Rényi Differential Privacy of the Subsampled Shuffle Model in Distributed Learning

Antonious M. Girgis
UCLA
amgirgis@g.ucla.edu

Deepesh Data
UCLA
deepesh.data@gmail.com

Suhas Diggavi
UCLA
suhasdiggavi@ucla.edu

Abstract

We study privacy in a distributed learning framework, where clients collaboratively build a learning model iteratively through interactions with a server from whom we need privacy. Motivated by stochastic optimization and the federated learning (FL) paradigm, we focus on the case where a small fraction of data samples are randomly sub-sampled in each round to participate in the learning process, which also enables privacy amplification. To obtain even stronger local privacy guarantees, we study this in the shuffle privacy model, where each client randomizes its response using a local differentially private (LDP) mechanism and the server only receives a random permutation (shuffle) of the clients' responses without their association to each client. The principal result of this paper is a privacy-optimization performance trade-off for discrete randomization mechanisms in this sub-sampled shuffle privacy model. This is enabled through a new theoretical technique to analyze the Rényi Differential Privacy (RDP) of the sub-sampled shuffle model. We numerically demonstrate that, for important regimes, with composition our bound yields significant improvement in privacy guarantee over the state-of-the-art approximate Differential Privacy (DP) guarantee (with strong composition) for sub-sampled shuffled models. We also demonstrate numerically significant improvement in privacy-learning performance operating point using real data sets. Despite these advances an open question is to bridge the gap between lower and upper privacy bounds in our RDP analysis.

1 Introduction

As learning moves towards the edge, there is a need to collaborate to build learning models[1], such as in federated learning [36, 44, 33]. In this framework, the collaboration is typically mediated by a server. In particular, we want to collaboratively build a learning model by solving an empirical risk minimization (ERM) problem (see (2) in Section 2). To obtain a model parametrized by θ using ERM, the commonly used mechanism is Stochastic Gradient Descent (SGD) [12]. However, one needs to solve this while enabling strong privacy guarantees on local data from the server, while also obtaining good learning performance, *i.e.,* a suitable privacy-learning performance operating point.

Differential privacy (DP) [18] is the gold standard notion of data privacy that gives a rigorous framework through quantifying the information leakage about individual training data points from the observed interactions. Though DP was originally proposed in a framework where data resides centrally [18], for distributed learning the more appropriate notion is of local differential privacy (LDP) [35, 17]. Here, each client randomizes its interactions with the server from whom the data is to be kept private (*e.g.,* see industrial implementations [23, 31, 16]). However, LDP mechanisms suffer

[1] This is because no client has access to enough data to build rich learning models locally and we do not want to directly share local data.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

from poor performance in comparison with the central DP mechanisms [17, 35, 32]. To overcome this, a new privacy framework using anonymization has been proposed in the so-called *shuffled model* [22, 25, 6, 26, 5, 15, 7, 8]. In the shuffled model, each client sends her private message to a secure shuffler that randomly permutes all the received messages before forwarding them to the server. This model enables significantly better privacy-utility performance by amplifying DP through this shuffling. Therefore, in this paper we consider the shuffle privacy framework for distributed learning.

In solving (2) using (distributed) gradient descent, each exchange leaks information about the local data, but we need as many steps as possible to obtain a good model; setting up the tension between privacy and performance. The goal is to obtain as many such interactions as possible for a given privacy budget. This is quantified through analyzing the privacy of the composition of privacy mechanisms. Abadi *et al.* [1] developed a framework for tighter analysis of such compositions, and this was later reformulated in terms of Rényi Differential Privacy (RDP) [37], and mapping this back to DP guarantee [38]. Therefore, studying RDP is important to obtaining strong composition privacy results, and is the focus of this paper.

In distributed (and federated) learning, a fraction of the data samples are sampled; for example, with random client participation and stochastic gradient descent (SGD), which can be written as

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{R}(\nabla f_i(\theta_t)),$$

where $\mathcal{R}$ is the local randomization mechanism and $\mathcal{I}$ are the indices of the sampled data. This is a subsampled mechanism that enables another privacy amplification opportunity; which, in several cases, is shown to yield a privacy advantage proportional to the subsampling rate; see [35, 42]. The central technical question addressed in this paper is how to analyze the RDP of an arbitrary discrete mechanism for the subsampled shuffle privacy model. This enables us to answer the overall question posed in this paper, which is an achievable privacy-learning performance trade-off point for solving (2) in the shuffled privacy model for distributed learning (see Figure 1). Our contributions are:

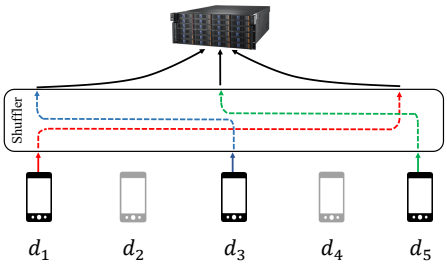


Figure 1: An iteration from the CLDP-SGD Algorithm, where 3 clients are randomly chosen at each iteration. Each client sends the private gradient $\mathcal{R}_p\left(g_t(d_i)\right)$ to the shuffler that randomly permutes the gradients before passing them to the server.

• We analyze the RDP of subsampled mechanisms in the shuffle framework by developing a novel bound applicable to any discrete ϵ_0-LDP mechanism as a function of the RDP order λ, subsampling rate γ, the LDP parameter ϵ_0, and the number of clients n; see Theorem 1. The bound is explicit and amenable to numerics, including all constants.[2] Furthermore, the bounds are valid for generic LDP mechanisms and *all* parameter regimes.[3] We also provide a lower bound for the RDP in Theorem 2. We prove our upper bound (Theorem 1) using the following novel analysis techniques: First, we reduce the problem of computing the RDP of sub-sampled shuffle mechanisms to the problem of computing ternary $|\chi|^\alpha$-DP [43] of shuffle (non sub-sampled) mechanisms; see Lemma 2. Then we reduce the computation of the ternary $|\chi|^\alpha$-DP of shuffle mechanisms for a *generic* triple of neighboring datasets to those that have a special structure (see Theorem 5) – this reduction step is one of the core technical results of this paper. Then we bound the ternary $|\chi|^\alpha$-DP of the shuffle mechanisms for triples of neighboring datasets having special structures by bounding the Pearson-Vajda divergence [43] using some concentration properties (see Theorem 6).

• Using the core technical result in Theorem 1, we analyze privacy-convergence trade-offs of the CLDP-SGD algorithm (see Algorithm 1) for Lipschitz convex functions in Theorem 3. This partially resolves an open question posed in [27], to extend their privacy analysis to RDP and significantly strengthening their privacy guatantees.

• Numerically, we save a factor $14\times$ in privacy (ϵ) over the best known results for approximate DP for shuffling [24] combined with strong composition [34] for $T = 10^5, \gamma = 0.001, n = 10^6$, and a factor of $2.5\times$ better than the best known RDP for shuffling bound [29] combined with the sub-sampling result in [43]. Translating these to privacy-performance operating point in distributed

[2] As emphasized in [43], "in differential privacy, constants matter".

[3] Some of the best known approximate DP bounds for the shuffle model [7, 24] are restricted to certain parameter regimes in terms of n, δ, ϵ_0, etc.

optimization, over the MNIST data set with ℓ_∞ clipping we numerically show gains: For the same privacy budget of $\epsilon = 1.4$, we get a test performance of 80% whereas using strong composition the test performance of [24] is 70%; furthermore, we achieves 90% accuracy with the total privacy budget $\epsilon = 2.91$, whereas, [24] (with strong composition) achieves the same accuracy with a total privacy budget of $\epsilon = 4.82$. See Section 4 and the supplementary material for more results.

Related work: We give a more complete literature review in Appendix A, and focus here on the works that are closest to the results presented in this paper.

Private optimization in the shuffled model: Recently, [21] and [27, 28] have proposed differentially private SGD algorithms for federated learning, where at each iteration, each client applies an LDP mechanism on the gradients with the existence of a secure shuffler between the clients and the central server. However, the privacy analyses in these works developed approximate DP using advanced composition theorems for DP (*e.g.,* [20, 34]), which are known to be loose for composition [1]. To the best of our knowledge, analyzing the private optimization framework using RDP and subsampling in the shuffled model is new to this paper.

Subsampled RDP: The works [38, 43, 45] have studied the RDP of subsampled mechanisms *without shuffling*. They demonstrated that this 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,[4] and is new to this paper. The RDP of the shuffled model was very recently studied in [29], but without incorporating subsampling, which poses new technical challenges, as directly bounding the RDP of subsampled shuffle mechanisms is non-trivial. We overcome this by reducing our problem of computing RDP to bounding the ternary $|\chi|^\alpha$-DP, and bounding the latter is a core technical contribution of our paper.

Paper organization: We give preliminaries and problem formulation in Section 2, main results (upper and lower bounds, and privacy-convergence tradeoff) in Section 3, numerical results in Section 4, proof of the upper bound in Section 5, and proof of the ternary DP of the shuffle model in Section 6. Omitted details/proofs from this paper are given in the supplementary material.

2 Preliminaries and Problem Formulation

We use several privacy definitions throughout this paper. Among these, the local and central differential privacy definitions are standard and we defer them to Appendix B. The other privacy definitions (Rényi DP and ternary $|\chi|^\alpha$-DP) are relatively less standard and we define them below.

We say that two datasets $\mathcal{D} = \{d_1, \ldots, d_n\} \in \mathcal{X}^n$ and $\mathcal{D}' = \{d'_1, \ldots, d'_n\} \in \mathcal{X}^n$ 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 $d_j = d'_j$.

Definition 1 ((λ, ϵ)-RDP (Rényi Differential Privacy) [37])**.** A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to have ϵ-Rényi differential privacy of order $\lambda \in (1, \infty)$ (in short, $(\lambda, \epsilon(\lambda))$-RDP), if for any neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$, the Rényi divergence of order λ between $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ is upper-bounded by $\epsilon(\lambda)$, *i.e.*,

$$D_\lambda(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) = \frac{1}{\lambda - 1} \log \left(\mathbb{E}_{\theta \sim \mathcal{M}(\mathcal{D}')} \left[\left(\frac{\mathcal{M}(\mathcal{D})(\theta)}{\mathcal{M}(\mathcal{D}')(\theta)} \right)^\lambda \right] \right) \leq \epsilon(\lambda), \tag{1}$$

where $\mathcal{M}(\mathcal{D})(\theta)$ denotes the probability that $\mathcal{M}$ on input $\mathcal{D}$ generates the output θ.

Definition 2 (ζ-Ternary $|\chi|^\alpha$-differential privacy [43])**.** A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to have ζ-ternary-$|\chi|^\alpha$-DP, if for any triple of mutually adjacent datasets $\mathcal{D}, \mathcal{D}', \mathcal{D}'' \in \mathcal{X}^n$ (*i.e.*, they mutually differ in the same location), the ternary-$|\chi|^\alpha$ divergence of $\mathcal{M}(\mathcal{D}), \mathcal{M}(\mathcal{D}'), \mathcal{M}(\mathcal{D}')$ is upper-bounded by $(\zeta(\alpha))^\alpha$ for all $\alpha \geq 1$ (where ζ is a function from $\mathbb{R}^+$ to $\mathbb{R}^+$), *i.e.*,

$$D_{|\chi|^\alpha}\left(\mathcal{M}(\mathcal{D}), \mathcal{M}(\mathcal{D}')||\mathcal{M}(\mathcal{D}'')\right) := \mathbb{E}_{\mathcal{M}(\mathcal{D}'')} \left[\left| \frac{\mathcal{M}(\mathcal{D}) - \mathcal{M}(\mathcal{D}')}{\mathcal{M}(\mathcal{D}'')} \right|^\alpha \right] \leq (\zeta(\alpha))^\alpha .$$

[4] One naive approach is to plug in the RDP analysis of shuffle model [29] into the results of [43]; however, our direct analysis of subsampled mechanisms yields better results in several interesting regimes; see Section 4.

The ternary $|\chi|^\alpha$-DP was proposed in [43] to characterize the RDP of the sub-sampled mechanism without shuffling. In this work, we analyze the ternary $|\chi|^\alpha$-DP of the shuffled mechanism to bound the RDP of the sub-sampled shuffle model.

We can use the following result for converting the RDP guarantees of a mechanism to its central DP guarantees. To the best of our knowledge, this result gives the best conversion.

