Improved Communication-Privacy Trade-offs in L_2 Mean Estimation under Streaming Differential Privacy

Wei-Ning Chen [1] Berivan Isik [1] Peter Kairouz [2] Albert No [3] Sewoong Oh [2 4] Zheng Xu [2]

Abstract

We study L_2 mean estimation under central differential privacy and communication constraints, and address two key challenges: firstly, existing mean estimation schemes that simultaneously handle both constraints are usually optimized for L_∞ geometry and rely on random rotation or Kashin's representation to adapt to L_2 geometry, resulting in suboptimal leading constants in mean square errors (MSEs); secondly, schemes achieving order-optimal communication-privacy trade-offs do not extend seamlessly to streaming differential privacy (DP) settings (e.g., tree aggregation or matrix factorization), rendering them incompatible with DP-FTRL type optimizers. In this work, we tackle these issues by introducing a novel privacy accounting method for the sparsified Gaussian mechanism that incorporates the randomness inherent in sparsification into the DP noise. Unlike previous approaches, our accounting algorithm directly operates in L_2 geometry, yielding MSEs that fast converge to those of the uncompressed Gaussian mechanism. Additionally, we extend the sparsification scheme to the matrix factorization framework under streaming DP and provide a precise accountant tailored for DP-FTRL type optimizers. Empirically, our method demonstrates at least a 100x improvement of compression for DP-SGD across various FL tasks.

1. Introduction

In federated learning (FL) (McMahan et al., 2016; Konečnỳ et al., 2016; Kairouz et al., 2021c), a server executes a specific learning task on data that is kept on clients' devices, avoiding the explicit collection of local raw datasets. This process typically involves the server iteratively gathering essential local model updates (such as noisy gradients) from the client side and subsequently updating the global model. While FL embodies the principle of data minimization by only requesting the minimal information necessary for model training, these local model updates may still contain sensitive information. As a result, additional privacy protection is necessary to prevent the trained model from possibly revealing individual information. Moreover, with the increase of model size, the exchange of local model updates becomes both memory and computation-intensive, leading to substantial latency and impeding the efficiency of training cycles. Consequently, it is desired to devise robust privacy protection mechanisms that simultaneously optimize communication efficiency.

In this paper, we study the L_2 mean estimation[1], a core sub-routine in the majority of FL schemes, subject to joint communication and differential privacy (DP) (Dwork et al., 2006) constraints. We consider two major types of DP optimization settings: (1) the classic DP-SGD type approach (Abadi et al., 2016) where independent DP noise is injected in each round of training, and (2) the DP-FTRL type approach (Guha Thakurta & Smith, 2013; Kairouz et al., 2021b; Denisov et al., 2022) where the DP noise is correlated across training rounds, the structure of which is intricately designed based on certain matrix factorization.

There has been a long thread of literature on distributed mean estimation (DME) under either or both privacy and communication constraints (Suresh et al., 2017; Konečnỳ et al., 2016; Agarwal et al., 2018; Chen et al., 2020; 2023; Shah et al., 2022; Feldman & Talwar, 2021; Isik et al., 2023a; Asi et al., 2023). Recent work by Chen et al. (2023) points out that, to achieve order-optimal mean square errors (MSEs) under joint constraints, it becomes imperative to integrate the inherent randomness utilized in compression (e.g., in sampling, sketching, or projection) into privacy analysis. Essentially, the implicit "compression noise" should be leveraged to amplify the DP guarantees, resulting

[*] Equal contribution [1] Stanford University [2] Google [3] Yonsei University [4] University of Washington. Correspondence to: Wei-Ning Chen <wnchen@stanford.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Here, L_2 refers to the L_2 geometry of the local model updates, i.e., $\|\boldsymbol{g}_i\|_2 \leq \Delta_2$ for all client i. This condition is typically maintained through the L_2 clipping step of the differential privacy mechanism.

in a significant reduction of DP noise. However, despite the coordinate subsampled Gaussian mechanism (CSGM) introduced by Chen et al. (2023) achieving order-optimal MSEs, it is crafted within the L_∞ geometry (i.e., assuming $\|\boldsymbol{g}_i\|_\infty \leq C_\infty$ for any local vector $\boldsymbol{g}_i$) and relies on random rotation or Kashin's representation to extend to L_2 mean estimation tasks. It is noteworthy that the bounded L_2 norm assumption is *strictly more robust* than the bounded L_∞ norm assumption (see Section 4.2), inevitably leading to larger MSEs with CSGM compared to the (uncompressed) Gaussian mechanism under equivalent DP guarantees.

A further challenge arises in CSGM (or, more broadly, general randomized compression schemes based on random projection, sampling, or sketching) when applied to streaming DP models (Guha Thakurta & Smith, 2013; Denisov et al., 2022; Jain et al., 2023), particularly in the context of DP-FTRL type optimization mechanisms based on tree aggregation (Honaker, 2015; Kairouz et al., 2021b) or matrix factorization (Denisov et al., 2022). In the streaming DP model, the DP noise injected in each round loses its independence. Instead, noise variables $\mathbf{Z} \in \mathbb{R}^T$ are correlated across T training rounds through a linear transform $\mathbf{B} \cdot \mathbf{Z}$, where $\mathbf{B} \in \mathbb{R}^{T \times T}$ is obtained from certain matrix factorization of the objective function that aims to minimize the overall distortions, such as MSEs. When the noise variables are correlated across rounds, they are no longer "aligned" with the randomness introduced in the local compression phase, as compression occurs locally and is thus independent across rounds. This complicates the analysis of privacy amplification, as privacy budgets cannot be accounted for round-wise, introducing what we term "temporal coupling." Moreover, the adaptive nature of DP-FTRL, where local gradients depend on the outputs of previous rounds, leads to the coupling of compression seeds that are typically introduced independently across dimensions. When analyzing the outputs over T rounds, this coupling, referred to as "spatial coupling," presents a significant challenge. Traditional privacy amplification tools (Balle et al., 2018; Zhu & Wang, 2019; Wang et al., 2019) fail in the face of such spatial and temporal coupling, necessitating a novel analysis approach.

Our contribution. In this work, we tackle both aforementioned challenges. Firstly, we introduce a novel privacy accounting method for the sparsified Gaussian mechanism. This method incorporates the inherent randomness from the sparsification phase into the DP noise. Unlike previous approaches in Chen et al. (2023), our accounting algorithm directly operates in L_2 geometry, resulting MSEs that converge fast to those of the uncompressed Gaussian mechanism. The key technique is to leverage the convexity of the Rényi DP profile of 1-dimensional subsampled Gaussian mechanism and extend it to multi-dimensional scenarios.

Secondly, we extend the application of the sparsified Gaussian mechanism to streaming DP settings, particularly within the matrix factorization DP-FTRL framework. We establish a Rényi privacy accounting theorem. While this theorem bears similarities to its non-streaming counterpart, the analysis necessitates a fundamentally different approach due to the spatial and temporal coupling inherent in the adaptive releases. A crucial step in our analysis involves decomposing the transcript (i.e., the collection of all releases across T training rounds), effectively transforming the adaptive releasing model into a non-adaptive one.

Although our analysis primarily revolves around the sparsified Gaussian mechanism (or coordinate subsampled Gaussian mechanism), it inherently encompasses a broader family of random projections, including subsampled randomized Hadamard transform (Ailon & Chazelle, 2006; Sarlos, 2006), and randomized Gaussian design (Wainwright, 2019, Section 6). These dimensionality reduction techniques can be viewed as a linear transform followed by a subsampling step. Additionally, by slightly lifting the dimension, these random designs exhibit deep connections to Kashin's representation, providing a uniform L_∞ bound, albeit with a larger leading constant (Lyubarskii & Vershynin, 2010).

Finally, we present comprehensive empirical results on the proposed L_2 sparsified Gaussian mechanism and sparsified Gaussian matrix factorization. Our results demonstrate a $100\times$ improvement in compression rates in various FL tasks (including FMNIST and Stackoverflow datasets). Moreover, our algorithm reduces the dimensionality of local model updates and hence can potentially be combined with other quantization or (scalar) lossless compression techniques (Alistarh et al., 2017; Isik et al., 2022; Mitchell et al., 2022).

2. Related Work

FL and DME. Federated learning (Konečnỳ et al., 2016; McMahan et al., 2016; Kairouz et al., 2019) emerges as a decentralized machine learning framework that provides data confidentiality by retaining clients' raw data on edge devices. In FL, communication between clients and the central server can quickly become a bottleneck (McMahan et al., 2016), so previous works have focused on compressing local model updates via gradient quantization (McMahan et al., 2016; Alistarh et al., 2017; Gandikota et al., 2019; Suresh et al., 2017; Wen et al., 2017; Wangni et al., 2018; Braverman et al., 2016), sparsification (Barnes et al., 2020; Hu et al., 2021; Farokhi, 2021; Isik et al., 2023b; Lin et al., 2018), or random projection (Rothchild et al., 2020; Vargaftik et al., 2021). To further enhance user privacy, FL is often combined with differential privacy (Dwork et al., 2006; Abadi et al., 2016; Agarwal et al., 2018; Hu et al., 2021).

Note that in this work, we consider FL (or more specifically, DME) under a *central*-DP setting where the server is trusted, which is different from the local DP model (Kasiviswanathan et al., 2011)[2] and the distributed DP model with secure aggregation (Bonawitz et al., 2016; Bell et al., 2020; Kairouz et al., 2021a; Agarwal et al., 2021; Chen et al., 2022b;a). When the secure aggregation is employed, local model updates cannot be compressed independently (Rothchild et al., 2020; Chen et al., 2023), and hence, the corresponding compression rates must be strictly higher than those without secure aggregation.

