k}$, and (b) follows from (C.10) together with the fact that there are $m(n-k)\binom{n-1}{k-1}m^{k-1} = k\binom{n-1}{k}m^k$ edges in the bipartite graph $G = (V_1 \cup V_2, E)$, where degree of vertices in V_1 is $m(n-k)$ and degree of vertices in V_2 is k.

This completes the proof of Lemma 4.3.1.

C.2 Proof of Lemma 4.4.1

Recall that the input dataset at client $i \in [n]$ is denoted by $\mathcal{D}_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\} \in \mathcal{X}^m$ and $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ denotes the entire dataset. Fix a time slot $t \in [T]$. Let $K_t = |\mathcal{U}_t|$ denote the random variable corresponding to the number of clients participating in the t'th time slot. Recall from (4.11) that the mechanism $\mathcal{M}_t$ on input dataset $\mathcal{D}$ can be defined as:

$$\mathcal{M}_t(\theta_t; \mathcal{D}) = \mathcal{H}_{K_t} \circ \text{samp}_{n,q}^{\text{iid}} \left(\mathcal{G}_1, \ldots, \mathcal{G}_n\right), \tag{C.11}$$

where $\mathcal{G}_i = \text{samp}_{m,1}^{\text{fix}} \left(\mathcal{R}(\boldsymbol{x}_{i1}^t), \ldots, \mathcal{R}(\boldsymbol{x}_{im}^t)\right)$ and $\boldsymbol{x}_{ij}^t = \nabla_{\theta_t} f(\theta_t; d_{ij}), \forall i \in [n], j \in [m]$. We define a mechanism $\mathcal{Z}\left(\mathcal{D}^{(t)}\right) = \mathcal{H}_{K_t}\left(\mathcal{R}\left(\boldsymbol{x}_1^t\right), \ldots, \mathcal{R}\left(\boldsymbol{x}_{K_t}^t\right)\right)$ which is a shuffling of K_t outputs of local mechanism $\mathcal{R}$, where $\mathcal{D}^{(t)}$ denotes an arbitrary set of K_t data points and we index $\boldsymbol{x}_i^t$'s from $i = 1$ to K_t just for convenience. From the amplification by shuffling result [BBG19d, Corollary 5.3.1], the mechanism $\mathcal{Z}$ is $(\tilde{\varepsilon}\left(K_t\right), \tilde{\delta})$-DP, where $\tilde{\delta} > 0$ is arbitrary, and, if $\varepsilon_0 \leq \frac{\log\left(K_t/\log\left(1/\tilde{\delta}\right)\right)}{2}$, then

$$\tilde{\varepsilon}\left(K_t\right) = \mathcal{O}\left(\min\{\varepsilon_0, 1\} e^{\varepsilon_0} \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{K_t}}\right). \tag{C.12}$$

Furthermore, when $\varepsilon_0 = \mathcal{O}\left(1\right)$, we get $\tilde{\varepsilon}\left(K_t\right) = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\log\left(1/\tilde{\delta}\right)}{K_t}}\right)$.

Let $\mathcal{T} \subseteq \{1, \ldots, n\}$ denote the identities of the K_t clients chosen at iteration t, and for $i \in \mathcal{T}$, let $\mathcal{T}_i \in \{1, \ldots, m\}$ denote the identity of the data point chosen at client i at time slot t. For any $\mathcal{T} \in \binom{[n]}{K_t}$ and $\mathcal{T}_i \in [m], i \in \mathcal{T}$, define $\overline{\mathcal{T}} = (\mathcal{T}, \mathcal{T}_i, i \in \mathcal{T})$, $\mathcal{D}^{\mathcal{T}_i} = \{d_{i\mathcal{T}_i}\}$ 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 $\mathcal{M}_t = \mathcal{Z}(\mathcal{D}^{\overline{\mathcal{T}}})$.

Define a dataset $\mathcal{D}' = (\mathcal{D}_1') \bigcup \left(\cup_{i=2}^{n} \mathcal{D}_i\right) \in \mathcal{X}^{(mn)}$, where $\mathcal{D}_1' = \{d_{11}', d_{12}, \ldots, d_{1m}\}$ 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 $(\overline{\varepsilon}, \overline{\delta})$-DP, we need to show that for an arbitrary subset $\mathcal{S}$ of the range of $\mathcal{M}_t$, we have

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] \leq e^{\overline{\varepsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] + \overline{\delta} \tag{C.13}$$

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] \leq e^{\overline{\varepsilon}} \Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] + \overline{\delta} \tag{C.14}$$

Note that both (C.13) and (C.14) are symmetric, so it suffices to prove only one of them. We prove (C.13) below.

For any $k \in [n]$, we define conditional probabilities as follows:

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

Note that when $K_t = 0$, i.e., no client participates, then we assume that the conditional probabilities $\Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S} \mid K_t = 0]$ and $\Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} \mid K_t = 0]$ are zero, as $\mathcal{D}'^{\overline{\mathcal{T}}} = \emptyset$ when $K_t = 0$. Therefore, in the rest of this section, we assume that K_t takes values in $[n] = \{1, 2, \ldots, m\}$. Now we expand $\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right]$:

$$\begin{aligned}
\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] &= \sum_{k=1}^{m} \Pr[K_t = k] \cdot \Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S} \mid K_t = k] \\
&= \sum_{k=1}^{m} \Pr[K_t = k] \left(\Pr[1 \in \mathcal{T}, 1 \in \mathcal{T}_1 \mid K_t = k] A_{11}(k)\right. \\
&\qquad \left. + \Pr[1 \in \mathcal{T}, 1 \notin \mathcal{T}_1 \mid K_t = k] A_{10}(k) + \Pr[1 \notin \mathcal{T} \mid K_t = k] A_0(k)\right)
\end{aligned} \tag{C.15}$$

Let $q' = \frac{1}{m}$, and for any $k \in [n]$, define $q_k = \frac{k}{n}$. With these, we can compute the conditional probabilities as

$$\begin{aligned}\Pr[1 \in \mathcal{T}, 1 \in \mathcal{T}_1 | K_t = k] &= \Pr[1 \in \mathcal{T} | K_t = k] \Pr[1 \in \mathcal{T}_1 | 1 \in \mathcal{T}] = q_k q' \\ \Pr[1 \in \mathcal{T}, 1 \notin \mathcal{T}_1 | K_t = k] &= \Pr[1 \in \mathcal{T} | K_t = k] \Pr[1 \notin \mathcal{T}_1 | 1 \in \mathcal{T}] = q_k (1 - q') \\ \Pr[1 \notin \mathcal{T} | K_t = k] &= 1 - q_k.\end{aligned}$$

Substituting these in (C.15) gives

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] = \sum_{k=1}^{n} \Pr[K_t = k]\left(q_k q' A_{11}(k) + q_k\left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)\right) \tag{C.16}$$

Similarly, we can show that

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}'\right) \in \mathcal{S}\right] = \sum_{k=1}^{n} \Pr[K_t = k]\left(q_k q' A'_{11}(k) + q_k\left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)\right) \tag{C.17}$$

Note that the mechanism $\mathcal{Z}$ is $(\tilde{\varepsilon}\left(K_t\right), \tilde{\delta})$-DP. Therefore, for every $k \in [n]$, we have

$$A_{11}(k) \leq e^{\tilde{\varepsilon}(k)} A'_{11}(k) + \tilde{\delta} \tag{C.18}$$

$$A_{11}(k) \leq e^{\tilde{\varepsilon}(k)} A_{10}(k) + \tilde{\delta} \tag{C.19}$$

Here (C.18) is straightforward, but proving (C.19) is obtained from eq(C.7). Since the mechanism $\mathcal{Z}$ is $(\tilde{\varepsilon}\left(K_t\right), \tilde{\delta})$-DP, in addition to (C.18)-(C.19), we also have the following inequality for every $k \in [n]$:

$$A_{11}(k) \leq e^{\tilde{\varepsilon}(k)} A_0(k) + \tilde{\delta} \tag{C.20}$$

Similar to (C.19), proving (C.20) is obtained from (C.8). Inequalities (C.18)-(C.20) together imply $A_{11}(k) \leq e^{\tilde{\varepsilon}(k)} \min\{A'_{11}(k), A_{10}(k), A_0(k)\} + \tilde{\delta}$.

Now we prove (C.13) for $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\bar{q}}{mn} \log(1/\tilde{\delta})}\right)$ (when $\varepsilon_0 = \mathcal{O}(1)$) and $\overline{\delta} = \bar{q}\tilde{\delta} + e^{-c'qn}$ for some constant $c' \in (0, 1)$, where $\bar{q} = \frac{q}{m}$.

$$\Pr\left[\mathcal{M}_t\left(\mathcal{D}\right) \in \mathcal{S}\right] = \sum_{k=1}^{n} \Pr[K_t = k] \cdot \Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S} \mid K_t = k]$$

$$
\begin{aligned}
&= \sum_{k=1}^{n} \Pr[K_t = k] \left(q_k q' A_{11}(k) + q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)\right) \\
&\leq \sum_{k=1}^{n} \Pr[K_t = k] \Big(q_k q' \Big(e^{\tilde{\varepsilon}(k)} \min\{A'_{11}(k), A_{10}(k), A_0(k)\} + \tilde{\delta}\Big) \\
&\quad +q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)) \\
&= \sum_{k=1}^{n} \Pr[K_t = k] \big(q_k q' \left((e^{\tilde{\varepsilon}(k)} - 1) \min\{A'_{11}(k), A_{10}(k), A_0(k)\}\right) \\
&\qquad + \min\{A'_{11}(k), A_{10}(k), A_0(k)\} + q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k) + q_k q' \tilde{\delta}\Big) \\
&\overset{\text{(a)}}{\leq} \sum_{k=1}^{n} \Pr[K_t = k] \big(q_k q'(e^{\tilde{\varepsilon}(k)} - 1) \min\{A'_{11}(k), A_{10}(k), A_0(k)\} + q_k q' A'_{11}(k) \\
&\qquad +q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k) + q_k q' \tilde{\delta}\Big) \\
&\overset{\text{(b)}}{\leq} \sum_{k=1}^{n} \Pr[K_t = k] \big(q_k q'(e^{\tilde{\varepsilon}(k)} - 1) \left(q_k q' A'_{11}(k) + q_k(1 - q') A_{10}(k) + (1 - q_k) A_0(k)\right)) \\
&\qquad + \left(q_k q' A'_{11}(k) + q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)\right) + q_k q' \tilde{\delta}\Big) \\
&= \sum_{k=1}^{n} \Pr[K_t = k] \left(\left(1 + q_k q' \left(e^{\tilde{\varepsilon}(k)} - 1\right)\right)\right) \\
&\qquad \left(q_k q' A'_{11}(k) + q_k \left(1 - q'\right) A_{10}(k) + \left(1 - q_k\right) A_0(k)\right) + q_k q' \tilde{\delta}\Big) \\
&= \sum_{k=1}^{n} \Pr[K_t = k] \left(e^{\ln(1 + q_k q'(e^{\tilde{\varepsilon}(k)} - 1))} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} \mid K_t = k] + q_k q' \tilde{\delta}\right) \\
&= \sum_{k=1}^{n} \Pr[K_t = k] e^{\ln(1 + q_k q'(e^{\tilde{\varepsilon}(k)} - 1))} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} \mid K_t = k] + \sum_{k=1}^{n} \Pr[K_t = k] q_k q' \tilde{\delta}
\end{aligned}
\tag{C.21}
$$

Here, (a) follows from $\min\{x, y, z\} \leq x$, and (b) follows from the fact that minimum is upper-bounded by the convex combination. Now we bound both the terms in (C.21) separately.

Bounding the first term of (C.21)**:** Let $p_k = \Pr[K_t = k]$ and $\mu_k = \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S} \mid K_t = k]$.

$$
\sum_{k=1}^{n} p_k e^{\ln(1 + q_k q'(e^{\tilde{\varepsilon}(k)} - 1))} \mu_k = \sum_{k < (1-\varepsilon)qn} p_k e^{\ln(1 + q_k q'(e^{\tilde{\varepsilon}(k)} - 1))} \mu_k + \sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k e^{\ln(1 + q_k q'(e^{\tilde{\varepsilon}(k)} - 1))} \mu_k
$$

$$+ \sum_{k>(1+\varepsilon)qn} p_k e^{\ln(1+q_k q'(e^{\tilde{\varepsilon}(k)}-1))} \mu_k,$$

where $\varepsilon \in (0,1)$ is a constant that we will decide later. Let $l = (1-\varepsilon)qn$ and $u = (1+\varepsilon)qn$. Substituting $\mu_k \leq 1$ in both the first and the third summation gives

$$\sum_{k=1}^{n} p_k e^{\ln(1+q_k q'(e^{\tilde{\varepsilon}(k)}-1))} \mu_k \leq \left((1 + q_l q'(e^{\tilde{\varepsilon}(1)} - 1)) \sum_{k<(1-\varepsilon)qn} p_k \right)$$
$$+ \left(e^{\ln(1+q_u q'(e^{\tilde{\varepsilon}(l)}-1))} \sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k \mu_k \right) + \left((1 + q_n q'(e^{\tilde{\varepsilon}(u)} - 1)) \sum_{k>(1+\varepsilon)qn} p_k \right) \tag{C.22}$$

First we bound $\sum_{k<(1-\varepsilon)qn} p_k$ and $\sum_{k>(1+\varepsilon)qn} p_k$, which are the tail probabilities of the binomial random variable. We can bound both these using the Chernoff bound as follows (where $\varepsilon \in (0,1)$):

$$\Pr[K_t \geq (1+\varepsilon)qn] \leq \exp(-qn\varepsilon^2/3) \tag{C.23}$$

$$\Pr[K_t \leq (1-\varepsilon)qn] \leq \exp(-qn\varepsilon^2/3) \tag{C.24}$$

Substituting these in (C.22) gives

$$\sum_{k=1}^{n} p_k e^{\ln(1+q_k q'(e^{\tilde{\varepsilon}(k)}-1))} \mu_k \leq \big((1 + q_l q'(e^{\tilde{\varepsilon}(1)} - 1)) + (1 + q_n q'(e^{\tilde{\varepsilon}(u)} - 1)) \big) \exp(-qn\varepsilon^2/3)$$
$$+ \left(e^{\ln(1+q_u q'(e^{\tilde{\varepsilon}(l)}-1))} \sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k \mu_k \right) \tag{C.25}$$

Note that $q_l, q' \leq 1, q_n = 1$ and $e^{\tilde{\varepsilon}(u)} \leq e^{\tilde{\varepsilon}(1)}$. Substituting these in (C.25) and also upper-bounding the last term trivially as $\sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k \mu_k \leq \sum_{k=1}^{n} p_k \mu_k$ and taking $\varepsilon = \frac{1}{2}$ gives

$$\sum_{k=1}^{n} p_k e^{\ln(1+q_k q'(e^{\tilde{\varepsilon}(k)}-1))} \mu_k \leq 2e^{\tilde{\varepsilon}(1)} \exp(-qn/12) + \left(e^{\ln(1+q_u q'(e^{\tilde{\varepsilon}(l)}-1))} \sum_{k=1}^{n} p_k \mu_k \right) \tag{C.26}$$

Now we bound both terms of (C.26) separately.

- We have from (C.12) that for $\varepsilon_0 = \mathcal{O}(1)$, we have $\tilde{\varepsilon}\,(1) = \varepsilon_0$. This implies $e^{\tilde{\varepsilon}(1)} = e^{\varepsilon_0}$. Note that this quantity is much smaller in comparison to, say, $\exp(qn/24)$, where qn is typically a large number (at least a few thousands) in cross device federated learning settings. Since $\varepsilon_0 = \mathcal{O}(1)$, we can bound the first term in (C.26) as

$$2e^{\tilde{\varepsilon}(1)} \exp(-qn/12) \leq \exp(-c'qn), \tag{C.27}$$

for some constant $c' \in (0,1)$.

- We can bound $e^{\ln(1+q_u q'(e^{\tilde{\varepsilon}(l)}-1))}$ as follows:

$$\ln(1+q_u q'(e^{\tilde{\varepsilon}(l)}-1)) \leq q_u q'(e^{\tilde{\varepsilon}(l)}-1) = \mathcal{O}\left(q_u q' \tilde{\varepsilon}\left(l\right)\right) = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{(1+\varepsilon)^2}{(1-\varepsilon)} \frac{\bar{q}}{mn} \log(1/\tilde{\delta})}\right), \tag{C.28}$$

  where $\varepsilon = \frac{1}{2}$, and $\bar{q} = qq' = \frac{q}{r}$.

Substituting the bounds from (C.27) and (C.28) into (C.26) gives

$$\sum_{k=1}^{n} p_k e^{\ln(1+q_k q'(e^{\tilde{\varepsilon}(k)}-1))} \mu_k \leq e^{\overline{\varepsilon}} \sum_{k=1}^{n} p_k \mu_k + \exp(-c'qn), \tag{C.29}$$

where $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\bar{q}}{mn} \log(1/\tilde{\delta})}\right)$.

Bounding the second term of (C.21)**:** In the following, $\bar{q} = qq' = \frac{q}{m}$.

$$\sum_{k=1}^{n} \Pr[K_t = k] q_k q' \tilde{\delta} = q'\tilde{\delta} \sum_{k=1}^{n} \Pr[K_t = k] \frac{k}{n} = q'\tilde{\delta} \frac{\mathbb{E}[K_t]}{n} = q'\tilde{\delta} q = \bar{q}\tilde{\delta}. \tag{C.30}$$

Substituting the bounds from (C.29) and (C.30) into (C.21) gives

$$\Pr[\mathcal{M}_t(\mathcal{D}) \in \mathcal{S}] \leq e^{\overline{\varepsilon}} \Pr[\mathcal{M}_t(\mathcal{D}') \in \mathcal{S}] + \overline{\delta}, \tag{C.31}$$

where $\overline{\varepsilon} = \mathcal{O}\left(\varepsilon_0 \sqrt{\frac{\bar{q}\log(1/\tilde{\delta})}{mn}}\right)$ and $\overline{\delta} = \bar{q}\tilde{\delta} + e^{-c'qn}$ for some constant $c' \in (0,1)$, where $\bar{q} = \frac{q}{m}$. This completes the proof of Lemma 4.4.1.

C.3 Proof of Claim 4.4.1

In the following $|\mathcal{U}_t| = K_t$.

$$\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} [\overline{\mathbf{g}}_t] = \mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} \left[\frac{1}{K_t} \sum_{i \in \mathcal{U}_t} \mathcal{R}_p\left(\nabla_{\theta_t} f(\theta_t; d_{ij_i})\right)\right]$$

$$
\begin{aligned}
&= \mathbb{E}_{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}} \left[\frac{1}{K_t} \sum_{i \in \mathcal{U}_t} \mathbb{E}_{j_i \in [m]} \left(\mathbb{E}_{\mathcal{R}_p}[\mathcal{R}_p \left(\nabla_{\theta_t} f(\theta_t; d_{ij_i}) \right)] \right) \right] \\
&\overset{\text{(a)}}{=} \mathbb{E}_{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}} \left[\frac{1}{K_t} \sum_{i \in \mathcal{U}_t} \mathbb{E}_{j_i \in [m]} \left(\nabla_{\theta_t} f(\theta_t; d_{ij_i}) \right) \right] \\
&\overset{\text{(b)}}{=} \mathbb{E}_{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}} \left[\frac{1}{K_t} \sum_{i \in \mathcal{U}_t} \nabla_{\theta_t} F_i(\theta_t) \right] \\
&\overset{\text{(c)}}{=} \nabla_{\theta_t} F(\theta_t),
\end{aligned}
$$

where (a) follows from the unbiasedness of the randomized mechanism $\mathcal{R}_p$, (b) follows because the mini-batch sampling of stochastic gradients gives unbiased gradient and that $F_i(\theta) = \frac{1}{m} \sum_{j=1}^{m} f(\theta; d_{ij})$ for $i \in [n]$, and (c) follows because i.i.d. sampling of clients gives unbiased global gradient and that $F(\theta) = \frac{1}{n} \sum_{i=1}^{n} F_i(\theta)$.

C.4 Proof of Lemma 4.4.2

Note that when we condition on $\{K_t = k\}$ – the event that a fixed number of k clients participate – the random variable $\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}$ is distributed as $\mathcal{U}_t \sim \text{samp}^{\text{fix}}_{n,k}$.

$$
\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} \|\overline{\mathbf{g}}_t\|_2^2 = \sum_{k=1}^{n} \Pr[K_t = k] \cdot \mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{fix}}_{n,k}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} [\|\overline{\mathbf{g}}_t\|_2^2 \mid K_t = k] \tag{C.32}
$$

To make the notation less cluttered, it will be convenient to define the following for any $k \in [n]$:

$$
\begin{aligned}
p_k &:= \Pr[K_t = k] \\
\mathcal{E}_k &:= \mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{fix}}_{n,k}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} [\|\overline{\mathbf{g}}_t\|_2^2 \mid K_t = k]
\end{aligned}
$$

Note that we do not update anything when no client participates, i.e., $\mathcal{E}_0 = 0$. Substituting these in (C.32), we get

$$
\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} \|\overline{\mathbf{g}}_t\|_2^2 = \sum_{k=1}^{n} p_k \mathcal{E}_k \tag{C.33}
$$

From Lemma 4.3.2, we have that:

$$\mathcal{E}_k \leq L^2 \left(1 + \frac{f_p(\varepsilon_0, b)}{qmn}\right), \tag{C.34}$$

where $f_p(\varepsilon_0, b)$ is the MSE of the private mechanism $\mathcal{R}_p$. It is clear from (C.34) that $\mathcal{E}_k$ is a non-increasing function of k. Thus, we have:

$$\begin{aligned}
\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} \|\overline{\mathbf{g}}_t\|_2^2 &= \sum_{k=1}^{n} p_k \mathcal{E}_k && \text{(from (C.33))} \\
&= \sum_{k<(1-\varepsilon)qn} p_k \mathcal{E}_k + \sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k \mathcal{E}_k + \sum_{k>(1+\varepsilon)qn} p_k \mathcal{E}_k \\
&\overset{\text{(e)}}{\leq} \left(\mathcal{E}_1 \sum_{k<(1-\varepsilon)qn} p_k\right) + \left(\mathcal{E}_{(1-\varepsilon)qn} \sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k\right) + \left(\mathcal{E}_{(1+\varepsilon)qn} \sum_{k>(1+\varepsilon)qn} p_k\right) \\
&\overset{\text{(f)}}{\leq} \left(\mathcal{E}_1 \sum_{k<(1-\varepsilon)qn} p_k\right) + \mathcal{E}_{(1-\varepsilon)qn} + \left(\mathcal{E}_1 \sum_{k>(1+\varepsilon)qn} p_k\right) \\
&= \mathcal{E}_{(1-\varepsilon)qn} + \mathcal{E}_1 \left(\Pr[K_t < (1-\varepsilon)qn] + \Pr[K_t > (1+\varepsilon)qn]\right). && \text{(C.35)}
\end{aligned}$$

In (e) we used the non-increasing property of $\mathcal{E}_k$, i.e., $\mathcal{E}_k \geq \mathcal{E}_{k+1}$ for any $k \in [n-1]$. In (f) we used $\mathcal{E}_{(1+\varepsilon)qn} \leq \mathcal{E}_1$ and that $\sum_{k=(1-\varepsilon)qn}^{(1+\varepsilon)qn} p_k \leq 1$. Both $\Pr[K_t < (1-\varepsilon)qn]$ and $\Pr[K_t > (1+\varepsilon)qn]$ are the tail probabilities of the binomial random variable. We can bound both these using the Chernoff bound from (C.24)–(C.23). Substituting the bounds from (C.23),(C.24) into (C.35) and taking $\varepsilon = \frac{1}{2}$ gives:

$$\begin{aligned}
\mathbb{E}_{\substack{\mathcal{U}_t \sim \text{samp}^{\text{iid}}_{n,q}, \mathcal{R}_p, \\ j_i \in [m], i \in \mathcal{U}_t}} \|\overline{\mathbf{g}}_t\|_2^2 &\leq \mathcal{E}_{\frac{qn}{2}} + 2\mathcal{E}_1 \exp(-qn/12) \\
&\leq L^2 \left(1 + \frac{2f_p(\varepsilon_0, b)}{qn}\right) \\
&\qquad + 2L^2 \left(1 + f_b(\varepsilon_0, b)\right) \exp(-qm/12) \\
&\leq L^2 \left(1 + \frac{2f_p(\varepsilon_0, b)}{qn}\right) + \exp(-c'qn), && \text{(C.36)}
\end{aligned}$$

where $c' > 0$ is a constant. In particular, $c' \geq 1/24$ if $2L^2(1 + f_p(\varepsilon_0, b)) \leq \exp(qm/24)$, which is easily satisfied in federated learning settings, where in each iteration, at least a few thousand clients send updates, i.e., qn is equal to a few thousands. Note that $\bar{q} = \frac{q}{m}$, so we have $qn = \bar{q}mn$. Substituting this in (C.36) yields (4.12), which completes the proof of Lemma 4.4.2.