Lemma 1 (From RDP to DP [13, 4])**.** *Suppose for any $\lambda > 1$, a mechanism $\mathcal{M}$ is $(\lambda, \epsilon(\lambda))$-RDP. Then, the mechanism $\mathcal{M}$ is (ϵ, δ)-DP, where $\delta > 0$ is arbitrary and ϵ is given by*

$$\epsilon = \min_{\lambda} \left(\epsilon(\lambda) + \frac{\log(1/\delta) + (\lambda - 1)\log(1 - 1/\lambda) - \log(\lambda)}{\lambda - 1} \right).$$

Problem formulation: We consider a distributed private learning setup comprising a set of n clients, where the ith client has a data point d_i drawn from a universe $\mathcal{X}$ for $i \in [n]$; see also Figure 1. Let $\mathcal{D} = (d_1, \ldots, d_n)$ denote the entire training dataset. The clients are connected to an untrusted server in order to solve the following empirical risk minimization (ERM) problem

$$\min_{\theta \in \mathcal{C}} \Big(F(\theta, \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} f(\theta, d_i) \Big), \tag{2}$$

where $\mathcal{C} \subset \mathbb{R}^d$ is a closed convex set, and $f : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is the loss function. Our goal is to construct a global learning model θ via stochastic gradient descent (SGD) while preserving privacy of individual data points in the training dataset $\mathcal{D}$ by providing strong DP guarantees.

We revisit the CLDP-SGD algorithm presented in [27] and described in Algorithm 1 to solve the ERM (2). In each step 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 and clips the ℓ_p-norm of the gradient $\nabla_{\theta_t} f(\theta_t, d_i)$ to apply the LDP mechanism $\mathcal{R}_p$, where $\mathcal{R}_p : \mathcal{B}_p^d \to \{0,1\}^b$ is an ϵ_0-LDP mechanism when inputs come from an ℓ_p-norm ball. In [27], the authors proposed different ϵ_0-LDP mechanisms for general ℓ_p-norm balls. After that, the shuffler randomly permutes the received k gradients $\{\mathcal{R}_p(\tilde{\mathbf{g}}_t(d_i))\}_{i \in \mathcal{U}_t}$ and sends them to the server. Finally, the server takes the average of the received gradients and updates the parameter vector. Our main contribution in this work is to present a stronger privacy analysis of the CLDP-SGD algorithm by characterizing the RDP of the sub-sampled shuffle model.

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

Input: Datasets $\mathcal{D} = (d_1, \ldots, d_n)$, LDP privacy parameter ϵ_0, gradient norm bound C, and learning rate schedule $\{\eta_t\}$.

Initialize: $\theta_0 \in \mathcal{C}$
for $t \in [T]$ **do**
  Client sampling: A random set $\mathcal{U}_t$ of k clients is chosen.
  for clients $i \in \mathcal{U}_t$ **do**
    Compute gradient: $\mathbf{g}_t(d_i) \leftarrow \nabla_{\theta_t} f(\theta_t, d_i)$
    Clip gradient: $\tilde{\mathbf{g}}_t(d_i) \leftarrow \mathbf{g}_t(d_i) \,/\, \max\left\{1, \frac{\|\mathbf{g}_t(d_i)\|_p}{C}\right\}$
    Client i sends $\mathcal{R}_p(\tilde{\mathbf{g}}_t(d_i))$ to the shuffler.
  end for
  Shuffling: The shuffler sends random permutation of $\{\mathcal{R}_p(\tilde{\mathbf{g}}_t(d_i)) : i \in \mathcal{U}_t\}$ to the server.
  Aggregate: $\overline{\mathbf{g}}_t \leftarrow \frac{1}{k} \sum_{i \in \mathcal{U}_t} \mathcal{R}_p(\tilde{\mathbf{g}}_t(d_i))$
  Descent Step: $\theta_{t+1} \leftarrow \prod_{\mathcal{C}} (\theta_t - \eta_t \overline{\mathbf{g}}_t)$, where $\prod_{\mathcal{C}}$ is the projection operator onto the set $\mathcal{C}$.
end for

Output: The model θ_T and the privacy parameters ϵ, δ.

3 Main Results

In this section, we present our main results. First, we characterize the RDP of the subsampled shuffle mechanism by presenting an upper bound in Theorem 1 and a lower bound in Theorem 2. We then present the privacy-convergence trade-offs of the CLDP-SGD Algorithm in Theorem 3.

Consider an arbitrary ϵ_0-LDP mechanism $\mathcal{R}$, whose range is a discrete set $[B] = \{1, \ldots, B\}$ for some $B \in \mathbb{N} := \{1, 2, 3, \ldots\}$. Here, $[B]$ could be the whole of $\mathbb{N}$. Let $\mathcal{M}(\mathcal{D})$ be a subsampled shuffle mechanism 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,[5] who randomly permutes the received k inputs and outputs the result. To formalize this, let $\mathcal{H}_k : \mathcal{Y}^k \to \mathcal{Y}^k$ denote the shuffling operation that takes k inputs and outputs their uniformly random permutation. Let $\mathrm{samp}_k^n : \mathcal{X}^n \to \mathcal{X}^k$ denote the sampling operation for choosing a random subset of k elements from a set of n elements. We define the subsampled-shuffle mechanism as

$$\mathcal{M}(\mathcal{D}) := \mathcal{H}_k \circ \mathrm{samp}_k^n \left(\mathcal{R}(d_1), \ldots, \mathcal{R}(d_n)\right). \tag{3}$$

Observe that each iteration of Algorithm 1 can be represented as an output of the subsampled shuffle mechanism $\mathcal{M}$. Thus, to analyze the privacy of Algorithm 1, it is sufficient to analyze the privacy of a sequence of identical T subsampled shuffle mechanisms, and then apply composition theorems.

Histogram notation. It will be useful to define the following notation. Since the output of $\mathcal{H}_k$ is a random permutation of the k outputs of $\mathcal{R}$ (subsampling is not important here), the server cannot associate the k 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 $\mathcal{A}_B^k$ as

$$\mathcal{A}_B^k = \left\{ \boldsymbol{h} = (h_1, \ldots, h_B) : \sum_{j=1}^{B} h_j = k \right\}, \tag{4}$$

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

Our main results for the RDP of the subsampled shuffled mechanism (defined in (3)) are given below. Our first result provides an upper bound (stated in Theorem 1 and proved in Section 5) and the second result provides a lower bound (stated in Theorem 2 and proved in Appendix D).

Theorem 1 (Upper Bound)**.** *For any $n \in \mathbb{N}$, $k \leq n$, $\epsilon_0 \geq 0$, and any integer $\lambda \geq 2$, the RDP of the subsampled shuffle mechanism $\mathcal{M}$ (defined in* (3)*) is upper-bounded by*

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

where $\overline{k} = \lfloor \frac{k-1}{2e^{\epsilon_0}} \rfloor + 1$, $\gamma = \frac{k}{n}$, and $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ is the Gamma function. The term Υ is given by $\Upsilon = \left(\left(1 + \gamma \frac{e^{2\epsilon_0} - 1}{e^{\epsilon_0}}\right)^\lambda - 1 - \lambda\gamma \frac{e^{2\epsilon_0} - 1}{e^{\epsilon_0}} \right) e^{-\frac{k-1}{8e^{\epsilon_0}}}$.

Theorem 2 (Lower Bound)**.** *For any $n \in \mathbb{N}$, $k \leq n$, $\epsilon_0 \geq 0$, and any integer $\lambda \geq 2$, the RDP of the subsampled shuffle mechanism $\mathcal{M}$ (defined in* (3)*) is lower-bounded by*

$$\epsilon(\lambda) \geq \frac{1}{\lambda - 1} \log \left(1 + \binom{\lambda}{2}\gamma^2 \frac{\left(e^{\epsilon_0} - 1\right)^2}{k e^{\epsilon_0}} + \sum_{j=3}^{\lambda} \binom{\lambda}{j} \gamma^j \left(\frac{\left(e^{2\epsilon_0} - 1\right)}{k e^{\epsilon_0}} \right)^j \mathbb{E}\left(m - \frac{k}{e^{\epsilon_0} + 1} \right)^j \right),$$

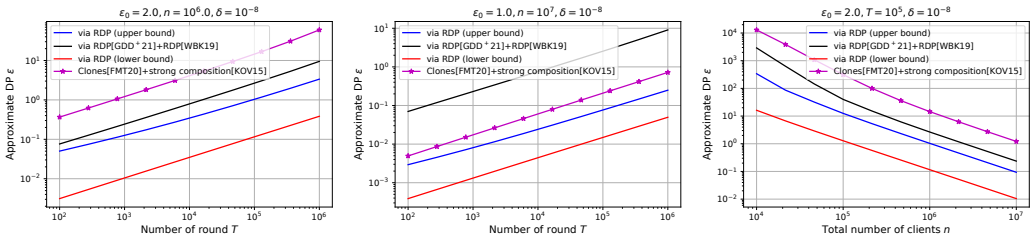
where expectation is taken w.r.t. the binomial r.v. $m \sim$ Bin (k, p) with parameter $p = \frac{1}{e^{\epsilon_0}+1}$.

Our CLDP-SGD algorithm and its privacy-convergence trade-offs (stated in Theorem 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., $\mathbb{E}\left[\mathcal{R}_p(\mathbf{x}) | \mathbf{x}\right] = \mathbf{x}$ for all $\mathbf{x} \in \mathcal{B}_p(a)$, where a is the radius of the ball $\mathcal{B}_p$. (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 variance: $\sup_{\mathbf{x} \in \mathcal{B}_p(a)} \mathbb{E}\|\mathcal{R}_p(\mathbf{x}) - \mathbf{x}\|_2^2 \leq G_p^2(a)$, where G_p^2 is a function from $\mathbb{R}^+$ to $\mathbb{R}^+$.

Girgis et al. [27] 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 and satisfy the above conditions. In this paper, achieving communication efficiency is not our goal (though we also achieve that since the ϵ_0-LDP mechanism $\mathcal{R}_p$ that we use takes values in a discrete set), as our main focus is on analyzing the RDP of the subsampled shuffle mechanism. If we use the ϵ_0-LDP mechanism $\mathcal{R}_p$ from [27], we would also get similar gains in communication as were obtained in [27].

The privacy-convergence trade-off of our algorithm $\mathcal{A}_{\text{cldp}}$ is given below.

[5] With a slight abuse of notation, in this paper we write $\mathcal{R}(d_i)$ to denote that $\mathcal{R}$ takes as its input the gradient computed on d_i using the current parameter vector.



(a) Approx. DP as a function of T for $\epsilon_0 = 2, \gamma = 0.001, n = 10^6$ (b) Approx. DP as a function of T for $\epsilon_0 = 1, \gamma = 0.001, n = 10^7$ (c) Approx. DP as a function of n for $\epsilon_0 = 2, \gamma n = 10^3, T = 10^5$

Figure 2: Comparison of several bounds on the Approximate (ϵ, δ)-DP for composition of a sequence of subsampled shuffle mechanisms for $\delta = 10^{-8}$: (i) Approximate DP obtained from our upper bound on the RDP in Theorem 1 (blue); (ii) Approximate DP obtained from our lower bound on the RDP in Theorem 2 (red); (iii) Approximate DP obtained from the upper bound on the RDP given in [29] with RDP amplification by subsampling from [43] (black); and (iv) Applying the strong composition theorem [34] after getting the Approximate DP of the shuffled model given in [24] with subsampling [42] (magenta).

Theorem 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* (2)*. 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*
$$\epsilon = \min_{\lambda} \left(T\epsilon(\lambda) + \frac{\log(1/\delta) + (\lambda - 1)\log(1 - 1/\lambda) - \log(\lambda)}{\lambda - 1} \right), \tag{5}$$
where $\epsilon(\lambda)$ is the RDP of the subsampled shuffle mechanism given in Theorem 1.

2. ***Convergence:*** *If we run $\mathcal{A}_{\text{cldp}}$ with $\eta_t = \frac{D}{G\sqrt{t}}$, where $G^2 = \max\{d^{1-\frac{2}{p}}, 1\}L^2 + \frac{G_p^2(L)}{\gamma n}$, we get*
$$\mathbb{E}\left[F(\theta_T)\right] - F(\theta^*) \leq \mathcal{O}\left(\frac{DG\log(T)}{\sqrt{T}}\right).$$

The proof outline of Theorem 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 (3), for which we have computed the RDP guarantees in Theorem 1. Now, for the privacy analysis of $\mathcal{A}_{\text{cldp}}$, we use the adaptive composition theorem from [37, Proposition 1] and then use the RDP to DP conversion given in Lemma 1. For the convergence analysis, we use a standard non-private SGD convergence result and compute the required parameters for that. See Appendix F for a complete proof of Theorem 3.