Streaming DP and DP-FTRL. In addition to the classic DP optimizers such as DP-SGD (Abadi et al., 2016) or DP-FedAvg (McMahan et al., 2016), we also study the online optimization settings such as DP-FTRL (Kairouz et al., 2021b) where the noise is correlated across rounds. This is motivated by the facts that (1) subsampling is often impractical in federated learning settings (Kairouz et al., 2021b; 2019), and (2) the correlated noise probably yields better utility compared to the independent noise (Choquette-Choo et al., 2023a;b). DP-FTRL algorithms are widely used in training production models in the cross-device FL system (Xu et al., 2023). The key component behind the DP-FTRL relies on the private releases under continual observation, an old problem dating back to (Dwork et al., 2010; Chan et al., 2012). Since then, several works have studied the continual release model and its applications (Upadhyay & Upadhyay, 2021; Choquette-Choo et al., 2022; 2023a; Henzinger et al., 2023; 2024). Kairouz et al. (2021b) originally used the efficient DP binary-tree estimator (Honaker, 2015) for the DP-FTRL algorithm, but later, a more general approach to cumulative sums based on matrix factorization (Hardt & Talwar, 2010; Li et al., 2015; Yuan et al., 2016; McKenna et al., 2018; Edmonds et al., 2020) was used. We, however, note that DP online optimization concerns *adaptive* inputs, that is, the future data points depend on previous outputs, and not all matrix mechanisms extend to the adaptive settings (Denisov et al., 2022), and it introduces challenges when incorporating compression into the privacy analysis. Indeed, to prove the adaptive DP guarantees of our algorithm, we need to handle the spatial and temporal dependency carefully. Finally, while the recent work Choquette-Choo et al. (2023b) also investigate privacy amplification through subsampling, their subsampling is conducted client-wise rather than coordinate-wise, as their scheme is not designed for compression. Consequently, Choquette-Choo et al. (2023b) do not encounter the spatial coupling issue as we do.

3. Preliminaries and Setups

In this section, we introduce the mathematical formulation of the problem and the DP models. We begin with DME in non-streaming DP, and then transition to the continual sum (or mean) problem within the streaming DP model.

3.1. DME and (Non-streaming) DP

Consider n clients, each with a local vector (e.g., local gradient or model update) $\boldsymbol{g}_i \in \mathbb{R}^d$ that satisfies $\|\boldsymbol{g}_i\|_2 \leq \Delta_2$ for some constant $\Delta_2 > 0$ (one can think of $\boldsymbol{g}_i$ as a clipped local gradient). A server aims to learn an estimate $\hat{\mu}$ of the mean $\mu(\boldsymbol{g}^n) \triangleq \frac{1}{n}\sum_i \boldsymbol{g}_i$ from $\boldsymbol{g}^n = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n)$ after communicating with the n clients. Toward this end, each client locally compresses $\boldsymbol{g}_i$ into a b-bit message $Y_i = \mathcal{E}_i\left(\boldsymbol{g}_i\right) \in \mathcal{Y}$ through a local encoder $\mathcal{E}_i : \mathbb{R}^d \mapsto \mathcal{Y}$ (where $|\mathcal{Y}| \leq 2^b$) and sends it to the central server, which upon receiving $Y^n = (Y_1, \ldots, Y_n)$ computes an estimate $\hat{\mu}\left(Y^n\right)$ that satisfies the following differential privacy:

Definition 3.1 (Differential Privacy (Dwork et al., 2006))**.** A mechanism (i.e., a randomized mapping) $\mathcal{M}(\boldsymbol{g}^n)$ is (ε, δ)-DP if for any neighboring datasets $\boldsymbol{g}^n \triangleq (\boldsymbol{g}_1, ..., \boldsymbol{g}_i, ..., \boldsymbol{g}_n)$, $\boldsymbol{h}^n \triangleq (\boldsymbol{g}_1, ..., \boldsymbol{g}_{i-1}, \boldsymbol{h}_i, \boldsymbol{g}_{i+1}, ..., \boldsymbol{g}_n)$, and measurable $\mathcal{S} \in \mathsf{range}\left(\mathcal{M}\right)$, it holds that

$$\Pr\left\{\mathcal{M}(\boldsymbol{g}^n) \in \mathcal{S}\right\} \leq e^{\varepsilon} \cdot \Pr\left\{\mathcal{M}(\boldsymbol{h}^n) \in \mathcal{S}\right\} + \delta,$$

where the probability is taken over the randomness of $\mathcal{M}(\cdot)$.

Our goal is to design schemes that minimize the MSE:

$$\min_{(\mathcal{E}_1,\ldots,\mathcal{E}_n,\hat{\mu})} \max_{\boldsymbol{g}^n} \mathbb{E}\left[\left\|\hat{\mu}\left(\mathcal{E}_1(\boldsymbol{g}_1), ..., \mathcal{E}_n(\boldsymbol{g}_n)\right) - \mu(\boldsymbol{g}^n)\right\|_2^2\right],$$

subject to b-bit communication and (ε, δ)-DP constraints.

The above DME task is closely related to FL with batched SGD (or other similar stochastic optimization methods, such as FedAvg (McMahan et al., 2016)). *In each round*, the server updates the global model using a noisy mean of local model updates. This estimate is typically derived through a DME primitive. As demonstrated in (Ghadimi & Lan, 2013), if the estimate remains unbiased in each round, convergence rates depend on the L_2 estimation error. Note that the DME procedure is invoked independently in each round, and the privacy budget is allocated for T rounds of training, differing from the online DP setting discussed below.

3.2. Streaming Differential Privacy

Next, we introduce the streaming DME problem and matrix mechanisms. To begin with, we first summarize the streaming DP setting (Denisov et al., 2022). A streaming mechanism $\mathcal{M}$ takes inputs $\boldsymbol{g}^{(1)}, \boldsymbol{g}^{(2)}, ..., \boldsymbol{g}^{(t)}$ and outputs $\boldsymbol{o}^{(t)}$ at time t. We denote the stream with T rounds in the following matrix form:

$$\mathbf{G} \triangleq \begin{bmatrix} —\boldsymbol{g}^{(1)}— \\ \vdots \\ —\boldsymbol{g}^{(T)}— \end{bmatrix},$$

[2] Another alternative to private DME is via local DP and shuffling. We provide a detailed discussion on this direction in Appendix B

and similarly for $\mathbf{H}$ and the adversary's view $\mathbf{O}$.

An adversary that adaptively defines two input sequences $\mathbf{G} = (\boldsymbol{g}^{(1)}, ..., \boldsymbol{g}^{(T)})$ and $\mathbf{H} = (\boldsymbol{h}^{(1)}, ..., \boldsymbol{h}^{(T)})$. The adversary must satisfy the promise that these sequences correspond to neighboring data sets. The privacy game proceeds in rounds. At round t, the adversary generates $\boldsymbol{g}^{(t)}$ and $\boldsymbol{h}^{(t)}$, *as a function of* $\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)}$. The game accepts these if the input streams defined so far are valid, meaning that there exist completions $\tilde{\mathbf{G}} = \left(\boldsymbol{g}^{(1)}, ..., \boldsymbol{g}^{(t)}, \tilde{\boldsymbol{g}}^{(t+1)}, ..., \tilde{\boldsymbol{g}}^{(T)}\right)$ and $\tilde{\mathbf{H}} = \left(\boldsymbol{h}^{(1)}, ..., \boldsymbol{h}^{(t)}, \tilde{\boldsymbol{h}}^{(t+1)}, ..., \tilde{\boldsymbol{h}}^{(T)}\right)$ such that $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{H}}$ are neighboring, in the following sense:

Definition 3.2 (Neighboring datasets)**.** Two data streams $\mathbf{G}$ and $\mathbf{H}$ in $\mathbb{R}^{T\times d}$ will be considered to be neighboring if they differ by a single row, with the ℓ_2-norm of the difference in this row at most Δ_2.

The game is parameterized by a bit $\mathsf{side} \in \{0, 1\}$, which is unknown to the adversary but constant throughout the game. The game hands either $\mathbf{G}$ or $\mathbf{H}$ to $\mathcal{M}$, depending on side. We say $\mathcal{M}$ is $(\alpha, \varepsilon(\alpha))$ Rényi DP if the adversary's views $\mathbf{O}$ under $\mathsf{side} = 0$ and $\mathsf{side} = 1$ is $\varepsilon(\alpha)$-indistinguishable under Rényi divergence at order α: $D_\alpha\left(P_{\mathbf{O}|\mathbf{G}} \middle\| P_{\mathbf{O}|\mathbf{H}}\right) \leq \varepsilon(\alpha)$.

Theorem 3.3 (Restated from Theorem 2.1 of Denisov et al. (2022))**.** *Let* $\mathbf{A} \in \mathbb{R}^{T\times T}$ *be a lower-triangular full-rank query matrix, and let* $\mathbf{A} = \mathbf{BC}$ *be any factorization with the following property: for any two neighboring streams of vectors* $\mathbf{G}, \mathbf{H} \in \mathbb{R}^{T\times d}$*, we have* $\left\|\mathbf{C}\left(\mathbf{G} - \mathbf{H}\right)\right\|_F \leq \kappa_2$*. Let* $\mathbf{Z} \in \mathbb{R}^{T\times d}$ *such that* $\mathbf{Z}_{i,j} \overset{i.i.d.}{\sim} \mathcal{N}(0, \kappa^2\sigma^2)$ *with* σ *large enough so that* $\mathcal{M}_{\mathsf{GMF}}(\mathbf{G}) = \mathbf{AG} + \mathbf{BZ} = \mathbf{B}\left(\mathbf{CG} + \mathbf{Z}\right)$ *satisfies* $(\alpha, \varepsilon(\alpha))$*-DP (or* ρ*-zCDP or* (ε, δ)*-approximate DP) in the non-adaptive continual release model. Then,* $\mathcal{M}$ *satisfies the same DP guarantee (with the same parameters) even when the rows of the inputs are chosen adaptively.*

3.3. DME and Matrix Mechanisms

Finally, we consider DME under the streaming DP model. In each round t, the server selects a batch of clients $B_t \in [N]$ and computes the empirical mean of their local vectors $\boldsymbol{g}^{(t)} = \sum_{i\in B_t} \boldsymbol{g}_i$. Note that $\boldsymbol{g}_i$ can depend on previous outputs $\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)}$. Our scheme assumes single-participation-per-epoch, that is, B_t disjoint with $B_{t'}$.