C.5 Proof of Theorem 4.5.1

We first analyze the RDP of a single global round $t \in [T]$ and then, we obtain the results from the composition of the RDP over total T global rounds. Recall that privacy leakage can happen through communicating $\{\boldsymbol{\mu}_i\}$ and $\{\psi_i^t\}$ and we privatize both of these. In the following, we do the privacy analysis of privatizing $\{\boldsymbol{\mu}_i\}$ and a similar analysis could be done for $\{\psi_i^t\}$ as well.

At each synchronization round $t \in [T]$, the server updates the global model $\boldsymbol{\mu}^{t+1}$ as follows:

$$\boldsymbol{\mu}^{t+1} = \frac{1}{K} \sum_{i \in \mathcal{K}t} \boldsymbol{\mu}_i^t, \tag{C.37}$$

where $\boldsymbol{\mu}_i^t$ is the update of the global model at the i-th client that is obtained by running τ local iterations at the i-th client. At each of the local iterations, the client clips the gradient $\boldsymbol{h}_i^t$ with threshold C_1 and adds a zero-mean Gaussian noise vector with variance $\sigma_{q_1}^2 \mathbb{I}_d$. When neglecting the noise added at the local iterations, the ℓ_2-norm sensitivity of updating the global model $\boldsymbol{\mu}_i^{t+1}$ at the synchronization round t is bounded by:

$$\Delta\boldsymbol{\mu} = \max_{\mathcal{K}^t, \mathcal{K}'^t} \|\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}'^{t+1}\|_2^2 \leq \frac{\tau C_1^2}{K^2}, \tag{C.38}$$

where $\mathcal{K}^t, \mathcal{K}'^t \subset [m]$ are neighboring sets that differ in only one client. Additionally, $\boldsymbol{\mu}^{t+1} = \frac{1}{K} \sum_{i \in \mathcal{K}t} \boldsymbol{\mu}_i^t$ and $\boldsymbol{\mu}'^{t+1} = \frac{1}{K} \sum_{i \in \mathcal{K}'t} \boldsymbol{\mu}_i^t$. Since we add i.i.d. Gaussian noises with variance $\sigma_{q_1}^2$ at each local iteration at each client, and then, we take the average of theses vectors over K clients, it is equivalent to adding a single Gaussian vector to the aggregated vectors with variance $\frac{\tau \sigma_{q_1}^2}{K}$. Thus, from the RDP of the sub-sampled Gaussian mechanism in [MTZ19, Table 1], [BDR18], we get that the global model $\boldsymbol{\mu}^{t+1}$ of a single global iteration of `DP-AdaPeD` is

$(\alpha, \varepsilon_t^{(1)}(\alpha))$-RDP, where $\varepsilon_t(\alpha)^(1)$ is bounded by:

$$\varepsilon_t^{(1)}(\alpha) = \left(\frac{K}{n}\right)^2 \frac{6\alpha C_1^2}{K\sigma_{q_1}^2}. \tag{C.39}$$

Similarly, we can show that the global parameter ψ^{t+1} at any synchronization round of DP-AdaPeD is $(\alpha, \varepsilon_t^{(2)}(\alpha))$-RDP, where $\varepsilon_t(\alpha)$ is bounded by:

$$\varepsilon_t^{(2)}(\alpha) = \left(\frac{K}{n}\right)^2 \frac{6\alpha C_2^2}{K\sigma_{q_2}^2}. \tag{C.40}$$

Using adaptive RDP composition in Lemma 2.1.4, we get that each synchronization round of DP-AdaPeD is $(\alpha, \varepsilon_t^{(1)}(\alpha) + \varepsilon_t^{(2)}(\alpha))$-RDP. Thus, by running DP-AdaPeD over T/τ synchronization rounds and from the composition of the RDP, we get that DP-AdaPeD is $(\alpha, \varepsilon(\alpha))$-RDP, where $\varepsilon(\alpha) = \left(\frac{T}{\tau}\right)(\varepsilon_t^{(1)}(\alpha) + \varepsilon_t^{(2)}(\alpha))$. This completes the proof of Theorem 4.5.1.

APPENDIX D

Omitted Details From Chapter 5

D.1 Proof of Corollary 1

In this section, we prove the simplified bound (stated in (5.5)) on the RDP of the shuffle model, provided that α, ε_0, n satisfy a certain condition. In particular, we will show that if α, ε_0, n satisfy $\alpha^4 e^{5\varepsilon_0} < \frac{n}{9}$, then

$$\varepsilon(\alpha) \leq \frac{1}{\alpha-1} \log\left(1 + \frac{\alpha^2\left(e^{\varepsilon_0}-1\right)^2}{\overline{n}e^{\varepsilon_0}}\right), \tag{D.1}$$

where $\overline{n} = \frac{n-1}{2e^{\varepsilon_0}} + 1$. In order to show (D.1), it suffices to prove the following (using which in (5.4) will yield (5.5)):

$$\sum_{i=3}^{\alpha} \binom{\alpha}{i} i\Gamma\left(i/2\right)\left(\frac{\left(e^{2\varepsilon_0}-1\right)^2}{2e^{2\varepsilon_0}\overline{n}}\right)^{i/2} + e^{\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}} \leq \binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{n}e^{\varepsilon_0}}. \tag{D.2}$$

First notice that $\binom{\alpha}{i} i\Gamma\left(i/2\right) \leq \alpha^i$ (see Claim D.1.1 on page 212). In order to show (D.2), it suffices to show

$$\sum_{i=3}^{\alpha}\left(\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{\left(2e^{2\varepsilon_0}\overline{n}\right)^{1/2}}\right)^i + e^{\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}} \leq \binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{n}e^{\varepsilon_0}}. \tag{D.3}$$

Note that there are $(\alpha-2)$ terms inside the summation. If we show that each of those terms is smaller than 1 (which would imply that the term corresponding to $i=3$ is the largest one), then the summation is at most $(\alpha-2)$ times the term with $i=3$. Further, if the additional exponential term in the LHS is upper-bounded by the term with $i=3$, then we can prove (D.3) by showing that $(\alpha-1)$ times the term with $i=3$ is upper-bounded by the RHS. These arguments are summarized in the following set of three inequalities:

$$\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{(2e^{2\varepsilon_0}\overline{n})^{1/2}} < 1 \tag{D.4}$$

$$e^{\varepsilon_0\alpha-\frac{n-1}{8e^{\varepsilon_0}}} \leq \left(\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{(2e^{2\varepsilon_0}\overline{n})^{1/2}}\right)^3 \tag{D.5}$$

$$(\alpha-1)\left(\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{(2e^{2\varepsilon_0}\overline{n})^{1/2}}\right)^3 \leq \binom{\alpha}{2}\frac{\left(e^{\varepsilon_0}-1\right)^2}{\overline{n}e^{\varepsilon_0}} \tag{D.6}$$

In the rest of this proof, we will derive the condition on ε_0, α, n such that (D.6) is satisfied. As we see later, the values of ε_0, α thus obtained will automatically satisfy (D.4) and (D.5).

By canceling same terms from both sides of (D.6), we get

$$\begin{aligned} \frac{\alpha^2\left(e^{2\varepsilon_0}-1\right)^3}{(2e^{\varepsilon_0}\overline{n})^{3/2}e^{3\varepsilon_0/2}} &\leq \frac{\left(e^{\varepsilon_0}-1\right)^2}{2\overline{n}e^{\varepsilon_0}} \\ \iff \alpha^2(e^{2\varepsilon_0}-1)(e^{\varepsilon_0}+1)^2 &\leq \sqrt{2\overline{n}e^{\varepsilon_0}}e^{3\varepsilon_0/2} \end{aligned} \tag{D.7}$$

For the LHS and the RHS, we respectively have

$$\begin{aligned} (e^{2\varepsilon_0}-1)(e^{\varepsilon_0}+1)^2 &= (e^{2\varepsilon_0}-1)(e^{2\varepsilon_0}+2e^{\varepsilon_0}+1) \\ &\leq e^{4\varepsilon_0}+2e^{3\varepsilon_0} \leq 3e^{4\varepsilon_0} \end{aligned} \tag{D.8}$$

$$2\overline{n}e^{\varepsilon_0} = n-1+2e^{\varepsilon_0} \geq n. \tag{D.9}$$

Therefore, in order to show (D.7), it suffices to show $3\alpha^2e^{4\varepsilon_0} \leq \sqrt{ne^{3\varepsilon_0}}$, which is equivalent to $\alpha^4e^{5\varepsilon_0} < \frac{n}{9}$. Thus, we have shown that $\alpha^4e^{5\varepsilon_0} < \frac{n}{9}$ implies (D.6).

Now we show that when $\alpha^4e^{5\varepsilon_0} < \frac{n}{9}$, (D.4) and (D.5) are automatically satisfied:

1. Proof of (D.4):

$$\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{\sqrt{2e^{2\varepsilon_0}\overline{n}}} \leq \frac{\alpha e^{2\varepsilon_0}}{\sqrt{2e^{\varepsilon_0}\overline{n}}} \leq \sqrt{\frac{\alpha^4e^{5\varepsilon_0}}{2e^{\varepsilon_0}\overline{n}}} \leq \sqrt{\frac{n/9}{n}} < 1.$$

   In the second inequality we used $\alpha \geq 1$ and in the penultimate inequality we used $2e^{\varepsilon_0}\overline{n} \geq n$ from (D.9).

2. Proof of (D.5): For this, first we upper-bound the LHS and lower-bound the RHS, and then note that the upper-bound is smaller than the lower-bound. For the upper-bound on $\exp(\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}})$, note that $\varepsilon_0\alpha \leq e^{5\varepsilon_0/4}\alpha = \left(e^{5\varepsilon_0}\alpha^4\right)^{1/4} < \left(\frac{n}{9}\right)^{1/4} = \frac{n^{1/4}}{\sqrt{3}}$. Also note that $e^{\varepsilon_0} \leq e^{5\varepsilon_0/4}\alpha < \frac{n^{1/4}}{\sqrt{3}}$, which implies $\frac{n-1}{8e^{\varepsilon_0}} = \frac{\sqrt{3}}{8}\frac{n-1}{n^{1/4}} \geq \frac{\sqrt{3}}{16}n^{3/4}$. Substituting these bounds in the exponent of $\exp(\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}})$, we get:

$$\begin{aligned}\exp\left(\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}}\right) &\leq \exp\left(\frac{n^{1/4}}{\sqrt{3}} - \frac{\sqrt{3}}{16}n^{3/4}\right) \\ &= \exp\left(-n^{3/4}\left(\frac{\sqrt{3}}{16} - \frac{1}{\sqrt{3n}}\right)\right) \\ &\leq \exp\left(-c'n^{3/4}\right),\end{aligned} \tag{D.10}$$

where $c' > 0$ is a constant even for small values of n. For example, for $n = 100$, we get $c' \geq \frac{1}{20}$.

For the lower-bound on $\left(\frac{\alpha\left(e^{2\varepsilon_0}-1\right)}{(2e^{2\varepsilon_0}\overline{n})^{1/2}}\right)^3$, note that $2e^{\varepsilon_0}\overline{n} = n-1+2e^{\varepsilon_0} \leq n-1+2\left(\frac{n}{9}\right)^{1/5} \leq 2n$, where $e^{\varepsilon_0} \leq \left(\frac{n}{9}\right)^{1/5}$ follows from $e^{5\varepsilon_0} \leq \alpha^4 e^{5\varepsilon_0} < \frac{n}{9}$. Now we show the lower bound:

$$\begin{aligned}\frac{\alpha^3(e^{2\varepsilon_0}-1)^3}{(2e^{2\varepsilon_0}\overline{n})^{3/2}} &\geq \frac{(e^{\varepsilon_0}-1)^3(e^{\varepsilon_0}+1)^3}{(2e^{\varepsilon_0}\overline{n})^{3/2}e^{3\varepsilon_0/2}} \\ &\geq \frac{(e^{\varepsilon_0}-1)^3 e^{3\varepsilon_0}}{(2n)^{3/2}e^{3\varepsilon_0/2}} \\ &\geq \frac{(e^{\varepsilon_0}-1)^3}{(2n)^{3/2}} \geq \frac{\varepsilon_0^3}{(2n)^{3/2}}\end{aligned} \tag{D.11}$$

Note that the upper-bound on $\exp(\varepsilon_0\alpha - \frac{n-1}{8e^{\varepsilon_0}})$ is exponentially small in $n^{3/4}$, whereas, the lower-bound on $\frac{\alpha^3(e^{2\varepsilon_0}-1)^3}{(2e^{2\varepsilon_0}\overline{n})^{3/2}}$ is inverse-polynomial in n. So, for sufficiently large n, (D.5) will be satisfied.

This completes the proof of Corollary 5.3.1

Claim D.1.1 (An Inequality for the Gamma Function)**.** For any $\alpha \in \mathbb{N}$ and $k \geq 3$, we have $\binom{\alpha}{k}k\Gamma(k/2) \leq \alpha^k$.

Proof. Note that for any $\alpha \in \mathbb{N}$ and $k \leq \alpha$, we have $\binom{\alpha}{k} = \frac{\alpha(\alpha-1)(\alpha-2)\ldots(\alpha-k+1)}{k!}$.

We show the claim separately for the cases when k is an even integer or not.

1. *When k is an even integer:* Since for any integer $n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$, so when k is an even integer, we have

$$\begin{aligned}\binom{\alpha}{k} k\Gamma(k/2) &= \frac{\alpha(\alpha-1)(\alpha-2)\dots(\alpha-k+1)}{k!} \times k \times (\frac{k}{2}-1)! \\ &\leq \alpha(\alpha-1)(\alpha-2)\dots(\alpha-k+1) \\ &\leq \alpha^k.\end{aligned}$$

2. *When k is an odd integer:* Note that for any integer $n \in \mathbb{N}$, we have $\Gamma\left(n+\frac{1}{2}\right) = \frac{(2n)!}{4^n n!}\sqrt{\pi}$; see [Wik]. Let $k = 2a+1$. Then

$$\begin{aligned}\binom{\alpha}{k} k\Gamma(k/2) &= \binom{\alpha}{k} k\Gamma(a+\frac{1}{2}) \\ &= \frac{\alpha(\alpha-1)(\alpha-2)\dots(\alpha-k+1)}{k!} \times k \times \frac{(2a)!}{4^a a!}\sqrt{\pi} \\ &= \alpha(\alpha-1)(\alpha-2)\dots(\alpha-k+1)\frac{\sqrt{\pi}}{4^a a!} \\ &\overset{(a)}{\leq} \alpha(\alpha-1)(\alpha-2)\dots(\alpha-k+1) \\ &\leq \alpha^k\end{aligned}$$

where (a) follows because $\frac{\sqrt{\pi}}{4^a a!} \leq 1$ when $a \geq 1 \iff k \geq 3$.

This proves Claim D.1.1. ■

D.2 Omitted Details from Section 5.3.1

D.2.1 Omitted Details from Section 5.3.1.1

Before proving (5.14), first we show an important property of E_m that we will use in the proof.

Lemma D.2.1. *E_m is a non-increasing function of m, i.e.,*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})}\left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})}\right)^{\alpha}\right] \le \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m})}\left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}^{(n)}_{m})(\boldsymbol{h})}\right)^{\alpha}\right], \tag{D.12}$$

where, for any $k \in \{m, m+1\}$, $\mathcal{D}^{(n)}_k = (d'_n, \dots, d'_n, d_n)$ and $\mathcal{D}'^{(n)}_k = (d'_n, \dots, d'_n, d'_n)$ with $|\mathcal{D}_k| = |\mathcal{D}'_k| = k$.

Proof. Lemma D.2.1 follows from Lemma 5.3.3 in a straightforward manner, as, unlike Lemma D.2.1, in Lemma 5.3.3 we consider arbitrary pairs of neighboring datasets. ■

Now we can prove (5.14).

Proof of (5.14).

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}')}\left[\left(\frac{\mathcal{M}(\mathcal{D})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}')(\boldsymbol{h})}\right)^{\alpha}\right] &\le \sum_{m=0}^{n-1} q_m E_m \\
&= \sum_{m<\lfloor(1-\gamma)q(n-1)\rfloor} q_m E_m + \sum_{m\ge\lfloor(1-\gamma)q(n-1)\rfloor} q_m E_m \\
&\overset{\text{(a)}}{\le} E_0 \sum_{m<\lfloor(1-\gamma)q(n-1)\rfloor} q_m + \sum_{m\ge\lfloor(1-\gamma)q(n-1)\rfloor} q_m E_m \\
&\overset{\text{(b)}}{\le} E_0 e^{-\frac{q(n-1)\gamma^2}{2}} + \sum_{m\ge\lfloor(1-\gamma)q(n-1)\rfloor} q_m E_m \\
&\overset{\text{(c)}}{\le} e^{\epsilon_0\alpha} e^{-\frac{q(n-1)\gamma^2}{2}} + \sum_{m\ge\lfloor(1-\gamma)q(n-1)\rfloor} q_m E_m \\
&\overset{\text{(d)}}{\le} e^{\epsilon_0\alpha} e^{-\frac{q(n-1)\gamma^2}{2}} + E_{(1-\gamma)q(n-1)}.
\end{aligned}$$

Here, steps (a) and (d) follow from the fact that E_m is a non-increasing function of m (see Lemma D.2.1). Step (b) follows from the Chernoff bound. In step (c), we used that $\mathcal{M}(d_n) = \mathcal{R}(d_n)$ and $\mathcal{M}(d'_n) = \mathcal{R}(d'_n)$, which together imply that

$$E_0 = \mathbb{E}\left[\left(\frac{\mathcal{M}(d_n)}{\mathcal{M}(d'_n)}\right)^{\alpha}\right] = \mathbb{E}\left[\left(\frac{\mathcal{R}(d_n)}{\mathcal{R}(d'_n)}\right)^{\alpha}\right] \le e^{\varepsilon_0\alpha},$$

where the inequality follows because $\mathcal{R}$ is an ε_0-LDP mechanism. ■

D.2.2 Proof of Theorem 5.3.6

Fix an arbitrary $m \in \mathbb{N}$. Let $(\mathcal{D}_m, \mathcal{D}'_m) \in \mathcal{D}^m_{\text{same}}$ and $\boldsymbol{p} = (p_1, \ldots, p_B), \boldsymbol{p}' = (p'_1, \ldots, p'_B)$ be the same as defined in the proof of Theorem 5.3.5 in Section 5.3.3.

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}'_m\right)(\boldsymbol{h})}{\mathcal{M}\left(\mathcal{D}_m\right)(\boldsymbol{h})}\right)^{\alpha}\right] &= \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\sum_{j=1}^{B}\frac{p'_j}{p_j}\frac{h_j}{m}\right)^{\alpha}\right] \\
&= \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(1+\sum_{j=1}^{B}\frac{p'_j}{p_j}\frac{h_j}{m}-1\right)^{\alpha}\right] \\
&\leq \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\exp\left(\alpha\left(\sum_{j=1}^{B}\frac{p'_j}{p_j}\frac{h_j}{m}-1\right)\right)\right], \qquad \text{(D.13)}
\end{aligned}$$

where the first equality uses (5.33) and the last inequality follows from $1 + x \leq e^x$. In (D.13), $\boldsymbol{h}$ is distributed according to $\mathcal{M}(\mathcal{D}_m) = \mathcal{H}_m(\mathcal{R}(d), \ldots, \mathcal{R}(d))$, where $\mathcal{H}_m$ denotes the shuffling operation on m elements and range of $\mathcal{R}$ is equal to $[B]$. Since all the m data points are identical, and all clients use independent randomness for computing $\mathcal{R}(d)$, we can assume, w.l.o.g., that $\mathcal{M}(\mathcal{D}_m)$ is a collection of m i.i.d. random variables $X_1, \ldots, X_m$, where $\Pr\left[X_i = j\right] = p_j$ for $j \in [B]$. Thus, we have (in the following, note that $\boldsymbol{h} = (h_1, \ldots, h_B)$ is a r.v.)

$$\begin{aligned}
\frac{1}{m}\sum_{j=1}^{B}\frac{p'_j}{p_j}h_j &= \frac{1}{m}\sum_{j=1}^{B}\frac{p'_j}{p_j}\sum_{i=1}^{m}\mathbf{1}_{\{X_i=j\}} \\
&= \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{B}\frac{p'_j}{p_j}\mathbf{1}_{\{X_i=j\}} = \frac{1}{m}\sum_{i=1}^{m}\frac{p'_{X_i}}{p_{X_i}}, \qquad \text{(D.14)}
\end{aligned}$$

where $\mathbf{1}_{\{.\}}$ denotes the indicator r.v. Substituting from (D.14) into (D.13), we get

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\exp\left(\alpha\left(\sum_{j=1}^{B}\frac{p'_j}{p_j}\frac{h_j}{m}-1\right)\right)\right] \\
&\qquad = \mathbb{E}_{X_1,\ldots,X_m}\left[\exp\left(\frac{\alpha}{m}\sum_{i=1}^{m}\left(\frac{p'_{X_i}}{p_{X_i}}-1\right)\right)\right] \\
&\qquad = \prod_{i=1}^{m}\mathbb{E}_{X_i}\left[\exp\left(\frac{\alpha}{m}\left(\frac{p'_{X_i}}{p_{X_i}}-1\right)\right)\right]
\end{aligned}$$

$$= \left(\mathbb{E}_{X\sim\boldsymbol{p}}\left[e^{\frac{\alpha}{m}\left(\frac{p'_X}{p_X}-1\right)}\right]\right)^{m} \tag{D.15}$$

where $\boldsymbol{p} = [p_1, \dots, p_B]$. From Taylor expansion of $e^x = 1 + \sum_{k=1}^{\infty} \frac{x^k}{k!}$, we get

$$\begin{aligned}
\mathbb{E}_{X\sim\boldsymbol{p}}\left[e^{\frac{\alpha}{m}\left(\frac{p'_X}{p_X}-1\right)}\right] &= 1 + \sum_{k=1}^{\infty}\frac{1}{k!}\mathbb{E}_{X\sim\boldsymbol{p}}\left[\left(\frac{\alpha}{m}\left(\frac{p'_X}{p_X}-1\right)\right)^{k}\right] \\
&= 1 + \sum_{k=1}^{\infty}\frac{1}{k!}\sum_{j=1}^{B} p_j\left(\frac{\alpha}{m}\left(\frac{p'_j}{p_j}-1\right)\right)^{k} \\
&= 1 + \sum_{k=2}^{\infty}\frac{1}{k!}\sum_{j=1}^{B} p_j\left(\frac{\alpha}{m}\left(\frac{p'_j}{p_j}-1\right)\right)^{k} \\
&\leq 1 + \sum_{k=2}^{\infty}\frac{1}{k!}\sum_{j=1}^{B} p_j\left(\frac{\alpha(e^{\varepsilon_0}-1)}{m}\right)^{k} \\
&= 1 + \sum_{k=1}^{\infty}\frac{1}{k!}\left(\frac{\alpha(e^{\varepsilon_0}-1)}{m}\right)^{k} - \frac{\alpha(e^{\varepsilon_0}-1)}{m} \\
&= e^{\frac{\alpha\left(e^{\epsilon_0}-1\right)}{m}} - \frac{\alpha\left(e^{\epsilon_0}-1\right)}{m}, \qquad\qquad (\text{D.16})
\end{aligned}$$

where the inequality follows from $\frac{p'_j}{p_j} \leq e^{\varepsilon_0}$, which holds for all $j \in [B]$. Substituting from (D.16) into (D.15), we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}_m)}\left[\left(\frac{\mathcal{M}\left(\mathcal{D}'_m\right)(\boldsymbol{h})}{\mathcal{M}\left(\mathcal{D}_m\right)(\boldsymbol{h})}\right)^{\alpha}\right] &\leq \left(e^{\frac{\alpha\left(e^{\epsilon_0}-1\right)}{m}} - \frac{\alpha\left(e^{\epsilon_0}-1\right)}{m}\right)^{m} \\
&= e^{\alpha\left(e^{\epsilon_0}-1\right)}\left[1 - \frac{\alpha\left(e^{\epsilon_0}-1\right)}{m}e^{\frac{-\alpha\left(e^{\epsilon_0}-1\right)}{m}}\right]^{m} \\
&\leq e^{\alpha\left(e^{\epsilon_0}-1\right)}e^{-\alpha\left(e^{\epsilon_0}-1\right)e^{\frac{-\alpha\left(e^{\epsilon_0}-1\right)}{m}}} \qquad (\text{since } 1-x \leq e^{-x}) \\
&= e^{\alpha\left(e^{\epsilon_0}-1\right)\left[1-e^{\frac{-\alpha\left(e^{\epsilon_0}-1\right)}{m}}\right]} \\
&\leq e^{\frac{\alpha^2\left(e^{\epsilon_0}-1\right)^2}{m}}. \qquad (\text{since } 1-e^{-x} \leq x)
\end{aligned}$$

This completes the proof of Theorem 5.3.6.