Remark 1. Note that our convergence bound is affected by the variance 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 $\mathcal{R}_2$ proposed in [11] that has variance $G_2^2(L) = 14L^2 d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$; and when f is L-Lipschitz continuous w.r.t. the ℓ_1-norm or ℓ_∞-norm, we can use the LDP mechanisms $\mathcal{R}_\infty$ or $\mathcal{R}_1$, respectively, proposed in [27] that have variances $G_\infty^2(L) = L^2 d^2\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$ and $G_1^2(L) = L^2 d\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2$, respectively. By plugging these variances $G_p^2(L)$ (for $p = 1, 2, \infty$) into Theorem 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 2. The privacy parameter in (5) 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 4 for more details.

4 Numerical Results

In this section, we present numerical experiments to show the performance of our bounds on RDP of the subsampled shuffle mechanism and its usage for getting approximate DP of Algorithm 1 for training machine learning models.

Composition of a sequence of subsampled shuffle models: In Figure 2, 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 shuffle mechanism for $t \in [T]$. In all our experiments reported in Figure 2, we fix $\delta = 10^{-8}$. We observe that our new bound on the RDP of the subsampled shuffle mechanism achieves a significant saving in total privacy ϵ compared to the state-of-the-art. For example, we save a factor of $14\times$ compared to the bound on DP [24] with strong composition theorem [34] and $2.5\times$ compared to the bound on the RDP given in [29] with subsampled RDP [43] in computing the overall privacy parameter ϵ for number of iterations $T = 10^5$, subsampling parameter $\gamma = 0.001$, LDP parameter $\epsilon_0 = 2$, and number of clients $n = 10^6$. We observe in Figure 2b that the bound given in [24] with the strong composition theorem [34] behaves better than the bound on the RDP [29] with subsampled RDP bound [43] when the number of subsampled clients per iteration is equal to $k = \gamma n = 10^4$; however, our bound beats both of them.[6] In Figure 2c, 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 [21, 39] and described in Table 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 Algorithm 1, 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 [27] with $\epsilon_0 = 1.5$. We run Algorithm 1 with $\delta = 10^{-5}$ for 200 epochs, with learning rate $\eta = 0.3$ for the first 70 epochs, and then decrease it to 0.18 in the remaining epochs.

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

Table 1: Model architecture for MNIST

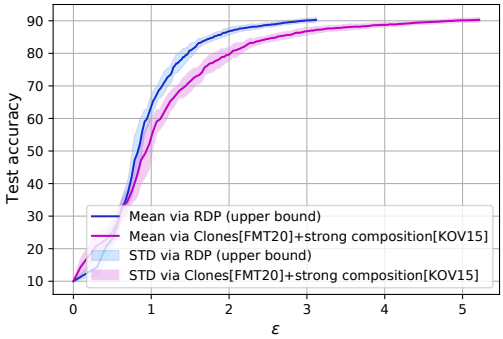


Figure 3: Privacy-Utility trade-offs on the MNIST dataset with ℓ_∞-norm clipping.

Figure 3 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 λ using our upper bound given in Theorem 1. For privacy analysis of [24], we first compute the privacy amplification by shuffling numerically given in [24]; then we compute its privacy obtained when amplified via subsampling [42]; and finally we use the strong composition theorem [34] to obtain the central privacy parameter ϵ.

We observe that we achieve an accuracy of $80\%(\pm 1.8)$ with a total privacy budget of $\epsilon = 1.4$ using our new privacy analysis, whereas, [24] achieves an accuracy of only $70.7\%(\pm 2.1)$ with the same privacy budget of $\epsilon = 1.4$ using the standard composition theorems. Furthermore, we can see that we achieves accuracy $90\%(\pm 0.5)$ with total privacy budget $\epsilon = 2.91$ using our new privacy analysis, whereas, [24] (together with the standard strong composition theorem) achieves the same accuracy with a total privacy budget of $\epsilon = 4.82$.

5 Proof of Theorem 1: Upper Bound

For any dataset $\mathcal{D}_k = (d_1, \ldots, d_k) \in \mathcal{X}^k$ containing of k data points, we define a shuffle mechanism $\mathcal{M}_{sh}(\mathcal{D}_k)$ as follows:

$$\mathcal{M}_{sh}(\mathcal{D}_k) = \mathcal{H}_k\left(\mathcal{R}\left(d_1\right), \ldots, \mathcal{R}\left(d_k\right)\right), \tag{6}$$

[6] In fact, there are several parameter regimes of great practical interest for which the results of [24] are not even valid; see Appendix G for more details on this, and also for more numerical comparisons.

where $\mathcal{H}_k$ takes k inputs and outputs a uniformly random permutation of them. Recall from (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 $\mathcal{M}(\mathcal{D}) := \mathcal{H}_k \circ \mathrm{samp}_k^n (\mathcal{R}(d_1), \ldots, \mathcal{R}(d_n))$.

The proof of Theorem 1 consists of two steps. First, we bound the ternary-$|\chi|^\alpha$-DP of the shuffle mechanism $\mathcal{M}_{sh}$ (see Theorem 4), which is the main technical contribution in this proof. Then, using this, we bound the RDP of the subsampled shuffle mechanism $\mathcal{M}$.

Theorem 4 (ζ-ternary-$|\chi|^\alpha$-DP of the shuffle mechanism $\mathcal{M}_{sh}$)**.** *For any integer $k \geq 2$, $\epsilon_0 > 0$, and all $\alpha \geq 2$, the ζ-ternary-$|\chi|^\alpha$-DP of the shuffle mechanism $\mathcal{M}_{sh}$ is bounded by:*

$$\zeta(\alpha)^\alpha \leq \begin{cases} 4\frac{(e^{\epsilon_0}-1)^2}{\overline{k}e^{\epsilon_0}} + (e^{\epsilon_0} - e^{-\epsilon_0})^\alpha e^{-\frac{k-1}{8e^{\epsilon_0}}} & \text{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{\overline{k}e^{2\epsilon_0}}\right)^{\alpha/2} + (e^{\epsilon_0} - e^{-\epsilon_0})^\alpha e^{-\frac{k-1}{8e^{\epsilon_0}}} & \text{otherwise}, \end{cases} \tag{7}$$

where $\overline{k} = \lfloor \frac{k-1}{2e^{\epsilon_0}} \rfloor + 1$ and $\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx$ is the Gamma function.

Theorem 4 is one of the core technical results of this paper, and we prove it in Section 6.

It was shown in [43, Proposition 16] that if a mechanism obeys ζ-ternary-$|\chi|^\alpha$-DP, then its subsampled version (with subsampling parameter γ) will obey $\gamma\zeta$-ternary-$|\chi|^\alpha$-DP. Using that result, the authors then bounded the RDP of the subsampled mechanism in [43, Eq. (9)]. Adapting that result to our setting, we have the following lemma.

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

Lemma 2 can be seen as a corollary to [43, Proposition 16 and Eq. (9)]. However, for completeness, we prove it in Appendix E.1. Substituting the bound on $\zeta(\alpha)$ from Theorem 4 into Lemma 2 together with some algebraic manipulation gives proves Theorem 1; see Appendix E.2 for details.

6 Proof of Theorem 4: Ternary $|\chi|^\alpha$-DP of the Shuffle Model

The proof has two main steps. 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.

The specific neighboring datasets to which we reduce our general problem has the following form:

$$\begin{aligned}\mathcal{D}_{\text{same}}^m = \{(\mathcal{D}_m, \mathcal{D}'_m, \mathcal{D}''_m) : \mathcal{D}_m = (d, \ldots, d, d) \in \mathcal{X}^m,\ \mathcal{D}'_m = (d, \ldots, d, d') \in \mathcal{X}^m, \text{ and} \\ \mathcal{D}''_m = (d, \ldots, d, d'') \in \mathcal{X}^m, \text{ where } d, d', d'' \in \mathcal{X}\},\end{aligned} \tag{8}$$

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 $\mathcal{D}_{m+1}^{(k)} = (d''_k, \ldots, d''_k, d_k)$, $\mathcal{D}_{m+1}^{\prime(k)} = (d''_k, \ldots, d''_k, d'_k)$, and $\mathcal{D}_{m+1}^{\prime\prime(k)} = (d''_k, \ldots, d''_k)$, each having $m+1$ elements. Observe that $(\mathcal{D}_{m+1}^{\prime\prime(k)}, \mathcal{D}_{m+1}^{\prime(k)}, \mathcal{D}_{m+1}^{(k)}) \in \mathcal{D}_{\text{same}}^m$.

The first step of the proof is given in the following theorem.

Theorem 5 (Reduction to the Special Case)**.** *Let $q = \frac{1}{e^{\epsilon_0}}$. We have:*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}'')}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}')(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}'')(\boldsymbol{h})}\right|^\alpha\right] \\ &\quad\leq \mathbb{E}_{m\sim\text{Bin}(k-1,q)}\left[\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_{m+1}^{\prime\prime(k)})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}_{m+1}^{(k)})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}_{m+1}^{\prime(k)})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_{m+1}^{\prime\prime(k)})(\boldsymbol{h})}\right|^\alpha\right]\right].\end{aligned} \tag{9}$$

We know (by Chernoff bound) that the binomial r.v. is concentrated around its mean, which implies that the terms in the RHS of (9) 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 (9) 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 6 ($|\chi|^{\alpha}$-DP for special case)**.** *For any $m \in \mathbb{N}$, integer $\alpha \geq 2$, and $(\mathcal{D}''_m, \mathcal{D}'_m, \mathcal{D}_m) \in \mathcal{D}^m_{same}$,*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})}\right|^{\alpha}\right] \leq \begin{cases} 4\frac{(e^{\epsilon_0}-1)^2}{me^{\epsilon_0}} & \text{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{2(e^{2\epsilon_0}-1)^2}{me^{2\epsilon_0}}\right)^{\alpha/2} & \text{otherwise.} \end{cases}$$

Missing details of how Theorem 4 follows from Theorems 5, 6 can be found in Appendix C.4.

Proof sketch of Theorem 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 $\boldsymbol{p}_i = \frac{1}{e^{\epsilon_0}}\boldsymbol{p}''_k + \left(1 - \frac{1}{e^{\epsilon_0}}\right)\tilde{\boldsymbol{p}}_i$, 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.) $1/e^{\epsilon_0}$ and according to $\tilde{\boldsymbol{p}}_i$ w.p. $(1 - 1/e^{\epsilon_0})$. As a result, the number of clients that sample from the distribution $\boldsymbol{p}''_k$ follows a binomial distribution $\text{Bin}(k-1, 1/e^{\epsilon_0})$. 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 $\tilde{\boldsymbol{p}}_1, \ldots, \tilde{\boldsymbol{p}}_{k-1}, \boldsymbol{p}_k, \boldsymbol{p}'_k, \boldsymbol{p}''_k$; see Lemma 4. Then using a joint convexity argument (see Lemma 3), 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 $\mathcal{M}_{sh}$ as in Lemma 4. Using a monotonicity argument (see Lemma 5), 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 6), which proves Theorem 5. See Appendix C.1 for a complete proof. ■

Proof sketch of Theorem 6. Consider $(\mathcal{D}''_m, \mathcal{D}'_m, \mathcal{D}_m) \in \mathcal{D}^m_{\text{same}}$ as in the statement of Theorem 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 [43], 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 [29], the authors 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. See Appendix C.3 for a complete proof. ■

7 Discussion

In this paper, we analyzed the Rényi differential privacy of the subsampled shuffle model by bounding the ternary $|\chi|^{\alpha}$-DP of the shuffle model. We numerically demonstrated the importance of our proposed bound, where we obtain a significant improvement over using the state-of-the-art in practical regimes. Furthermore, we used our privacy analysis to study the privacy-accuracy trade-offs on the MNIST dataset, where we obtained 90% accuracy with total privacy budget of $\epsilon = 2.91$, which is an improvement over an analysis yielding 4.82, using standard strong composition theorem.

Closing the gap (shown numerically) between our lower bound in Theorem 2 and the achievable upper bound in Theorem 1 is an important unresolved question. Another direction to explore would be to analyze the RDP of the subsampled shuffle model for different sub-sampling techniques such as Poisson subsampling [45], random check-in [9], or client self-sampling [30].

Societal Impact. Collaborative learning comes with significant societal risks of privacy violations, which is the main topic addressed in this paper. However, such learning is only as good as the data used for training, and if the data is not unbiased, this could lead to significant issues related to fairness and could also lead to societally undesirable outcomes. Such an issue is exacerbated when privacy is guaranteed on the data used for training, making a-priori fairness checks on data infeasible. This can be ameliorated by properly testing models finally obtained against fairness criteria and rejecting models that fail the test. This paper did not consider the issue of robustness to security, and this could also be an important societal issue in collaborative learning, where a small subset of users could insert malicious inputs to disrupt the learning process or worse bias the learned model covertly. This could also lead to negative outcomes. This issue of robustness to malicious participants has been studied in several papers, and incorporating this into the framework of the paper is an important future research topic.