The goal of matrix mechanisms is to continually release a differentially private version of $\mathbf{AG}$ while minimizing the overall MSE: $\left\|\widehat{\mathbf{AG}} - \mathbf{AG}\right\|_F^2$. Here, $\mathbf{A} \in \mathbb{R}^{T\times T}$ must be a lower triangular matrix in order to ensure causality. In online optimization, the matrix $\mathbf{A}$ is determined by update rules. For instance, in simple SGD with a fixed step size $\eta > 0$, the model is updated as follows:

$$\boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} - \eta\boldsymbol{g}^{(t)} = \boldsymbol{w}^{(0)} - \eta\sum_{\tau=1}^{t}\boldsymbol{g}^{(\tau)},$$

resulting in the corresponding $\mathbf{A}$ being the prefix-sum matrix satisfying $[\mathbf{A}]_{t,t'} = \mathbb{1}_{\{t\leq t'\}}$. In general, one can leverage the matrix mechanism within the DP-FTRL framework (Kairouz et al., 2021b, Algorithm 1) and further incorporate momentum (Denisov et al., 2022).

To ensure privacy, instead of directly privatizing data matrix $\mathbf{G}$ (which results in a DP-SGD type scheme), we leverage the factorization $\mathbf{A} = \mathbf{BC}$ for $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{T\times T}$. If $(\alpha, \varepsilon(\alpha))$-DP is preserved for $\mathbf{CG} + \mathbf{Z}$, then the same level of DP holds for $\mathbf{AG} + \mathbf{BZ}$ as well. Notbaly, in the online optimization setting, local vectors $\boldsymbol{g}^{(t)}$ are *adaptively* generated and depend on $(\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)})$. Denisov et al. (2022, Theorem 2.1) shows that for Gaussian mechanism (i.e., $[\mathbf{Z}]_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for some $\sigma^2 > 0$), the non-adaptive DP guarantee (meaning that $\mathbf{G}$ is independent with the previous private outputs $\mathbf{O}$) implies the same level of adaptive DP.

To optimize the error, Li et al. (2015); Yuan et al. (2016); Denisov et al. (2022) formulate the factorization $\mathbf{A} = \mathbf{BC}$ as a convex optimization problem :

$$\min_{\mathbf{B}:\mathbf{A}=\mathbf{BC},\, \Delta(\mathbf{C})=1} \left\|\mathbf{B}\right\|_F^2, \tag{1}$$

where $\Delta\left(\mathbf{C}\right) \triangleq \max_{t\in[T]} \left\|\mathbf{C}_{[:,t]}\right\|_2^2$ is the sensitivity of $\mathbf{C}$. In this work, while we plug in the optimal factorization in our scheme (specifically solved via the fixed point method in Denisov et al. (2022)), our results hold for general factorization.

Our objective is to devise a local compression mechanism satisfying two criteria:

- $\widehat{\mathbf{AG}}$ satisfies adaptive streaming DP;
- $\widehat{\mathbf{AG}}$ is a function of locally compressed vectors $\mathcal{E}_i\left(\boldsymbol{g}_i\right)$ that can be described in b bits.

Remark 3.4. *In the streaming scenario, the cohort size* $|B_t|$ *solely impacts the sensitivity of the mean function each round. For simplicity in privacy analysis, we assume* $|B_t| = 1$ *(non-batched SGD). Nevertheless, our results extend to general batch sizes, as outlined in the main theorems.*

Notation. In the non-streaming setting, we employ $\boldsymbol{g}_i$ (or $\boldsymbol{h}_i$) to represent the local (row) vector at client i. In the streaming scenario, $\boldsymbol{g}^{(t)}$ (or $\boldsymbol{h}^{(t)}$) denotes the averaged row vectors of clients at round t. Matrices are denoted by capital bold-faced symbols; for instance, $\mathbf{G} \in \mathbb{R}^{T\times d}$ represents the matrix form of the stream $(\boldsymbol{g}^{(1)}, ..., \boldsymbol{g}^{(T)})$, where the t-th row of $\mathbf{G}$ is $\boldsymbol{g}^{(t)}$. When the context is clear, we may use $\mathbf{G}$ to refer to the stream itself. Additionally, we use $\boldsymbol{g}_j^{(t)}$ or $\mathbf{G}_{t,j}$ interchangeably to indicate the (t, j)-th entry of $\mathbf{G}$, with $t \in [T]$ and $j \in [d]$[3].

4. Differentially Private $\mathbf{L_2}$ Mean Estimation

In this section, we consider the non-streaming DME problem described in Section 3.1. To reduce communication costs under central DP, previous work of Chen et al. (2023) proposes a coordinate-subsampled Gaussian mechanism (CSGM), which random sparsifies each local vector in a coordinate-wise fashion, followed by server aggregation and the addition of Gaussian noise. While aligning with several gradient compression techniques, CSGM significantly enhances privacy guarantees by incorporating the randomness introduced in the sparsification phase into privacy analysis.

However, a notable drawback in Chen et al. (2023) emerges within the L_∞ geometry assumption that requires $\|\boldsymbol{g}_i\|_\infty \leq \Delta_\infty$. It is crucial to note that, in general, the L_∞ assumption is weaker than the L_2 assumption described in Section 3.1. To extend to the L_2 scenario, Chen et al. (2023) employs random rotation (or Kashin's representation) and L_∞ clipping to pre-process local vectors. This approach, however, results in larger Mean Squared Errors (MSEs) compared to the uncompressed Gaussian mechanism under equivalent Differential Privacy (DP) guarantees.

Algorithm 1 L_2-CSGM

```
Input: users' data $\boldsymbol{g}_1, ..., \boldsymbol{g}_n$, sampling parameters $\gamma \triangleq b/d$, DP parameters $(\alpha, \varepsilon(\alpha))$.
for user $i \in [n]$ do
    Draw $\boldsymbol{s}_i \overset{\text{i.i.d.}}{\sim} \mathsf{Bern}^{\otimes d}(\gamma)$ via shared randomness.
    Compress and send $\boldsymbol{g}_i \odot \boldsymbol{s}_i$ to the server (where $\odot$ denotes the entry-wise product).
end for
Server computes the noisy mean
    $\hat{\mu}_{\mathsf{CSGM}}(\boldsymbol{g}^n; \boldsymbol{s}^n, Z) \triangleq \frac{1}{n\gamma}\left(\sum_{i=1}^n \boldsymbol{g}_i \odot \boldsymbol{s}_i + Z\right), \quad (2)$
where $Z \sim \mathcal{N}\left(0, \sigma^2 \mathbb{I}_d\right)$ and $\sigma^2$ is computed according to (3) in Theorem 4.1.
Return: $\hat{\mu}_{\mathsf{CSGM}}$.
```

To address the geometry issue in Chen et al. (2023), we introduce a slight modification (see Algorithm 1) to the CSGM scheme and present an enhanced analysis of its Rényi DP profile, yielding a significantly improved guarantee. To differentiate between the two schemes, we term our proposed version as L_2-CSGM, while the original one in Chen et al. (2023) is referred to as L_∞-CSGM. Our main result in this section is the following privacy upper bound for the L_2-CSGM mean estimation scheme.

Theorem 4.1. *Let $\boldsymbol{g}_1, ..., \boldsymbol{g}_n \in \mathbb{S}^{d-1}(\Delta_2)$ (i.e., $\|\boldsymbol{g}_i\|_2 \leq \Delta_2$), and $\|\boldsymbol{g}_i\|_\infty \leq \Delta_\infty$ for all $i \in [n]$. Let $\hat{\mu}_{\mathsf{CSGM}}(\boldsymbol{g}^n)$ be defined as in (2) with $\boldsymbol{s}_1, ..., \boldsymbol{s}_n \overset{i.i.d.}{\sim} \mathsf{Bern}(\gamma)^{\otimes d}$, and $Z \sim \mathcal{N}\left(0, \sigma^2 \mathbb{I}_d\right)$. Then $\hat{\mu}_{\mathsf{CSGM}}$ satisfies $(\alpha, \varepsilon(\alpha))$-Rényi DP, for all integer α and*

$$\varepsilon(\alpha) \geq \frac{\Delta_2^2/\Delta_\infty^2}{\alpha - 1} \log\Big((1-\gamma)^{\alpha-1}(\gamma(\alpha-1)+1) + \sum_{\ell=1}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell e^{(\ell-1)\ell\frac{\Delta_\infty^2}{2\sigma^2}}\Big). \quad (3)$$

While L_2-CSGM also employs L_∞ clipping, we do not account for privacy budgets directly based on the L_∞ clipping norm Δ_∞ (which is the case in L_∞-CSGM). Instead, we consider both Δ_2 and Δ_∞, with L_∞ serving to "mitigate" the regime on which the privacy amplification lemma operates. In L_2-CSGM, the L_∞ clipping norm only influences higher-order terms in the final guarantees, and a slight increase in Δ_∞ does not alter the privacy guarantee asymptotically with increasing dimension d. In the subsequent subsection, we demonstrate that, for any $\alpha > 0$ and under the same MSE constraint, the Rényi DP guarantee of L_2-CSGM converges to that of the (uncompressed) Gaussian mechanism as $d \to \infty$.