D.3 Omitted Details from Section 5.3.2

D.3.1 Proof of Lemma 5.3.1

We only show (5.23); (5.24) can be shown similarly. For convenience, for any $\mathcal{C} \subseteq [n-1]$, define

$$\begin{aligned} \mathcal{P}'_{|\mathcal{C}|,n} &= \{\boldsymbol{p}'_n, \ldots, \boldsymbol{p}'_n\} \text{ with } |\mathcal{P}'_{|\mathcal{C}|,n}| = |\mathcal{C}|, \\ \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} &= \{\tilde{\boldsymbol{p}}_i : i \in [n-1] \setminus \mathcal{C}\}. \end{aligned}$$

With these notations, we can write $\mathcal{P}_{\mathcal{C}} = \mathcal{P}'_{|\mathcal{C}|,n} \bigcup \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}_n\}$ and $\mathcal{P}'_{\mathcal{C}} = \mathcal{P}'_{|\mathcal{C}|,n} \bigcup \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}'_n\}$. Note that $\boldsymbol{p}_i = q\boldsymbol{p}'_n + (1-q)\tilde{\boldsymbol{p}}_i$ for all $i \in [n-1]$. For any $i \in [n-1]$, define the following random variable $\widehat{\boldsymbol{p}}_i$:

$$\widehat{\boldsymbol{p}}_i = \begin{cases} \boldsymbol{p}'_n & \text{w.p. } q, \\ \widetilde{\boldsymbol{p}}_i & \text{w.p. } 1-q. \end{cases}$$

Note that $\mathbb{E}[\widehat{\boldsymbol{p}}_i] = \boldsymbol{p}_i$. For any subset $\mathcal{C} \subseteq [n-1]$, define an event $\mathcal{E}_{\mathcal{C}} := \{\widehat{\boldsymbol{p}}_i = \boldsymbol{p}'_n \text{ for } i \in \mathcal{C} \text{ and } \widehat{\boldsymbol{p}}_i = \widetilde{\boldsymbol{p}}_i \text{ for } i \in [n-1] \setminus \mathcal{C}\}$. Since $\widehat{\boldsymbol{p}}_1, \ldots, \widehat{\boldsymbol{p}}_{n-1}$ are independent random variables, we have $\Pr[\mathcal{E}_{\mathcal{C}}] = q^{|\mathcal{C}|}(1-q)^{n-|\mathcal{C}|-1}$.

Consider an arbitrary $\boldsymbol{h} \in \mathcal{A}_B^n$. Define a random variable $U(\mathcal{P})$ over $\mathcal{A}_B^n$ whose distribution is equal to $F(\mathcal{P})$.

$$\begin{aligned} F(\mathcal{P})(\boldsymbol{h}) &= \Pr[U(\mathcal{P}) = \boldsymbol{h}] \\ &= \Pr[U(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}, \boldsymbol{p}_n) = \boldsymbol{h}] \\ &= \Pr\left[U\left(\mathbb{E}[\widehat{\boldsymbol{p}}_1], \ldots, \mathbb{E}[\widehat{\boldsymbol{p}}_{n-1}], \boldsymbol{p}_n\right) = \boldsymbol{h}\right] \\ &= \sum_{\mathcal{C}\subseteq[n-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U\left(\mathbb{E}[\widehat{\boldsymbol{p}}_1], \ldots, \mathbb{E}[\widehat{\boldsymbol{p}}_{n-1}], \boldsymbol{p}_n\right) = \boldsymbol{h} \mid \mathcal{E}_{\mathcal{C}}\right] \\ &\overset{\text{(e)}}{=} \sum_{\mathcal{C}\subseteq[n-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U\left(\mathcal{P}'_{|\mathcal{C}|,n} \bigcup \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}_n\}\right) = \boldsymbol{h}\right] \\ &= \sum_{\mathcal{C}\subseteq[n-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U(\mathcal{P}_{\mathcal{C}}) = \boldsymbol{h}\right] \end{aligned}$$

$$= \sum_{\mathcal{C}\subseteq[n-1]} q^{|\mathcal{C}|}(1-q)^{n-|\mathcal{C}|-1} \Pr\left[U(\mathcal{P}_{\mathcal{C}}) = \boldsymbol{h}\right],$$
$$= \sum_{\mathcal{C}\subseteq[n-1]} q^{|\mathcal{C}|}(1-q)^{n-|\mathcal{C}|-1} F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h}) \tag{D.17}$$

where, $\mathcal{P}'_{|\mathcal{C}|,n}$ and $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}}$ in the RHS of (e) are defined in the statement of the claim. Since the above calculation holds for every $\boldsymbol{h} \in \mathcal{A}_B^n$, we have proved (5.23).

D.3.2 Proof of Lemma 5.3.2

For simplicity of notation, let $P = F(\mathcal{P})$ and $Q = F(\mathcal{P}')$. Note that $\mathbb{E}_Q\left[\left(\frac{P}{Q}\right)^{\alpha}\right] = \int P^{\alpha}Q^{1-\alpha}d\mu$, which is also called the Hellinger integral. In order to prove the lemma, it suffices to show that $\int P^{\alpha}Q^{1-\alpha}d\mu$ is jointly convex in (P,Q), i.e., if $P_\lambda = \lambda P_0 + (1-\lambda)P_1$ and $Q_\lambda = \lambda Q_0 + (1-\lambda)Q_1$ for some $\lambda \in [0,1]$, then the following holds

$$\int P_\lambda^{\alpha} Q_\lambda^{1-\alpha} d\mu \leq \lambda \int P_0^{\alpha} Q_0^{1-\alpha} d\mu + (1-\lambda) \int P_1^{\alpha} Q_1^{1-\alpha} d\mu. \tag{D.18}$$

Proof of (D.18) is implicit in the proof of [EH14, Theorem 13]. However, for completeness, we prove (D.18) in Lemma D.3.1 below.

Since $P = F(\mathcal{P})$ and $Q = F(\mathcal{P}')$ are convex combinations of $P_{\mathcal{C}} = F(\mathcal{P}_{\mathcal{C}})$ and $Q_{\mathcal{C}} = F(\mathcal{P}'_{\mathcal{C}})$, respectively, with same coefficients, repeated application of (D.18) implies (5.25).

Lemma D.3.1. *For $\alpha \geq 1$, the Hellinger integral $\int P^{\alpha}Q^{1-\alpha}d\mu$ is jointly convex in (P,Q), i.e., if $P_\lambda = \lambda P_0 + (1-\lambda)P_1$ and $Q_\lambda = \lambda Q_0 + (1-\lambda)Q_1$ for some $\lambda \in [0,1]$, then we have*

$$\int P_\lambda^{\alpha} Q_\lambda^{1-\alpha} d\mu \leq \lambda \int P_0^{\alpha} Q_0^{1-\alpha} d\mu + (1-\lambda) \int P_1^{\alpha} Q_1^{1-\alpha} d\mu. \tag{D.19}$$

Proof. Let $f(x) = x^{\alpha}$. It is easy to show that for any $\alpha \geq 1$, $f(x)$ is a convex function when $x > 0$. This implies that for any point $\omega \in \Omega$ in the sample space, we have

$$f\left(\frac{P_\lambda(\omega)}{Q_\lambda(\omega)}\right) = f\left(\frac{\lambda P_0(\omega)}{Q_\lambda(\omega)} + \frac{(1-\lambda)P_1(\omega)}{Q_\lambda(\omega)}\right)$$
$$= f\left(\frac{\lambda Q_0(\omega)}{Q_\lambda(\omega)}\frac{P_0(\omega)}{Q_0(\omega)} + \frac{(1-\lambda)Q_1(\omega)}{Q_\lambda(\omega)}\frac{P_1(\omega)}{Q_1(\omega)}\right)$$

$$\leq \frac{\lambda Q_0(\omega)}{Q_\lambda(\omega)} f\left(\frac{P_0(\omega)}{Q_0(\omega)}\right) + \frac{(1-\lambda)Q_1(\omega)}{Q_\lambda(\omega)} f\left(\frac{P_1(\omega)}{Q_1(\omega)}\right),$$

where the last inequality follows from the convexity of $f(x)$. By multiplying both sides with $Q_\lambda(\omega)$ and substituting the definition of $f(x) = x^\alpha$, we get

$$P_\lambda^\alpha(\omega) Q_\lambda^{1-\alpha}(\omega) \leq \lambda P_0^\alpha(\omega) Q_0^{1-\alpha}(\omega) + (1-\lambda) P_1^\alpha(\omega) Q_1^{1-\alpha}(\omega).$$

By integrating this equality, we get (D.19). ■

D.3.3 Proof of Lemma 5.3.3

First we show that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^\alpha\right]$ is convex in $\boldsymbol{p}_i$ for any $i \in [n-1]$.

Note that due to the independence of $\mathcal{R}$ on different data points, for any $\boldsymbol{h} = (h_1, \ldots, h_B) \in \mathcal{A}_B^n$, we can recursively write the distributions $F(\mathcal{P})(\boldsymbol{h})$ and $F(\mathcal{P}')(\boldsymbol{h})$ (which are defined in (5.18)) as follows:

$$F(\mathcal{P})(\boldsymbol{h}) = \sum_{j=1}^{B} p_{ij} F(\mathcal{P}_{-i})(\widetilde{\boldsymbol{h}}_j), \qquad \forall i \in [n] \tag{D.20}$$

$$F(\mathcal{P}')(\boldsymbol{h}) = \sum_{j=1}^{B} p_{ij} F(\mathcal{P}'_{-i})(\widetilde{\boldsymbol{h}}_j) = \sum_{j=1}^{B} p'_{nj} F(\mathcal{P}'_{-n})(\widetilde{\boldsymbol{h}}_j),\ \forall i \in [n-1], \tag{D.21}$$

where $\widetilde{\boldsymbol{h}}_j = (h_1, \ldots, h_{j-1}, h_j - 1, h_{j+1}, \ldots, h_B)$ for any $j \in [B]$. Here, $F(\mathcal{P}_{-i})$, $F(\mathcal{P}'_{-i})$ are distributions over $\mathcal{A}_B^{n-1}$.[1]

Fix any $i \in [n-1]$ and also fix arbitrary $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{i-1}, \boldsymbol{p}_{i+1}, \ldots$
$, \boldsymbol{p}_n, \boldsymbol{p}'_n$. Take any $\lambda \in [0,1]$, and consider $\boldsymbol{p}_i^\lambda = \lambda \boldsymbol{p}_i^0 + (1-\lambda)\boldsymbol{p}_i^1$. Let $\mathcal{P}_\lambda = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^\lambda, \ldots, \boldsymbol{p}_n)$, $\mathcal{P}_0 = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^0, \ldots, \boldsymbol{p}_n)$, and $\mathcal{P}_1 = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^1, \ldots, \boldsymbol{p}_n)$. Similarly, let $\mathcal{P}'_\lambda = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^\lambda, \ldots, \boldsymbol{p}'_n)$, $\mathcal{P}'_0 = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^0, \ldots, \boldsymbol{p}'_n)$, and $\mathcal{P}'_1 = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_i^1, \ldots, \boldsymbol{p}'_n)$. With these definitions, we have $\mathcal{P}_\lambda = \lambda\mathcal{P}_0 + (1-\lambda)\mathcal{P}_1$. Note that $(\mathcal{P}_\lambda)_{-i} = (\mathcal{P}_0)_{-i} = (\mathcal{P}_1)_{-i}$.

[1] We assume that $F(\mathcal{P}_{-i})(\widetilde{\boldsymbol{h}}_j) = 0$ and $F(\mathcal{P}'_{-i})(\widetilde{\boldsymbol{h}}_j) = 0$ if $h_j - 1 < 0$.

Then, from the recursive definitions of $F\left(\mathcal{P}\right)$ and $F\left(\mathcal{P}'\right)$ (given in (D.20) and (D.21), respectively), for any $\boldsymbol{h} \in \mathcal{A}_B^n$, we get

$$\begin{aligned}
&F(\mathcal{P}_\lambda)(\boldsymbol{h}) = \sum_{j=1}^{B} p_{ij}^{\lambda} F\left((\mathcal{P}_\lambda)_{-i}\right)(\widetilde{\boldsymbol{h}}_j) \\
&= \lambda \sum_{j=1}^{B} p_{ij}^{0} F\left((\mathcal{P}_\lambda)_{-i}\right)(\widetilde{\boldsymbol{h}}_j) + (1-\lambda) \sum_{j=1}^{B} p_{ij}^{1} F\left((\mathcal{P}_\lambda)_{-i}\right)(\widetilde{\boldsymbol{h}}_j) \quad (\text{since } \boldsymbol{p}_i^{\lambda} = \lambda \boldsymbol{p}_i^{0} + (1-\lambda)\boldsymbol{p}_i^{1}) \\
&= \lambda \sum_{j=1}^{B} p_{ij}^{0} F\left((\mathcal{P}_0)_{-i}\right)(\widetilde{\boldsymbol{h}}_j) + (1-\lambda) \sum_{j=1}^{B} p_{ij}^{1} F\left((\mathcal{P}_1)_{-i}\right)(\widetilde{\boldsymbol{h}}_j) \\
&\hfill (\text{since } \left(\mathcal{P}_\lambda\right)_{-i} = \left(\mathcal{P}_0\right)_{-i} = \left(\mathcal{P}_1\right)_{-i}) \\
&= \lambda F(\mathcal{P}_0)(\boldsymbol{h}) + (1-\lambda) F(\mathcal{P}_1)(\boldsymbol{h}).
\end{aligned}$$

Similarly, we can show that $F(\mathcal{P}'_\lambda)(\boldsymbol{h}) = \lambda F(\mathcal{P}'_0)(\boldsymbol{h}) + (1-\lambda)F(\mathcal{P}'_1)(\boldsymbol{h})$.

Thus we have shown that

$$\begin{aligned}
F\left(\mathcal{P}_\lambda\right) &= \lambda F\left(\mathcal{P}_0\right) + (1-\lambda)\, F\left(\mathcal{P}_1\right) \\
F\left(\mathcal{P}'_\lambda\right) &= \lambda F\left(\mathcal{P}'_0\right) + (1-\lambda)\, F\left(\mathcal{P}'_1\right).
\end{aligned}$$

From Lemma D.3.1, we have that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\alpha}\right]$ is jointly convex in $F\left(\mathcal{P}\right)$ and $F\left(\mathcal{P}'\right)$. As a result, we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim F\left(\mathcal{P}'_\lambda\right)}\left[\left(\frac{F\left(\mathcal{P}_\lambda\right)(\boldsymbol{h})}{F\left(\mathcal{P}'_\lambda\right)(\boldsymbol{h})}\right)^{\alpha}\right] \leq{}& \lambda \mathbb{E}_{\boldsymbol{h}\sim F\left(\mathcal{P}'_0\right)}\left[\left(\frac{F\left(\mathcal{P}_0\right)(\boldsymbol{h})}{F\left(\mathcal{P}'_0\right)(\boldsymbol{h})}\right)^{\alpha}\right] \\
&+ (1-\lambda)\, \mathbb{E}_{\boldsymbol{h}\sim F\left(\mathcal{P}'_1\right)}\left[\left(\frac{F\left(\mathcal{P}_1\right)(\boldsymbol{h})}{F\left(\mathcal{P}'_1\right)(\boldsymbol{h})}\right)^{\alpha}\right]
\end{aligned} \tag{D.22}$$

Thus, we have shown that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\alpha}\right]$ is convex in $\boldsymbol{p}_i$ for any $i \in [n-1]$. Now we are ready to prove Lemma 5.3.3.

The LDP constraints put some restrictions on the set of values that the distribution $\boldsymbol{p}_i$ can take; however, the maximum value that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\alpha}\right]$ takes can only increase when we remove those constraints. We instead maximize it w.r.t. $\boldsymbol{p}_i$ over the simplex

$\Delta_B := \{(p_{i1}, \ldots, p_{iB}) : p_{ij} \geq 0 \text{ for } j \in [B] \text{ and } \sum_{j=1}^{B} p_{ij} = 1\}$. This implies

$$\mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})} \right)^{\alpha} \right] \leq \max_{\boldsymbol{p}_i \in \Delta_B} \mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})} \right)^{\alpha} \right] \tag{D.23}$$

Substituting from (D.20) and (D.21) into (D.23), we get

$$\mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})} \right)^{\alpha} \right] \leq \tag{D.24}$$

$$\max_{\boldsymbol{p}_i \in \Delta_B} \mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{\sum_{j=1}^{B} p_{ij} F(\mathcal{P}_{-i})(\widetilde{\boldsymbol{h}}_j)}{\sum_{j=1}^{B} p_{ij} F(\mathcal{P}'_{-i})(\widetilde{\boldsymbol{h}}_j)} \right)^{\alpha} \right] \tag{D.25}$$

Since maximizing a convex function over a polyhedron attains its maximum value at one of its vertices, and there are B vertices in the simplex Δ_B, which are of the form $p_{ij^*} = 1$ for some $j^* \in [B]$ and $p_{ik} = 0$ for all $k \neq j^*$, we have

$$\begin{aligned} &\max_{\boldsymbol{p}_i \in \Delta_B} \mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{\sum_{j=1}^{B} p_{ij} F(\mathcal{P}_{-i})(\widetilde{\boldsymbol{h}}_j)}{\sum_{j=1}^{B} p_{ij} F(\mathcal{P}'_{-i})(\widetilde{\boldsymbol{h}}_j)} \right)^{\alpha} \right] \\ &\overset{\text{(a)}}{=} \mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}')} \left[\left(\frac{F(\mathcal{P}_{-i})(\widetilde{\boldsymbol{h}}_{j^*})}{F(\mathcal{P}'_{-i})(\widetilde{\boldsymbol{h}}_{j^*})} \right)^{\alpha} \right] \\ &\overset{\text{(b)}}{=} \mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}'_{-i})} \left[\left(\frac{F(\mathcal{P}_{-i})(\boldsymbol{h})}{F(\mathcal{P}'_{-i})(\boldsymbol{h})} \right)^{\alpha} \right] \end{aligned}$$

Since the i'th data point deterministically maps to the j^*'th output by the mechanism $\mathcal{R}$, the expectation term in the RHS of (a) has no dependence on the i'th data point, so we can safely remove that, which gives (b). This proves Lemma 5.3.3.

D.3.4 Proof of Corollary 5.3.3

Recall from Lemma 5.3.1 and the notation defined in Appendix D.3, that for any $\mathcal{C} \subseteq [n-1]$, we have $\mathcal{P}_{\mathcal{C}} = \mathcal{P}'_{|\mathcal{C}|,n} \bigcup \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}_n\}$ and $\mathcal{P}'_{\mathcal{C}} = \mathcal{P}'_{|\mathcal{C}|,n} \bigcup \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}'_n\}$, where $\mathcal{P}'_{|\mathcal{C}|,n} = \{\boldsymbol{p}'_n, \ldots, \boldsymbol{p}'_n\}$ with $|\mathcal{P}'_{|\mathcal{C}|,n}| = |\mathcal{C}|$ and $\widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}} = \{\tilde{\boldsymbol{p}}_i : i \in [n-1] \setminus \mathcal{C}\}$.

Now, repeatedly applying Lemma 5.3.3 over the set of distributions $\tilde{\boldsymbol{p}}_i \in \widetilde{\mathcal{P}}_{[n-1]\setminus\mathcal{C}}$, we get that

$$\mathbb{E}_{\boldsymbol{h} \sim F(\mathcal{P}'_{\mathcal{C}})} \left[\left(\frac{F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h})}{F(\mathcal{P}'_{\mathcal{C}})(\boldsymbol{h})} \right)^{\alpha} \right]$$

$$\begin{aligned}
&\leq \mathbb{E}_{\boldsymbol{h}\sim F\left(\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}'_n\}\right)}\left[\left(\frac{F\left(\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}_n\}\right)(\boldsymbol{h})}{F\left(\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}'_n\}\right)(\boldsymbol{h})}\right)^{\alpha}\right]\\
&= \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})}\left[\left(\frac{\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})}\right)^{\alpha}\right]
\end{aligned}$$

In the last equality, we used that $\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}_n\}$ has $|\mathcal{C}|+1 = m+1$ distributions which are associated with the $(m+1)$ data points $\{d'_n, \ldots, d'_n, d_n\}$ (m of them are equal to d'_n); similarly, $\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}'_n\}$ also has $|\mathcal{C}|+1 = m+1$ distributions which are associated with the $(m+1)$ data points $\{d'_n, \ldots, d'_n, d'_n\}$ (all of them are equal to d'_n). This implies that for every $\boldsymbol{h} \in \mathcal{A}_B^{m+1}$, $F\left(\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}_n\}\right)(\boldsymbol{h})$ and $F\left(\mathcal{P}'_{|\mathcal{C}|,n}\bigcup\{\boldsymbol{p}'_n\}\right)(\boldsymbol{h})$ are distributionally equal to $\mathcal{M}(\mathcal{D}^{(n)}_{m+1})(\boldsymbol{h})$ and $\mathcal{M}(\mathcal{D}'^{(n)}_{m+1})(\boldsymbol{h})$, respectively. This proves Corollary 5.3.3.

D.4 Omitted Details from Section 5.3.3

D.4.1 Proof of Lemma 5.3.4

For simplicity of notation, let μ_0, μ_1 denote the distributions $\mathcal{M}(\mathcal{D}_m), \mathcal{M}(\mathcal{D}'_m)$, respectively. As shown in (5.33), for any $\boldsymbol{h} \in \mathcal{A}_B^m$, we have

$$X(\boldsymbol{h}) = m\left(\frac{\mu_1(\boldsymbol{h})}{\mu_0(\boldsymbol{h})} - 1\right) = \left(\sum_{j=1}^{B} a_j h_j\right) - m,$$

where $a_j = \frac{p'_j}{p_j} \in [e^{-\varepsilon_0}, e^{\varepsilon_0}]$ for all $j \in [B]$.

Now we show the three properties.