Acknowledgment

This work was supported in part by NSF grants #2007714 and #1955632 and a Google Faculty research award and an Amazon Research Award.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

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

[3] S. Asoodeh, J. Liao, F. P. Calmon, O. Kosut, and L. Sankar. Three variants of differential privacy: Lossless conversion and applications. *IEEE Journal on Selected Areas in Information Theory*, 2(1):208–222, 2021.

[4] B. Balle, G. Barthe, M. Gaboardi, J. Hsu, and T. Sato. Hypothesis testing interpretations and renyi differential privacy. In S. Chiappa and R. Calandra, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pages 2496–2506. PMLR, 2020.

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

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

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

[8] B. Balle, J. Bell, A. Gascón, and K. Nissim. Private summation in the multi-message shuffle model. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 657–676, 2020.

[9] B. Balle, P. Kairouz, B. McMahan, O. D. Thakkar, and A. Thakurta. Privacy amplification via random check-ins. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. 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.

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

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

[12] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[13] C. L. Canonne, G. Kamath, and T. Steinke. The discrete gaussian for differential privacy. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. 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.

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

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

[16] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3574–3583, Red Hook, NY, USA, 2017. Curran Associates Inc.

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

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

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

[20] C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[21] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, S. Song, K. Talwar, and A. Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618, 2020.

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

[23] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[24] V. Feldman, A. McMillan, and K. 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*. IEEE, 2021.

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

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

[27] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. Theertha Suresh. Shuffled model of differential privacy in federated learning. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2521–2529. PMLR, 13–15 Apr 2021.

[28] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. 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.

[29] A. M. Girgis, D. Data, S. Diggavi, A. T. Suresh, and P. Kairouz. On the renyi differential privacy of the shuffle model. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*. ACM, 2021.

[30] A. M. Girgis, D. Data, and S. N. Diggavi. Differentially private federated learning with shuffling and client self-sampling. In *IEEE International Symposium on Information Theory, ISIT 2021, Melbourne, Australia, July 12-20, 2021*, pages 338–343. IEEE, 2021.

[31] A. Greenberg. Apple's 'differential privacy'is about collecting your data—but not your data. *Wired, June*, 13, 2016.

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

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

[34] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

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

[36] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[37] I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[38] I. Mironov, K. Talwar, and L. Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

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

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

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

[42] J. Ullman. Cs7880. rigorous approaches to data privacy. 2017.

[43] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[44] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[45] Y. Zhu and Y.-X. Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642. PMLR, 2019.

Supplementary Material

A Literature Review

We give the most relevant work related to the paper and review some of the main developments in differentially private learning below.

Private Optimization: In [14], Chaudhuri et al. studied *centralized* privacy-preserving machine learning algorithms for convex optimization problem. In [10], 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 [1], the authors have generalized the private SGD algorithm proposed in [10] for non-convex optimization framework. In addition, the authors have proposed a new analysis technique, called moment accounting, to improve on the strong composition theorems to compute the central differential privacy guarantee for iterative algorithms. However, the works mentioned, [14, 10, 1], assume that there exists a trusted server that collects the clients' data. This motivates other works to design a distributed SGD algorithms, where each client perturbs her own data without needing a trusted server.

Distributed learning under local differential privacy (LDP) has studied in [2, 21, 27]. In [2] the authors proposed a communication-efficient algorithm for learning models under local differential privacy. In [21], the authors have proposed a distributed local-differential-privacy gradient descent algorithm, a newly proposed anonymization/shuffling framework [7] is used to amplify the privacy. In [27], the authors proposed communication efficient algorithms for general ℓ_p-norm stetting under local differential privacy constraints, where they use recent results on amplification by shuffling to boost the privacy-utility trade-offs of the distributed learning algorithms.

Shuffled privacy model: The shuffled model of privacy has been of significant recent interest [22, 25, 6, 26, 5, 15, 7, 8]. However, most of the existing works in literature [22, 7, 24] only characterize the approximate DP of the shuffled model. Recently, the authors in [29] proposed a novel bound on the RDP of the shuffled model, where they show that the RDP provides a significant saving in computing the total privacy budget for a composition of a sequence of shuffled mechanisms. However, the work [29] does not characterize the RDP of the subsampled shuffle mechanism. We can compute a bound on the RDP of the subsampled shuffle mechanism by combining the bound of the RDP of the shuffle mechanism in [29] with the bound of the subsampled RDP mechanism in [43]. However, we show numerically that our new bound on the subsampled shuffle mechanism outperforms this bound.

Rényi differential privacy: The work of Abadi *et al.* [1] provided a new analysis technique to improve on the strong composition theorems. Inherently, this used Rényi divergence, and was later formalized in [37] which defined Rényi differential privacy (RDP). Several works [38, 43, 45] have shown that analyzing the RDP of subsampled mechanisms provides a tighter bound on the total privacy loss than the bound that can be obtained using the standard strong composition theorems. In this paper, we analyze the RDP of the subsampled shuffle model, where we can bound the approximate DP of a sequence of subsampled shuffle models using the transformation from RDP to approximate DP [1, 43, 13, 3]. We show that our RDP analysis provides a better bound on the total privacy loss of composition than the bound that can be obtained using the standard strong composition theorems and the bound that can be obtained by combining the RDP bound of the shuffle model in [29] with the subsampled RDP mechanism in [43].

B Additional Preliminaries

Here, we give local and central differential privacy definitions that we use throughout this work.

Definition 3 (Local Differential Privacy - LDP [35])**.** For $\epsilon_0 \geq 0$, 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

$$\Pr[\mathcal{R}(d) \in \mathcal{S}] \leq e^{\epsilon_0} \Pr[\mathcal{R}(d') \in \mathcal{S}], \qquad \forall \mathcal{S} \subset \mathcal{Y}. \tag{10}$$

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 $d_j = d'_j$.

Definition 4 (Central Differential Privacy - DP [18, 19])**.** For $\epsilon, \delta \geq 0$, a randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to be (ϵ, δ)-differentially private (in short, (ϵ, δ)-DP), if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ and every subset $\mathcal{S} \subseteq \mathcal{Y}$, we have

$$\Pr\left[\mathcal{M}(\mathcal{D}) \in \mathcal{S}\right] \leq e^{\epsilon_0} \Pr\left[\mathcal{M}(\mathcal{D}') \in \mathcal{S}\right] + \delta. \tag{11}$$

C Omitted Details from Section 6

C.1 Proof of Theorem 5: Reduction to the Special Case

First, we prove the joint-convexity of the ternary $|\chi|^\alpha$-divergence as it is important in the following proof.

Lemma 3 (Joint-convexity of the ternary $|\chi|^\alpha$-divergence)**.** *For all $\alpha \geq 1$, the ternary-$|\chi|^\alpha$-divergence $\mathbb{E}\left[\left|\frac{P-Q}{R}\right|^\alpha\right]$ is jointly convex in P, Q and R. In other words, if $P_a = aP_0 + (1-a)P_1$, $Q_a = aQ_0 + (1-a)Q_1$, and $R_a = aR_0 + (1-a)R_1$ for some $a \in [0,1]$, then the following holds*

$$\mathbb{E}_{R_a}\left[\left|\frac{P_a - Q_a}{R_a}\right|^\alpha\right] \leq a\mathbb{E}_{R_0}\left[\left|\frac{P_0 - Q_0}{R_0}\right|^\alpha\right] + (1-a)\mathbb{E}_{R_1}\left[\left|\frac{P_1 - Q_1}{R_1}\right|^\alpha\right] \tag{12}$$

Proof. First, observe that $g(x, y) = |x - y|$ is jointly convex on $\mathbb{R}^2$, i.e., if $x_a = ax_0 + (1-a)x_1$ and $y_a = ay_0 + (1-a)y_1$, we have

$$\begin{aligned} |x_a - y_a| &= |a(x_0 - y_0) + (1-a)(x_1 - y_1)| \\ &\leq a|x_0 - y_0| + (1-a)|x_1 - y_1| \end{aligned} \tag{13}$$

Let $f(x, y) = x^j / y^{j-1}$, which is jointly convex on $\mathbb{R}_+^2$ for $j \geq 1$; see [43, Lemma 20] for a proof. Thus, we get

$$\frac{|P_a - Q_a|^j}{R_a^{j-1}} \leq \frac{(a|P_0 - Q_0| + (1-a)|P_1 - Q_1|)^j}{(aR_0 + (1-a)R_1)^{j-1}} \quad \leq a\frac{|P_0 - Q_0|^j}{R_0^{j-1}} + (1-a)\frac{|P_1 - Q_1|^j}{R_1^{j-1}}, \tag{14}$$

where the first inequality is obtained from (13) and the second inequality is obtained from the convexity of $f(x, y)$. ■

Now, we prove Theorem 5. Our proof is an adaptation of the proof of [29, Theorem 4]. The difference comes from the fact that [29, Theorem 4] was for Renyi divergence, whereas, here we are working with ternary $|\chi|^\alpha$-divergence. This changes some details and we provide a full proof of Theorem 5 below.

Let $\boldsymbol{p}_i := (p_{i1}, \ldots, p_{iB})$, $\boldsymbol{p}'_k := (p'_{k1}, \ldots, p'_{kB})$, $\boldsymbol{p}''_k := (p''_{k1}, \ldots, p''_{kB})$ denote the probability distributions over $\mathcal{Y}$ when the input to $\mathcal{R}$ is d_i, d'_k, and d''_k respectively, where $p_{ij} = \Pr[\mathcal{R}(d_i) = j]$ for all $j \in [B]$ and $i \in [n]$. Let $\mathcal{P} = \{\boldsymbol{p}_i : i \in [k]\}$, $\mathcal{P}' = \{\boldsymbol{p}_i : i \in [k-1]\} \bigcup \{\boldsymbol{p}'_k\}$, and $\mathcal{P}'' = \{\boldsymbol{p}_i : i \in [k-1]\} \bigcup \{\boldsymbol{p}''_k\}$.

For $i \in [k-1]$, let $\mathcal{P}_{-i} = \mathcal{P} \setminus \{\boldsymbol{p}_i\}$ and also $\mathcal{P}_{-k} = \mathcal{P} \setminus \{\boldsymbol{p}_k\}$. Here, $\mathcal{P}, \mathcal{P}', \mathcal{P}''$ correspond to the datasets $\mathcal{D} = \{d_1, \ldots, d_k\}, \mathcal{D}' = \{d_1, \ldots, d_{k-1}, d'_k\}$, and $\mathcal{D}'' = \{d_1, \ldots, d_{k-1}, d''_k\}$ respectively, and for any $i \in [k]$, $\mathcal{P}_{-i}$ corresponds to the dataset $\mathcal{D}_{-i} = \{d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_k\}$.

For any collection $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\}$ of k distributions, we define $F(\mathcal{P})$ to be the distribution over $\mathcal{A}_B^k$ (which is the set of histograms on B bins with k elements as defined in (4)) 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 $\boldsymbol{h} \in \mathcal{A}_B^k$, define

$$\mathcal{U}_{\boldsymbol{h}} := \left\{ (\mathcal{U}_1, \ldots, \mathcal{U}_B) : \mathcal{U}_1, \ldots, \mathcal{U}_B \subseteq [k] \text{ s.t. } \bigcup_{j=1}^{B} \mathcal{U}_j = [k] \text{ and } |\mathcal{U}_j| = h_j, \forall j \in [B] \right\}. \tag{15}$$

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 $[B]$ – here $\mathcal{U}_j$'s are disjoint for all $j \in [B]$. Note also that $|\mathcal{U}_{\boldsymbol{h}}| = \binom{k}{\boldsymbol{h}} = \frac{k!}{h_1! h_2! \ldots h_B!}$. It is easy to verify that for any $\boldsymbol{h} \in \mathcal{A}_B^k$, $F(\mathcal{P})(\boldsymbol{h})$ is equal to

$$F(\mathcal{P})(\boldsymbol{h}) = \sum_{(\mathcal{U}_1, \ldots, \mathcal{U}_B) \in \mathcal{U}_{\boldsymbol{h}}} \prod_{j=1}^{B} \prod_{i \in \mathcal{U}_j} p_{ij} \tag{16}$$

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

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

$$\boldsymbol{p}_i = q\boldsymbol{p}''_k + (1-q)\,\tilde{\boldsymbol{p}}_i, \tag{17}$$

where $q = \frac{1}{e^{\epsilon_0}}$. The distribution $\tilde{\boldsymbol{p}}_i = [\tilde{p}_{i1}, \ldots, \tilde{p}_{iB}]$ is given by $\tilde{p}_{ij} = \frac{p_{ij} - qp''_{kj}}{1-q}$, where it is easy to verify that $\tilde{p}_{ij} \geq 0$ and $\sum_{j=1}^{B} \tilde{p}_{ij} = 1$.