4.1. Compared to the Gaussian Mechanism

We begin with the following lemma that computes the MSE of $\hat{\mu}_{\mathsf{CSGM}}$.

Corollary 4.2. *Under the hypotheses of Theorem 4.1, let $\hat{\mu}_{\mathsf{CSGM}}$ be defined as in Algorithm 1. Then the MSE of $\hat{\mu}_{\mathsf{CSGM}}$ is bounded by*

$$\mathsf{MSE}(\hat{\mu}_{\mathsf{CSGM}}) \triangleq \mathbb{E}\left[\|\hat{\mu}_{\mathsf{CSGM}} - \mu\|_2^2\right] \leq \frac{\sigma^2}{n^2\gamma^2} + \frac{\Delta_2^2}{n\gamma}.$$

On the other hand, the MSE of the (uncompressed) Gaussian mechanism $\hat{\mu}_{\mathsf{GM}}$ is $\mathsf{MSE}(\hat{\mu}_{\mathsf{GM}}) = \sigma^2/n^2$. It can be shown that under the same MSE constraints, the Renyi DP of L_2-CSGM converges to that of the Gaussian mechanism in the following sense:

Lemma 4.3. *For any fixed sparsification rate γ and Renyi DP order α, let σ_{GM}^2 and σ_{CSGM}^2 be chosen such that $\mathsf{MSE}(\hat{\mu}_{\mathsf{GM}}) = \mathsf{MSE}(\hat{\mu}_{\mathsf{CSGM}})$, i.e., $\sigma_{\mathsf{GM}}^2 = \frac{\sigma_{\mathsf{CSGM}}^2}{\gamma^2} + \frac{n\Delta_2^2}{\gamma}$. Then, it holds that $\varepsilon_{\mathsf{CSGM}}(\alpha) \to \varepsilon_{\mathsf{GM}}(\alpha)$ as $\Delta_\infty^2/\Delta_2^2 \to 0$, where $\varepsilon_{\mathsf{GM}}(\alpha) = \Delta_2^2\alpha/\sigma^2$ is the Rényi DP bound of the Gaussian mechanism, and $\varepsilon_{\mathsf{CSGM}}(\alpha)$ is defined in (3).*

It is worth noting that, in general, the Δ_∞/Δ_2 ratio decreases rapidly as d increases, leading to $\varepsilon_{\mathsf{CSGM}}(\alpha) \to \varepsilon_{\mathsf{GM}}(\alpha)$ as $d \to \infty$. For instance, by utilizing random rotation for preprocessing local vectors, with high probability, $\Delta_\infty/\Delta_2 = O\left(\frac{\log d}{\sqrt{d}}\right)$. If further employing Kashin's representation (Lyubarskii & Vershynin, 2010), then $\Delta_\infty/\Delta_2 = O(1/\sqrt{d})$ with probability 1.

[3] In general, we use $t \in [T]$ as the time index, $j \in [d]$ as the coordinate (spatial) index, and $i \in [n]$ as the client index for subscripts and superscripts.

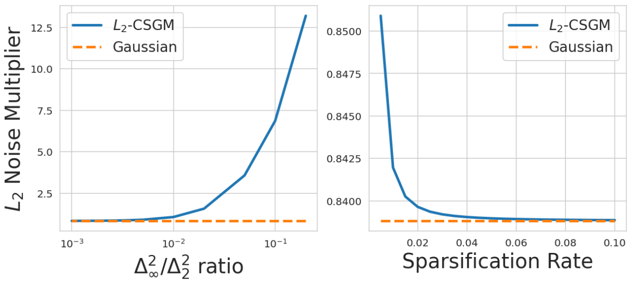

Figure 1. Noise multipliers (defined as σ/Δ_2) of CSGM and GM with $\varepsilon = 5.0$, $\delta = 10^{-8}$ and $\gamma = 0.01$. On the left, we fix the sparsification rate $\gamma = 0.01$. The numerical result indicates that as the ratio decreases, the noise multiplier of CSGM converges to that of the GM. Equivalently, this implies that $\varepsilon_{\mathsf{CSGM}}(\alpha) \to \varepsilon_{\mathsf{GM}}(\alpha)$ if one fixes the MSEs of both schemes. On the right, we fix the Δ_2/Δ_∞ ratio to be 1000 and plot the noise multipliers.

On the other hand, if we calibrate the noise based on Δ_∞ as in L_∞-CGSM, the constant in Rényi DP will not match that of the uncompressed Gaussian mechanism, which we elaborate on in the next subsection.

4.2. Compared to L_∞-CSGM (Chen et al., 2023).

To compare the L_2 and L_∞-CSGM, first observe that the Rényi DP bound in (3) can be expressed as

$$\varepsilon_{\mathsf{CGSM,L_2}}(\alpha) = \Delta_2^2/\Delta_\infty^2 \cdot D_\alpha\left(\Delta_\infty S + Z \| Z\right),$$

where $Z \sim \mathcal{N}(0, \sigma^2)$ and $S \sim \mathsf{Bern}(\gamma)$. On the other hand, the Rényi DP bound of L_∞-CSGM in Chen et al. (2023) is

$$\varepsilon_{\mathsf{CGSM,L_\infty}}(\alpha) = d \cdot D_\alpha\left(\Delta_\infty S + Z \| Z\right).$$

As a result, the ratio between two Rényi DP bounds is $\frac{d\Delta_\infty^2}{\Delta_2^2} > 1$ (because $\|\boldsymbol{g}\|_\infty \leq \Delta_\infty$ implies $\|\boldsymbol{g}\|_2^2 \leq d\Delta_\infty^2$). When employing random rotation, this ratio is $O(\log(d))$ with high probability; with Kashin's representation, this ratio remains constant, but the constant is non-negligible (for instance, in Chen et al. (2020), the constant is set to be around 2). The sub-optimality gap between L_∞-CSGM and the (uncompressed) Gaussian mechanism makes it undesirable in practical FL tasks, emphasizing the necessity of L_2-CSGM.

5. Matrix Factorization Mechanism with Local Sparsification under Streaming DP

Moving on, we delve into the streaming DP setting, specifically focusing on the matrix mechanism detailed in Section 3.2 and Section 3.3.

In the context of matrix mechanisms, the objective is to continually release a DP version of $\mathbf{AG}$, where each row of $\mathbf{G}$ may depend on previous outputs $\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)}$. To minimize the overall MSE, $\left\|\widehat{\mathbf{AG}} - \mathbf{AG}\right\|_F^2$, we factorize $\mathbf{A}$ into $\mathbf{BC}$ and designing DP mechanisms according to $\mathbf{CG}$, as discussed in Section 3.3. Notably, our scheme adopts the optimal factorization for the prefix sum matrix, addressing the optimization problem (1).

We aim to devise a matrix factorization scheme that simultaneously compresses local gradients $\mathbf{G}$. In this approach, instead of transmitting $\mathbf{G}$ to the server, clients send $\mathsf{compress}(\mathbf{G})$, with compression applied row-wise (i.e., client-wise). A tempting strategy is to employ the local sparsification technique in CSGM and enhance privacy using Theorem 4.1[4]: $\mathcal{M}_{\mathsf{SGMF}}(\mathbf{G}) \triangleq \mathbf{A}\left(\mathbf{S} \odot \mathbf{G}\right) + \mathbf{BZ}$, where $\mathbf{A} = \mathbf{BC}$ is a factorization, $[\mathbf{Z}]_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ and $[\mathbf{S}]_{t,j} \overset{\text{i.i.d.}}{\sim} \mathsf{Bern}(\gamma)$ for $t \in [T]$ and $j \in [d]$. However, the privacy analysis encounters two challenges:

- In matrix mechanisms, a local vector $\boldsymbol{g}^{(t)}$ may persist across all T rounds. Consequently, the randomness introduced in local sparsification steps at the t-th round might affect other rounds, resulting in what we term as temporal coupling. Unlike in Denisov et al. (2022), where the temporal coupling of isotropic Gaussian noise can be circumvented due to rotational invariance, local sparsification or sampling breaks this invariance, rendering Theorem 2.1 of Denisov et al. (2022) inapplicable.
- In the streaming scenario, the sampling variable $\boldsymbol{s}_j^{(t)}$ for the j-th coordinate in the t-round may influence the j'-th coordinate later due to adaptivity. For instance, $\boldsymbol{g}^{(t+1)}$ can depend on the t-th output $\boldsymbol{o}^{(t)}$, which, in turn, is a function of $\boldsymbol{s}_{j'}^{(t)}$ for all $j' \in [d]$. This introduces "spatial correlation," which does not appear in the non-streaming setting (e.g., Theorem 4.1).