1. The mean of the random variable X is given by

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[X(\boldsymbol{h})\right] &= m\mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[\frac{\mu_1(\boldsymbol{h})}{\mu_0(\boldsymbol{h})} - 1\right]\\
&= m\sum_{\boldsymbol{h}\in\mathcal{A}_B^m}\mu_0(\boldsymbol{h})\left(\frac{\mu_1(\boldsymbol{h})}{\mu_0(\boldsymbol{h})} - 1\right)\\
&= m\sum_{\boldsymbol{h}\in\mathcal{A}_B^m}\left(\mu_1(\boldsymbol{h}) - \mu_0(\boldsymbol{h})\right) = 0
\end{aligned}$$

2. The variance of the random variable X is given by

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[X\left(\boldsymbol{h}\right)^2\right] &= \mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[\left(\sum_{j=1}^{B} a_j h_j - m\right)^2\right] \\
&= m^2 \mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[\sum_{j=1}^{B}\sum_{l=1}^{B} a_j a_l \frac{h_j h_l}{m^2} - 2\sum_{j=1}^{B} a_j \frac{h_j}{m} + 1\right] \\
&= m^2 \mathbb{E}_{\boldsymbol{h}\sim\mu_0}\left[\sum_{j=1}^{B} a_j^2 \frac{h_j^2}{m^2} + \sum_{j=1}^{B}\sum_{l\neq j} a_j a_l \frac{h_j h_l}{m^2} - 2\sum_{j=1}^{B} a_j \frac{h_j}{m} + 1\right] \\
&= m^2 \Big[\sum_{j=1}^{B} \frac{(p_j')^2}{p_j^2} \frac{\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_j^2]}{m^2} + \sum_{j=1}^{B}\sum_{l\neq j} \frac{p_j' p_l'}{p_j p_l} \frac{\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_j h_l]}{m^2} \\
&\qquad - 2\sum_{j=1}^{B} \frac{p_j'}{p_j} \frac{\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_j]}{m} + 1\Big] \\
&\overset{\text{(b)}}{=} m^2 \Big[\sum_{j=1}^{B} \frac{(p_j')^2}{p_j^2} \frac{(m p_j (1-p_j) + m^2 p_j^2)}{m^2} \\
&\qquad + \sum_{j=1}^{B}\sum_{l\neq j} \frac{p_j' p_l'}{p_j p_l} \frac{(-m p_j p_l + m^2 p_j p_l)}{m^2} - 2\sum_{j=1}^{B} \frac{p_j'}{p_j} \frac{p_j m}{m} + 1\Big] \\
&= m^2 \Big[\sum_{j=1}^{B} \left(\frac{(p_j')^2\,(1-p_j)}{p_j m} + (p_j')^2\right) \\
&\qquad + \sum_{j=1}^{B}\sum_{l\neq j} \left(-\frac{p_j' p_l'}{m} + p_j' p_l'\right) - 1\Big] \\
&= m^2 \Big[\frac{1}{m}\left(\sum_{j=1}^{B} \frac{(p_j')^2 (1-p_j)}{p_j} - \sum_{j=1}^{B}\sum_{l\neq j} p_j' p_l'\right) \\
&\qquad + \sum_{j=1}^{B} (p_j')^2 + \sum_{j=1}^{B}\sum_{l\neq j} p_j' p_l' - 1\Big] \\
&= m^2 \Big[\frac{1}{m}\left(\sum_{j=1}^{B} \frac{(p_j')^2}{p_j} - \sum_{j=1}^{B} (p_j')^2 - \sum_{j=1}^{B}\sum_{l\neq j} p_j' p_l'\right) \\
&\qquad + \sum_{j=1}^{B} (p_j')^2 + \sum_{j=1}^{B}\sum_{l\neq j} p_j' p_l' - 1\Big] \\
&\overset{\text{(c)}}{=} m \left(\sum_{j=1}^{B} \frac{(p_j')^2}{p_j} - 1\right).
\end{aligned}$$

Here, step (b) uses properties of multinomial distribution: $\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_j] = mp_j$, $\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_j^2] = mp_j(1-p_j) + m^2p_j^2$, and $\mathbb{E}_{\boldsymbol{h}\sim\mu_0}[h_jh_l] = -mp_jp_l + m^2p_jp_l$ for $j \neq l$. Step (c) follows because $\sum_{j=1}^{B}(p'_j)^2 + \sum_{j=1}^{B}\sum_{l\neq j} p'_jp'_l = \left(\sum_{j=1}^{B} p'_j\right)^2 = 1$, as $\boldsymbol{p'} = (p'_1, \ldots, p'_B)$ is a probability distribution.

3. Let Y_i denote the random variable associated with the output of the local randomizer at the i'th client. So, $\Pr\left[Y_i = j\right] = p_j$ for $j \in [B]$. Recall that h_j denote the number of clients that map to the j'th element from $[B]$. This implies that for any $j \in [B]$, we have $h_j = \sum_{i=1}^{m} \mathbf{1}_{\{Y_i=j\}}$. For any $i \in [m]$, define a random variable $X_i = \left(\sum_{j=1}^{B} a_j\mathbf{1}_{\{Y_i=j\}}\right) - 1$, where $a_j = \frac{p'_j}{p_j}$. Observe that $X_1, \ldots, X_m$ are zero mean i.i.d. random variables, because for any $i \in [m]$, we have $\mathbb{E}\left[X_i\right] = \left(\sum_{j=1}^{B} a_jp_j\right) - 1 = 0$. With these definitions, we can equivalently represent $X(\boldsymbol{h}) = \left(\sum_{j=1}^{B} a_jh_j\right) - m$ as $X(\boldsymbol{h}) = \sum_{i=1}^{m} X_i$, which is the sum of m zero mean i.i.d. r.v.s. Furthermore, since $a_j \in [e^{-\epsilon_0}, e^{\epsilon_0}]$ for any $j \in [B]$, we have $X_i \in [e^{-\epsilon_0} - 1, e^{\epsilon_0} - 1]$. Since any bounded r.v. $Z \in [a, b]$ is a sub-Gaussian r.v. with parameter $\frac{(b-a)^2}{4}$ (see [RH15, Lemma 1.8])), we have that X_i is a sub-Gaussian r.v. with parameter $\nu^2 = \frac{\left(e^{\epsilon_0} - e^{-\epsilon_0}\right)^2}{4}$, i.e.,

$$\mathbb{E}\left[e^{sX_i}\right] \leq e^{\frac{s^2\nu^2}{2}}, \qquad \forall s \in \mathbb{R}.$$

It follows that $X\left(\mathbf{h}\right) = \sum_{i=1}^{m} X_i$ is also a sub-Gaussian random variable with parameter $m\nu^2$. The remaining steps are similar to bound the moments of a sub-Gaussian random variable. We write them here for completeness. From Chernoff bound we get

$$\begin{aligned}\Pr\left[X \geq t\right] &\leq \min_{s\geq 0} \frac{\mathbb{E}\left[e^{sX}\right]}{e^{st}} \\ &\leq \min_{s\geq 0} \frac{e^{\frac{s^2m\nu^2}{2}}}{e^{st}} \\ &\overset{\text{(b)}}{\leq} e^{-\frac{t^2}{2m\nu^2}}\end{aligned}$$

where (b) follows by setting $s = \frac{t}{m\nu^2}$. Similarly, we can bound the term $\Pr\left[-X \geq t\right]$. Thus, we get

$$\Pr\left[|X| \geq t\right] \leq 2e^{-\frac{t^2}{2m\nu^2}}$$

Hence, the i'th moment of the random variable X can be bounded by

$$\begin{aligned}
\mathbb{E}\left[X^i\right] &\leq \mathbb{E}\left[|X|^i\right] \\
&= i\int_0^\infty t^{i-1}\Pr\left[|X|\geq t\right]dt \\
&\leq 2i\int_0^\infty t^{i-1}e^{-\frac{t^2}{2m\nu^2}}dt \\
&\overset{\text{(b)}}{=} i\left(2m\nu^2\right)^{i/2}\int_0^\infty u^{i/2-1}e^{-u}du \\
&= i\left(2m\nu^2\right)^{i/2}\Gamma\left(i/2\right),
\end{aligned}$$

where step (b) follows by setting $u = \frac{t^2}{2m\nu^2}$ (change of variables). In the last step, $\Gamma\left(z\right) = \int_0^\infty x^{z-1}e^{-x}dx$ denotes the Gamma function. Thus, we conclude that for every $i \geq 3$, we have $\mathbb{E}\left[|X|^i\right] \leq i\Gamma\left(i/2\right)\left(2m\nu^2\right)^{i/2}$, where $\nu^2 = \frac{\left(e^{\varepsilon_0}-e^{-\epsilon_0}\right)^2}{4}$.

This completes the proof of Lemma 5.3.4.

D.4.2 Proof of Lemma 5.3.5

For any $(\boldsymbol{p},\boldsymbol{p}')\in\mathcal{T}_{\varepsilon_0}$, define $f(\boldsymbol{p},\boldsymbol{p}') = \sum_{j=1}^{B}\frac{(p_j')^2}{p_j}$. Since the function $g\left(x,y\right) = \frac{x^2}{y}$ is convex in (x,y) for $y>0$, it implies that the objective function $f(\boldsymbol{p},\boldsymbol{p}')$ is also convex in $(\boldsymbol{p},\boldsymbol{p}')$. It is easy to verify that $\mathcal{T}_{\varepsilon_0}$ is a polytope.

Since we maximize a convex function $f(\boldsymbol{p},\boldsymbol{p}')$ over a polytope $\mathcal{T}_{\varepsilon_0}$, the optimal solution is one of the vertices of the polytope. Note that any vertex $(\boldsymbol{p},\boldsymbol{p}')$ of the polytope in B dimensions satisfies all the B LDP constraints (i.e., $e^{-\varepsilon_0}\leq\frac{p_j}{p_j'}\leq e^{\varepsilon_0}, j=1,\ldots,B$) with equality. Without loss of generality, assume that the optimal solution $(\tilde{\boldsymbol{p}},\tilde{\boldsymbol{p}}')$ is a vertex such that $\frac{\tilde{p}_j'}{\tilde{p}_j} = e^{\varepsilon_0}$ for $j=1,\ldots,l$ and $\frac{\tilde{p}_j'}{\tilde{p}_j} = e^{-\varepsilon_0}$ for $j=l+1,\ldots,B$, for some $l\in[B]$. Thus, we have

$$\begin{aligned}
1 = \sum_{j=1}^{B}\tilde{p}_j' &= e^{\varepsilon_0}\sum_{j=1}^{l}\tilde{p}_j + e^{-\varepsilon_0}\sum_{j=l+1}^{B}\tilde{p}_j \\
&= e^{\varepsilon_0}\sum_{j=1}^{l}\tilde{p}_j + e^{-\varepsilon_0}\Big(1-\sum_{j=1}^{l}\tilde{p}_j\Big) = e^{-\varepsilon_0} + \left(e^{\varepsilon_0}-e^{-\varepsilon_0}\right)\sum_{j=1}^{l}\tilde{p}_j
\end{aligned}$$

Rearranging the above gives $\sum_{j=1}^{l} \tilde{p}_j = \frac{1}{e^{\varepsilon_0}+1}$. This implies $\sum_{j=1}^{l} \tilde{p}'_j = \frac{e^{\varepsilon_0}}{e^{\varepsilon_0}+1}$, which in turn implies $\sum_{j=l+1}^{B} \tilde{p}'_j = \frac{1}{e^{\varepsilon_0}+1}$. Now the result follows from the following set of equalities:

$$\begin{aligned}
f\left(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{p}}'\right) &= \sum_{j=1}^{B} \frac{(\tilde{p}'_j)^2}{\tilde{p}_j} = \sum_{j=1}^{l} \frac{\tilde{p}'_j}{\tilde{p}_j}\tilde{p}'_j + \sum_{j=l+1}^{B} \frac{\tilde{p}'_j}{\tilde{p}_j}\tilde{p}'_j \\
&= e^{\varepsilon_0} \sum_{j=1}^{l} \tilde{p}'_j + e^{-\varepsilon_0} \sum_{j=l+1}^{B} \tilde{p}'_j \\
&= \frac{e^{2\varepsilon_0}}{e^{\varepsilon_0}+1} + \frac{1}{e^{\varepsilon_0}\left(e^{\varepsilon_0}+1\right)} = \frac{(e^{\varepsilon_0})^3+1}{e^{\varepsilon_0}(e^{\varepsilon_0}+1)} = \frac{\left(e^{\varepsilon_0}-1\right)^2}{e^{\varepsilon_0}} + 1,
\end{aligned}$$

where the last equality uses the identity $x^3 + 1 = (x+1)(x^2 - x + 1)$. This completes the proof of Lemma 5.3.5.

APPENDIX E

Omitted Details From Chapter 6

E.1 Regret and Privacy Analysis of The Central DP Model (Proof of Theorem 6.3.1)

In this section, we prove the regret bound and the privacy guarantees of the central DP algorithm. We present the privacy analysis in Section E.1.1 and the regret analysis in Section E.1.2.

E.1.1 Privacy Analysis

We first show that Algorithm 6.3.1 is ε-DP. Let $\bar{r}_i = [\bar{r}_{ia_1}, ..., \bar{r}_{ia_{|\mathcal{C}_i|}}]$, $\hat{r}_i = [\hat{r}_{ia_1}, ..., \hat{r}_{ia_{|\mathcal{C}_i|}}] = \bar{r}_i + z_i, z_i = [z_{ia_1}, ..., z_{ia_{|\mathcal{C}_i|}}]$, where $a_1, ..., a_{|\mathcal{C}_i|}$ is an enumeration of the elements of $\mathcal{C}_i$. We construct the concatenated reward vector denoted by $\bar{r} = [\bar{r}_1, ..., \bar{r}_{\log(T)-1}]$, and let $\hat{r} = [\hat{r}_1, ..., \hat{r}_{\log(T)-1}] = \bar{r} + z, z = [z_1, ..., z_{\log(T)-1}]$.

Now consider two neighboring sequence of rewards $\mathcal{R}, \mathcal{R}'$, that only differ in r_k, r'_k, with corresponding concatenated reward vectors $\bar{r}, \bar{r}'$. We notice that each reward in $\mathcal{R}$ appears once in $\bar{r}$, and similarly for $\mathcal{R}', \bar{r}'$. Thus, we get:

$$\|\bar{r} - \bar{r}'\|_1 \leq \max_{r_k, r'_k} |r_k - r'_k| \leq 1, \tag{E.1}$$

where the last inequality follows from Assumption 2 with bounded rewards $|r_k| \leq 1$. Then, from [DMN06, Theorem 3.6], $\hat{r}$ is ε-DP. We notice that the output of Algorithm 6.3.1 depends on $r_1, ..., r_T$ only through $\hat{r}$. Hence, by post processing, Algorithm 6.3.1 is ε-DP.

E.1.2 Regret Analysis

We next prove the regret bound of Algorithm 6.3.1 for stochastic linear bandits.

Our analysis follows the known confidence bound technique in [ACF95] by designing confidence intervals (in step 5) that take into consideration the privacy effect.

Let $K = (3T)^d$ be the size of the $\frac{1}{T}$-net set $\mathcal{N}_{1/T}$ from Lemma 6.3.1. We first bound the following regret:

$$\tilde{R}_T = T \max_{a \in \mathcal{N}_{1/T}} \langle a, \theta_* \rangle - \sum_{t=1}^{T} \langle a_t, \theta_* \rangle, \tag{E.2}$$

where $a_1, a_2, \dots, a_T \in \mathcal{N}_{1/T}$. We then bound the regret R_T by showing that we only loose a constant term when we choose actions from $\mathcal{N}_{1/T}$ instead of the bigger set $\mathcal{A}$.

We start with a set of actions $\mathcal{A}_0 = \mathcal{N}_{1/T}$ with cardinality $|\mathcal{A}_0| = K$. Furthermore, we have $|\mathcal{A}_i| \leq |\mathcal{A}_{i-1}|$, and hence, we get $|\mathcal{A}_i| \leq K$ for all $i \in [\log(T)]$.

For given batch $i \in [\log(T)]$, let $\mathcal{C}_i$ be the core set of $\mathcal{A}_i$ that has at most Bd actions. At the ith batch, each action $a \in \mathcal{C}_i$ is picked n_{ia} times, where $n_{ia} = \lceil \pi_i(a) q^i \rceil$. Let $\mathcal{G}$ be the good event $\left\{ \left| \langle a, \hat{\theta}_i - \theta_* \rangle \right| < \gamma_i \ \forall i \in [\log T] \ \forall a \in \mathcal{A}_i \right\}$. Lemma E.1.1 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof, we condition on the event $\mathcal{G}$.

We first show that the best action $a_* = \arg\max_{a \in \mathcal{N}_{1/T}} \langle a, \theta_* \rangle$ will not be eliminated at any batch $i \in [\log T]$; this is because the elimination criterion will not hold for the optimal action a_*:

$$\langle a, \hat{\theta}_i \rangle - \langle a_*, \hat{\theta}_i \rangle < (\langle a, \theta_* \rangle + \gamma_i) - (\langle a_*, \theta_* \rangle - \gamma_i) \leq 2\gamma_i \qquad \forall a \in \mathcal{A}_i \ \forall i \in [\log T]. \tag{E.3}$$

For each sub-optimal action $a \in \mathcal{A}_0$ with $\Delta_a = \langle a_* - a, \theta_* \rangle$, let i be the smallest integer for which $\gamma_i < \frac{\Delta_a}{4}$. From the triangle inequality, we get that

$$\langle a_*, \hat{\theta}_i \rangle - \langle a, \hat{\theta}_i \rangle \geq (\langle a_*, \hat{\theta}_* \rangle - \gamma_i) - (\langle a, \hat{\theta}_i \rangle + \gamma_i) = \Delta_a - 2\gamma_i > 2\gamma_i. \tag{E.4}$$

This implies that a will be eliminated before the beginning of batch $i+1$. Hence, each action $a \in \mathcal{A}_{i+1}$ at batch $i+1$ has a gap at most $4\gamma_i$. Let $n_i = \sum_{a \in \mathcal{C}_i} n_{ia} \leq Bd + q^i$ denote the total number of rounds at the i-th batch. Note that the number of batches is upper bounded by $\log T$ since $\sum_{i=1}^{\log T} q^i \geq T$. When $q^i < Bd$, the regret can be bounded by $2Bd$, and when $q^i \geq Bd$, we bound $n_i \leq 2q^i$. Thus, there is universal constants C', C such that the total regret in (E.2) can be bounded as

$$\begin{aligned}
\tilde{R}_T &\leq 2Bd\log(T) + \sum_{i=1}^{\log T} 4n_i\gamma_{i-1} && \text{(E.5)}\\
&\leq 2Bd\log(T) + \sum_{i=1}^{\log T} 8q^i \left(\sqrt{\frac{4d}{q^{i-1}} \log\left(4KT^2\right)} + \frac{2Bd^2 + 2d\log\left(4KT^2\right)}{\varepsilon q^{i-1}} \right)\\
&\leq C' \left(d\log(T) + d\sqrt{\log T} \sum_{i=1}^{\log T} q^{(i-1)/2} + \frac{d^2 \log^2 T}{\varepsilon} \right) q\\
&\overset{(a)}{\leq} C'q \left(d\log(T) + d\sqrt{\log T} q^{\log T/2} + \frac{d^2\log^2 T}{\varepsilon} \right)\\
&\overset{(b)}{\leq} C'q \left(d\log(T) + d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon} \right)\\
&\overset{(c)}{\leq} C \left(d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon} \right), && \text{(E.6)}
\end{aligned}$$

where step (a) follows from the sum of a geometric series and $q > 1$, step (b) uses $q = (2T)^{1/\log T}$, and step (c) follows from the facts $q \leq e^2$, $\log T = O(\sqrt{T})$.

Hence, with probability at least $1 - \frac{1}{T}$ the regret in (E.2) is bounded as

$$\tilde{R}_T \leq C \left(d\sqrt{T\log T} + \frac{d^2\log^2 T}{\varepsilon} \right). \tag{E.7}$$

Next, we bound the exact regret R_T. Observe that the first step in our Algorithm is to use the finite $\frac{1}{T}$-net set $\mathcal{N}_{1/T}$ of actions. Thus, for any round $t \in [T]$ and any action $a \in \mathcal{A}$, there exists an action $a' \in \mathcal{N}_{1/T}$ such that $\|a - a'\| \leq \frac{1}{T}$. As a result, we get

$\langle a, \theta_* \rangle - \langle a', \theta_* \rangle \leq \|a - a'\| \|\theta_*\| \leq \frac{1}{T}$, where $\|\theta_*\| \leq 1$. Hence, there is a universal constant C such that we can bound the regret R_T as

$$\begin{aligned} R_T &= T \max_{a \in \mathcal{A}} \langle a, \theta_* \rangle - \sum_{t=1}^{T} \langle a_t, \theta_* \rangle \\ &= \left[T \max_{a \in \mathcal{A}} \langle a, \theta_* \rangle - T \max_{a' \in \mathcal{N}_{1/T}} \langle a', \theta_* \rangle \right] + \left[T \max_{a' \in \mathcal{N}_{1/T}} \langle a', \theta_* \rangle - \sum_{t=1}^{T} \langle a_t, \theta_* \rangle \right] \\ &\leq T \frac{1}{T} + \tilde{R}_T \\ &= 1 + \tilde{R}_T. \end{aligned} \tag{E.8}$$

Hence, with probability at least $1 - \frac{1}{T}$ the regret R_T is bounded as

$$R_T \leq C \left(d\sqrt{T \log T} + \frac{d^2 \log^2 T}{\varepsilon} \right). \tag{E.9}$$

This concludes the proof of Theorem 6.3.1.

Lemma E.1.1. *Let $\hat{\theta}_i$ be the least square estimate of θ_* at the end of the ith batch of Algorithm 6.3.1. Then, we have that*

$$\Pr \left[\left| \langle a, \hat{\theta}_i - \theta_* \rangle \right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i \right] \leq \frac{1}{T}, \tag{E.10}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i} \log \left(4KT^2\right)} + \frac{2Bd^2 + 2d \log\left(4KT^2\right)}{\varepsilon q^i}$.

Proof. Let $\hat{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} \hat{r}_{ia} a$ be the private estimate of θ_* and $\overline{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} \overline{r}_{ia} a$ be the non-private estimate of θ_* as $\{\overline{r}_{ia}\}$ are the non-private rewards, where $V_i = \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top$. From [Chapter 21, Eqn 21.1], for each $a \in \mathcal{A}_i$, we get:

$$\Pr \left[\langle a, \overline{\theta}_i - \theta_* \rangle \geq \sqrt{2 \|a\|^2_{V_i^{-1}} \log \left(\frac{1}{\beta} \right)} \right] \leq \beta, \tag{E.11}$$

where $\beta \in (0, 1)$ and $\|a\|^2_{V_i^{-1}} = a^\top V_i^{-1} a$. Let $V_i(\pi_i) = \sum_{a \in \mathcal{C}_i} \pi_i(a) a a^\top$ and hence we have

$$V_i = \sum_{a \in \mathcal{C}_i} n_{ia} a a^\top \geq q^i \sum_{a \in \mathcal{C}_i} \pi_i(a) a a^\top = q^i V_i(\pi_i). \tag{E.12}$$

Observe that for any symmetric random variable x if $\Pr[x \geq t] \leq \beta$, then $\Pr[|x| \geq t] = \Pr[x \geq t] + \Pr[-x \geq t] \leq 2\beta$. Thus, from lemma 6.3.2, we have $\|a\|^2_{V_i^{-1}} = \frac{1}{q^i} a^\top V_i(\pi_i)^{-1} a \leq \frac{2d}{q^i}$ for each $a \in \mathcal{A}_i$. By setting $\beta = \frac{1}{4KT^2}$ and $\|a\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{A}_i$ in (E.11), we get that:

$$\Pr\left[\left| \langle a, \bar{\theta}_i - \theta_* \rangle \right| \geq \sqrt{\frac{4d}{q^i} \log\left(4KT^2\right)} \right] \leq \frac{1}{2KT^2}, \tag{E.13}$$

for each $a \in \mathcal{A}_i$. Now, we compute the effect of the privacy in estimating θ_* by bounding difference $\langle a, \bar{\theta}_i - \hat{\theta}_i \rangle$. Observe that $\hat{r}_{ia} = \bar{r}_{ia} + z_{ia}$, where $z_{ia} \sim \mathsf{Lap}(\frac{1}{\varepsilon})$, and hence, we can write $\hat{\theta}_i - \bar{\theta}_i = V_i^{-1} \sum_{a \in \mathcal{C}_i} z_{ia} a$. Thus, for any $\alpha \in \mathcal{A}_i$, we have that:

$$\langle \alpha, \hat{\theta}_i - \bar{\theta}_i \rangle = \sum_{a \in \mathcal{C}_i} \alpha^\top V_i^{-1} a z_{ia}, \tag{E.14}$$

where $\alpha^\top V_i^{-1} a \leq \max_{b \in \mathcal{A}_i} \|b\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{C}_i$ that holds from the fact that V_i is positive semi-definite. From Lemma E.1.2 presented at the end of the section, by setting $b = \varepsilon$, $n = Bd$, $c = \frac{2d}{q^i}\sqrt{n}$, and $t = 2\frac{Bd^2}{\varepsilon q^i} + \frac{2d\log\left(4KT^2\right)}{\varepsilon q^i}$, we get that:

$$\Pr\left[\left| \langle a, \bar{\theta}_i - \hat{\theta}_i \rangle \right| \geq 2\frac{Bd^2}{\varepsilon q^i} + \frac{2d \log\left(4KT^2\right)}{\varepsilon q^i} \right] \leq \frac{1}{2KT^2}, \tag{E.15}$$

Then, by the union bound and triangle inequality we have that

$$\Pr\left[\left| \langle a, \hat{\theta}_i - \theta_* \rangle \right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i \right] \leq \frac{1}{T}, \tag{E.16}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i} \log\left(4KT^2\right)} + \frac{2Bd^2 + 2d\log\left(4KT^2\right)}{\varepsilon q^i}$. This concludes the proof of Lemma E.1.1. ■

Lemma E.1.2. *Let $x_i = l_i z_i$ for $i \in [n]$, where $z_i \sim \mathsf{Lap}(1/b)$ and l_i, c are constants such that $c^2 \geq \sum_{i=1}^n |l_i|^2$. Let $\bar{x} = \sum_{i=1}^n x_i$. We have that*

$$\Pr[\bar{x} \geq t] \leq \begin{cases} \exp\left(-\frac{t^2 b^2}{2c^2}\right) & \text{if } t \leq \frac{c^2}{b l_{\max}} \\ \exp\left(\frac{c^2}{2 l_{\max}^2} - \frac{b}{l_{\max}} t\right) & \text{if } t > \frac{c^2}{b l_{\max}} \end{cases}, \tag{E.17}$$

where $l_{\max} = \max_i l_i$.