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

$$\mathcal{P}_{\mathcal{C}} = \{\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_{k-1}\} \bigcup \{\boldsymbol{p}_k\}, \tag{18}$$

$$\mathcal{P}'_{\mathcal{C}} = \{\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_{k-1}\} \bigcup \{\boldsymbol{p}'_k\}, \tag{19}$$

$$\mathcal{P}''_{\mathcal{C}} = \{\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_{k-1}\} \bigcup \{\boldsymbol{p}''_k\}, \tag{20}$$

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

$$\hat{\boldsymbol{p}}_i = \begin{cases} \boldsymbol{p}''_k & \text{if } i \in \mathcal{C}, \\ \tilde{\boldsymbol{p}}_i & \text{if } i \in [k-1] \setminus \mathcal{C}. \end{cases} \tag{21}$$

In the following lemma, we show that $F(\mathcal{P}), F(\mathcal{P}')$, and $F(\mathcal{P}'')$ can be written as convex combinations of $\{F(\mathcal{P}_{\mathcal{C}}) : \mathcal{C} \subseteq [k-1]\}, \{F(\mathcal{P}'_{\mathcal{C}}) : \mathcal{C} \subseteq [n-1]\}$, and $\{F(\mathcal{P}''_{\mathcal{C}}) : \mathcal{C} \subseteq [k-1]\}$, respectively, where for any $\mathcal{C} \subseteq [k-1]$, $F(\mathcal{P}_{\mathcal{C}}), F(\mathcal{P}'_{\mathcal{C}})$. and $F(\mathcal{P}''_{\mathcal{C}})$ can be computed analogously as in (16).

Lemma 4 (Mixture Interpretation)**.** *$F(\mathcal{P}), F(\mathcal{P}')$, and $F(\mathcal{P}'')$ can be written as the following convex combinations:*

$$F(\mathcal{P}) = \sum_{\mathcal{C} \subseteq [k-1]} q^{|\mathcal{C}|}(1-q)^{k-|\mathcal{C}|-1} F(\mathcal{P}_{\mathcal{C}}), \tag{22}$$

$$F(\mathcal{P}') = \sum_{\mathcal{C} \subseteq [k-1]} q^{|\mathcal{C}|}(1-q)^{k-|\mathcal{C}|-1} F(\mathcal{P}'_{\mathcal{C}}), \tag{23}$$

$$F(\mathcal{P}'') = \sum_{\mathcal{C} \subseteq [k-1]} q^{|\mathcal{C}|}(1-q)^{k-|\mathcal{C}|-1} F(\mathcal{P}''_{\mathcal{C}}), \tag{24}$$

where $\mathcal{P}_{\mathcal{C}}, \mathcal{P}'_{\mathcal{C}}, \mathcal{P}''_{\mathcal{C}}$ are defined in (18)-(21).

We present a proof of Lemma 4 in Appendix C.2.

From Lemma 3 and Lemma 4, we get

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

$$\leq \sum_{\mathcal{C}\subseteq[k-1]} q^{|\mathcal{C}|}\,(1-q)^{k-|\mathcal{C}|-1}\,\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}''_{\mathcal{C}})}\left[\left|\frac{F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h})-F(\mathcal{P}'_{\mathcal{C}})(\boldsymbol{h})}{F(\mathcal{P}''_{\mathcal{C}})(\boldsymbol{h})}\right|^{\alpha}\right]. \quad (25)$$

For any $\mathcal{C}\subseteq[k-1]$, let $\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}=\{\tilde{\boldsymbol{p}}_i : i\in[k-1]\setminus\mathcal{C}\}$. With this notation, note that $\mathcal{P}_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}=\{\boldsymbol{p}''_k,\ldots,\boldsymbol{p}''_k\}\bigcup\{\boldsymbol{p}_k\}$, $\mathcal{P}'_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}=\{\boldsymbol{p}''_k,\ldots,\boldsymbol{p}''_k\}\bigcup\{\boldsymbol{p}'_k\}$, and $\mathcal{P}''_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}=\{\boldsymbol{p}''_k,\ldots,\boldsymbol{p}''_k\}\bigcup\{\boldsymbol{p}''_k\}$ are a triple of specific neighboring distributions, each containing $|\mathcal{C}|+1$ distributions. In other words, if we define $\mathcal{D}^{(k)}_{|\mathcal{C}|+1}=(d''_k,\ldots,d''_k,d_k)$, $\mathcal{D}'^{(k)}_{|\mathcal{C}|+1}=(d''_k,\ldots,d''_k,d'_k)$, and $\mathcal{D}''^{(k)}_{|\mathcal{C}|+1}=(d''_k,\ldots,d''_k,d''_k)$, each having $(|\mathcal{C}|+1)$ data points (note that $(\mathcal{D}''^{(k)}_{|\mathcal{C}|+1},\mathcal{D}'^{(k)}_{|\mathcal{C}|+1},\mathcal{D}^{(k)}_{|\mathcal{C}|+1})\in\mathcal{D}^{|\mathcal{C}|+1}_{\text{same}}$), then the mechanisms $\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{|\mathcal{C}|+1})$, $\mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{|\mathcal{C}|+1})$, and $\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{|\mathcal{C}|+1})$ will have distributions $F(\mathcal{P}_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}})$, $F(\mathcal{P}'_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}})$, and $F(\mathcal{P}''_{\mathcal{C}}\setminus\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}})$, respectively.

Now, since $(\mathcal{D}''^{(k)}_{|\mathcal{C}|+1},\mathcal{D}'^{(k)}_{|\mathcal{C}|+1},\mathcal{D}^{(k)}_{|\mathcal{C}|+1})\in\mathcal{D}^{|\mathcal{C}|+1}_{\text{same}}$, if we remove the effect of distributions in $\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}$ in the RHS of (25), we would be able to bound the RHS of (25) using the ternary $|\chi|^{\alpha}$-divergence 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 $\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}$ in the RHS (25).

The following lemma holds for arbitrary triples $(\mathcal{P},\mathcal{P}',\mathcal{P}'')$ of neighboring distributions $\mathcal{P}=\{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_{k-1},\boldsymbol{p}_k\}$, $\mathcal{P}'=\{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_{k-1},\boldsymbol{p}'_k\}$, and $\mathcal{P}''=\{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_{k-1},\boldsymbol{p}''_k\}$, where we show that the ternary $|\chi|^{\alpha}$-divergence $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}'')}\left[\left|\frac{F(\mathcal{P})(\boldsymbol{h})-F(\mathcal{P}')(\boldsymbol{h})}{F(\mathcal{P}'')(\boldsymbol{h})}\right|^{\alpha}\right]$ does not decrease when we eliminate a distribution $\boldsymbol{p}_i$ (i.e., remove the data point d_i from the datasets) for any $i\in[k-1]$.

Lemma 5 (Monotonicity)**.** *For any $i\in[k-1]$, we have*

$$\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}'')}\left[\left|\frac{F(\mathcal{P})(\boldsymbol{h})-F(\mathcal{P}')(\boldsymbol{h})}{F(\mathcal{P}'')(\boldsymbol{h})}\right|^{\alpha}\right]\leq\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}''_{-i})}\left[\left|\frac{F(\mathcal{P}_{-i})(\boldsymbol{h})-F(\mathcal{P}'_{-i})(\boldsymbol{h})}{F(\mathcal{P}''_{-i})(\boldsymbol{h})}\right|^{\alpha}\right]. \quad (26)$$

Proof. This can be proved along the lines of the proof of [29, Lemma 5], which shows that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\lambda}\right]\leq\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}'_{-i})}\left[\left(\frac{F(\mathcal{P}_{-i})(\boldsymbol{h})}{F(\mathcal{P}'_{-i})(\boldsymbol{h})}\right)^{\lambda}\right]$ holds for all $i\in[k-1]$. This is a result about Renyi divergence, and the only property of the Renyi divergence that is used in the proof of [29, Lemma 5] is that $\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}')}\left[\left(\frac{F(\mathcal{P})(\boldsymbol{h})}{F(\mathcal{P}')(\boldsymbol{h})}\right)^{\lambda}\right]$ is convex in $\boldsymbol{p}_i$ for any $i\in[k-1]$.

Note that Lemma 5 is about the ternary $|\chi|^{\alpha}$-divergence, and the required convexity about this follows from Lemma 3. So, following the proof of [29, Lemma 5] and using Lemma 3, proves Lemma 5. ■

Now, for any given $\mathcal{C}\subseteq[k-1]$, by eliminating the distributions $\tilde{\boldsymbol{p}}_i$ in $\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}=\{\tilde{\boldsymbol{p}}_i : i\in[k-1]\setminus\mathcal{C}\}$ from $\mathcal{P}_C$, $\mathcal{P}'_C$, and $\mathcal{P}''_C$ (by repeatedly applying Lemma 5), we get that

$$\mathbb{E}_{\boldsymbol{h}\sim F(\mathcal{P}''_{\mathcal{C}})}\left[\left|\frac{F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h})-F(\mathcal{P}'_{\mathcal{C}})(\boldsymbol{h})}{F(\mathcal{P}''_{\mathcal{C}})(\boldsymbol{h})}\right|^{\alpha}\right]$$
$$\leq\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m+1})(\boldsymbol{h})-\mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{m+1})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}\right|^{\alpha}\right], \quad (27)$$

where $m=|\mathcal{C}|$. By substituting from (27) into (25) completes the proof of Theorem 5.

C.2 Proof of Lemma 4

This can be proved along the lines of the proof of [29, Lemma 3], and we prove it below for completeness.

We only show (22); (23) and (24) can be shown similarly.

For convenience, for any $\mathcal{C} \subseteq [k-1]$, define

$$\mathcal{P}''_{|\mathcal{C}|,k} = \{\boldsymbol{p}''_k, \ldots, \boldsymbol{p}''_k\} \text{ with } |\mathcal{P}''_{|\mathcal{C}|,k}| = |\mathcal{C}|,$$
$$\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}} = \{\tilde{\boldsymbol{p}}_i : i \in [k-1] \setminus \mathcal{C}\}.$$

With these notations, we can write $\mathcal{P}_{\mathcal{C}} = \mathcal{P}''_{|\mathcal{C}|,k} \bigcup \widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}_k\}$, and $\mathcal{P}'_{\mathcal{C}} = \mathcal{P}''_{|\mathcal{C}|,k} \bigcup \widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}'_k\}$, and $\mathcal{P}''_{\mathcal{C}} = \mathcal{P}''_{|\mathcal{C}|,k} \bigcup \widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}''_k\}$.

Note that $\boldsymbol{p}_i = q\boldsymbol{p}'_k + (1-q)\tilde{\boldsymbol{p}}_i$ for all $i \in [k-1]$. For any $i \in [k-1]$, define the following random variable $\widehat{\boldsymbol{p}}_i$:

$$\widehat{\boldsymbol{p}}_i = \begin{cases} \boldsymbol{p}''_k & \text{w.p. } q, \\ \widetilde{\boldsymbol{p}}_i & \text{w.p. } 1-q. \end{cases}$$

Note that $\mathbb{E}[\widehat{\boldsymbol{p}}_i] = \boldsymbol{p}_i$. For any subset $\mathcal{C} \subseteq [k-1]$, define an event $\mathcal{E}_{\mathcal{C}} := \{\widehat{\boldsymbol{p}}_i = \boldsymbol{p}''_k \text{ for } i \in \mathcal{C} \text{ and } \widehat{\boldsymbol{p}}_i = \widetilde{\boldsymbol{p}}_i \text{ for } i \in [k-1] \setminus \mathcal{C}\}$. 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 $\mathcal{A}_B^k$ whose distribution is equal to $F(\mathcal{P})$.

$$\begin{aligned}
F(\mathcal{P})(\boldsymbol{h}) &= \Pr[U(\mathcal{P}) = \boldsymbol{h}] \\
&= \Pr[U(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}, \boldsymbol{p}_n) = \boldsymbol{h}] \\
&= \Pr\left[U\left(\mathbb{E}[\widehat{\boldsymbol{p}}_1], \ldots, \mathbb{E}[\widehat{\boldsymbol{p}}_{k-1}], \boldsymbol{p}_k\right) = \boldsymbol{h}\right] \\
&= \sum_{\mathcal{C}\subseteq[k-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U\left(\mathbb{E}[\widehat{\boldsymbol{p}}_1], \ldots, \mathbb{E}[\widehat{\boldsymbol{p}}_{k-1}], \boldsymbol{p}_k\right) = \boldsymbol{h} \mid \mathcal{E}_{\mathcal{C}}\right] \\
&\overset{\text{(e)}}{=} \sum_{\mathcal{C}\subseteq[k-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U\left(\mathcal{P}'_{|\mathcal{C}|,k} \bigcup \widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}} \bigcup \{\boldsymbol{p}_k\}\right) = \boldsymbol{h}\right] \\
&= \sum_{\mathcal{C}\subseteq[k-1]} \Pr[\mathcal{E}_{\mathcal{C}}] \Pr\left[U(\mathcal{P}_{\mathcal{C}}) = \boldsymbol{h}\right] \\
&= \sum_{\mathcal{C}\subseteq[k-1]} q^{|\mathcal{C}|}(1-q)^{k-|\mathcal{C}|-1} \Pr\left[U(\mathcal{P}_{\mathcal{C}}) = \boldsymbol{h}\right], \\
&= \sum_{\mathcal{C}\subseteq[k-1]} q^{|\mathcal{C}|}(1-q)^{k-|\mathcal{C}|-1} F(\mathcal{P}_{\mathcal{C}})(\boldsymbol{h}) \qquad (28)
\end{aligned}$$

where, $\mathcal{P}'_{|\mathcal{C}|,k}$ and $\widetilde{\mathcal{P}}_{[k-1]\setminus\mathcal{C}}$ in the RHS of (e) are defined in the statement of the claim.