Algorithm 2 Sparsified Gaussian Matrix Factorization

```
Input: Local vectors $\mathbf{g}^{(1)}, ..., \boldsymbol{g}^{(T)}$, noise scale $\sigma$, sparsification rate $\gamma$, factorization $\mathbf{A} = \mathbf{B} \cdot \mathbf{C}$.
//See Alg. 3 for cohort_size > 1.
for Each client $t$ at time $t$ do
    Generate $d$ independent binary masks $\mathbf{s}^{(t)} \in \{0,1\}^d$: for any $j \in [d]$, $\mathbf{s}_j^{(t)} \overset{\text{i.i.d.}}{\sim} \mathsf{Ber}(\gamma)$;
    Compute $\tilde{\mathbf{g}}^{(t)} = \boldsymbol{g}^{(t)} \odot \mathbf{s}^{(t)}$ and sends it to the server;
end for
Server samples Gaussian noise: $[\mathbf{Z}]_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $t \in [T]$ and $j \in [d]$.
Server computes the noisy outputs: $\mathbf{A} \cdot \tilde{\mathbf{G}} + \mathbf{B} \cdot \mathbf{Z}$.
```

In this section, we demonstrate that despite both temporal and spatial couplings, we can still achieve the same "am-

[4] Throughout this section, we assume a cohort size of 1 for simplicity. Our results naturally extend to general scenarios, and the full scheme is presented in Algorithm 3 in Appendix A, in which each client adopts an independent sampling mask.

plification effect" as in Theorem 4.1. Our primary result is the Rényi Differential Privacy (DP) bound for the sparsified Gaussian matrix factorization outlined in Algorithm 2 (which can be seen as a direct extension of L_2-CSGM to the streaming DP setting).

Theorem 5.1. *Let $\mathbf{A} \in \mathbb{R}^{T \times T}$ be a lower-triangular full-rank query matrix, and let $\mathbf{A} = \mathbf{BC}$ be any factorization for some $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{T \times T}$, with $\Delta(\mathbf{C}) = \max_{t \in [T]} \left\| \boldsymbol{c}^{(t)} \right\|_2$. Let $\mathbf{G}$ be the data matrix and Δ_2 and Δ_∞ be the L_2 and L_∞ norm bounds of $\mathbf{G}$, i.e., $\left\| \boldsymbol{g}^{(t)} \right\|_2 \leq \Delta_2$ and $\left\| \boldsymbol{g}^{(t)} \right\|_\infty \leq \Delta_\infty$ (recall that $\boldsymbol{g}^{(t)}$ denotes the t-th row of $\mathbf{G}$). Then, the $\mathcal{M}_{\mathsf{SGMF}}$ in Algorithm 2 satisfies adaptive $(\alpha, \varepsilon(\alpha))$-Rényi DP for any $\alpha \geq 1$ and*

$$\varepsilon\left(\alpha\right) \geq \frac{\kappa_2^2/\kappa_\infty^2}{\alpha - 1} \log\Bigg(\left(1-\gamma\right)^{\alpha-1}\left(\gamma(\alpha-1)+1\right) + \sum_{\ell=1}^{\alpha} \binom{\alpha}{\ell} \left(1-\gamma\right)^{\alpha-\ell} \gamma^{\ell} e^{(\ell-1)\ell\frac{\kappa_\infty^2}{2\sigma^2}} \Bigg). \tag{4}$$

where $\kappa_2 = \Delta(\mathbf{C}) \cdot \Delta_2$ and $\kappa_\infty = \Delta(\mathbf{C}) \cdot \Delta_\infty$ are the L_2 and L_∞ sensitivities.

A couple of remarks follow. Firstly, the class of matrix mechanisms encompasses tree-based methods as a special case, such as online or full-honaker tree aggregation (Honaker, 2015) used in Kairouz et al. (2021b). Therefore, Theorem 5.1 also applies to these results. Second, while Choquette-Choo et al. (2023b) also investigate privacy amplification through subsampling, their subsampling is conducted client-wise rather than coordinate-wise, as their scheme does not aim for compression. Consequently, Choquette-Choo et al. (2023b) do not encounter the spatial coupling issue. Finally, our scheme assumes single participation per epoch, and in practice, this can be done by shuffling and restarting the mechanism each epoch, similar to the TreeRestart approach in Kairouz et al. (2021b).

5.1. Proof of Theorem 5.1

Next, we prove Theorem 5.1. The proof begins with the LQ decomposition trick in Denisov et al. (2022), followed by a careful decoupling of the joint distribution on $\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t)}$.

Reparameterization. Let $\mathbf{B} = \mathbf{L} \cdot \mathbf{Q}$ be the LQ decomposition of the matrix $\mathbf{B}$. Consider a different lower-triangular factorization: $\tilde{\mathcal{M}}\left(\mathbf{G}\right) = \mathbf{L}\left(\mathbf{QC}\left(\mathbf{G} \odot \mathbf{S}\right) + \mathbf{Z}\right)$, where $\mathbf{Z}_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$, $\mathbf{Q}$ is orthonormal, and both $\mathbf{L}$ and $\mathbf{QC}$ are lower-triangular. Since $\mathbf{QC}$ is lower-triangular, $\mathbf{QC}\left(\mathbf{G} \odot \mathbf{S}\right) + \mathbf{Z}$ can operate in the continuous release model, as row t of $\mathbf{QC}(\mathbf{G} \odot \mathbf{S})$ depends only on the first t rows of $\mathbf{G}$. Following from the same argument in Denisov et al. (2022, Theorem 2.1), it suffices to show the desired DP guarantee (4) on $\mathbf{QC}\left(\mathbf{G} \odot \mathbf{S}\right) + \mathbf{Z}$ since we can always replace $\mathbf{Z}$ with $\mathbf{QZ}$ due to the rotational invariance of isotropic Gaussian distribution. For notational convenience, we denote $\mathbf{QC} \triangleq \mathbf{M}$ in the remaining proof (note that $\mathbf{M}$ is lower triangular).

Joint density of the transcript. Next, we show the mechanism $\mathbf{M}\left(\mathbf{G} \odot \mathbf{S}\right) + \mathbf{Z}$ is an instance of the standard (subsampled) Gaussian mechanism for computing an adaptive function in the continuous release model with a guaranteed bound on the global L_2 and L_∞ sensitivities. Let $\mathbf{G}$ and $\mathbf{H}$ be any two neighboring data streams (defined in Definition 3.2) that additionally satisfy the following L_∞ condition: $\max_{t \in [T],\, j \in [d]} \left| \mathbf{g}_j^{(t)} - \mathbf{h}_j^{(t)} \right| \leq \Delta_\infty$. Without loss of generality, we assume that $\mathbf{G}$ and $\mathbf{H}$ differ at $t = 1$, and thus when analyzing the privacy guarantees, we condition on the realization $(\mathbf{s}^{(2)}, ..., \mathbf{s}^{(T)}) = (\check{\mathbf{s}}^{(2)}, ..., \check{\mathbf{s}}^{(T)})$ and all the potential randomness used in the optimization algorithm, treating them as deterministic. The only randomness that will be accounted for in the privacy analysis is $\mathbf{s}^{(1)}$ and $\mathbf{Z}$.

Given the data stream $\mathbf{G}$, the output transcript $\mathbf{O} \triangleq \left(\boldsymbol{o}^{(1)}, \boldsymbol{o}^{(2)}, ..., \boldsymbol{o}^{(T)}\right) \in \mathbb{R}^{d \times T}$ is computed as follows:

$$\boldsymbol{o}^{(1)} = \underbrace{\mathbf{M}_{11}\left(\boldsymbol{g}^{(1)} \odot \mathbf{s}^{(1)}\right) + \mathbf{Z}^{(1)}}_{\triangleq \mathbf{p}^{(1)}};$$

$$\boldsymbol{o}^{(t)} = \underbrace{\mathbf{M}_{t1}\left(\boldsymbol{g}^{(1)} \odot \mathbf{s}^{(1)}\right) + \mathbf{Z}^{(t)}}_{\triangleq \boldsymbol{p}^{(t)}} + \underbrace{\sum_{\tau=2}^{t} \mathbf{M}_{t\tau}\left(\boldsymbol{g}^{(\tau)} \odot \check{\mathbf{s}}^{(\tau)}\right)}_{\triangleq \boldsymbol{q}^{(t)}},$$

for all $t \geq 1$. Our goal is to control $D_\alpha\left(P_{\mathbf{O}|\mathbf{G}} \middle\| P_{\mathbf{O}|\mathbf{H}}\right)$, where $P_{\mathbf{O}|\mathbf{G}}$ denotes the distribution of transcript $\mathbf{O}$ under data stream $\mathbf{G}$ and $P_{\mathbf{O}|\mathbf{H}}$ denotes the distribution of $\mathbf{O}$ under $\mathbf{H}$. Note that the randomness used to compute the above divergence only includes $\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(T)}$ and $\mathbf{s}^{(1)}$, as we have conditioned on all other (irrelevant) external randomness, including $\check{\mathbf{s}}^{(2)}, ..., \check{\mathbf{s}}^{(T)}$.

Decoupling the joint distribution. The main challenge here, compared to the uncompressed Gaussian mechanism in Denisov et al. (2022), is the spatial and temporal coupling on the joint distribution $P_{\mathbf{O}|\mathbf{G}}$. To see this, observe that $\boldsymbol{o}_{j'}^{(t)}$ implicitly depends on the j-th sampling variable $\mathbf{s}_j^{(1)}$ through $\boldsymbol{g}^{(2)}, ..., \boldsymbol{g}^{(t-1)}$ (which are functions of $\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)}$). As a result, the joint distribution of $\mathbf{O}$ is a mixture of product distributions, so the scheme cannot be reduced into a simple subsampled Gaussian mechanism.

To address this issue, we introduce the following decomposition trick on the transcript $\boldsymbol{o}^{(t)}$ to decouple the complicated spatial and temporal correlation. For all $t \geq 1$, write $\boldsymbol{p}^{(t)} \triangleq \mathbf{M}_{t1}\left(\boldsymbol{g}^{(1)} \odot \mathbf{s}^{(1)}\right) + \mathbf{Z}^{(t)}$, $\boldsymbol{q}^{(1)} \triangleq 0$, and

$$\boldsymbol{q}^{(t)} \triangleq \sum_{\tau=2}^{t} \mathbf{M}_{t\tau}\left(\boldsymbol{g}^{(\tau)} \odot \check{\mathbf{s}}^{(\tau)}\right),$$

so that $\boldsymbol{o}^{(t)} = \boldsymbol{p}^{(t)} + \boldsymbol{q}^{(t)}$.