The proof is provided in App. E.2.

E.2 Proof of Lemma E.1.2

Lemma. *Let $x_i = l_i z_i$ for $i \in [n]$, where $z_i \sim \mathsf{Lap}(1/b)$ and l_i, c are constants such that $c^2 \geq \sum_{i=1}^n |l_i|^2$. Let $\bar{x} = \sum_{i=1}^n x_i$. We have that*

$$\Pr[\bar{x} \geq t] \leq \begin{cases} \exp\left(-\frac{t^2 b^2}{2c^2}\right) & \textit{if } \ t \leq \frac{c^2}{b l_{\max}} \\ \exp\left(\frac{c^2}{2 l_{\max}^2} - \frac{b}{l_{\max}} t\right) & \textit{if } \ t > \frac{c^2}{b l_{\max}} \end{cases}, \tag{E.18}$$

where $l_{\max} = \max_i l_i$.

Proof. The proof follows from the concentration results of the Laplace distribution (e.g., see). We have that

$$\begin{aligned} \Pr\left[\bar{x} \geq t\right] &= \Pr\left[\exp\left(\lambda \bar{x}\right) \geq e^{\lambda t}\right] & \forall\, \lambda \geq 0 \\ &\overset{(a)}{\leq} \frac{\mathbb{E}\left[\exp\left(\lambda \bar{x}\right)\right]}{e^{\lambda t}} \\ &\overset{(b)}{=} \frac{\prod_{i=1}^n \mathbb{E}\left[e^{\lambda x_i}\right]}{e^{\lambda t}} \\ &\overset{(c)}{\leq} \frac{\prod_{i=1}^n e^{\lambda^2 \frac{l_i^2}{2b^2}}}{e^{\lambda t}} & \forall\, 0 \leq \lambda \leq \frac{b}{l_{\max}} \\ &= \frac{e^{\lambda^2 \frac{c^2}{2b^2}}}{e^{\lambda t}} & \forall\, 0 \leq \lambda \leq \frac{b}{l_{\max}} \end{aligned} \tag{E.19}$$

where $l_{\max} = \max_i l_i$, step (a) follows from Markov's inequality and step (b) follows from the fact that $z_1, \ldots, z_n$ are independent Laplace random variables. Step (c) follows from the fact that z_i is sub-exponential random variable with proxy $\frac{l_i^2}{2b^2}$. By choosing $\lambda = \frac{tb^2}{c^2}$ when $t < \frac{c^2}{b l_{\max}}$ and $\lambda = \frac{b}{l_{\max}}$ when $t > \frac{c^2}{b l_{\max}}$, we get that

$$\Pr[\bar{x} \geq t] \leq \begin{cases} \exp\left(-\frac{t^2 b^2}{2c^2}\right) & \text{if } \ t \leq \frac{c^2}{b l_{\max}} \\ \exp\left(\frac{c^2}{2 l_{\max}^2} - \frac{b}{l_{\max}} t\right) & \text{if } \ t > \frac{c^2}{b l_{\max}} \end{cases} \tag{E.20}$$

This completes the proof of Lemma E.1.2. ■

E.3 Regret and Privacy Analysis of The local DP Model (Proof of Theorem 6.4.1)

E.3.1 Privacy Analysis

The privacy proof is straightforward. For any client, since the reward is bounded by $|r| \leq 1$, the output $\hat{r} = r + \mathsf{Lap}(1/\varepsilon_0)$ is ε_0-LDP from [DMN06, Theorem 3.6].

E.3.2 Regret Analysis

We next prove the regret bound of Algorithm 6.3.2 for stochastic linear bandits with LDP. Our proof is similar to the proofs of the central DP Algorithm presented in Section E.1.2. Let $\tilde{R}_T$ be the regret defined in (E.2).

Let $\mathcal{G}$ be the good event $\left\{\left|\langle a, \hat{\theta}_i - \theta_* \rangle\right| < \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right\}$. Lemma E.3.1 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof we condition on the event $\mathcal{G}$. When $q^i < \max\{Bd, 2\log(4KT^2)\}$, the regret can be bounded by $\max\{Bd, 2\log(4KT^2)\}$, and when $q^i \geq \max\{Bd, 2\log(4KT^2)\}$, we bound $n_i \leq 2q^i$, and hence,

$$\gamma_i \leq \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{2d}{\varepsilon_0}\sqrt{\frac{\log(4KT^2)}{q^i}} \leq (1 + \frac{1}{\varepsilon_0})2d\sqrt{\frac{\log(4KT^2)}{q^i}}.$$

By following similar steps as in the central DP, we can show that there is universal constants C', C such that the total regret in (E.2) can be bounded as

$$\begin{aligned}
\tilde{R}_T &\leq (Bd + 2\log(4KT^2))\log(T) + \sum_{i=1}^{\log T} 4n_i\gamma_{i-1} \\
&\leq (Bd + 2\log(4KT^2))\log(T) + (1 + \frac{1}{\varepsilon_0})2d\sum_{i=1}^{\log T} 8q^i\sqrt{\frac{1}{q^{i-1}}\log\left(4KT^2\right)}
\end{aligned}$$

$$
\begin{aligned}
&\leq C'(1+\frac{1}{\varepsilon_0})\left(d\sqrt{d}\log^2(T)+d\sqrt{d\log T}\sum_{i=1}^{\log T}q^{(i-1)/2}\right)q\\
&\overset{(a)}{\leq} C'(1+\frac{1}{\varepsilon_0})q\left(d\sqrt{d}\log^2(T)+d\sqrt{d\log T}q^{\log T/2}\right)\\
&\overset{(b)}{\leq} C'(1+\frac{1}{\varepsilon_0})q\left(d\sqrt{d}\log^2(T)+d\sqrt{dT\log T}\right)\\
&\overset{(c)}{\leq} C(1+\frac{1}{\varepsilon_0})\left(d\sqrt{dT\log T}\right),
\end{aligned}
\tag{E.21}
$$

where step (a) follows from the sum of a geometric series and $q > 1$, step (b) uses $q = (2T)^{1/\log T}$, and step (c) follows from the facts $q \leq e^2$, $\log^2 T = O(\sqrt{T})$.

Hence, following similar steps as in the proof of the central DP algorithm, with probability at least $1-\frac{1}{T}$ the regret is bounded as

$$
R_T \leq \tilde{R}_T + 1 \leq C(1+\frac{1}{\varepsilon_0})\left(d\sqrt{dT\log T}\right). \tag{E.22}
$$

Lemma E.3.1. *Let $\hat{\theta}_i$ be the least square estimate of θ_* at the end of the ith batch of Algorithm 6.3.2. Then, we have that*

$$
\Pr\left[\left|\langle a,\hat{\theta}_i-\theta_*\rangle\right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \tag{E.23}
$$

where $\gamma_i = \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)} + \frac{1}{q^i\varepsilon_0}\sqrt{2dn_i\log(4KT^2)}$.

Proof. Let $\hat{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\hat{r}_{ia}a$ be the private estimate of θ_* and $\overline{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}\overline{r}_{ia}a$ be the non-private estimate of θ_* as $\{\overline{r}_{ia}\}$ are the non-private rewards, where $V_i = \sum_{a\in\mathcal{C}_i}n_{ia}aa^\top$ and $\hat{r}_{ia} = \sum_{j=1}^{n_{ia}}\hat{r}_{ia}^{(j)}$. Similar to the central DP in Section 6.3, we have that

$$
\Pr\left[\left|\langle a,\bar{\theta}_i-\theta_*\rangle\right| \geq \sqrt{\frac{4d}{q^i}\log\left(4KT^2\right)}\right] \leq \frac{1}{2KT^2}, \tag{E.24}
$$

for each $a \in \mathcal{A}_i$. Now, we compute the effect of the LDP in estimating θ_* by bounding difference $\langle a,\bar{\theta}_i-\hat{\theta}_i\rangle$. Observe that $\hat{r}_{ia} = \sum_{j=1}^{n_{ia}}\hat{r}_{ia}^{(j)} = \bar{r}_{ia} + z_{ia}$, where $\bar{r}_{ia} = \sum_{j=1}^{n_{ia}}r_{ia}^{(j)}$ and $z_{ia} = \sum_{j=1}^{n_{ia}}z_{ia}^{(j)}$, where $z_{ia}^{(j)} \sim \mathsf{Lap}(\frac{1}{\varepsilon_0})$. Hence, we can write $\hat{\theta}_i - \bar{\theta}_i = V_i^{-1}\sum_{a\in\mathcal{C}_i}z_{ia}a$. Thus, for any $\alpha \in \mathcal{A}_i$, we have that:

$$
\langle \alpha,\hat{\theta}_i-\bar{\theta}_i\rangle = \sum_{a\in\mathcal{C}_i}\sum_{j=1}^{n_{ia}}\alpha^\top V_i^{-1}az_{ia}^{(j)}, \tag{E.25}
$$

where $\alpha^\top V_i^{-1} a \leq \max_{b \in \mathcal{A}_i} \|b\|^2_{V_i^{-1}} \leq \frac{2d}{q^i}$ for each $a \in \mathcal{C}_i$ that holds from the fact that V_i is positive semi-definite. We also have that

$$\sum_{a \in \mathcal{C}_i} \sum_{j=1}^{n_{ia}} (\alpha^\top V_i^{-1} a)^2 = \sum_{a \in \mathcal{C}_i} \sum_{j=1}^{n_{ia}} \alpha^\top V_i^{-1} a a^\top V_i^{-1} \alpha = \alpha^\top V_i^{-1} \alpha \leq \frac{2d}{q^i} \tag{E.26}$$

From Lemma E.1.2 presented in Section 6.3, by setting $b = \varepsilon_0$, $n = n_i$, $c^2 = \frac{2d}{q^i}$, and $t = \frac{1}{q^i \varepsilon_0} \sqrt{2dn_i \log(4KT^2)}$, we get that:

$$\Pr\left[\left|\langle a, \bar{\theta}_i - \hat{\theta}_i \rangle\right| \geq \frac{1}{q^i \varepsilon_0} \sqrt{2dn_i \log(4KT^2)}\right] \leq \frac{1}{2KT^2}, \tag{E.27}$$

Then, by the union bound and triangle inequality we have that

$$\Pr\left[\left|\langle a, \hat{\theta}_i - \theta_* \rangle\right| > \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right] \leq \frac{1}{T}, \tag{E.28}$$

where $\gamma_i = \sqrt{\frac{4d}{q^i} \log\left(4KT^2\right)} + \frac{1}{q^i \varepsilon_0} \sqrt{2dn_i \log(4KT^2)}$. This concludes the proof of Lemma E.3.1.

■

E.4 Regret and Privacy Analysis of The Shuffled Model (Proof of Theorem 6.5.1)

E.4.1 Privacy Analysis

We note that the data of each user j can be represented as $\cup_{a \in \mathcal{C}_i} \{(a, r_a^{(j)})\}$. We observe that our scheme is equivalent to performing the following steps

- Each user $j \in [n_i]$ sends its data $\mathcal{D}_j = \cup_{a \in \mathcal{C}_i} \{(a, r_a^{(j)})\}$ to the shuffler.
- The shuffler randomly permutes the sets $\mathcal{D}_1, ..., \mathcal{D}_{n_i}$ to get $\mathcal{D}_{\pi(1)}, ..., \mathcal{D}_{\pi(n_i)}$.
- The shuffler reveals n_i action reward pairs $(a_1, \hat{r}_{ia_1}), ..., (a_{n_i}, \hat{r}_{ia_{n_i}})$, where $(a_j, \hat{r}_{ia_j}) \in \mathcal{D}_{\pi(j)}$, and $\hat{r}_{ia_j}$ is the LDP version of r_{ia_j} ($\hat{r}_{ia_j} = r_{ia_j} + \mathsf{Lap}(\frac{1}{\varepsilon_0^{(i)}})$).

Hence, we shuffle the data, then feed it to an LDP mechanism with LDP parameter $\varepsilon_0^{(i)}$ (as proved in Theorem 6.4.1). As a result, it follows from [FMT22] that the output of the shuffler is (ε_i, δ)-DP where

$$\varepsilon_i = \log\left(1 + \frac{e^{\varepsilon_0^{(i)}} - 1}{e^{\varepsilon_0^{(i)}} + 1}\left(\frac{8\sqrt{e^{\varepsilon_0^{(i)}}\log(4/\delta)}}{\sqrt{n_i}} + \frac{8e^{\varepsilon_0^{(i)}}}{n_i}\right)\right). \tag{E.29}$$

By the choice of $\varepsilon_0^{(i)}$ as an inverse of the function $f_{n_i,\delta}$, we have that $\varepsilon_i = \varepsilon$ for all $i \in [\log T]$.

We observe that for any neighboring datasets D, D', there is only one user data that is different between D, D'. That user appears in exactly one batch. It follows that Algorithm 6.4.1 is (ε, δ)-DP.

E.4.2 Regret Analysis

We next prove the regret bound of Algorithm 6.4.1 for stochastic linear bandits in the shuffled model. Our proof is similar to the proofs of the LDP Algorithm presented in Section E.3.2. Let $\tilde{R}_T$ be the regret defined in (E.2).

Let $\mathcal{G}$ be the good event $\left\{\left|\langle a, \hat{\theta}_i - \theta_*\rangle\right| < \gamma_i \ \forall i \in [\log T] \forall a \in \mathcal{A}_i\right\}$. Lemma E.3.1 shows that the event $\mathcal{G}$ holds with probability at least $1 - \frac{1}{T}$. In the remaining part of the proof we condition on the event $\mathcal{G}$. When $q^i < Bd$, the regret can be bounded by Bd. By following similar steps as in the central DP, we can show that there is universal constants C' such that the total regret in (E.2) can be bounded as

$$\begin{aligned}
\tilde{R}_T &\leq Bd\log(T) + \sum_{i=1}^{\log T} 4n_i\gamma_{i-1} \\
&\overset{(a)}{\leq} Bd\log(T) + \sum_{i=1}^{\log T} 8q^i\sqrt{\frac{4d}{q^{i-1}}\log\left(4KT^2\right)} + C'\frac{2d}{\varepsilon}\sum_{i=1}^{\log T} 8q\sqrt{\log\left(4KT^2\right)\log(1/\delta)} \\
&\leq C\left(d\sqrt{T\log T} + \frac{(d\log T)^{3/2}\sqrt{\log(1/\delta)}}{\varepsilon}\right),
\end{aligned} \tag{E.30}$$

where step (a) follows from the fact that from the privacy analysis, when $\varepsilon_0^{(i)} \leq 1$, we get that $\varepsilon = O(\varepsilon_0^{(i)}\sqrt{\frac{\log(1/\delta)}{n_i}})$.

Hence, following similar steps as in the proof of the central DP algorithm, with probability at least $1 - \frac{1}{T}$ the regret is bounded as

$$R_T \leq \tilde{R}_T + 1 \leq C \left(d\sqrt{T \log T} + \frac{(d \log T)^{3/2} \sqrt{\log(1/\delta)}}{\varepsilon} \right). \tag{E.31}$$

APPENDIX F

Omitted Details From Chapter 7

F.1 Lower Bound on The Minimax Risk Estimation Using Fisher Information

In this section, we introduce an alternative proof of Theorem 7.3.1. Our proof is inspired by the approach in [BHO19] that uses Fisher information to bound the minimax risk estimation under communication constraints. The main idea of our proof is to formulate a non-convex optimization problem to bound the Fisher information matrix under privacy and randomness constraints. Let $\overline{\mathcal{P}} \subset \Delta_k$ be a subset of simplex Δ_k defined by

$$\overline{\mathcal{P}} = \left\{ \mathbf{p} \in \mathbb{R}^k : \sum_{j=1}^{k} p_j = 1,\ \frac{1}{k} \leq p_j \leq \frac{2}{k},\ p_{j+k/2} = \frac{2}{k} - p_j,\ \forall j \in [k/2] \right\}.$$

For every $\mathbf{p} \in \overline{\mathcal{P}}$, the number of free variables is $k/2$, where each parameter $p_{j+k/2}$ is associated with the variable p_j, $\forall\ j \in [k/2]$. For a given distribution $\mathbf{p} \in \Delta_k$, we define the marginal distribution on the output Y as

$$\mathbf{M}\left(y|\mathbf{p}\right) = \sum_{j=1}^{k} Q\left(Y = y|X = j\right) p_j. \tag{F.1}$$

Let $S_{\mathbf{p}}\left(y\right)$ denote the $k/2$-vector score function of Y given by

$$\begin{aligned} S_{\mathbf{p}}\left(y\right) &= \left[S_{p_1}\left(y\right), \ldots, S_{p_{k/2}}\left(y\right)\right] \\ &= \left[\frac{\partial \log\left(\mathbf{M}\left(y|\mathbf{p}\right)\right)}{\partial p_1}, \ldots, \frac{\partial \log\left(\mathbf{M}\left(y|\mathbf{p}\right)\right)}{\partial p_{k/2}}\right]. \end{aligned} \tag{F.2}$$

Then, the Fisher information matrix for estimating $\mathbf{p} \in \overline{\mathcal{P}}$ from Y is given by

$$I_Y(\mathbf{p}) = \mathbb{E}\left[S_{\mathbf{p}}(y)\, S_{\mathbf{p}}(y)^T\right], \tag{F.3}$$

where the expectation is taken over the randomness in the output Y. Now, consider the following inequalities

$$\begin{aligned} r^{\ell_2^2}_{\varepsilon,R,n,k} &= \inf_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \inf_{\hat{\mathbf{p}}} \sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\ell_2^2\left(\hat{\mathbf{p}}\left(\mathbf{Y}^n\right), \mathbf{p}\right)\right] \\ &\geq \inf_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \inf_{\hat{\mathbf{p}}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} \mathbb{E}\left[\ell_2^2\left(\hat{\mathbf{p}}\left(\mathbf{Y}^n\right), \mathbf{p}\right)\right] \\ &\overset{(a)}{\geq} \frac{(k/2)^2}{\sup\limits_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \sup\limits_{\mathbf{p}\in\overline{\mathcal{P}}} \operatorname{Tr}\left(I_{Y^n}(\mathbf{p})\right) + \frac{k}{2}\pi^2} \end{aligned} \tag{F.4}$$

where $I_{Y^n}(\mathbf{p})$ denotes the Fisher information matrix for estimating $\mathbf{p}$ from $Y^n = [Y_1, \ldots, Y_n]$, and $\operatorname{Tr}(I_{Y^n}(\mathbf{p}))$ denotes the trace of the Fisher information matrix $I_{Y^n}(\mathbf{p})$. Step (a) follows from the van Trees inequality [BHO19][Eqn.4-8]. Our goal is to bound the term $\sup_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} \operatorname{Tr}(I_{Y^n}(\mathbf{p}))$. For a given distribution $\mathbf{p} \in \overline{\mathcal{P}}$, the random variables $Y_1, \ldots, Y_n$ are independent. As a result, the trace of the Fisher information matrix for estimating $\mathbf{p}$ from $Y_1, \ldots, Y_n$ is bounded by

$$\begin{aligned} &\sup_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} \operatorname{Tr}\left(I_{Y^n}(\mathbf{p})\right) \\ &\qquad \overset{(a)}{=} \sup_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} \sum_{i=1}^{n} \operatorname{Tr}\left(I_{Y_i}(\mathbf{p})\right) \\ &\qquad \leq \sup_{\{Q_i \in \mathcal{Q}_{(\varepsilon,R)}\}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} n \sup_{i\in[n]} \operatorname{Tr}\left(I_{Y_i}(\mathbf{p})\right) \\ &\qquad \overset{(b)}{\leq} \begin{cases} 2nk\frac{e^{\varepsilon}(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2nk\frac{p_R^2(e^{\varepsilon}-1)^2}{e^{\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \end{aligned} \tag{F.5}$$

where step (a) follows from the chain rule of the Fisher information [Zam98][Lemma 1]. Step (b) follows from Lemma F.1.1 presented below. Substituting from (F.5) into (F.4), we get

$$r^{\ell_2^2}_{\varepsilon,R,n,k} \geq \begin{cases} \frac{k(e^{\varepsilon}+1)^2}{16ne^{\varepsilon}(e^{\varepsilon}-1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ \frac{ke^{\varepsilon}}{16np_R^2(e^{\varepsilon}-1)^2} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \tag{F.6}$$

for $n \geq 4\frac{e^{\varepsilon}}{p_R^2(e^{\varepsilon}-1)^2}$.

Lemma F.1.1. *For any (ε, R)-LDP mechanism, the trace of the Fisher information matrix $I_Y(\mathbf{p})$ is bounded by*

$$\sup_{Q\in\mathcal{Q}_{(\varepsilon,R)}} \sup_{\mathbf{p}\in\overline{\mathcal{P}}} Tr\left(I_Y(\mathbf{p})\right) \leq \begin{cases} 2k\frac{e^{\varepsilon}(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2k\frac{p_R^2(e^{\varepsilon}-1)^2}{e^{\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \tag{F.7}$$

where $H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right)$ is the Shannon entropy, and $p_R < 0.5$ denotes the inverse Shannon entropy $p_R = h^{-1}(R)$.

Proof. For a given distribution $\mathbf{p} \in \overline{\mathcal{P}}$, we have

$$\begin{aligned} S_{p_j}(y) &= \frac{\partial \log\left(\mathbf{M}(y|\mathbf{p})\right)}{\partial p_j} \\ &= \frac{Q(y|j) - Q(y|j+k/2)}{\mathbf{M}(y|\mathbf{p})}, \end{aligned} \tag{F.8}$$

for $j \in [k/2]$. By taking the expectation with respect to Y, we get

$$\mathbb{E}\left[S_{p_j}(Y)^2\right] = \sum_{y\in\mathcal{Y}} \frac{\left(Q(y|j) - Q(y|j+k/2)\right)^2}{\sum_{j'=1}^{k} Q(y|j')\, p_{j'}} \tag{F.9}$$

Thus, the trace of the Fisher information matrix is given by

$$\begin{aligned} \text{Tr}\left(I_Y(\mathbf{p})\right) &= \sum_{j=1}^{k/2} \mathbb{E}\left[S_{p_j}(Y)^2\right] \\ &= \sum_{j=1}^{k/2} \sum_{y\in\mathcal{Y}} \frac{\left(Q(y|j) - Q(y|j+k/2)\right)^2}{\sum_{j'=1}^{k} Q(y|j')\, p_{j'}} \\ &\leq \frac{k}{2} \max_{j\in[k/2]} \sum_{y\in\mathcal{Y}} \frac{\left(Q(y|j) - Q(y|j+k/2)\right)^2}{\sum_{j'=1}^{k} Q(y|j')\, p_{j'}} \\ &\overset{(a)}{\leq} ke^{\varepsilon} \max_{j\in[k/2]} \sum_{y\in\mathcal{Y}} \frac{\left(Q(y|j) - Q(y|j+k/2)\right)^2}{Q(y|j) + Q(y|j+k/2)} \\ &\overset{(b)}{\leq} \begin{cases} 2k\frac{e^{\varepsilon}(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2} & \text{if } R \geq H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \\ 2k\frac{p_R^2(e^{\varepsilon}-1)^2}{e^{\varepsilon}} & \text{if } R < H_2\left(\frac{e^{\varepsilon}}{e^{\varepsilon}+1}\right) \end{cases} \end{aligned} \tag{F.10}$$

where step (a) follows from the fact that $Q\left(y|j'\right) \geq e^{-\varepsilon} Q\left(y|j\right)$ and $Q\left(y|j'\right) \geq e^{-\varepsilon} Q\left(y|j+k/2\right), \forall j' \in [k]$. Thus, we have

$$\begin{aligned}\sum_{j'=1}^{k} Q\left(y|j'\right) p_{j'} &\geq e^{-\varepsilon} \frac{Q\left(y|j\right) + Q_i\left(y|j+k/2\right)}{2} \sum_{j'=1}^{k} p_{j'} \\ &= e^{-\varepsilon} \frac{Q\left(y|j\right) + Q\left(y|j+k/2\right)}{2}\end{aligned} \tag{F.11}$$

Step (b) follows from Lemma 7.3.2 presented at the end of Section 7.3.1. This completes the proof of Lemma F.1.1. ■

F.1.1 Proof of Lemma 7.3.1

We start our proof by Assoud's method.