Since the above calculation holds for every $\boldsymbol{h} \in \mathcal{A}_B^k$, we have proved (22).

C.3 Proof of Theorem 6: Ternary $|\chi|^\alpha$-DP of the Special Case

First, we present the following standard inequality which is important to our proof.

Lemma 6. *Let $x, y \in \mathbb{R}$ be any two real numbers. Then, for all $j \geq 1$, we have*

$$|x+y|^j \leq 2^{j-1}\left(|x|^j + |y|^j\right). \qquad (29)$$

Proof. The proof is simple from the convexity of the function $f(x) = x^j$ for $j \geq 1$.

$$|x+y|^j = 2^j \left|\frac{x+y}{2}\right|^j \leq 2^j \left(\frac{|x|+|y|}{2}\right)^j \leq 2^j \left(\frac{|x|^j + |y|^j}{2}\right) = 2^{j-1}\left(|x|^j + |y|^j\right),$$

where the second inequality is obtained from the Jensen's inequality and the fact that the function $f(x) = x^j$ is convex on $\mathbb{R}^+$ for all $j \geq 1$. ■

From Lemma 6, we get the following corollary.

Corollary 1. Fix an arbitrary $m \in \mathbb{N}$ and consider any three mutually neighboring datasets $\mathcal{D}_m,\ \mathcal{D}'_m,\ \mathcal{D}''_m$, where $\mathcal{D}_m = (d, \dots, d) \in \mathcal{X}^m$, $\mathcal{D}'_m = (d, \dots, d, d') \in \mathcal{X}^m$ and $\mathcal{D}''_m = (d, \dots, d, d'') \in \mathcal{X}^m$. The ternary $|\chi|^\alpha$-DP is bounded by

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})}\right|^\alpha\right] \\
&\quad \le 2^{\alpha-1}\left(\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right|^\alpha\right] + \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right|^\alpha\right]\right).
\end{aligned} \tag{30}$$

Proof. Fix any $\boldsymbol{h} \in \mathcal{A}_B^m$, and take $x = \left(\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)$, $y = -\left(\frac{\mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)$. Then applying Lemma 6 and taking expectation w.r.t. $\boldsymbol{h} \sim \mathcal{M}_{sh}(\mathcal{D}_m)$ will yield Corollary 1. ■

Remark 3. Observe that the proof of Corollary 1 does not require $\mathcal{D}_m, \mathcal{D}'_m, \mathcal{D}''_m$ to be special triple of neighboring datasets such that $(\mathcal{D}_m, \mathcal{D}'_m, \mathcal{D}''_m) \in \mathcal{D}^m_{\text{same}}$. In fact, Corollary 1 holds for any triple of distributions p, q, r over the same domain, for which we can show $\mathbb{E}_r[\left|\frac{p-q}{r}\right|^\alpha] \le 2^{\alpha-1}\left(\mathbb{E}_r[\left|\frac{p}{r} - 1\right|^\alpha] + \mathbb{E}_r[\left|\frac{q}{r} - 1\right|^\alpha]\right)$.

Now, in order to prove Theorem 6, it suffices to bound the expectation terms on the RHS of (30). This is what we do in the lemma below.

Lemma 7. *For any pair of the special pair of neighboring datasets $\mathcal{D}_m, \mathcal{D}'_m$, where $\mathcal{D}_m = (d, \dots, d) \in \mathcal{X}^m$ and $\mathcal{D}'_m = (d, \dots, d, d') \in \mathcal{X}^m$, we have*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right|^\alpha\right] \le \begin{cases} \frac{(e^{\epsilon_0}-1)^2}{me^{\epsilon_0}} & \text{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{\left(e^{2\epsilon_0}-1\right)^2}{2me^{2\epsilon_0}}\right)^{\alpha/2} & \text{otherwise.} \end{cases}$$

Substituting the bound from Lemma 7 into Corollary 1, we get

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}_m)}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}''_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})}\right|^\alpha\right] \le \begin{cases} 4\frac{(e^{\epsilon_0}-1)^2}{me^{\epsilon_0}} & \text{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{me^{2\epsilon_0}}\right)^{\alpha/2} & \text{otherwise,} \end{cases}$$

which completes the proof of Theorem 6.

Proof of Lemma 7. Consider any pair of neighboring datasets $\mathcal{D}_m,\ \mathcal{D}'_m$, where $\mathcal{D}_m = (d, \dots, d) \in \mathcal{X}^m$, $\mathcal{D}'_m = (d, \dots, d, d') \in \mathcal{X}^m$. Let $\boldsymbol{p} = (p_1, \dots, p_B)$ and $\boldsymbol{p}' = (p'_1, \dots, p'_B)$ 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

$$e^{-\epsilon_0} \le \frac{p_j}{p'_j} \le e^{\epsilon_0}, \qquad \forall j \in [B]. \tag{31}$$

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

$$\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h}) = \boldsymbol{MN}(m, \boldsymbol{p}, \boldsymbol{h}) = \binom{m}{\boldsymbol{h}} \prod_{j=1}^{B} p_j^{h_j}, \tag{32}$$

$$\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h}) = \sum_{j=1}^{B} p'_j \boldsymbol{MN}\left(m-1, \boldsymbol{p}, \widetilde{\boldsymbol{h}}_j\right), \tag{33}$$

where $\boldsymbol{MN}(m, \boldsymbol{p}, \boldsymbol{h})$ denotes the Multinomial distribution with $\binom{m}{\boldsymbol{h}} = \frac{m!}{h_1! \cdots h_B!}$.

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 $X(\boldsymbol{h}) := m\left(\frac{\mathcal{M}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right)$. Thus, we get that $\mathbb{E}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}'_m)(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}_m)(\boldsymbol{h})} - 1\right|^\alpha\right] = \frac{1}{m^\alpha}\mathbb{E}\left[|X|^\alpha\right]$. From [29, Lemmas 6, 7], we get that

$$\frac{1}{m^\alpha}\mathbb{E}\left[|X|^\alpha\right] \leq \begin{cases} \frac{(e^{\epsilon_0}-1)^2}{me^{\epsilon_0}} & \text{if } \alpha = 2, \\ \alpha\Gamma(\alpha/2)\left(\frac{\left(e^{2\epsilon_0}-1\right)^2}{2me^{2\epsilon_0}}\right)^{\alpha/2} & \text{otherwise.} \end{cases} \tag{34}$$

This completes the proof of Lemma 7. ■

C.4 Completing the Proof of Theorem 4

For simplicity of notation, for any $m \in \{0, 1, \ldots, n-1\}$, define

$$q_m := \binom{k-1}{m} q^m (1-q)^{k-m-1}$$

$$E_m := \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m+1})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{m+1})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}\right|^\alpha\right].$$

First we show an important property of E_m that we will use in the proof.

Lemma 8. *E_m is a non-increasing function of m, i.e.,*

$$\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m+1})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m+1})(\boldsymbol{h})}\right|^\alpha\right] \leq \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m})}\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D}^{(k)}_{m})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}'^{(k)}_{m})(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}''^{(k)}_{m})(\boldsymbol{h})}\right|^\alpha\right], \tag{35}$$

where, for any $l \in \{m, m+1\}$, $\mathcal{D}^{(k)}_l = (d''_k, \ldots, d''_k, d_k)$, $\mathcal{D}'^{(k)}_l = (d''_k, \ldots, d''_k, d'_k)$, and $\mathcal{D}''^{(k)}_l = (d''_k, \ldots, d''_k, d''_k)$, each having l elements.

Proof. Lemma 8 follows from Lemma 5 in a straightforward manner, as Lemma 5 is for arbitrary triples of adjacent datasets, whereas, Lemma 8 is for triples of adjacent datasets having special structures. ■

Thus, we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}(\mathcal{D}'')}&\left[\left|\frac{\mathcal{M}_{sh}(\mathcal{D})(\boldsymbol{h}) - \mathcal{M}_{sh}(\mathcal{D}')(\boldsymbol{h})}{\mathcal{M}_{sh}(\mathcal{D}'')(\boldsymbol{h})}\right|^\alpha\right] \\
&\leq \sum_{m=0}^{k-1} q_m E_m \\
&= \sum_{m<\lfloor(1-\gamma)q(k-1)\rfloor} q_m E_m + \sum_{m\geq\lfloor(1-\gamma)q(k-1)\rfloor} q_m E_m \\
&\overset{(a)}{\leq} E_0 \sum_{m<\lfloor(1-\gamma)q(k-1)\rfloor} q_m + \sum_{m\geq\lfloor(1-\gamma)q(k-1)\rfloor} q_m E_m \\
&\overset{(b)}{\leq} E_0 e^{-\frac{q(k-1)\gamma^2}{2}} + \sum_{m\geq\lfloor(1-\gamma)q(k-1)\rfloor} q_m E_m \\
&\overset{(c)}{\leq} e^{\epsilon_0\alpha} e^{-\frac{q(k-1)\gamma^2}{2}} + \sum_{m\geq\lfloor(1-\gamma)q(k-1)\rfloor} q_m E_m \\
&\overset{(d)}{\leq} (e^{\epsilon_0} - e^{-\epsilon_0})^\alpha e^{-\frac{q(k-1)\gamma^2}{2}} + E_{(1-\gamma)q(k-1)}.
\end{aligned} \tag{36}$$

Here, steps (a) and (d) follow from the fact that E_m is a non-increasing function of m (see Lemma 8). Step (b) follows from the Chernoff bound. In step (c), we used that $\mathcal{M}_{sh}(d_k) = \mathcal{R}(d_k)$, $\mathcal{M}_{sh}(d'_k) = \mathcal{R}(d'_k)$, and $\mathcal{M}_{sh}(d''_k) = \mathcal{R}(d''_k)$ which together imply that $E_0 = (e^{\epsilon_0} - e^{-\epsilon_0})^\alpha$, where the inequality follows because $\mathcal{R}$ is an ϵ_0-LDP mechanism. By choosing $\gamma = 0.5$ completes the proof of Theorem 4.