The key observation is that, conditioned on the realization $\check{\mathbf{s}}^{(2)}, ..., \check{\mathbf{s}}^{(T)}$, $\mathbf{Q} \triangleq (\boldsymbol{q}^{(1)}, ..., \boldsymbol{q}^{(T)})$ is a *deterministic* function of $\mathbf{P} \triangleq (\boldsymbol{p}^{(1)}, ..., \boldsymbol{p}^{(T)})$. To see this, note that

$$\begin{aligned}\boldsymbol{q}^{(t)} &= f(\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)}) \\ &= g\left((\boldsymbol{p}^{(1)}, \boldsymbol{q}^{(1)}), ..., (\boldsymbol{p}^{(t-1)}, \boldsymbol{q}^{(t-1)})\right)\end{aligned}$$

for some functions f and g. Also notice that $\boldsymbol{q}^{(1)} = 0$. Thus, by induction, $\boldsymbol{q}^{(t)}$ is a function of $\boldsymbol{p}^{(1)}, ..., \boldsymbol{p}^{(t-1)}$.

As a result, the overall transcript $\mathbf{O} = \mathbf{P} + \mathbf{Q}(\mathbf{P})$ can be viewed as a post-processing of $\mathbf{P}$, so by data processing inequality, it holds that

$$D_\alpha\left(P_{\mathbf{O}|\mathbf{G}} \middle\| P_{\mathbf{O}|\mathbf{H}}\right) \leq D_\alpha\left(P_{\mathbf{P}|\mathbf{G}} \middle\| P_{\mathbf{P}|\mathbf{H}}\right). \tag{5}$$

Since the $\mathbf{P} = (\boldsymbol{p}^{(1)}, ..., \boldsymbol{p}^{(T)})$ does not have spatial coupling, in the sense that $\boldsymbol{p}_j^{(t)}$ is independent of $\mathbf{s}_{j'}^{(1)}$ for all $t \in [T]$ and $j, j' \in [d]$, $j \neq j'$, we can invoke the argument of Denisov et al. (2022) along with privacy amplification by subsampling, summarized as in the following lemma.

Lemma 5.2. *Let* $\mathbf{P}$ *be defined as above. Then, it holds that*

$$\begin{aligned}D_\alpha\left(P_{\mathbf{P}|\mathbf{G}} \middle\| P_{\mathbf{P}|\mathbf{H}}\right) \leq \frac{\kappa_2^2/\kappa_\infty^2}{\alpha - 1} \log\Big((1-\gamma)^{\alpha-1} \cdot \\ (\gamma(\alpha-1)+1) + \sum_{\ell=1}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell e^{(\ell-1)\ell\frac{\kappa_\infty^2}{2\sigma^2}}\Big).\end{aligned}$$

We remark that (5) implies that among all possible adaptive dependencies of $\boldsymbol{g}^{(t)}(\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)})$, the transcript $\mathbf{O}$ is statistically dominated by the independent one, that is, $\boldsymbol{g}^{(t)}$ remains constant regardless of previous outputs $(\boldsymbol{o}^{(1)}, ..., \boldsymbol{o}^{(t-1)})$. □

6. Empirical Evaluation

We provide empirical evaluations on the privacy-utility trade-offs for both DP-SGD (under a non-streaming setting) and DP-FTRL type (with matrix mechanisms (Denisov et al., 2022)) algorithms. We mainly compare the L_2-CGSM (Algorithm 2) and sparsified Gaussian matrix factorization (Algorithm 2) with the uncompressed Gaussian mechanism (Balle & Wang, 2018). We convert the Rényi DP bounds to (ε, δ)-DP via the conversion lemma from Canonne et al. (2020) for a fair comparison.

Datasets and models. We run experiments on the full Federated EMNIST (Cohen et al., 2017) and Stack Overflow (Authors., 2019) dataset. F-EMNIST has 62 classes and $N = 3400$ clients with a total of $671,585$ training samples. Inputs are single-channel $(28, 28)$ images. The Stack Overflow (SO) dataset is a large-scale text dataset based on responses to questions asked on the site Stack Overflow. There are over 10^8 data samples unevenly distributed across $N = 342,477$ clients. We focus on the next word prediction (NWP) task: given a sequence of words, predict the next words in the sequence.

On F-EMNIST, we experiment with a (4 layer) Convolutional Neural Network (CNN) used by Kairouz et al. (2021a) (with around 1 million parameters). On SONWP, we experiment with a 4 million parameters (4 layer) long-short term memory (LSTM) model – the same as prior work Andrew et al. (2021); Kairouz et al. (2021a). In both cases, clients train for 1 local epoch using SGD. Only the server uses momentum.

Additionally, for local model updates, we perform random rotation and L_∞-clipping, with $\Delta_\infty = \Delta_2\sqrt{2\log(d \cdot n)/d}$, where d is the model dimension (i.e., # trainable parameters) and n is the cohort size in each training round.

L_2-CSGM for DP-SGD. In Figure 2, we report the accuracy of L_2-CSGM (Algorithm 1) as well as the uncompressed Gaussian mechanism.

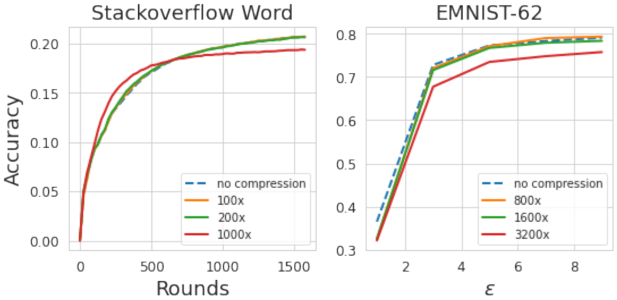

Figure 2. Accuracy of GM and CSGM, with $\delta = 10^{-5}$ for F-EMNIST and $\delta = 10^{-6}$ for SONWP. The resulting Δ_∞/Δ_2 value is $6.4 \cdot 10^{-3}$ for F-EMNIST and $3.3 \cdot 10^{-3}$ for SONWP.

Sparsified Gaussian Matrix Mechanism for DP-FTRL. In Figure 3, we report the accuracy of SGMF (Algorithm 1) and the uncompressed matrix mechanism. We use the same optimal factorization as in Denisov et al. (2022) with $T = 32$ for 16 epochs, and we restart the mechanism and shuffling clients every epoch as in the TreeRestart approach in Kairouz et al. (2021b). We observe that for the matrix mechanism, the compression rates are, in general, less than DP-FedAvg, and in addition, the performance is more sensitive to server learning rates and L_2 clip norms.

7. Conclusion

Our work addresses challenges in L_2 mean estimation under central DP and communication constraints. We introduce a novel L_2 Rényi DP accounting algorithm for the sparsified Gaussian mechanism that significantly improves upon previous ones based on L_∞ sensitivity. We also extend the scheme and accountant to the streaming setting, providing

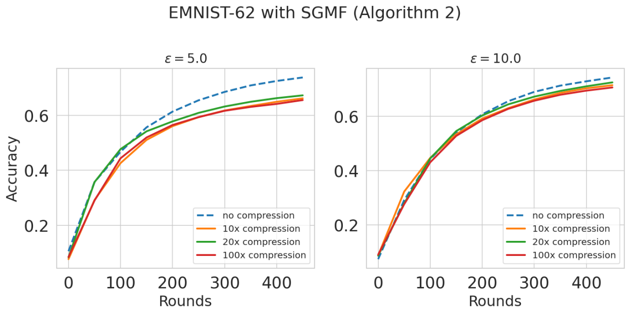

Figure 3. Accuracy of MF and SGMF, with $\delta = 10^{-5}$, cohort size $n = 100$, clipped norm $\Delta_2 = 1.0$, and server learning rate 0.1.

an adaptive DP bound that handles spatial and temporal couplings of privacy loss unique to adaptive settings. Empirical evaluations on diverse federated learning tasks showcase a 100x enhancement in compression. Notably, our scheme focuses on reducing the dimensionality of local model updates, and hence it can potentially be combined with other gradient quantization or compression techniques, thereby promising heightened compression efficiency.

Impact Statement

This paper introduces a new privacy accounting method for the sparsified Gaussian mechanism which improves the communication-privacy-utility trade-offs in federated learning. By contributing to the decentralized, accessible, and trustworthy machine learning efforts, we expect that this work will have only positive impact on the society.

Acknowledgements

This work was done during Wei-Ning's internship at Google. Wei-Ning is supported in part by NSF Award #CCF-2213223. Berivan is supported by a Google PhD Fellowship. Albert is supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program #IITP-2024-RS-2022-00156295.

References

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

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

Agarwal, N., Kairouz, P., and Liu, Z. The Skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34: 5052–5064, 2021.

Ailon, N. and Chazelle, B. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 557–563, 2006.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems 30*, pp. 1709–1720, 2017.

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34: 17455–17466, 2021.

Asi, H., Feldman, V., and Talwar, K. Optimal algorithms for mean estimation under local differential privacy. In *International Conference on Machine Learning*, pp. 1046–1056. PMLR, 2022.

Asi, H., Feldman, V., Nelson, J., Nguyen, H., and Talwar, K. Fast optimal locally private mean estimation via random projections. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=K3JgUvDSYX.

Authors., T. T. F. Tensorflow federated stack overflow dataset, 2019.

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

Balle, B., Barthe, G., and Gaboardi, M. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

Barnes, L. P., Inan, H. A., Isik, B., and Özgür, A. rtop-k: A statistical estimation approach to distributed sgd. *IEEE Journal on Selected Areas in Information Theory*, 1(3): 897–907, 2020.