Lemma F.1.2. *(Assouad's Method [DJW18]) For the family of distributions $\left\{\mathbf{p}^{\nu} : \nu \in \mathcal{V} = \{-1,1\}^{k/2}\right\}$, and a loss function $\ell\left(\hat{\mathbf{p}}, \mathbf{p}\right) = \sum_{j=1}^{k} \phi\left(\hat{p}_j - p_j\right)$ defined in Section 7.3.1, we have*

$$\begin{aligned}r^{\ell}_{\varepsilon,R,n,k}\left(Q^n\right) &= \inf_{\hat{\mathbf{p}}} \sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}\right)\right] \\ &\geq \phi\left(\delta\right) \sum_{j=1}^{k/2} \left(1 - ||\mathbf{M}^n_{+j} - \mathbf{M}^n_{-j}||_{TV}\right)\end{aligned} \tag{F.12}$$

For completeness, we present the proof of Lemma F.1.2 in Appendix F.3. Let $\{e_j\}_{j=1}^{k/2}$ be the standard basis of $\mathbb{R}^{k/2}$. Consider now the following inequalities:

$$\begin{aligned}
\sum_{j=1}^{k/2}\left(1-\|\mathbf{M}_{+j}^{n}-\mathbf{M}_{-j}^{n}\|_{\mathrm{TV}}\right) &\overset{(a)}{\geq} \sum_{j=1}^{k/2}\left(1-\frac{1}{|\mathcal{V}|}\sum_{\nu:\nu_j=1}||\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu}\right)-\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu-2e_j}\right)||_{\mathrm{TV}}\right)\\
&\geq \sum_{j=1}^{k/2}\left(1-\sup_{\nu:\nu_j=1}||\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu}\right)-\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu-2e_j}\right)||_{\mathrm{TV}}\right)\\
&\overset{(b)}{\geq} \sum_{j=1}^{k/2}\left(1-\sup_{\nu:\nu_j=1}\sqrt{\frac{1}{2}D_{\mathrm{KL}}\left(\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu}\right)||\left(\prod_{i=1}^{n}\mathbf{M}_i^{\nu-2e_j}\right)\right)}\right)\\
&\overset{(c)}{\geq} \sum_{j=1}^{k/2}\left(1-\sqrt{\frac{1}{2}\sup_{\nu:\nu_j=1}\sum_{i=1}^{n}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)\\
&= \frac{k}{2}\left(1-\frac{2}{k}\sum_{j=1}^{k/2}\sqrt{\frac{1}{2}\sup_{\nu:\nu_j=1}\sum_{i=1}^{n}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)\\
&\overset{(d)}{\geq} \frac{k}{2}\left(1-\sqrt{\frac{1}{k}\sum_{j=1}^{k/2}\sup_{\nu:\nu_j=1}\sum_{i=1}^{n}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)\\
&\geq \frac{k}{2}\left(1-\sqrt{\frac{n}{2}\sup_{j\in[k/2]}\sup_{i\in[n]}\sup_{\nu:\nu_j=1}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)
\end{aligned} \tag{F.13}$$

where step (a) follows from the triangular inequality. Step (b) follows from Pinsker's inequality that states that for any two distributions $\mathbf{P}$ and $\mathbf{Q}$, we get $\|\mathbf{P}-\mathbf{Q}\|_{\mathrm{TV}}\leq\sqrt{\frac{1}{2}D\left(\mathbf{P}||\mathbf{Q}\right)}$ [Tsy08, Lemma 2.5]. Step (c) follows from the properties of KL-divergence. Step (d) follows from the concavity of function $\sqrt{x}$. Substituting from (F.13) into (F.12), we get

$$\begin{aligned}
r_{\varepsilon,R,n,k}^{\ell} &= \inf_{\{Q_i\in\mathcal{Q}_{(\varepsilon,R)}\}} r_{\varepsilon,R,n,k}^{\ell}\left(Q^n\right)\\
&\geq \inf_{\{Q_i\in\mathcal{Q}_{(\varepsilon,R)}\}}\phi\left(\delta\right)\frac{k}{2}\left(1-\sqrt{\frac{n}{2}\sup_{j\in[k/2]}\sup_{i\in[n]}\sup_{\nu:\nu_j=1}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)\\
&= \phi\left(\delta\right)\frac{k}{2}\left(1-\sqrt{\frac{n}{2}\sup_{j\in[k/2]}\sup_{i\in[n]}\sup_{\nu:\nu_j=1}\sup_{Q_i\in\mathcal{Q}_{(\varepsilon,R)}}D_{\mathrm{KL}}\left(\mathbf{M}_i^{\nu}||\mathbf{M}_i^{\nu-2e_j}\right)}\right)
\end{aligned} \tag{F.14}$$

Hence the proof is completed.

F.2 Proof of Lemma 7.3.3

Lemma F.2.1. *The optimal solution of the non-convex optimization problem **P1** is obtained when the the output size is $m = 2$.*

Proof. Note that if $m = 1$, then the optimal value of **P1** will be zero, and hence, we have $m \geq 2$. In the following, we prove that the optimal solution is achievable at $m = 2$. Let

$$f\left(\mathbf{q}_j^m, \mathbf{q}_{j+k/2}^m\right) = \sum_{l=1}^{m} \frac{\left(q_{l,j} - q_{l,j+k/2}\right)^2}{q_{l,j} + q_{l,j+k/2}}$$

denote the objective function of the problem **P1**, where $\mathbf{q}_j^m = [q_{1,j}, \ldots, q_{m,j}]$ and $\mathbf{q}_{j+k/2}^m = \left[q_{1,j+k/2}, \ldots, q_{m,j+k/2}\right]$. Suppose that the optimal solution is obtained at $m > 2$. In other words, there exist two distributions $\mathbf{q}_j^m$ and $\mathbf{q}_{j+k/2}^m$ with size $m > 2$ that maximize the objective function $f\left(\mathbf{q}_j^m, \mathbf{q}_{j+k/2}^m\right)$ and satisfy the constraints (7.21). We prove that if $\mathbf{q}_j^m$ and $\mathbf{q}_{j+k/2}^m$ are optimal, then there exist two distributions $\tilde{\mathbf{q}}_j^{m-1}$ and $\tilde{\mathbf{q}}_{j+k/2}^{m-1}$ with support size $m-1$ that satisfy the problem constraints and achieve at least the same objective value as $\mathbf{q}_j^m$ and $\mathbf{q}_{j+k/2}^m$. Let $\tilde{\mathbf{q}}_j^{m-1} = [q_{1,j}, \ldots, q_{m-2,j}, q_{m-1,j} + q_{m,j}]$ and $\tilde{\mathbf{q}}_{j+k/2}^{m-1} = \left[q_{1,j+k/2}, \ldots, q_{m-2,j+k/2}, q_{m-1,j} + q_{m,j+k/2}\right]$. We can easily verify that $H\left(\tilde{\mathbf{q}}_j^{m-1}\right) \leq R$ as $H\left(\mathbf{q}_j^m\right) \leq R$ and $H\left(\tilde{\mathbf{q}}_{j+k/2}^{m-1}\right) \leq R$ as $H\left(\mathbf{q}_{j+k/2}^m\right) \leq R$. Furthermore, we have

$$e^{-\varepsilon} = e^{-\varepsilon} \frac{q_{m-1,j+k/2} + q_{m,j+k/2}}{q_{m-1,j+k/2} + q_{m,j+k/2}} \leq \frac{q_{m-1,j} + q_{m,j}}{q_{m-1,j+k/2} + q_{m,j+k/2}} \leq e^{\varepsilon} \frac{q_{m-1,j+k/2} + q_{m,j+k/2}}{q_{m-1,j+k/2} + q_{m,j+k/2}} = e^{\varepsilon} \tag{F.15}$$

Hence, the distributions $\tilde{\mathbf{q}}_j^{m-1}$ and $\tilde{\mathbf{q}}_{j+k/2}^{m-1}$ satisfy the constraints of the problem **P1**. Consider the following inequalities

$$\begin{aligned}
&f\left(\tilde{\mathbf{q}}_j^m, \tilde{\mathbf{q}}_{j+k/2}^{m-1}\right) - f\left(\mathbf{q}_j^m, \mathbf{q}_{j+k/2}^m\right) \\
&\quad = \frac{\left(q_{m-1,j} + q_{m,j} - q_{m-1,j+k/2} + q_{m,j+k/2}\right)^2}{q_{m-1,j} + q_{m,j} + q_{m-1,j+k/2} + q_{m,j+k/2}} - \left[\frac{\left(q_{m-1,j} - q_{m-1,j+k/2}\right)^2}{q_{m-1,j} + q_{m-1,j+k/2}} + \frac{\left(q_{m,j} - q_{m,j+k/2}\right)^2}{q_{m,j} + q_{m,j+k/2}}\right] \\
&\quad \overset{(a)}{\geq} \frac{\left(q_{m-1,j} + q_{m,j} - q_{m-1,j+k/2} + q_{m,j+k/2}\right)^2}{q_{m-1,j} + q_{m,j} + q_{m-1,j+k/2} + q_{m,j+k/2}} - 2\frac{\left(\frac{q_{m-1,j}+q_{m,j}}{2} - \frac{q_{m-1,j+k/2}+q_{m,j+k/2}}{2}\right)^2}{\frac{q_{m-1,j}+q_{m,j}}{2} + \frac{q_{m-1,j+k/2}+q_{m,j+k/2}}{2}} \\
&\quad = 0
\end{aligned} \tag{F.16}$$

where step (a) follows from the convexity of the function $(x-y)^2/(x+y)$ for $x, y \in [0:1]$. Hence the distributions $\tilde{\mathbf{q}}_j^m, \tilde{\mathbf{q}}_{j+k/2}^{m-1}$ have at least the same objective value as $\mathbf{q}_j^m$ and $\mathbf{q}_{j+k/2}^m$. ■

F.3 Proof of Lemma F.1.2

Consider an arbitrary estimator $\hat{\mathbf{p}}$, then we have

$$\begin{aligned}
\sup_{\mathbf{p}\in\Delta_k} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}\right)\right] &\geq \sup_{\nu\in\mathcal{V}} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}^\nu\right)\right] \\
&\geq \frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}} \mathbb{E}\left[\ell\left(\hat{\mathbf{p}}\left(Y^n\right), \mathbf{p}^\nu\right)\right] \\
&\geq \phi\left(\delta\right)\frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}} \mathbb{E}\left[\sum_{j=1}^{k/2} \mathbb{1}\left(\psi_j\left(Y^n\right) \neq \nu_j\right)\right] \\
&\geq \phi\left(\delta\right)\sum_{j=1}^{k/2}\left(\frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}:\nu_j=+1} \mathbb{E}\left[\mathbb{1}\left(\psi_j\left(Y^n\right)\neq +1\right)\right] + \frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}:\nu_j=-1} \mathbb{E}\left[\mathbb{1}\left(\psi_j\left(Y^n\right)\neq -1\right)\right]\right) \\
&\geq \phi\left(\delta\right)\sum_{j=1}^{k/2}\inf_{\psi}\left(\frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}:\nu_j=+1} \Pr\left[\psi_j\left(Y^n\right)\neq +1\right] + \frac{1}{|\mathcal{V}|}\sum_{\nu\in\mathcal{V}:\nu_j=-1} \Pr\left[\psi_j\left(Y^n\right)\neq -1\right]\right) \\
&= \phi\left(\delta\right)\sum_{j=1}^{k/2}\frac{1}{2}\inf_{\psi}\left(\mathbf{M}_{+j}^n\left[\psi_j\left(\mathbf{Y}^n\right)\neq +1\right] + \mathbf{M}_{+j}^n\left[\psi_j\left(\mathbf{Y}^n\right)\neq -1\right]\right) \\
&\geq \phi\left(\delta\right)\sum_{j=1}^{k/2}\left(1 - ||\mathbf{M}_{+j}^n - \mathbf{M}_{-j}^n||_{\mathrm{TV}}\right)
\end{aligned} \tag{F.17}$$

where $\psi = \left(\psi_1, \ldots, \psi_{k/2}\right)$ is a vector of test functions.

F.4 Proof of Lemma 7.4.1

We claim that the conditional distribution on $Y_i^j | X_i$ is given by

$$\Pr\left[Y_i^j = 1 | X_i\right] = \begin{cases} \frac{e^{\varepsilon_j}}{e^{\varepsilon_j}+1} & \text{if } X_i \in B_i \\ \frac{1}{e^{\varepsilon_j}+1} & \text{if } X_i \notin B_i \end{cases} \tag{F.18}$$

which is ε_j-LDP. We prove our claim by induction. For the basis step, we can easily verify that Y_i^1 defined in (7.37) follows the conditional distribution in (F.18). For the induction step, suppose that our claim is true for j. Observe that $Y_i^{j+1} = Y_i^j \oplus U_i^{j+1}$. Hence, we have

$$\begin{aligned} &\Pr\left[Y_i^{j+1} = 1 | X_i \in B_i\right] \\ &= \Pr\left[Y_i^{j+1} = 1 | X_i \in B_i, Y_i^j = 1\right] \Pr\left[Y_i^j = 1 | X_i \in B_i\right] \\ &\qquad\qquad + \Pr\left[Y_i^{j+1} = 1 | X_i \in B_i, Y_i^j = 0\right] \Pr\left[Y_i^j = 0 | X_i \in B_i\right] \\ &= \Pr\left[U_i^{j+1} = 0\right] \Pr\left[Y_i^j = 1 | X_i \in B_i\right] + \Pr\left[U_i^{j+1} = 1\right] \Pr\left[Y_i^j = 0 | X_i \in B_i\right] \\ &= \left(1 - q_{j+1}\right)\left(1 - z_j\right) + q_{j+1} z_j \\ &= 1 - z_{j+1} = \frac{e^{\varepsilon_{j+1}}}{e^{\varepsilon_{j+1}} + 1} \end{aligned} \tag{F.19}$$

Similarly, we can prove that $\Pr\left[Y_i^{j+1} = 1 | X_i \notin B_i\right] = z_{j+1} = \frac{1}{e^{\varepsilon_{j+1}}+1}$. Hence, the proof is completed.

F.5 Proof of Lemma 7.5.1

In order to recover X from Y and U, it is required that each input database $x \in [k]$ is mapped to y with a different value of key U for every output $y \in [k]$. Let $y = x \oplus u$ for all $x \in [k]$ and $u \in [k]$, where $x \oplus y = [(x + u - 2) \bmod k] + 1$. Note that the set $[k]$ along with the operation $\oplus$ forms a group[1]. The private mechanism Q is defined as follows

$$Q\left(y|x\right)=q_{u}, \tag{F.20}$$

for $y=x\oplus u$. Note that an input x is mapped to each output y with a different value of the key $U=(k-x+2)\oplus y$. Moreover, for a given output y, we can easily see that each input $x\in[k]$ is mapped to y with a different value of the key U. Hence, it is possible to recover X from Y and U. Furthermore, for any two inputs $x,x'\in\mathcal{X}$, we have

$$\sup_{y\in[k]}\frac{Q\left(y|x\right)}{Q\left(y|x'\right)}\leq\frac{q_{\max}}{q_{\min}}\overset{(a)}{\leq}e^{\varepsilon}, \tag{F.21}$$

where $q_{\max}=\max\limits_{j\in[k]}q_j$ and $q_{\min}=\min\limits_{j\in[k]}q_j$. Step (a) follows from the assumption that $\frac{q_{\max}}{q_{\min}}\leq e^{\varepsilon}$. Thus, the mechanism Q is an ε-LDP-Rec mechanism.

F.5.1 Proof of Lemma 7.5.2

Before we present the proof of Lemma 7.5.2, we provide the following lemma whose proof is in Appendix F.9.

Lemma F.5.1. *Let $U\in\mathcal{U}=\{u_1,\ldots,u_m\}$ be a random variable with size m having a distribution $\mathbf{q}=[q_1,\ldots,q_m]$, where $q_1\geq\cdots\geq q_m$. Then, the random variable $U'\in\mathcal{U}'=\{u_1,\ldots,u_{m-1}\}$ with distribution $\mathbf{q}'=\left[q'_1,\ldots,q'_{m-1}\right]$ has an entropy*

$$H\left(U\right)\geq H\left(U'\right), \tag{F.22}$$

where $q'_j=q_j/\left(1-q_m\right)$ for $j\in\{1,\ldots,m-1\}$.

This lemma shows that if we trim the last symbol that has the lowest probability from a distribution, and normalize the remaining probabilities, then we get a distribution that has lower entropy.

[1] It is exactly the group defined on integers $\{0,\ldots,k-1\}$ with modulo-k operation, but we subtract -2 before taking mod$k$ and adding one to fit modulo-k operation with the set $[k]=\{1,\ldots,k\}$

The main idea of the proof of Lemma 7.5.2 is that we do some reduction steps to get a new random key U' with a support size equal to the input size from the random key U. In addition, this new random key U' has lower entropy than the entropy of the original random key U. First, we give an example to illustrate the idea, and then we proceed to the general proof.

Example F.5.1. Suppose that a random key $U \in \{1, 2, \ldots, 6\}$ has a distribution $\mathbf{q} = [q_1, \ldots, q_6]$, where $q_1 \geq \cdots \geq q_6$. The random key U is used to design an ε-LDP-Rec mechanism Q with input $X \in \{1, 2, 3\}$. Suppose that there exists an output y such that $X = x$ is mapped to y when $U \in \mathcal{U}_{yx}$, where $\mathcal{U}_{y1} = \{6\}$, $\mathcal{U}_{y2} = \{2, 3\}$, and $\mathcal{U}_{y3} = \{1\}$. Hence, $Q\left(y|X = 1\right) = q_6$, $Q\left(y|X = 2\right) = q_2 + q_3$, and $Q\left(y|X = 3\right) = q_1$. Let $\mathcal{U}_y = \bigcup_{x\in[3]} \mathcal{U}_{yx} = \{1, 2, 3, 6\}$, and $\overline{\mathcal{U}}_y = \mathcal{U} \setminus \mathcal{U}_y = \{4, 5\}$. Let $\tilde{\mathbf{q}} = [q_6, q_2 + q_3, q_1, q_4, q_5]$, where the first three elements are $Q\left(y|X = i\right)$ for $i \in [3]$ and the remaining elements represent q_u for $u \in \overline{\mathcal{U}}_y$. Then, we sort the distribution $\tilde{\mathbf{q}}$ in a descending order to get $\tilde{\mathbf{q}}^{\downarrow} = [q_2 + q_3, q_1, q_4, q_5, q_6]$, where $\tilde{q}_i^{\downarrow}$ denotes the ith largest component in $\tilde{\mathbf{q}}$. Consider a random key $\tilde{U} \in \{1, 2, 3, 4, 5\}$ having a distribution $\tilde{\mathbf{q}}^{\downarrow}$. Observe that $H\left(\tilde{U}\right) \leq H\left(U\right)$, since $\tilde{U}$ can be represented as a function of U. Furthermore, we have $\frac{q_2+q_3}{q_1} \leq \frac{q_2+q_3}{q_4} \leq \frac{q_2+q_3}{q_6} \leq e^{\varepsilon}$, since Q is an ε-LDP mechanism, and $q_4 \geq q_6$. Consider a random key U' having a distribution $\mathbf{q}' = \left[\frac{q_2+q_3}{1-(q_5+q_6)}, \frac{q_1}{1-(q_5+q_6)}, \frac{q_4}{1-(q_5+q_6)}\right]$ obtained by trimming sequentially the last two symbols of the random key $\tilde{U}$. By applying Lemma F.5.1 twice on the distribution $\tilde{\mathbf{q}}^{\downarrow}$, we get that $H\left(U\right) \geq H\left(\tilde{U}\right) \geq H\left(U'\right)$. Furthermore, we have $q'_{\max}/q'_{\min} \leq e^{\varepsilon}$. Thus, from Lemma 7.5.1, we can construct an ε-LDP-Rec mechanism with input $X \in [3]$ and an output $Y \in [3]$ using the random key U', where $H\left(U\right) \geq H\left(U'\right)$.

We now present the general proof. Let $U \in \mathcal{U} = \{u_1, \ldots, u_m\}$ be a random key with size $m > k$ having a distribution $\mathbf{q} = [q_1, \ldots, q_m]$. Without loss of generality, assume that $q_1 \geq \cdots \geq q_m$. Let Q be an ε-LDP-Rec mechanism designed using a random key U with input $X \in [k]$ and an output $Y \in \mathcal{Y}$. Let $\mathcal{U}_{yx} \subset \mathcal{U}$ be a subset of keys such that the input $X = x$ is mapped to $Y = y$ when $U \in \mathcal{U}_{yx}$ for all $x \in [k]$ and $y \in \mathcal{Y}$. As a result the private mechanism Q can be represented by $Q\left(y|X = x\right) = \sum_{u\in\mathcal{U}_{yx}} q_u$.

Observe that for given y, we have $\mathcal{U}_{yx}\bigcap\mathcal{U}_{yx'}=\phi$, otherwise we cannot recover X from Y and U, since there would be x and x' mapped to y with the same key value. Let $\mathcal{U}_y=\bigcup_{x\in[k]}\mathcal{U}_{yx}$, and hence, $\mathcal{U}_y\subseteq\mathcal{U}$. Furthermore, for given y, we have $Q\left(y|X=x\right)/Q\left(y|X=x'\right)\leq e^{\varepsilon}$, since Q is an ε-LDP mechanism.

Consider an output $y\in\mathcal{Y}$ such that $u_1\in\mathcal{U}_y$. Let $\overline{\mathcal{U}}_y=\mathcal{U}\setminus\mathcal{U}_y$ be an indexed set with size $l=|\overline{\mathcal{U}}_y|$, where $\overline{\mathcal{U}}_y\left(j\right)$ denotes the jth element in $\overline{\mathcal{U}}_y$. Consider a distribution $\tilde{\mathbf{q}}=\left[\tilde{q}_1,\ldots,\tilde{q}_{l+k}\right]$ designed as follows $\tilde{q}_j=Q\left(y|X=j\right)$ for all $j\in[k]$ and $\tilde{q}_j=q_{\overline{\mathcal{U}}_y(j-k)}$ for all $i\in\{k+1,\ldots,k+l\}$. We can sort the distribution $\tilde{\mathbf{q}}$ in a descending order to get $\tilde{\mathbf{q}}^{\downarrow}=\left[\tilde{q}_1^{\downarrow},\ldots,\tilde{q}_{l+k}^{\downarrow}\right]$, where $\tilde{q}_i^{\downarrow}$ denotes the ith largest component in $\tilde{\mathbf{q}}$. Let $\tilde{U}$ be a random key drawn from a distribution $\tilde{\mathbf{q}}^{\downarrow}$. We have the following two properties on the distribution $\tilde{\mathbf{q}}^{\downarrow}$:

1. $H\left(U\right)\geq H\left(\tilde{U}\right)$.
2. $\frac{\tilde{q}_1^{\downarrow}}{\tilde{q}_k^{\downarrow}}\leq e^{\varepsilon}$.

The first property is straightforward, since the random key $\tilde{U}$ can be represented as a function of U. Observe that $u_1\in\mathcal{U}_y$, and $q_1\geq q_u$ for all $u\in\overline{\mathcal{U}}_y$. Hence, $\tilde{q}_1^{\downarrow}$ is one of the first k elements in $\tilde{\mathbf{q}}$. Thus, we get

$$\frac{\tilde{q}_1^{\downarrow}}{\tilde{q}_k^{\downarrow}}\overset{(a)}{\leq}\frac{\tilde{q}_{\max}}{\tilde{q}_{\min}}\leq e^{\varepsilon}$$

where $\tilde{q}_{\max}=\max_{j\in[k]}\tilde{q}_j=\tilde{q}_1^{\downarrow}$ and $\tilde{q}_{\min}=\min_{j\in[k]}\tilde{q}_j$. If q_u for $u\in\overline{\mathcal{U}}_y$ is one of the first k elements in $\tilde{\mathbf{q}}^{\downarrow}$, i.e, $q_u>\tilde{q}_{\min}$, then inequality (a) is still valid.