D Proof of Theorem 2 (Lower Bound)

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 $\Pr[\mathcal{R}(d) = d] = \frac{e^{\epsilon_0}}{e^{\epsilon_0}+1}$ for $d \in \mathcal{X}$. It is easy to verify that $\mathcal{R}$ is an ϵ_0-LDP mechanism. For simplicity, let $p = \frac{1}{e^{\epsilon_0}+1}$. Consider two neighboring datasets $\mathcal{D},\ \mathcal{D}' \in \{0,1\}^k$, where $\mathcal{D} = (0, \ldots, 0, 0)$ and $\mathcal{D}' = (0, \ldots, 0, 1)$. Let $m \in \{0, \ldots, k\}$ denote the number of ones in the output of the shuffler. We define two distributions

$$\begin{aligned}
\mu_0(m) &= \binom{k}{m} p^m (1-p)^{k-m}, \\
\mu_1(m) &= (1-p)\binom{k-1}{m-1} p^{m-1}(1-p)^{k-m} + p\binom{k-1}{m} p^m (1-p)^{k-m-1}.
\end{aligned} \tag{37}$$

As argued on page 5, since the output of the shuffled mechanism $\mathcal{M}$ can be thought of as the distribution of the number of ones in the output, we have that $m \sim \mathcal{M}(\mathcal{D})$ is distributed as a Binomial random variable $\text{Bin}(k, p)$. Thus, we have

$$\begin{aligned}
\mathcal{M}(\mathcal{D})(m) &= \mu_0(m) \\
\mathcal{M}(\mathcal{D}')(m) &= (1-\gamma)\mu_1(m) + \gamma\mu_0(m)
\end{aligned}$$

It will be useful to compute $\frac{\mu_1(m)}{\mu_0(m)} - 1$ for the calculations later.

$$\begin{aligned}
\frac{\mu_1(m)}{\mu_0(m)} - 1 &= \frac{(1-p)\binom{k-1}{m-1} p^{m-1}(1-p)^{k-m} + p\binom{k-1}{m} p^m (1-p)^{k-m-1}}{\binom{k}{m} p^m (1-p)^{k-m}} - 1 \\
&= \frac{m}{k}\frac{(1-p)}{p} + \frac{(k-m)}{k}\frac{p}{(1-p)} - 1 \\
&= \frac{m}{k} e^{\epsilon_0} + \frac{(k-m)}{k} e^{-\epsilon_0} - 1 \\
&= \frac{m}{k}\left(e^{\epsilon_0} - e^{-\epsilon_0}\right) + e^{-\epsilon_0} - 1 \\
&= \frac{m}{k}\left(\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right) - \left(\frac{e^{\epsilon_0}-1}{e^{\epsilon_0}}\right) \\
&= \left(\frac{e^{2\epsilon_0}-1}{ke^{\epsilon_0}}\right)\left(m - \frac{k}{e^{\epsilon_0}+1}\right)
\end{aligned} \tag{38}$$

Thus, we have that

$$\begin{aligned}
\mathbb{E}_{m\sim\mathcal{M}(\mathcal{D})}\left[\left(\frac{\mathcal{M}(\mathcal{D}')(m)}{\mathcal{M}(\mathcal{D})(m)}\right)^\lambda\right] &= \mathbb{E}\left[\left(1 + \gamma\left(\frac{\mu_1(m)}{\mu_0(m)} - 1\right)\right)^\lambda\right] \\
&\overset{(a)}{=} 1 + \sum_{i=1}^{\lambda}\binom{\lambda}{i}\gamma^i \mathbb{E}\left[\left(\frac{\mu_1(m)}{\mu_0(m)} - 1\right)^i\right] \\
&\overset{(b)}{=} 1 + \sum_{i=2}^{\lambda}\binom{\lambda}{i}\gamma^i \mathbb{E}\left[\left(\frac{\mu_1(m)}{\mu_0(m)} - 1\right)^i\right] \\
&= 1 + \sum_{i=2}^{\lambda}\binom{\lambda}{i}\gamma^i \left(\frac{(e^{2\epsilon_0}-1)}{ke^{\epsilon_0}}\right)^i \mathbb{E}\left[\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^i\right] \qquad \text{(from (38))} \\
&\overset{(c)}{=} 1 + \binom{\lambda}{2}\gamma^2 \frac{(e^{\epsilon_0}-1)^2}{ke^{\epsilon_0}} + \sum_{i=3}^{\lambda}\binom{\lambda}{i}\gamma^i \left(\frac{(e^{2\epsilon_0}-1)}{ke^{\epsilon_0}}\right)^i \mathbb{E}\left[\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^i\right].
\end{aligned}$$

Here, step (a) from the polynomial expansion $(1+x)^k = \sum_{m=0}^{k} \binom{k}{m} x^m$, step (b) follows because the term corresponding to $i = 1$ is zero (i.e., $\mathbb{E}_{m\sim\mu_0}\left[\left(\frac{\mu_1(m)}{\mu_0(m)} - 1\right)\right] = 0$), and step (c) from the from the fact that $\mathbb{E}_{m\sim\mu_0}\left[\left(m - \frac{k}{e^{\epsilon_0}+1}\right)^2\right] = kp(1-p) = \frac{ke^{\epsilon_0}}{(e^{\epsilon_0}+1)^2}$, which is equal to the variance of the Binomial random variable. This completes the proof of Theorem 2.

E Omitted Details from Section 5

E.1 Proof of Lemma 2

Consider arbitrary neighboring datasets $\mathcal{D} = (d_1, \ldots, d_n) \in \mathcal{X}^n$ and $\mathcal{D}' = (d_1, \ldots, d_{n-1}, d'_n) \in \mathcal{X}^n$. 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. Thus, without loss of generality, we deal with sets $\mathcal{P}$ and $\mathcal{P}'$ throughout this section instead of dealing with $\mathcal{D}$ and $\mathcal{D}'$. Thus, we write $\mathcal{M}(\mathcal{P}) \triangleq \mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{P}') \triangleq \mathcal{M}(\mathcal{D}')$.

We bound the Rényi divergence between $\mathcal{M}(\mathcal{P})$ and $\mathcal{M}(\mathcal{P}')$. For given a set $\mathcal{S} \subset [n]$ with $|\mathcal{S}| = \gamma n = k$, we define two sets $\mathcal{P}^{\mathcal{S}}, \mathcal{P}'^{\mathcal{S}}$, having k distributions each, as follows:

$$\mathcal{P}^{\mathcal{S}} = \{\boldsymbol{p}_i : i \in \mathcal{S}\}, \tag{39}$$

$$\mathcal{P}'^{\mathcal{S}} = \{\boldsymbol{p}_i : i \in \mathcal{S}\}, \tag{40}$$

Observe that when $n \notin \mathcal{S}$, we have that $\mathcal{P}^{\mathcal{S}} = \mathcal{P}'^{\mathcal{S}}$. For given k distributions $\mathcal{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k)$, we define a shuffle mechanism $\mathcal{M}_{sh}(\mathcal{P})$ as follows:

$$\mathcal{M}_{sh}(\mathcal{P}) = \mathcal{H}_k(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k). \tag{41}$$

Thus, the mechanisms $\mathcal{M}(\mathcal{P})$ and $\mathcal{M}(\mathcal{P}')$ can be defined by:

$$\begin{aligned}\mathcal{M}(\mathcal{P}) &= \frac{1}{\binom{n}{k}} \sum_{\mathcal{S}\subset[n]} \mathcal{M}_{sh}\left(\mathcal{P}^{\mathcal{S}}\right) \\ &= \gamma P_E + (1-\gamma) P_{E^c} \end{aligned} \tag{42}$$

$$\begin{aligned}\mathcal{M}(\mathcal{P}') &= \frac{1}{\binom{n}{k}} \sum_{\mathcal{S}\subset[n]} \mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right) \\ &= \gamma Q_E + (1-\gamma) P_{E^c}, \end{aligned} \tag{43}$$

where $P_E = \frac{1}{\binom{n-1}{k-1}} \sum_{\substack{\mathcal{S}\subset[n] \\ n\in\mathcal{S}}} \mathcal{M}_{sh}\left(\mathcal{P}^{\mathcal{S}}\right)$, $P_{E^c} = \frac{1}{\binom{n-1}{k}} \sum_{\substack{\mathcal{S}\subset[n] \\ n\notin\mathcal{S}}} \mathcal{M}_{sh}\left(\mathcal{P}^{\mathcal{S}}\right)$, and $Q_E = \frac{1}{\binom{n-1}{k-1}} \sum_{\substack{\mathcal{S}\subset[n] \\ n\in\mathcal{S}}} \mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right)$. Hence, from the polynomial expansion, we get:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{P}')}\left[\left(\frac{\mathcal{M}(\mathcal{P})(\boldsymbol{h})}{\mathcal{M}(\mathcal{P}')(\boldsymbol{h})}\right)^{\lambda}\right] &= \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{P}')}\left[\left(1 + \frac{\mathcal{M}(\mathcal{P})(\boldsymbol{h})}{\mathcal{M}(\mathcal{P}')(\boldsymbol{h})} - 1\right)^{\lambda}\right] \\ &= 1 + \sum_{j=2}^{\lambda} \binom{\lambda}{j} \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{P}')}\left[\left(\frac{\mathcal{M}(\mathcal{P})(\boldsymbol{h}) - \mathcal{M}(\mathcal{P}')(\boldsymbol{h})}{\mathcal{M}(\mathcal{P}')(\boldsymbol{h})}\right)^{j}\right] \\ &= 1 + \sum_{j=2}^{\lambda} \binom{\lambda}{j} \gamma^j \mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{P}')}\left[\left(\frac{P_E(\boldsymbol{h}) - Q_E(\boldsymbol{h})}{\mathcal{M}(\mathcal{P}')(\boldsymbol{h})}\right)^{j}\right] \end{aligned} \tag{44}$$

Now, we borrow the trick used in [43] to bound each term in the right hand side in (44). For completeness, we repeat their definitions and proofs here. We define an auxiliary dummy variable $i \sim \text{Unif}\,(1,\ldots,k)$ that is independent to everything else. Furthermore, we define two functions $g(\mathcal{S},i)$ and $g'(\mathcal{S},i)$ as follows:

$$g(\mathcal{S},i) = \begin{cases} \mathcal{M}_{sh}\left(\mathcal{P}^{\mathcal{S}}\right) & \text{if } n \in \mathcal{S} \\ \mathcal{M}_{sh}\left(\mathcal{P}^{\mathcal{S}\cup\{n\}\setminus\mathcal{S}(i)}\right) & \text{otherwise} \end{cases} \tag{45}$$

$$g'(\mathcal{S},i) = \begin{cases} \mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right) & \text{if } n \in \mathcal{S} \\ \mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}\cup\{n\}\setminus\mathcal{S}(i)}\right) & \text{otherwise} \end{cases} \tag{46}$$

Observe that $\mathbb{E}_{\mathcal{S},i}\left[g(\mathcal{S},i)\right] = P_E$, $\mathbb{E}_{\mathcal{S},i}\left[g'(\mathcal{S},i)\right] = Q_E$, and $\mathbb{E}_{\mathcal{S},i}\left[\mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right)\right] = \mathcal{M}\left(\mathcal{P}\right)$. As a result, we get that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}(\mathcal{P}')}\left[\left(\frac{P_E\left(\boldsymbol{h}\right)-Q_E\left(\boldsymbol{h}\right)}{\mathcal{M}\left(\mathcal{P}'\right)\left(\boldsymbol{h}\right)}\right)^j\right] &\leq \sum_{\boldsymbol{h}} \frac{\left|P_E\left(\boldsymbol{h}\right)-Q_E\left(\boldsymbol{h}\right)\right|^j}{\left(\mathcal{M}\left(\mathcal{P}'\right)\left(\boldsymbol{h}\right)\right)^{j-1}} \\
&\leq \sum_{\boldsymbol{h}} \mathbb{E}_{\mathcal{S},i}\left[\frac{\left|g(\mathcal{S},i)\left(\boldsymbol{h}\right)-g'(\mathcal{S},i)\left(\boldsymbol{h}\right)\right|^j}{\left(\mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right)\left(\boldsymbol{h}\right)\right)^{j-1}}\right] \\
&= \mathbb{E}_{\mathcal{S},i}\mathbb{E}_{\boldsymbol{h}\sim\mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right)}\left[\left(\frac{\left|g(\mathcal{S},i)\left(\boldsymbol{h}\right)-g'(\mathcal{S},i)\left(\boldsymbol{h}\right)\right|}{\mathcal{M}_{sh}\left(\mathcal{P}'^{\mathcal{S}}\right)\left(\boldsymbol{h}\right)}\right)^j\right] \\
&\leq \left(\zeta(j)\right)^j.
\end{aligned}$$

Here, step (a) follows from Jensen's inequality and the convexity of the function x^j/y^{j-1} (see [43, Lemma 20]). Step (b) follows from Fubini's theorem. The last inequality is obtained by taking the supremum over all possible neighboring datasets $\mathcal{D}$, $\mathcal{D}'$. This completes the proof of Lemma 2.

E.2 Combining Theorem 4 and Lemma 2 for Proving Theorem 1

In this section, we complete the remaining calculation from Section 5 of combining the results of Theorem 4 and Lemma 2 and proving Theorem 1.