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1253–1269, 2020.

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

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Braverman, M., Garg, A., Ma, T., Nguyen, H. L., and Woodruff, D. P. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM Symposium on Theory of Computing*, pp. 1011–1020, 2016.

Canonne, C. L., Kamath, G., and Steinke, T. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Chan, T. H. H., Li, M., Shi, E., and Xu, W. Differentially private continual monitoring of heavy hitters from distributed streams. In *Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings 12*, pp. 140–159. Springer, 2012.

Chen, W.-N., Kairouz, P., and Ozgur, A. Breaking the communication-privacy-accuracy trilemma. *Advances in Neural Information Processing Systems*, 33, 2020.

Chen, W.-N., Choo, C. A. C., Kairouz, P., and Suresh, A. T. The fundamental price of secure aggregation in differentially private federated learning. In *International Conference on Machine Learning*, pp. 3056–3089. PMLR, 2022a.

Chen, W.-N., Ozgur, A., and Kairouz, P. The Poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 3490–3506. PMLR, 2022b.

Chen, W.-N., Song, D., Ozgur, A., and Kairouz, P. Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=izNfcaHJk0.

Choquette-Choo, C. A., McMahan, H. B., Rush, K., and Thakurta, A. Multi-epoch matrix factorization mechanisms for private machine learning. *arXiv preprint arXiv:2211.06530*, 2022.

Choquette-Choo, C. A., Dvijotham, K., Pillutla, K., Ganesh, A., Steinke, T., and Thakurta, A. Correlated noise provably beats independent noise for differentially private learning. *arXiv preprint arXiv:2310.06771*, 2023a.

Choquette-Choo, C. A., Ganesh, A., Steinke, T., and Thakurta, A. Privacy amplification for matrix mechanisms. *arXiv preprint arXiv:2310.15526*, 2023b.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Denisov, S., McMahan, H. B., Rush, J., Smith, A., and Guha Thakurta, A. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *Advances in Neural Information Processing Systems*, 35: 5910–5924, 2022.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 715–724, 2010.

Edmonds, A., Nikolov, A., and Ullman, J. The power of factorization mechanisms in local and central differential privacy. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 425–438, 2020.

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

Farokhi, F. Gradient sparsification can improve performance of differentially-private convex machine learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1695–1700. IEEE, 2021.

Feldman, V. and Talwar, K. Lossless compression of efficient private local randomizers. In *International Conference on Machine Learning*, pp. 3208–3219. PMLR, 2021.

Feldman, V., McMillan, A., and Talwar, K. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2022.

Feldman, V., McMillan, A., and Talwar, K. Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4966–4981. SIAM, 2023.

Gandikota, V., Kane, D., Maity, R. K., and Mazumdar, A. vqSGD: Vector quantized stochastic gradient descent, 2019.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Girgis, A., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2521–2529. PMLR, 2021.

Girgis, A. M. and Diggavi, S. Multi-message shuffled privacy in federated learning. *arXiv preprint arXiv:2302.11152*, 2023.

Guha Thakurta, A. and Smith, A. (nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013.

Hardt, M. and Talwar, K. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 705–714, 2010.

Henzinger, M., Upadhyay, J., and Upadhyay, S. Almost tight error bounds on differentially private continual counting. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 5003–5039. SIAM, 2023.

Henzinger, M., Upadhyay, J., and Upadhyay, S. A unifying framework for differentially private sums under continual observation. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 995–1018. SIAM, 2024.

Honaker, J. Efficient use of differentially private binary trees. *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2:26–27, 2015.

Hu, R., Gong, Y., and Guo, Y. Federated learning with sparsification-amplified privacy and adaptive optimization. In Zhou, Z.-H. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1463–1469. International Joint

Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/202. URL https://doi.org/10.24963/ijcai.2021/202. Main Track.

Isik, B., Weissman, T., and No, A. An information-theoretic justification for model pruning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3821–3846. PMLR, 2022.

Isik, B., Chen, W.-N., Ozgur, A., Weissman, T., and No, A. Exact optimality of communication-privacy-utility tradeoffs in distributed mean estimation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=7ETbK9lQd7.

Isik, B., Pase, F., Gunduz, D., Weissman, T., and Michele, Z. Sparse random networks for communication-efficient federated learning. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=k1FHgri5y3-.

Jain, P., Raskhodnikova, S., Sivakumar, S., and Smith, A. The price of differential privacy under continual observation. In *International Conference on Machine Learning*, pp. 14654–14678. PMLR, 2023.

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

Kairouz, P., Liu, Z., and Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 5201–5212. PMLR, 2021a.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021b.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021c.

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

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Li, C., Miklau, G., Hay, M., McGregor, A., and Rastogi, V. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB journal*, 24: 757–781, 2015.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, B. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SkhQHMW0W.

Lyubarskii, Y. and Vershynin, R. Uncertainty principles and vector quantization. *IEEE Transactions on Information Theory*, 56(7):3491–3501, 2010.

McKenna, R., Miklau, G., Hay, M., and Machanavajjhala, A. Optimizing error of high-dimensional statistical queries under differential privacy. *arXiv preprint arXiv:1808.03537*, 2018.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. Communication-efficient learning of deep networks from decentralized data (2016). *arXiv preprint arXiv:1602.05629*, 2016.

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

Mitchell, N., Ballé, J., Charles, Z., and Konečnỳ, J. Optimizing the communication-accuracy trade-off in federated learning with rate-distortion theory. *arXiv preprint arXiv:2201.02664*, 2022.

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., and Arora, R. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pp. 8253–8265. PMLR, 2020.

Sarlos, T. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pp. 143–152. IEEE, 2006.

Shah, A., Chen, W.-N., Balle, J., Kairouz, P., and Theis, L. Optimal compression of locally differentially private mechanisms. In *International Conference on Artificial Intelligence and Statistics*, pp. 7680–7723. PMLR, 2022.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3329–3337. JMLR.org, 2017.

Upadhyay, J. and Upadhyay, S. A framework for private matrix analysis in sliding window model. In *International Conference on Machine Learning*, pp. 10465–10475. PMLR, 2021.

Vargaftik, S., Ben-Basat, R., Portnoy, A., Mendelson, G., Ben-Itzhak, Y., and Mitzenmacher, M. Drive: One-bit distributed mean estimation. *Advances in Neural Information Processing Systems*, 34:362–377, 2021.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

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

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pp. 1299–1309, 2018.

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

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems*, pp. 1509–1519, 2017.

Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., and Zhang, Y. Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*, 2023.

Yuan, G., Yang, Y., Zhang, Z., and Hao, Z. Optimal linear aggregate query processing under approximate differential privacy. *CoRR, abs/1602.04302*, 2016.

Zhu, Y. and Wang, Y.-X. Poission subsampled Rényi differential privacy. In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.

A. Sparsified Gaussian Matrix Factorization for General Cohort Size

In this section, we present the full SGMF schemes with a general cohort size. Note that while we allow more than one client per FL round, each client only participates *once*.

Algorithm 3 Sparsified Gaussian Matrix Factorization with Full Cohort Size

```
Input: Local vectors $\mathbf{g}^{(1)}, ..., \boldsymbol{g}^{(T)}$, noise scale $\sigma$, sparsification rate $\gamma$, factorization $\mathbf{A} = \mathbf{B} \cdot \mathbf{C}$.
for Each cohort $\mathcal{B}_t$ at time $t$ do
  for Each client $i$ in cohort $\mathcal{B}_t$ do
    Generates an independent binary mask $\mathbf{s}^{(t,i)} \in \mathsf{Ber}(\gamma)^{\otimes d}$; Send $\tilde{\mathbf{g}}^{(t,i)} = \boldsymbol{g}^{(t,i)} \odot \mathbf{s}^{(t,i)}$ to the server;
  end for
end for
Server computes $\tilde{G} \in \mathbb{R}^{T \times d}$, where the $t$-th row is $\tilde{\mathbf{g}}^{(t)} = \frac{1}{\gamma|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}^{(t,i)}$;
Server samples Gaussian noise: $[\mathbf{Z}]_{t,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $t \in [T]$ and $j \in [d]$.
Server computes the noisy mean: $\mathbf{A} \cdot \tilde{\mathbf{G}} + \mathbf{B} \cdot \mathbf{Z}$.
```

B. Additional Details on Communication-Efficient DME with Local DP

An alternative method for achieving communication-efficient DME under central DP involves employing local DP mechanisms (Warner, 1965; Kasiviswanathan et al., 2011) with privacy amplification through shuffling (Erlingsson et al., 2019; Girgis et al., 2021; Feldman et al., 2022; 2023). It is worth noting that, under a $\varepsilon_{\mathsf{Local}}$-DP constraint, the optimal local randomizer is the privUnit mechanism (Bhowmick et al., 2018; Asi et al., 2022). This mechanism can be efficiently compressed using a pseudo-random generator (PRG) (Feldman & Talwar, 2021) or random projection (Asi et al., 2023) (without going through quantization or L_∞ clipping). Combining these local DP schemes with a multi-message shuffler has been proven to achieve order-optimal privacy-accuracy-utility trade-offs (Chen et al., 2023; Girgis & Diggavi, 2023), requiring less trust assumption on the server.

However, as pointed out in Chen et al. (2023), this local DP approach involves privacy amplification by shuffling lemmas that exhibit large leading constants compared to CGSM. Furthermore, privUnit is designed and optimized under *pure* DP, leaving its optimality under approximate or Rényi DP unclear. Additionally, to our best knowledge, there is currently no privacy amplification lemma known for transforming local Rényi DP into central Rényi DP. Hence, even if one adopts an optimal *local R'enyi* DP scheme and combines it with shuffling, it remains uncclear whether the resulting privacy guarantee is order-optimal. Lastly, Chen et al. (2023) empirically demonstrates a non-negligible gap in Mean Squared Errors (MSEs) between shuffling-based methods and L_∞-CGSM.