Now, let $U'\in[k]$ be a random key drawn from a distribution $\mathbf{q}'=\left[q_1',\ldots,q_k'\right]$, where $q_j'=\frac{\tilde{q}_j^{\downarrow}}{\sum_{j=1}^{k}\tilde{q}_j^{\downarrow}}$. Observe that $\mathbf{q}'$ is obtained by applying Lemma F.5.1 l times on $\tilde{\mathbf{q}}^{\downarrow}$ to trim sequentially the last l symbols of $\tilde{U}$ that have the lowest l probabilities. Thus, we get that $H\left(U\right)\geq H\left(\tilde{U}\right)\geq H\left(U'\right)$. Furthermore, from the second property, we have $q_{\max}'/q_{\min}'=\frac{\tilde{q}_1^{\downarrow}}{\tilde{q}_k^{\downarrow}}\leq e^{\varepsilon}$. Thus, from Lemma 7.5.1, we can construct an ε-LDP-Rec mechanism with input $X\in[k]$ and an output $Y\in[k]$ using the random key U', and $H\left(U\right)\geq H\left(U'\right)$. This completes the proof.

F.6 Omitted Details from Section 7.5.1

First we prove the first necessary condition of Theorem 7.5.1. As mentioned in Section 7.5.1, we prove this in two parts: First we show $|\mathcal{Y}| \geq |\mathcal{X}|$ using the recoverability constraint and then $|\mathcal{U}| \geq |\mathcal{Y}|$ using the privacy constraint.

$|\mathcal{Y}| \geq |\mathcal{X}|$: Observe that the output Y of the private mechanism Q can be represented as a function of the input X and the random key U, i.e., $Y = f(X, U)$. Fix the value of the random key $U = u$ for an arbitrary $u \in \mathcal{U}$. Then, for each value of $x \in \mathcal{X}$, the function $f(X, U)$ should generate a different output Y in order to be able to recover X from Y and U. In other words, each input $x \in \mathcal{X}$ should be mapped to a different output $y \in \mathcal{Y}$ for the same value of the random key $u \in \mathcal{U}$. Otherwise, there exists two inputs mapped with the same key value to the same output. As a result, it is required that the output size is at least the same as the input size: $|\mathcal{Y}| \geq |\mathcal{X}|$.

$|\mathcal{U}| \geq |\mathcal{Y}|$: Let $\mathcal{Y}(x) \subseteq \mathcal{Y}$ be a subset of outputs such that input $X = x$ is mapped with non-zero probability to every $y \in \mathcal{Y}(x)$. We claim that $\mathcal{Y}(x) = \mathcal{Y}$ for all $x \in \mathcal{X}$ for any ε-LDP-Rec mechanism. In other words, we claim that each input $x \in \mathcal{X}$ should be mapped with non-zero probability to every output $y \in \mathcal{Y}$. We prove our claim by contradiction. Suppose that there exist $x, x' \in \mathcal{X}$ such that $\mathcal{Y}(x) \neq \mathcal{Y}(x')$. Thus, there exists $y \in \mathcal{Y}(x) \setminus \mathcal{Y}(x')$ or $y \in \mathcal{Y}(x') \setminus \mathcal{Y}(x)$. Hence, we have $\frac{Q(y|x)}{Q(y|x')} \to \infty$ or $\frac{Q(y|x')}{Q(y|x)} \to \infty$ which violates the privacy constraints. Therefore, $\mathcal{Y}(x) = \mathcal{Y}(x') = \mathcal{Y}$ for all $x, x' \in \mathcal{X}$. However, for a given $x \in \mathcal{X}$, we have $|\mathcal{Y}(x)| \leq |\mathcal{U}|$, since each input $x \in \mathcal{X}$ can be mapped with non-zero probability to at most $|\mathcal{U}|$ outputs. Thus, we get that the random key size is at least the same as the output size: $|\mathcal{U}| \geq |\mathcal{Y}| \geq |\mathcal{X}|$.

Hence, the first condition is necessary to design an ε-LDP-Rec mechanism. This completes the proof of the first necessary condition of Theorem 7.5.1.

Now, assuming $q_1 \leq q_2 \leq \ldots \leq q_k$, we show $q_k/q_1 \leq e^{\varepsilon}$. This will be required to prove the second necessary condition to prove Theorem 7.5.1.

$q_k/q_1 \leq e^{\varepsilon}$: We prove our claim by contradiction. Suppose that $q_k/q_1 > e^{\varepsilon}$. Consider a certain output $y \in \mathcal{Y}$ such that there exists $x \in \mathcal{X}$ mapped to y when $U = u_k$ with probability q_k. Note that each sample $x \in \mathcal{X}$ should be mapped using a different value of the key to each output $y \in \mathcal{Y}$ in order to be able to recover the sample X from Y and U. In our case, there are $k-1$ remaining inputs to be mapped to y with different values of keys; however, none of these $k-1$ inputs can be mapped to y with $U = u_1$, since $q_k/q_1 > e^{\varepsilon}$, which violates the privacy constraint. Hence, we have $k-1$ inputs mapped to y using at most $k-2$ values of keys. Thus, there would exist at least two inputs mapped to output y with the same key value. Therefore, we cannot recover X from y given U. As a result, we should have $q_k/q_1 \leq e^{\varepsilon}$.

F.7 Proof of Lemma 7.6.1

To simplify the proof, we assume that $[k] = \{0, \ldots, k-1\}$. Let $\mathcal{X}^T = [k]^T$ denote the input dataset, and $Y^T = \left(Y^{(1)}, \ldots, Y^{(T)}\right)$ be the output of the private mechanism Q that takes a value from a set $\mathcal{Y}^T = [k]^T$. In order to recover X^T from Y^T and U, it is required that each input database $\mathbf{x} \in \mathcal{X}^T$ is mapped to each output $\mathbf{y} \in [k]^T$ with a different value of key U. Let the random key U be drawn from an ε-DP distribution $\mathbf{q}$. Hence, there exists a bijective function $f : \mathcal{X}^T \to [k]^T$ such that

$$\frac{q_{f(\mathbf{x})}}{q_{f(\mathbf{x}')}} \leq e^{\varepsilon}. \tag{F.23}$$

for every neighboring databases $\mathbf{x}, \mathbf{x}' \in [k]^T$. Let Q be a private mechanism defined as follows

$$Q\left(\mathbf{y}|\mathbf{x}\right) = q_{f(\mathbf{x} \oplus \mathbf{y})}. \tag{F.24}$$

where $\mathbf{x} \oplus \mathbf{y} = \left(x^{(1)} \oplus y^{(1)}, \ldots, x^{(T)} \oplus y^{(T)}\right)^2$, and $x^{(j)} \oplus y^{(j)} = \left[\left(x^{(j)} + y^{(j)}\right) \bmod k\right]$ which is an addition between $x^{(j)}$ and $y^{(j)}$ in a finite group of order k. For a fixed $\mathbf{y} \in \mathcal{Y}^T$, we can easily see that $f\left(\mathbf{x} \oplus \mathbf{y}\right) \neq f\left(\hat{\mathbf{x}} \oplus \mathbf{y}\right)$ for any $\mathbf{x} \neq \hat{\mathbf{x}}$ and $\mathbf{x}, \hat{\mathbf{x}} \in [k]^T$, since $\mathbf{x} \oplus \mathbf{y} \neq \hat{\mathbf{x}} \oplus \mathbf{y}$ and

[2] We apply elementwise operation $\oplus$ on the vectors $\mathbf{x}$ and $\mathbf{y}$.

f is a bijection. Hence, for every output $\mathbf{y} \in [k]^T$, each input database $\mathbf{x} \in \mathcal{X}^T$ is mapped to an output $\mathbf{y}$ with a different value of key U. Thus, we can recover X^T from Y^T and U. For a fixed $\mathbf{x} \in [k]^T$, we can see that $f(\mathbf{x} \oplus \mathbf{y}) \neq f(\mathbf{x} \oplus \hat{\mathbf{y}})$ for any $\mathbf{y} \neq \hat{\mathbf{y}}$ and $\mathbf{y}, \hat{\mathbf{y}} \in [k]^T$, since $\mathbf{x} \oplus \mathbf{y} \neq \mathbf{x} \oplus \hat{\mathbf{y}}$ and f is a bijection. Hence $Q(\mathbf{y}|\mathbf{x})$ is a valid conditional distribution for each $\mathbf{x} \in [k]^T$. It remains to prove that the private mechanism Q given in (F.24) is ε-DP. In the following, we prove that for every output $\mathbf{y}$, and every neighboring databases $\mathbf{x}, \tilde{\mathbf{x}} \in [k]^T$, we have

$$\frac{Q(\mathbf{y}|\mathbf{x})}{Q(\mathbf{y}|\tilde{\mathbf{x}})} \leq e^{\varepsilon} \tag{F.25}$$

Therefore, the private mechanism Q is ε-DP. The proof is by induction. For the basis step, observe that each input database $\mathbf{x}$ is mapped to $\mathbf{y}_0 = [0, \ldots, 0]$ with probability $q_{f(\mathbf{x})}$ for $\mathbf{x} \in [k]^T$. Thus, for every neighboring databases $\mathbf{x}, \tilde{\mathbf{x}} \in [k]^T$, we get

$$\frac{Q(\mathbf{y}_0|\mathbf{x})}{Q(\mathbf{y}_0|\tilde{\mathbf{x}})} = \frac{q_{f(\mathbf{x})}}{q_{f(\tilde{\mathbf{x}})}} \overset{(a)}{\leq} e^{\varepsilon} \tag{F.26}$$

where step (a) follows from the assumption that the distribution $\mathbf{q}$ satisfies ε-DP. For the induction step, suppose there exists an output $\mathbf{y} \in [k]^T$ that satisfies (F.25). Let $\tilde{\mathbf{y}}$ be a neighboring output to $\mathbf{y}$, i.e., $\tilde{\mathbf{y}}$ and $\mathbf{y}$ are different in only one element. Without loss of generality, let $y^{(i)} \neq \tilde{y}^{(i)}$ while $y^{(j)} = \tilde{y}^{(j)}$ for $j \neq i$. Then, for every neighboring databases $\mathbf{x}, \tilde{\mathbf{x}} \in [k]^T$, we get

$$\begin{aligned} \frac{Q(\tilde{\mathbf{y}}|\mathbf{x})}{Q(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})} &= \frac{q_{f(\mathbf{x}\oplus\tilde{\mathbf{y}})}}{q_{f(\tilde{\mathbf{x}}\oplus\tilde{\mathbf{y}})}} \\ &= \frac{q_{f(\underline{\mathbf{x}}\oplus\mathbf{y})}}{q_{f(\underline{\tilde{\mathbf{x}}}\oplus\mathbf{y})}} \\ &\overset{(a)}{\leq} e^{\varepsilon} \end{aligned} \tag{F.27}$$

where $\underline{\mathbf{x}} = \left(\underline{x}^{(1)}, \ldots, \underline{x}^{(T)}\right)$ such that $\underline{x}^{(j)} = x^{(j)}$ for $j \neq i$ and $\underline{x}^{(i)} = \left[\left(k + x^{(i)} + y^{(i)} - \tilde{y}^{(i)}\right) \bmod k\right]$. Similarly, $\underline{\tilde{\mathbf{x}}} = \left(\underline{\tilde{x}}^{(1)}, \ldots, \underline{\tilde{x}}^{(T)}\right)$ such that $\underline{\tilde{x}}^{(j)} = \tilde{x}^{(j)}$ for $j \neq i$ and $\underline{\tilde{x}}^{(i)} = \left[\left(k + \tilde{x}^{(i)} + y^{(i)} - \tilde{y}^{(i)}\right) \bmod k\right]$. Since $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are neighboring databases, then $\underline{\mathbf{x}}$ and $\underline{\tilde{\mathbf{x}}}$ are also neighboring databases. Step (a) follows from the assumption that $\mathbf{y}$ satisfy (F.25). From the basic step along with the induction step, we conclude that the mechanism Q given in (F.24) is ε-DP-Rec mechanism. Hence, the proof is completed.

F.7.1 Proof of the first necessary condition ($|\mathcal{U}| \geq |\mathcal{Y}^T| \geq |\mathcal{X}^T|$) of Theorem 7.6.1

We prove it in two parts: first we show $|\mathcal{Y}^T| \geq |\mathcal{X}^T|$, and then we show $|\mathcal{U}| \geq |\mathcal{Y}^T|$.

$|\mathcal{Y}^T| \geq |\mathcal{X}^T|$**:** Note that the output is a deterministic function of the input and the random key, i.e., $Y^T = f(X^T, U)$ for some deterministic function f. This implies that, for any fixed $u \in \mathcal{U}$, the function $f(\mathbf{x}, u)$ should generate a different output $\mathbf{y} \in \mathcal{Y}^T$ for different values of $\mathbf{x} \in \mathcal{X}^T$, which implies that $|\mathcal{Y}^T| \geq |\mathcal{X}^T|$.

$|\mathcal{U}| \geq |\mathcal{Y}^T|$**:** Let $\mathcal{Y}(\mathbf{x}) \subseteq \mathcal{Y}^T$ be a subset of outputs such that the input $X^T = \mathbf{x}$ is mapped with non-zero probability to every $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$. We claim that $\mathcal{Y}(\mathbf{x}) = \mathcal{Y}^T$ for all $\mathbf{x} \in \mathcal{X}^T$ for any ε-DP-Rec mechanism. In other words, we claim that each input $\mathbf{x} \in \mathcal{X}^T$ should be mapped with non-zero probability to every output $\mathbf{y} \in \mathcal{Y}^T$. We prove our claim by contradiction. Suppose that there exist two neighboring $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^T$ such that $\mathcal{Y}(\mathbf{x}) \neq \mathcal{Y}(\mathbf{x}')$. Thus, there exists $\mathbf{y} \in \mathcal{Y}(\mathbf{x}) \setminus \mathcal{Y}(\mathbf{x}')$ or $\mathbf{y} \in \mathcal{Y}(\mathbf{x}') \setminus \mathcal{Y}(\mathbf{x})$. Hence, we have $\frac{Q(\mathbf{y}|\mathbf{x})}{Q(\mathbf{y}|\mathbf{x}')} \to \infty$ or $\frac{Q(\mathbf{y}|\mathbf{x}')}{Q(\mathbf{y}|\mathbf{x})} \to \infty$ which violates the privacy constraints. Therefore, $\mathcal{Y}(\mathbf{x}) = \mathcal{Y}(\mathbf{x}') = \mathcal{Y}^T$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^T$. Given $\mathbf{x} \in \mathcal{X}^T$, we have that $|\mathcal{Y}(\mathbf{x})| \leq |\mathcal{U}|$, where $|\mathcal{U}|$ is the maximum number of possible keys. Thus, the random key size is at least the same as the output size: $|\mathcal{U}| \geq |\mathcal{Y}^T|$.

Hence, the first condition of Theorem 7.6.1 is necessary to design an ε-DP-Rec mechanism.

F.8 Proof of Lemma 7.6.2

Let $g_i = i\,(k)^{(T-1)}$ for $i \in \{0, \ldots, k\}$. Observe that databases $\mathbf{x}_1, \ldots, \mathbf{x}_{g_1}$ have $x^{(1)} = 1$ and the databases $\mathbf{x}_{g_1+1}, \ldots, \mathbf{x}_{g_2}$ have $x^{(1)} = 2$. Generally, the databases $\mathbf{x}_{g_{i-1}+1}, \ldots, \mathbf{x}_{g_i}$ have $x^{(1)} = i$. Let $C_i = \sum_{a=g_{i-1}+1}^{g_i} P_a^{\mathbf{y}}$ for $i \in [k]$. Consider the following inequalities that we will prove next

$$H(\mathbf{P}^{\mathbf{y}}) = -\sum_{a=1}^{k^T} P_a^{\mathbf{y}} \log\left(P_a^{\mathbf{y}}\right)$$

$$= \sum_{i=1}^{k} C_i \left[- \sum_{a=g_{i-1}+1}^{g_i} \frac{P_a^{\mathbf{y}}}{C_i} \log\left(\frac{P_a^{\mathbf{y}}}{C_i} \right) \right] - \sum_{i=1}^{k} C_i \log\left(C_i\right) \tag{F.28}$$

$$\geq \sum_{i=1}^{k} C_i H\left(U_{\min,T-1}\right) - \sum_{i=1}^{k} C_i \log\left(C_i\right)) \tag{F.29}$$

$$\geq \sum_{i=1}^{k} C_i H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right) \tag{F.30}$$

$$= H\left(U_{\min,T-1}\right) + H\left(U_{\min,1}\right) \tag{F.31}$$

We begin with inequality (F.29). Observe that the k^{T-1} databases $\mathbf{x}_{g_{i-1}+1}, \ldots, \mathbf{x}_{g_i}$ have the same value of the first sample $x^{(1)} = i$, and hence these k^{T-1} databases cover all possible databases in $\mathcal{X}^{T-1}$. Consider a random variable U^{T-1} drawn according to the distribution $\mathbf{P}_{T-1} = \left[\frac{P^{\mathbf{y}}_{g_{i-1}+1}}{C_i}, \ldots, \frac{P^{\mathbf{y}}_{g_i}}{C_i}\right]$. This is a valid distribution with support size k^{T-1}. Furthermore, since the distribution $\mathbf{P}^{\mathbf{y}}$ is ε-DP, then the distribution $\mathbf{P}_{T-1}$ is also ε-DP. From Lemma 7.6.1, the random key U^{T-1} can be used to construct an ε-DP-Rec mechanism with the possibility to recover the databases $X^{T-1} = \left(x^{(2)}, \ldots, x^{(T)}\right)$ from the output of the mechanism and the random key U^{T-1}. Hence, we get

$$H\left(U^{T-1}\right) \geq H\left(U_{\min,T-1}\right). \tag{F.32}$$

This proves inequality (F.29). Now, observe that databases $\mathbf{x}_i, \mathbf{x}_{g_1+i}, \ldots, \mathbf{x}_{g_{k-1}+i}$ are neighboring databases for each $i \in \left[k^{T-1}\right]$, since they are only different in the value of the first sample $x^{(1)}$. Since the mechanism Q is ε-DP-Rec, we have

$$e^{-\varepsilon} \leq \frac{P^{\mathbf{y}}_{g_a+i}}{P^{\mathbf{y}}_{g_j+i}} \leq e^{\varepsilon} \qquad \forall a, j \in \{0, \ldots, k-1\} \tag{F.33}$$

Thus, we get

$$e^{-\varepsilon} \leq \frac{\sum_{i=g_{a-1}+1}^{g_a} P_i^{\mathbf{y}}}{e^{\varepsilon} \sum_{i=g_{a-1}+1}^{g_a} P_i^{\mathbf{y}}} \leq \frac{C_a}{C_j} = \frac{\sum_{i=g_{a-1}+1}^{g_a} P_i^{\mathbf{y}}}{\sum_{i=g_{j-1}+1}^{g_j} P_i^{\mathbf{y}}} \leq \frac{e^{\varepsilon} \sum_{i=g_{j-1}+1}^{g_j} P_i^{\mathbf{y}}}{\sum_{i=g_{j-1}+1}^{g_j} P_i^{\mathbf{y}}} \leq e^{\varepsilon} \qquad \forall a, j \in [k] \tag{F.34}$$

Consider a random key U^1 that has a distribution $\mathbf{C} = [C_1, \ldots, C_k]$, where $C_a = \sum_{i=g_{a-1}+1}^{g_a} P_i^{\mathbf{y}}$. From Lemma 7.5.1, the random key U^1 can be used to construct an ε-LDP-Rec mechanism with the possibility to recover the sample X_1 from the output of the mechanism and the random key U^1. Hence from Theorem 7.5.1, we have

$$H\left(U^1\right) \geq H\left(U_{\min,1}\right). \tag{F.35}$$

This proves inequality (F.30), and completes the proof of Lemma 7.6.2.

F.9 Proof of Lemma F.5.1

For the random variable U', the distribution $\mathbf{q}' = \left[q'_1, \ldots, q'_{m-1}\right]$ is given by

$$q'_j = \frac{q_j}{1-q_m}. \tag{F.36}$$

Note that the distribution $\mathbf{q}'$ is a valid distribution on U' since $\sum_{j=1}^{m-1} q'_j = \sum_{j=1}^{m-1} \frac{q_j}{1-q_m} = 1$. Now, we can bound the difference between $H(U) - H(U')$ as follows

$$\begin{aligned}
H(U) - H(U') &= \sum_{j=1}^{m-1} q'_j \log\left(q'_j\right) - \sum_{j=1}^{m} q_j \log\left(q_j\right) \\
&= \sum_{j=1}^{m-1} \frac{q_j}{1-q_m} \log\left(\frac{q_j}{1-q_m}\right) - \sum_{j=1}^{m} q_j \log\left(q_j\right) \\
&= \sum_{j=1}^{m-1} \frac{q_j}{1-q_m} \left[\log\left(\frac{q_j}{1-q_m}\right) - \log\left(q_j^{(1-q_m)}\right)\right] - q_m \log\left(q_m\right) \\
&= \sum_{j=1}^{m-1} \frac{q_j}{1-q_m} \left[-\log\left(\frac{1-q_m}{q_j^{q_m}}\right)\right] - q_m \log\left(q_m\right) \\
&> -\log\left(\sum_{j=1}^{m-1} q_j^{(1-q_m)}\right) - q_m \log\left(q_m\right) && \text{(F.37)} \\
&\geq -\left(1-q_m\right)\log\left(1-q_m\right) - q_m \log\left(m-1\right) - q_m \log\left(q_m\right) && \text{(F.38)} \\
&\geq \min\left(0, \log\left(\frac{m}{m-1}\right)\right) && \text{(F.39)} \\
&\geq 0 && \text{(F.40)}
\end{aligned}$$

where (F.37) follows from the fact that $-\log(.)$ is a strictly convex function and $q_j/1-q_m > 0$ for $j \in [m-1]$. The inequality (F.38) follows from solving the convex problem

$$\begin{aligned} \max_{\{q_j\}_{j=1}^{m-1}} & \sum_{j=1}^{m-1} q_j^{(1-q_m)} \\ s.t. & \sum_{j=1}^{m-1} q_j = 1 - q_m \\ & q_j \geq q_m \ \forall j \in [m-1] \end{aligned} \tag{F.41}$$

Note that x^a is a concave function on $x \in \mathbb{R}_+$ for $0 \leq a \leq 1$. Therefore, the objective function in (F.41) is concave in $\{q_j\}$. By solving the optimization problem in (F.41), we get $q_j^* = \frac{1-q_m}{m-1} \geq q_m$ for all $j \in [m-1]$ and $\sum_{j=1}^{m-1} q_j^{(1-q_m)} \leq \frac{(1-q_m)^{(1-q_m)}}{(m-1)^{(-q_m)}}$. Since $\log(x)$ is a monotonic function, we get $-\log\left(\sum_{j=1}^{m-1} q_j^{(1-q_m)}\right) \geq -(1-q_m)\log(1-q_m) - q_m \log(m-1)$. The inequality (F.39) follows from the fact that $-(1-q_m)\log(1-q_m) - q_m\log(m-1) - q_m\log(q_m) = H(q_m) - q_m\log(m-1)$ is a concave function of q_m. The minimum of a concave function is one of the vertices, where $q_m \in \{0, \frac{1}{m}\}$. Hence, the proof is completed.

REFERENCES

[ACF95] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. "Gambling in a rigged casino: The adversarial multi-armed bandit problem." In *Proceedings of IEEE 36th annual foundations of computer science*, pp. 322–331. IEEE, 1995.

[ACF18] Jayadev Acharya, Clément L Canonne, Cody Freitag, and Himanshu Tyagi. "Test without trust: Optimal locally private distribution testing." *arXiv preprint arXiv:1808.02174*, 2018.

[ACG16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy." In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

[AFT22] Hilal Asi, Vitaly Feldman, and Kunal Talwar. "Optimal algorithms for mean estimation under local differential privacy." In *International Conference on Machine Learning*, pp. 1046–1056. PMLR, 2022.