Substituting the bound on $\zeta(\alpha)$ from Theorem 4 into Lemma 2, we get

$$\begin{aligned}
\epsilon\left(\lambda\right) \leq \frac{1}{\lambda-1}\log\Bigg[1 &+ \binom{\lambda}{2}\gamma^2\left(4\frac{\left(e^{\epsilon_0}-1\right)^2}{\overline{k}e^{\epsilon_0}} + (e^{\epsilon_0}-e^{-\epsilon_0})^2 e^{-\frac{k-1}{8e^{\epsilon_0}}}\right) \\
&+ \sum_{\alpha=3}^{\lambda}\binom{\lambda}{\alpha}\gamma^\alpha\left(\alpha\Gamma(\alpha/2)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{\overline{k}e^{2\epsilon_0}}\right)^{\alpha/2} + (e^{\epsilon_0}-e^{-\epsilon_0})^\alpha e^{-\frac{k-1}{8e^{\epsilon_0}}}\right)\Bigg] \\
= \frac{1}{\lambda-1}\log\Bigg[1 &+ 4\binom{\lambda}{2}\gamma^2\frac{\left(e^{\epsilon_0}-1\right)^2}{\overline{k}e^{\epsilon_0}} + \sum_{\alpha=3}^{\lambda}\binom{\lambda}{\alpha}\gamma^\alpha\alpha\Gamma(\alpha/2)\left(\frac{2\left(e^{2\epsilon_0}-1\right)^2}{\overline{k}e^{2\epsilon_0}}\right)^{\alpha/2} + \Upsilon\Bigg],
\end{aligned} \tag{47}$$

where $\Upsilon = \sum_{\alpha=2}^{\lambda}\binom{\lambda}{\alpha}\gamma^\alpha(e^{\epsilon_0}-e^{-\epsilon_0})^\alpha e^{-\frac{k-1}{8e^{\epsilon_0}}} = \left(\left(1+\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right)^\lambda - 1 - \lambda\gamma\frac{e^{2\epsilon_0}-1}{e^{\epsilon_0}}\right)e^{-\frac{k-1}{8e^{\epsilon_0}}}$.

The above expression in (47) is the bound given in Theorem 1.

F Proof of Theorem 3: Privacy-Convergence Tradeoff

In this section, we prove the privacy-convergence tradeoff of Algorithm 1 and prove Theorem 3.

The privacy part is straightforward from conversion from RDP to approximate DP using Lemma 1 and Theorem 1. Now, we prove the convergence rate.

At iteration $t \in [T]$ of Algorithm 1, server averages the k received gradients and obtains $\overline{\mathbf{g}}_t = \frac{1}{k}\sum_{i\in\mathcal{U}_t}\mathcal{R}_p\left(\tilde{\mathbf{g}}_t\left(d_i\right)\right)$ and then updates the parameter vector as $\theta_{t+1} \leftarrow \prod_{\mathcal{C}}\left(\theta_t - \eta_t\overline{\mathbf{g}}_t\right)$. Observe that the mechanism $\mathcal{R}_p$ is unbiased and has a bounded variance: $\sup_{\mathbf{x}\in\mathcal{B}_p(L)}\mathbb{E}\|\mathcal{R}_p\left(\mathbf{x}\right) - \mathbf{x}\|_2^2 \leq G_p^2(L)$. As a result, 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 subsampling of clients 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 9. *For any $d \in \mathcal{X}$, if the function $f\left(\theta;.\right) : \mathcal{C}\times\mathcal{X} \to \mathbb{R}$ is convex and L-Lipschitz continuous with respect to the ℓ_g-norm, which is the dual of the ℓ_p-norm (i.e., $\frac{1}{p}+\frac{1}{g}=1$), then we have*

$$\mathbb{E}\|\overline{\mathbf{g}}_t\|_2^2 \leq L^2\max\{d^{1-\frac{2}{p}},1\}\left(1+\frac{cd}{qn}\left(\frac{e^{\epsilon_0}+1}{e^{\epsilon_0}-1}\right)^2\right), \tag{48}$$

where c is a global constant: $c=4$ if $p\in\{1,\infty\}$ and $c=14$ if $p\notin\{1,\infty\}$.

Proof. Under the conditions of the lemma, we have from [40, 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

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

Step (a) follows from the fact that $\|\nabla_{\theta_t}F\left(\theta_t\right)\| \leq L$ together with the norm inequality $\|\boldsymbol{u}\|_q \leq \|\boldsymbol{u}\|_p \leq d^{\frac{1}{p}-\frac{1}{q}}\|\boldsymbol{u}\|_q$ for $1\leq p\leq q\leq\infty$. Step (b) follows from the assumption that $\mathcal{R}_p$ has bounded variance. Step (c) uses $\gamma = \frac{k}{n}$. ■

Now, we can use standard SGD convergence results for convex functions. In particular, we use the following result from [41].

Lemma 10 (SGD Convergence [41])**.** *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*

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq 2DG\frac{2+\log\left(T\right)}{\sqrt{T}} = \mathcal{O}\left(DG\frac{\log\left(T\right)}{\sqrt{T}}\right). \tag{49}$$

As shown in Lemma 9 and above that Algorithm 1 satisfies the premise of Lemma 10. Now, using the bound on G^2 from Lemma 9, we have that the output θ_T of Algorithm 1 satisfies

$$\mathbb{E}\left[F\left(\theta_T\right)\right] - F\left(\theta^*\right) \leq \mathcal{O}\left(DG\frac{\log\left(T\right)}{\sqrt{T}}\right), \tag{50}$$

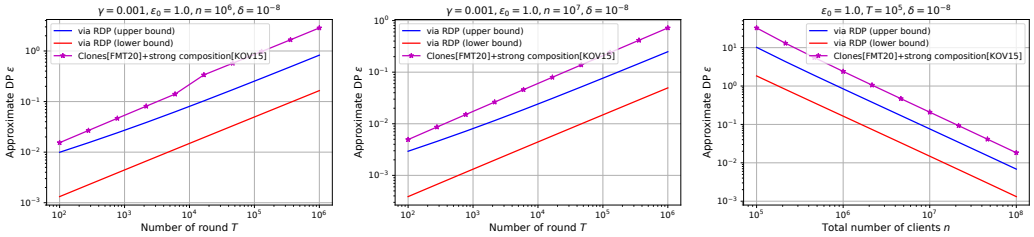
where $G^2 = \max\{d^{1-\frac{2}{p}},1\}L^2 + \frac{G_p^2(L)}{\gamma n}$. This completes the proof of the third part of Theorem 3.

G Additional Numerical Results

In this section, we present additional numerical experiments of our bounds on RDP of the subsampled shuffle mechanism and its usage for getting approximate DP of Algorithm 1 for training machine learning models.

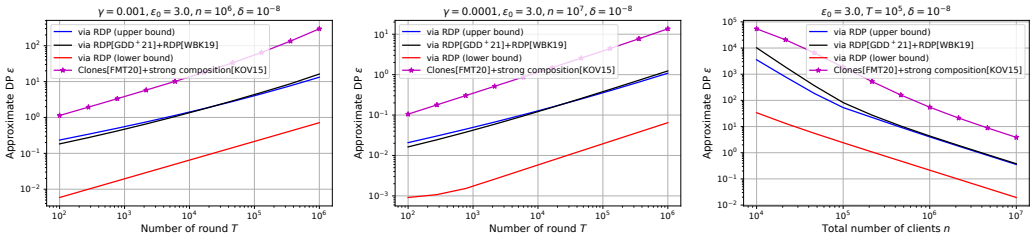
Composition of a sequence of subsampled shuffle models: Observe that our RDP bound presented in 1 is general for any values of LDP parameter ϵ_0, number of clients n, and RDP order $\lambda \geq 2$. On the other hand, the results of the privacy amplification by shuffling presented in [24] is valid under the condition on the LDP parameter:

$$\epsilon_0 \leq \log\left(\frac{n}{16\log(2/\delta)}\right). \tag{51}$$



(a) Approx. DP as a function of T for $\epsilon_0 = 1, \gamma = 0.001, n = 10^6$ (b) Approx. DP as a function of T for $\epsilon_0 = 1, \gamma = 0.001, n = 10^7$ (c) Approx. DP as a function of n for $\epsilon_0 = 1, \gamma n = 10^4, T = 10^5$

Figure 4: Comparison of our bound on the Approximate (ϵ, δ)-DP (blue) for composition of a sequence of subsampled shuffle mechanisms for $\delta = 10^{-8}$ with applying the strong composition theorem [34] after getting the Approximate DP of the shuffled model given in [24] with subsampling [42] (magenta).



(a) Approx. DP as a function of T for $\epsilon_0 = 3, \gamma = 0.001, n = 10^6$ (b) Approx. DP as a function of T for $\epsilon_0 = 3, \gamma = 0.0001, n = 10^7$ (c) Approx. DP as a function of n for $\epsilon_0 = 3, \gamma n = 10^3, T = 10^5$

Figure 5: Comparison of several bounds on the Approximate (ϵ, δ)-DP for composition of a sequence of subsampled shuffle mechanisms for $\delta = 10^{-8}$: (i) Approximate DP obtained from our upper bound on the RDP in Theorem 1 (blue); (ii) Approximate DP obtained from our lower bound on the RDP in Theorem 2 (red); (iii) Approximate DP obtained from the upper bound on the RDP given in [29] with RDP amplification by subsampling from [43] (black); and (iv) Applying the strong composition theorem [34] after getting the Approximate DP of the shuffled model given in [24] with subsampling [42] (magenta).

Furthermore, the results of the privacy amplification by shuffling presented in [7] is valid under the condition on the LDP parameter:

$$\epsilon_0 \leq \frac{1}{2} \log \left(\frac{n}{\log(1/\delta)} \right). \tag{52}$$

Thus, if the conditions in (51)- (52) do not hold, then the results in [24, 7] have a privacy bound: $\epsilon = \epsilon_0$ and $\delta = 0$. For example, when total number of clients $n = 10^6$, LDP parameter $\epsilon_0 = 3$, and we choose uniformly at random $k = 1000$ clients at each iteration, then both conditions (51)- (52) do not hold.

In Figure 4, we plot additional results for comparison between our bound on the approximate DP for a composition of T mechanisms $(\mathcal{M}_1, \ldots, \mathcal{M}_T)$ with the bound obtained by applying the strong composition theorem [34] after getting the approximate DP of the shuffled model given in [24] with subsampling [42]. In Figure 5, we plot our bound 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 shuffle mechanism for $t \in [T]$. In all our experiments reported in Figure 5, we fix $\delta = 10^{-8}$ and $\epsilon_0 = 3$, where we consider the cases in which the conditions (51)- (52) do not hold. Thus, we compare our results with the bound: $\epsilon = \epsilon_0$ and $\delta = 0$.

We observe that our new bound on the RDP of the subsampled shuffle mechanism achieves a significant saving in total privacy ϵ compared to the bound in [24, 7]. For example, we save a factor of $17\times$ compared to the results in [24, 7] with the strong composition theorem [34] in computing the overall privacy parameter ϵ for number of iterations $T = 10^5$, subsampling parameter $\gamma = 0.001$,

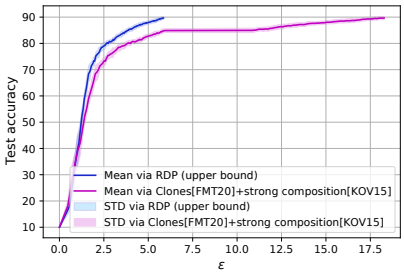


Figure 6: Privacy-Utility trade-offs on the MNIST dataset with ℓ_2-norm clipping.

LDP parameter $\epsilon_0 = 3$, and number of clients $n = 10^6$. We observe that our bound on the over all privacy parameters almost matches the bound on the RDP given in [29] with subsampled RDP [43] when subsampling parameter $\gamma = 0.001$, LDP parameter $\epsilon_0 = 3$, and number of clients $n = 10^6$.

Distributed private learning: We numerically evaluate the proposed privacy-learning performance on training machine learning models. We train a simple neural network on MNIST dataset that was also used in [21, 39] and described in Table 1. We assume that we have $n = 60,000$ clients, where each client has one sample. At each step of the Algorithm 1, we choose uniformly at random $2,000$ clients, where each client clips the ℓ_2-norm of the gradient with clipping parameter $C = 0.005$ and applies the $\mathcal{R}_2$ ϵ_0-LDP mechanism (Privunit) proposed in [17] with $\epsilon_0 = 2$. We run Algorithm 1 with $\delta = 10^{-5}$ for 200 epochs, with learning rate $\eta = 12$ for the first 30 epochs, and then decrease it to 4 in the next 30 epochs. We decrease the learning rate to 3.5 for the remaining epochs.

Figure 6 plots the mean and the standard deviation of privacy-accuracy trade-offs averaged over 4 runs. We observe that we achieve an accuracy of $81.15\%(\pm 0.7)$ with a total privacy budget of $\epsilon = 3$ using our new privacy analysis, whereas, [24] achieves an accuracy of only $76.46\%(\pm 1.9)$ with the same privacy budget of $\epsilon = 3$ using the standard composition theorems. Furthermore, we can see that we achieves accuracy $89.7\%(\pm 0.5)$ with total privacy budget $\epsilon = 5.8$ using our new privacy analysis, whereas, [24] (together with the standard strong composition theorem) achieves the same accuracy with a total privacy budget of $\epsilon = 18.3$.

Computation resources: For our experiments, we used a server which has 6 Nvidia RTX2080Ti GPU's and Intel Xeon Gold 6230 CPU @ 2.10GHz CPU's. The longest epoch time is 400 seconds for training on MNIST.