[AGL17] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. "QSGD: Communication-efficient SGD via gradient quantization and encoding." In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

[App17] Apple. "Differential Privavy." 2017.

[APS11] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. "Improved algorithms for linear stochastic bandits." *Advances in neural information processing systems*, **24**, 2011.

[AS19] Jayadev Acharya and Ziteng Sun. "Communication Complexity in Locally Private Distribution Estimation and Heavy Hitters." In *International Conference on Machine Learning (ICML)*, pp. 51–60, 2019.

[ASY18] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. "cpSGD: Communication-efficient and differentially-private distributed SGD." In *Advances in Neural Information Processing Systems*, pp. 7564–7575, 2018.

[ASZ19] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. "Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication." In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1120–1129, 2019.

[AZZ21] Durmus Alp Emre Acar, Yue Zhao, Ruizhao Zhu, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. "Debiasing Model Updates for Improving Personalized Federated Training." In *International Conference on Machine Learning*, pp. 21–31. PMLR, 2021.

[BBG19a] Borja Balle, Gilles Barthe, Marco Gaboardi, and Joseph Geumlek. "Privacy amplification by mixing and diffusion mechanisms." In *Advances in Neural Information Processing Systems*, pp. 13277–13287, 2019.

[BBG19b] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. "Differentially private summation with multi-message shuffling." *arXiv preprint arXiv:1906.09116*, 2019.

[BBG19c] Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. "Improved summation from shuffling." *arXiv preprint arXiv:1909.11225*, 2019.

[BBG19d] Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. "The privacy blanket of the shuffle model." In *Annual International Cryptology Conference*, pp. 638–667. Springer, 2019.

[BBG20a] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. "Hypothesis Testing Interpretations and Renyi Differential Privacy." In Silvia Chiappa and Roberto Calandra, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2496–2506. PMLR, 2020.

[BBG20b] Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. "Private summation in the multi-message shuffle model." In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 657–676, 2020.

[BC20] Victor Balcer and Albert Cheu. "Separating Local & Shuffled Differential Privacy via Histograms." In Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs, editors, *1st Conference on Information-Theoretic Cryptography, ITC 2020, June 17-19, 2020, Boston, MA, USA*, volume 163 of *LIPIcs*, pp. 1:1–1:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[BDF18] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. "Protection against reconstruction and its applications in private federated learning." *arXiv preprint arXiv:1812.00984*, 2018.

[BDK19] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations." In *Advances in Neural Information Processing Systems*, pp. 14695–14706, 2019.

[BDR18] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. "Composable and versatile privacy via truncated CDP." In *ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 74–86, 2018.

[BEM17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. "Prochlo: Strong Privacy for Analytics in the Crowd." In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP)*, pp. 441–459. ACM, 2017.

[BHO19] Leighton Pate Barnes, Yanjun Han, and Ayfer Ozgur. "Learning Distributions from their Samples under Communication Constraints." *arXiv preprint arXiv:1902.02890*, 2019.

[BKM20] Borja Balle, Peter Kairouz, H Brendan McMahan, Om Thakkar, and Abhradeep Thakurta. "Privacy amplification via random check-ins." *arXiv preprint arXiv:2007.06605*, 2020.

[BNO08] Amos Beimel, Kobbi Nissim, and Eran Omri. "Distributed private data analysis: Simultaneously solving how and what." In *Annual International Cryptology Conference*, pp. 451–468. Springer, 2008.

[BNS17] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. "Practical locally private heavy hitters." In *Advances in Neural Information Processing Systems*, pp. 2288–2296, 2017.

[BNS18] Mark Bun, Jelani Nelson, and Uri Stemmer. "Heavy hitters and the structure of local privacy." In *ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 435–447, 2018.

[BR19] Djallel Bouneffouf and Irina Rish. "A survey on practical applications of multi-armed and contextual bandits." *arXiv preprint arXiv:1904.10040*, 2019.

[BRC17] Djallel Bouneffouf, Irina Rish, and Guillermo A Cecchi. "Bandit models of human behavior: Reward processing in mental disorders." In *International Conference on Artificial General Intelligence*, pp. 237–248. Springer, 2017.

[BS15] Raef Bassily and Adam Smith. "Local, private, efficient protocols for succinct histograms." In *STOC*, pp. 127–135, 2015.

[BS16] Mark Bun and Thomas Steinke. "Concentrated differential privacy: Simplifications, extensions, and lower bounds." In *Theory of Cryptography Conference (TCC)*, pp. 635–658. Springer, 2016.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. "Private empirical risk minimization: Efficient algorithms and tight error bounds." In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

[BUV14] Mark Bun, Jonathan Ullman, and Salil Vadhan. “Fingerprinting codes and the price of approximate differential privacy.” In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 1–10, 2014.

[BWA18] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. “signSGD: Compressed optimisation for non-convex problems.” In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

[BZH06] Michael Barbaro, Tom Zeller, and Saul Hansell. “A face is exposed for AOL searcher no. 4417749.” *New York Times*, **9**(2008):8, 2006.

[CCK22] Wei-Ning Chen, Christopher A Choquette Choo, Peter Kairouz, and Ananda Theertha Suresh. “The Fundamental Price of Secure Aggregation in Differentially Private Federated Learning.” In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3056–3089, 17–23 Jul 2022.

[CDW18] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. “Leaf: A benchmark for federated settings.” *arXiv preprint arXiv:1812.01097*, 2018.

[CGK21] Alisa Chang, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. “Locally private k-means in one round.” In *International Conference on Machine Learning*, pp. 1441–1451. PMLR, 2021.

[CJM22] Albert Cheu, Matthew Joseph, Jieming Mao, and Binghui Peng. “Shuffle Private Stochastic Convex Optimization.” In *International Conference on Learning Representations (ICLR)*, 2022.

[CKM18] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. “Expanding the reach of federated learning by reducing client resource requirements.” *arXiv preprint arXiv:1812.07210*, 2018.

[CKO20] Wei-Ning Chen, Peter Kairouz, and Ayfer Ozgur. “Breaking the communication-privacy-accuracy trilemma.” *Advances in Neural Information Processing Systems*, **33**:3312–3324, 2020.

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. “The Discrete Gaussian for Differential Privacy.” In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. "Differentially private empirical risk minimization." *Journal of Machine Learning Research*, **12**(3), 2011.

[CSS12] TH Hubert Chan, Elaine Shi, and Dawn Song. "Optimal lower bound for differentially private multi-party aggregation." In *European Symposium on Algorithms*, pp. 277–288. Springer, 2012.

[CSU19] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. "Distributed Differential Privacy via Shuffling." In *Advances in Cryptology - EUROCRYPT 2019*, volume 11476, pp. 375–403. Springer, 2019.

[CTW21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. "Extracting training data from large language models." In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

[CZ22] Sayak Ray Chowdhury and Xingyu Zhou. "Shuffle Private Linear Contextual Bandits." *arXiv preprint arXiv:2202.05567*, 2022.

[DJW18] John C Duchi, Michael I Jordan, and Martin J Wainwright. "Minimax optimal procedures for locally private estimation." *Journal of the American Statistical Association*, **113**(521):182–201, 2018.

[DKM20] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. "Adaptive Personalized Federated Learning." *arXiv preprint arXiv:2003.13461*, 2020.

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. "Collecting Telemetry Data Privately." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, p. 3574–3583, Red Hook, NY, USA, 2017. Curran Associates Inc.

[DLM12] Yevgeniy Dodis, Adriana López-Alt, Ilya Mironov, and Salil P. Vadhan. "Differential Privacy with Imperfect Randomness." In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO*, volume 7417, pp. 497–516. Springer, 2012.

[DMN06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating noise to sensitivity in private data analysis." In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

[DR14] Cynthia Dwork and Aaron Roth. "The algorithmic foundations of differential privacy." *Foundations and Trends® in Theoretical Computer Science*, **9**(3–4):211–407, 2014.

[DR19] John Duchi and Ryan Rogers. “Lower bounds for locally private estimation via communication complexity.” *arXiv preprint arXiv:1902.00582*, 2019.

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. “Boosting and differential privacy.” In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60. IEEE, 2010.

[DTN20] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. “Personalized Federated Learning with Moreau Envelopes.” In *Advances in Neural Information Processing Systems*, 2020.

[DWJ13] John Duchi, Martin J Wainwright, and Michael I Jordan. “Local privacy and minimax bounds: Sharp rates for probability estimation.” In *Advances in Neural Information Processing Systems*, pp. 1529–1537, 2013.

[Dwo19] Cynthia Dwork. “Differential Privacy and the US Census.” In *ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 1–1, 2019.

[EFM19] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. “Amplification by shuffling: From local to central differential privacy via anonymity.” In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

[EFM20a] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. “Encode, Shuffle, Analyze Privacy Revisited: Formalizations and Empirical Evaluation.” *CoRR*, **abs/2001.03618**, 2020.

[EFM20b] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. “Encode, shuffle, analyze privacy revisited: formalizations and empirical evaluation.” *arXiv preprint arXiv:2001.03618*, 2020.

[EH14] Tim van Erven and Peter Harremoës. “Rényi Divergence and Kullback-Leibler Divergence.” *IEEE Trans. Inf. Theory*, **60**(7):3797–3820, 2014.

[EKM21] Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. “Regret bounds for batched bandits.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7340–7348, 2021.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. “RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response.” In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067, 2014.

[FMO20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. "Personalized Federated Learning: A Meta-Learning Approach." In *Advances in Neural Information Processing Systems*, 2020.

[FMT22] Vitaly Feldman, Audra McMillan, and Kunal Talwar. "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling." In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2022.

[FMT23] Vitaly Feldman, Audra McMillan, and Kunal Talwar. "Stronger Privacy Amplification by Shuffling for Renyi and Approximate Differential Privacy." In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 4966–4981. SIAM, 2023.

[FT21] Vitaly Feldman and Kunal Talwar. "Lossless compression of efficient private local randomizers." In *International Conference on Machine Learning*, pp. 3208–3219. PMLR, 2021.

[FW56] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming." *Naval research logistics quarterly*, **3**(1-2):95–110, 1956.

[GBD20] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, **33**:16937–16947, 2020.

[GCP22] Evrard Garcelon, Kamalika Chaudhuri, Vianney Perchet, and Matteo Pirotta. "Privacy Amplification via Shuffling for Linear Contextual Bandits." In *International Conference on Algorithmic Learning Theory*, pp. 381–407. PMLR, 2022.

[GD23] Antonious M Girgis and Suhas Diggavi. "Multi-Message Shuffled Privacy in Federated Learning." *arXiv preprint arXiv:2302.11152*, 2023.

[GDC20] Antonious M Girgis, Deepesh Data, Kamalika Chaudhuri, Christina Fragouli, and Suhas N Diggavi. "Successive refinement of privacy." *IEEE Journal on Selected Areas in Information Theory*, **1**(3):745–759, 2020.

[GDD21a] Antonious Girgis, Deepesh Data, Suhas N. Diggavi, Peter Kairouz, and Ananda Theertha Suresh. "Shuffled Model of Differential Privacy in Federated Learning." In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2521–2529. PMLR, 2021.

[GDD21b] Antonious M Girgis, Deepesh Data, and Suhas Diggavi. "Differentially private federated learning with shuffling and client self-sampling." In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 338–343. IEEE, 2021.

[GDD21c] Antonious M Girgis, Deepesh Data, and Suhas Diggavi. “Renyi differential privacy of the subsampled shuffle model in distributed learning.” *Advances in Neural Information Processing Systems*, **34**:29181–29192, 2021.

[GDD21d] Antonious M. Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. “Shuffled Model of Federated Learning: Privacy, Accuracy and Communication Trade-Offs.” *IEEE Journal on Selected Areas in Information Theory*, **2**(1):464–478, 2021.

[GDD21e] Antonious M Girgis, Deepesh Data, Suhas Diggavi, Ananda Theertha Suresh, and Peter Kairouz. “On the renyi differential privacy of the shuffle model.” In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2321–2341, 2021.

[GDD22] Antonious M Girgis, Deepesh Data, and Suhas Diggavi. “Distributed user-level private mean estimation.” In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2196–2201. IEEE, 2022.

[GGK19] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. “On the Power of Multiple Anonymous Messages.” *IACR Cryptol. ePrint Arch.*, **2019**:1382, 2019.

[GKM20] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. “Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead.” In *International Conference on Machine Learning*, pp. 3505–3514. PMLR, 2020.

[GKM21a] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. “User-Level Differentially Private Learning via Correlated Sampling.” In *Advances in Neural Information Processing Systems*, 2021.

[GKM21b] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. “Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message.” In *International Conference on Machine Learning*, pp. 3692–3701. PMLR, 2021.

[GPV19] Badih Ghazi, Rasmus Pagh, and Ameya Velingker. “Scalable and differentially private distributed aggregation in the shuffled model.” *arXiv preprint arXiv:1906.08320*, 2019.

[Gre16] Andy Greenberg. “Apple’s ‘differential privacy’is about collecting your data—but not your data.” *Wired, June*, **13**, 2016.

[HGF22] Osama A. Hanna, Antonious M. Girgis, Christina Fragouli, and Suhas N. Diggavi. “Differentially Private Stochastic Linear Bandits: (Almost) for Free.” *CoRR*, **abs/2207.03445**, 2022. Posted July 13, 2022 on arxiv.

[HGL20] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. "Personalized federated learning with differential privacy." *IEEE Internet of Things Journal*, **7**(10):9530–9539, 2020.

[HKR12] Justin Hsu, Sanjeev Khanna, and Aaron Roth. "Distributed private heavy hitters." In *International Colloquium on Automata, Languages, and Programming*, pp. 461–472. Springer, 2012.

[HLW21] Yuxuan Han, Zhipeng Liang, Yang Wang, and Jiheng Zhang. "Generalized linear bandits with local differential privacy." volume 34, 2021.

[HLY21] Ziyue Huang, Yuting Liang, and Ke Yi. "Instance-optimal mean estimation under differential privacy." *Advances in Neural Information Processing Systems*, **34**:25993–26004, 2021.

[HR20] Filip Hanzely and Peter Richtárik. "Federated Learning of a Mixture of Global and Local Models." *arXiv preprint arXiv:2002.05516*, 2020.

[HYF22] Osama Hanna, Lin Yang, and Christina Fragouli. "Learning from Distributed Users in Contextual Linear Bandits Without Sharing the Context." *Advances in Neural Information Processing Systems*, **35**:11049–11062, 2022.

[HYF23] Osama A Hanna, Lin Yang, and Christina Fragouli. "Contexts can be cheap: Solving stochastic contextual bandits with linear bandit algorithms." In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 1791–1821. PMLR, 2023.

[HZZ22] Jiahao He, Jiheng Zhang, and Rachel Zhang. "A reduction from linear contextual bandit lower bounds to estimation lower bounds." In *International Conference on Machine Learning*, pp. 8660–8677. PMLR, July 2022.

[IKO06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Cryptography from anonymity." In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 239–248. IEEE, 2006.

[JRS21] Prateek Jain, John Rush, Adam Smith, Shuang Song, and Abhradeep Guha Thakurta. "Differentially Private Model Personalization." In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[Kai19] Peter Kairouz et al. "Advances and Open Problems in Federated Learning." *CoRR*, **abs/1912.04977**, 2019.

[Kas77] Boris S Kashin. "Diameters of some finite-dimensional sets and classes of smooth functions." *Math. USSR, Izv*, **11**(2):317–333, 1977.

[KBR16] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. "Discrete distribution estimation under local privacy." *arXiv preprint arXiv:1602.07387*, 2016.

[KLN11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. "What can we learn privately?" *SIAM Journal on Computing*, **40**(3):793–826, 2011.

[KLS21] Peter Kairouz, Ziyu Liu, and Thomas Steinke. "The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation." In *Proceedings International Conference on Machine Learning, ICML*, volume 139, pp. 5201–5212, 2021.

[KMS21] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. "Practical and private (deep) learning without sampling or shuffling." In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021.

[KOV14] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. "Extremal mechanisms for local differential privacy." In *Advances in neural information processing systems*, pp. 2879–2887, 2014.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. "The composition theorem for differential privacy." In *International conference on machine learning*, pp. 1376–1385. PMLR, 2015.

[LAS14] Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. "Communication efficient distributed machine learning with the parameter server." *Advances in Neural Information Processing Systems*, **27**, 2014.

[LHB21] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. "Ditto: Fair and robust federated learning through personalization." In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

[LKC20] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. "Differentially Private Meta-Learning." In *International Conference on Learning Representations*, 2020.

[LQS12] Ninghui Li, Wahbeh Qardaji, and Dong Su. "On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy." In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32–33, 2012.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[LSA21] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. "Learning with user-level privacy." *Advances in Neural Information Processing Systems*, **34**:12466–12479, 2021.

[LSY20] Yuhan Liu, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Michael Riley. "Learning discrete distributions: user vs item-level privacy." *Advances in Neural Information Processing Systems*, **33**:20965–20976, 2020.

[LV10] Yurii Lyubarskii and Roman Vershynin. "Uncertainty principles and vector quantization." *IEEE Transactions on Information Theory*, **56**(7):3491–3501, 2010.

[LZJ22] Fengjiao Li, Xingyu Zhou, and Bo Ji. "Differentially private linear bandits with partial distributed feedback." In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pp. 41–48. IEEE, September 2022.

[MAE18] H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. "A general approach to adding differential privacy to iterative training procedures." *arXiv preprint arXiv:1812.06210*, 2018.

[MGP15] Jérémie Mary, Romaric Gaudel, and Philippe Preux. "Bandits and recommender systems." In *International Workshop on Machine Learning, Optimization and Big Data*, pp. 325–336. Springer, 2015.

[Mir17] Ilya Mironov. "Rényi differential privacy." In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

[MMR17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

[MMR20] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. "Three Approaches for Personalization with Applications to Federated Learning." *arXiv preprint arXiv:2002.10619*, 2020.

[MSU22] Audra McMillan, Adam Smith, and Jon Ullman. "Instance-optimal differentially private estimation." *arXiv preprint arXiv:2210.15819*, 2022.

[MT20] Prathamesh Mayekar and Himanshu Tyagi. "Limits on Gradient Compression for Stochastic Optimization." *IEEE International Symposium on Information Theory (ISIT)*, 2020.

[MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. "R\'enyi Differential Privacy of the Sampled Gaussian Mechanism." *arXiv preprint arXiv:1908.10530*, 2019.

[NS08] Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets." In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125. IEEE, 2008.

[NSR11] Arvind Narayanan, Elaine Shi, and Benjamin IP Rubinstein. “Link prediction by de-anonymization: How we won the kaggle social network challenge.” In *The 2011 International Joint Conference on Neural Networks*, pp. 1825–1834. IEEE, 2011.

[OGD22] Kaan Ozkara, Antonious M Girgis, Deepesh Data, and Suhas Diggavi. “A Statistical Framework for Personalized Federated Learning and Estimation: Theory, Algorithms, and Privacy.” In *The Eleventh International Conference on Learning Representations*, 2022.

[OSD21] Kaan Ozkara, Navjot Singh, Deepesh Data, and Suhas Diggavi. “QuPeD: Quantized Personalization via Distillation with Applications to Federated Learning.” *Advances in Neural Information Processing Systems*, **34**, 2021.

[PRC16] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. “Batched bandit problems.” *The Annals of Statistics*, pp. 660–681, 2016.

[PTS20] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. “Tempered Sigmoid Activations for Deep Learning with Differential Privacy.” *arXiv preprint arXiv:2007.14191*, 2020.

[QYY16] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. “Heavy hitter estimation over set-valued data with local differential privacy.” In *CCS*, pp. 192–203. ACM, 2016.

[RH15] Phillippe Rigollet and Jan-Christian Hütter. “High dimensional statistics.” *Lecture notes for course 18S997*, **813**:814, 2015.

[Ros83] J Ben Rosen. “Global minimization of a linearly constrained concave function by partition of feasible domain.” *Mathematics of Operations Research*, **8**(2):215–230, 1983.

[RT10] Paat Rusmevichientong and John N Tsitsiklis. “Linearly parameterized bandits.” *Mathematics of Operations Research*, **35**(2):395–411, 2010.

[RYW18] Anna N Rafferty, Huiji Ying, and Joseph Jay Williams. “Bandit assignment for educational experiments: Benefits to students versus statistical power.” In *International Conference on Artificial Intelligence in Education*, pp. 286–290. Springer, 2018.

[RZL20] Wenbo Ren, Xingyu Zhou, Jia Liu, and Ness B Shroff. “Multi-armed bandits with local differential privacy.” *arXiv preprint arXiv:2007.03121*, 2020.

[SC13] Anand D. Sarwate and Kamalika Chaudhuri. “Signal Processing and Machine Learning with Differential Privacy: Algorithms and Challenges for Continuous Data.” *IEEE Signal Process. Mag.*, **30**(5):86–94, 2013.

[SCJ18] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. "Sparsified SGD with memory." In *Advances in Neural Information Processing Systems*, pp. 4447–4458, 2018.

[SFK17] Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. "Distributed mean estimation with limited communication." In *International conference on machine learning*, pp. 3329–3337. PMLR, 2017.

[Sha49] Claude E Shannon. "Communication theory of secrecy systems." *Bell system technical journal*, **28**(4):656–715, 1949.

[Sha12] Shai Shalev-Shwartz et al. "Online learning and online convex optimization." *Foundations and Trends® in Machine Learning*, **4**(2):107–194, 2012.

[SS18] Roshan Shariff and Or Sheffet. "Differentially private contextual linear bandits." volume 31, 2018.

[SS19] Touqir Sajed and Or Sheffet. "An optimal private stochastic-mab algorithm based on optimal private stopping rule." In *International Conference on Machine Learning*, pp. 5579–5588. PMLR, 2019.

[SSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models." In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

[Swe97] Latanya Sweeney. "Guaranteeing anonymity when sharing medical data, the Datafly System." In *Proceedings of the AMIA Annual Fall Symposium*, p. 51. American Medical Informatics Association, 1997.

[Swe02] Latanya Sweeney. "k-anonymity: A model for protecting privacy." *International journal of uncertainty, fuzziness and knowledge-based systems*, **10**(05):557–570, 2002.

[SZ13] Ohad Shamir and Tong Zhang. "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes." In *International conference on machine learning*, pp. 71–79, 2013.

[TKM21] Jay Tenenbaum, Haim Kaplan, Yishay Mansour, and Uri Stemmer. "Differentially private multi-armed bandits in the shuffle model." *Advances in Neural Information Processing Systems*, **34**, 2021.

[Tsy08] Alexandre B Tsybakov. *Introduction to nonparametric estimation.* Springer Science & Business Media, 2008.

[Ull17] Jonathan Ullman. "CS7880. Rigorous approaches to data privacy." 2017.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[War65] Stanley L Warner. "Randomized response: A survey technique for eliminating evasive answer bias." *Journal of the American Statistical Association*, **60**(309):63–69, 1965.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. "Subsampled Rényi differential privacy and analytical moments accountant." In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235. PMLR, 2019.

[WHW16] Shaowei Wang, Liusheng Huang, Pengzhan Wang, Yiwen Nie, Hongli Xu, Wei Yang, Xiang-Yang Li, and Chunming Qiao. "Mutual information optimally local private discrete distribution estimation." *arXiv preprint arXiv:1607.08025*, 2016.

[Wik] Wikipedia. "Gamma function." `https://en.wikipedia.org/wiki/Gamma_function`.

[WSZ19] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. "Beyond inferring class representatives: User-level privacy leakage from federated learning." In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520. IEEE, 2019.

[YB18] M. Ye and A. Barg. "Optimal Schemes for Discrete Distribution Estimation Under Locally Differential Privacy." *IEEE Transactions on Information Theory*, **64**(8):5662–5676, Aug 2018.

[Zam98] Ram Zamir. "A proof of the Fisher information inequality via a data processing argument." *IEEE Transactions on Information Theory*, **44**(3):1246–1250, 1998.

[ZCH20] Kai Zheng, Tianle Cai, Weiran Huang, Zhenguo Li, and Liwei Wang. "Locally differentially private (contextual) bandits learning." volume 33, pp. 12300–12310, 2020.

[ZLH19] Ligeng Zhu, Zhijian Liu, and Song Han. "Deep leakage from gradients." *Advances in neural information processing systems*, **32**, 2019.

[ZSF21] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. "Personalized Federated Learning with First Order Model Optimization." In *International Conference on Learning Representations*, 2021.

[ZW19] Yuqing Zhu and Yu-Xiang Wang. "Poission subsampled rényi differential privacy." In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.