C. Additional Details for the Experiments

In this section, we provide additional details of the experiments. We mainly compare the L_2-CGSM (Algorithm 2) and sparsified Gaussian matrix factorization (Algorithm 2) with the uncompressed Gaussian mechanism (Balle & Wang, 2018). We convert the Rényi DP bounds to (ε, δ)-DP via the conversion lemma from Canonne et al. (2020) for a fair comparison.

Datasets and models. We run experiments on the full Federated EMNIST (Cohen et al., 2017) and Stack Overflow (Authors., 2019) dataset. F-EMNIST has 62 classes and $N = 3400$ clients with a total of $671,585$ training samples. Inputs are single-channel $(28, 28)$ images. The Stack Overflow (SO) dataset is a large-scale text dataset based on responses to questions asked on the site Stack Overflow. There are over 10^8 data samples unevenly distributed across $N = 342,477$ clients. We focus on the next word prediction (NWP) task: given a sequence of words, predict the next words in the sequence.

On F-EMNIST, we experiment with a (4 layer) Convolutional Neural Network (CNN), which is used by Kairouz et al. (2021a). The architecture is slightly smaller and has $d \leq 2^{20}$ parameters to reduce the zero padding required by the randomized Hadamard transform used for flattening and L_∞ clipping (see Algorithm 1). The requirement can be potentially removed if one uses a randomized Fourier transform instead. On SONWP, we experiment with a 4 million parameters (4 layer) long-short term memory (LSTM) model – the same architecture as prior work Andrew et al. (2021); Kairouz et al. (2021a). In both cases, clients train for 1 local epoch using SGD. Only the server uses momentum.

For each local model update, we perform random rotation (based on randomized Hadamard transform) and L_∞ clipping, with $\Delta_\infty = \Delta_2\sqrt{2\log(d \cdot n)/d}$, where d is the model dimension (i.e., # trainable parameters) and n is the cohort size in each training round.

L_2-CSGM for DP-SGD. In Figure 4 and Figure 5, we present the accuracy results of the L_2-CSGM algorithm (Algorithm 1) applied to F-EMNIST with varying cohort sizes, juxtaposed with the performance of the uncompressed Gaussian mechanism. Notably, our findings reveal that, on the whole, we can achieve compression exceeding 100x without a significant compromise in accuracy. Furthermore, as the cohort size n increases, the impact of compression on utility diminishes. This implies that greater compression is feasible with larger values of n. Similarly, in Figure 6, we delineate the accuracy outcomes for the Stack Overflow next-word prediction task across diverse ε values, maintaining a constant cohort size of 1000.

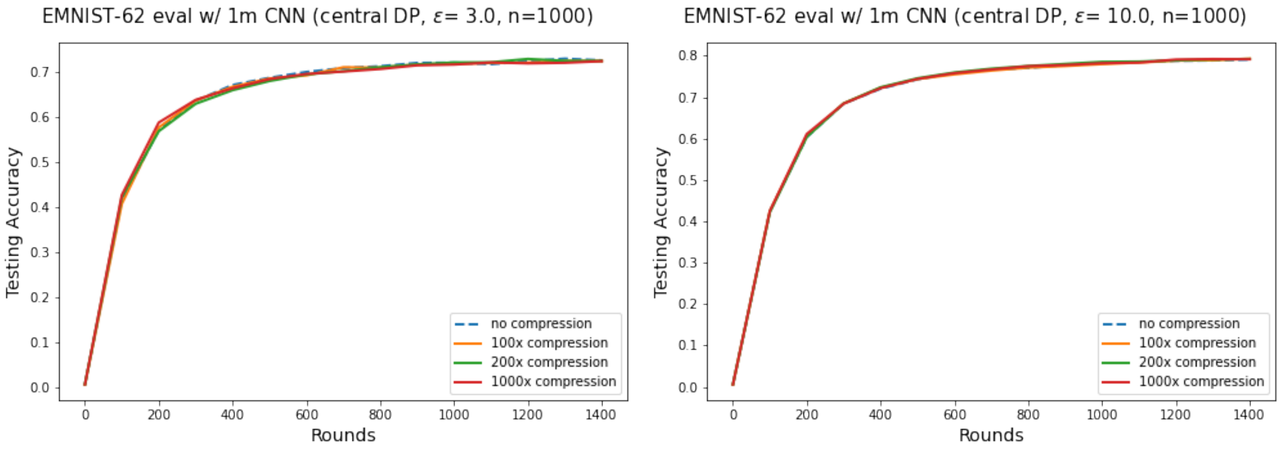

Figure 4. Accuracy of GM and CSGM, with $\delta = 10^{-5}$ and cohort size 1000. The Δ_∞/Δ_2 ratio is $6.4 \cdot 10^{-3}$ for F-EMNIST.

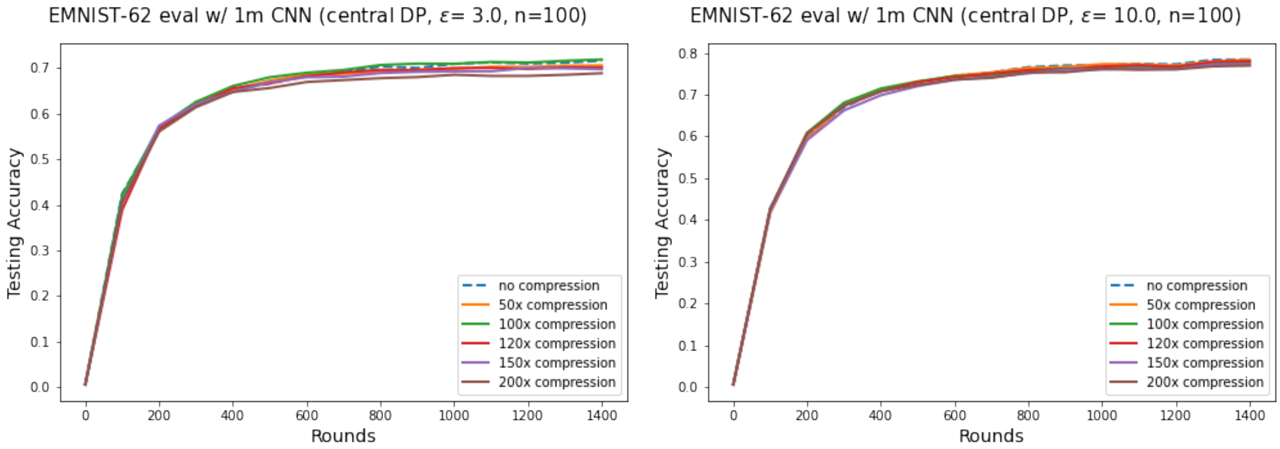

Figure 5. Accuracy of GM and CSGM, with $\delta = 10^{-5}$ and cohort size 100. The Δ_∞/Δ_2 ratio is $6.4 \cdot 10^{-3}$ for F-EMNIST.

Sparsified Gaussian Matrix Mechanism for DP-FTRL. In Figure 7 and Figure 8, we report the accuracy of SGMF (Algorithm 2) and the uncompressed matrix mechanism. We use the same factorization as in Denisov et al. (2022) with $T = 32$ for 16 epochs (due to the limited amount of clients), and we restart the mechanism and shuffle clients every epoch as in the TreeRestart approach in Kairouz et al. (2021b). We observe that for the matrix mechanism, the compression rates are generally less than DP-FedAvg, and the performance is more sensitive to server learning rates and L_2 clip norms.

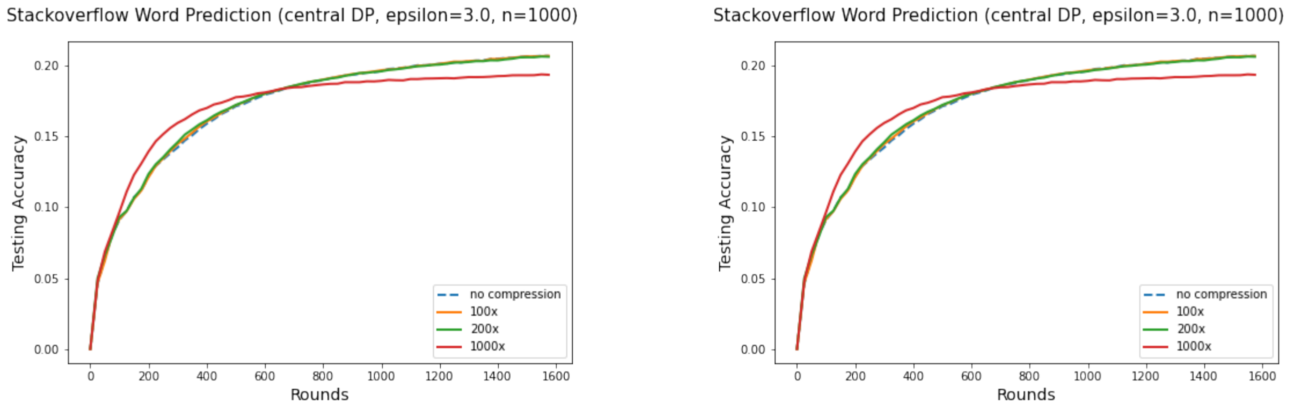

Figure 6. Accuracy of GM and CSGM, with $\delta = 10^{-5}$ and cohort size 100. The Δ_∞/Δ_2 ratio is $6.4 \cdot 10^{-3}$ for F-EMNIST.

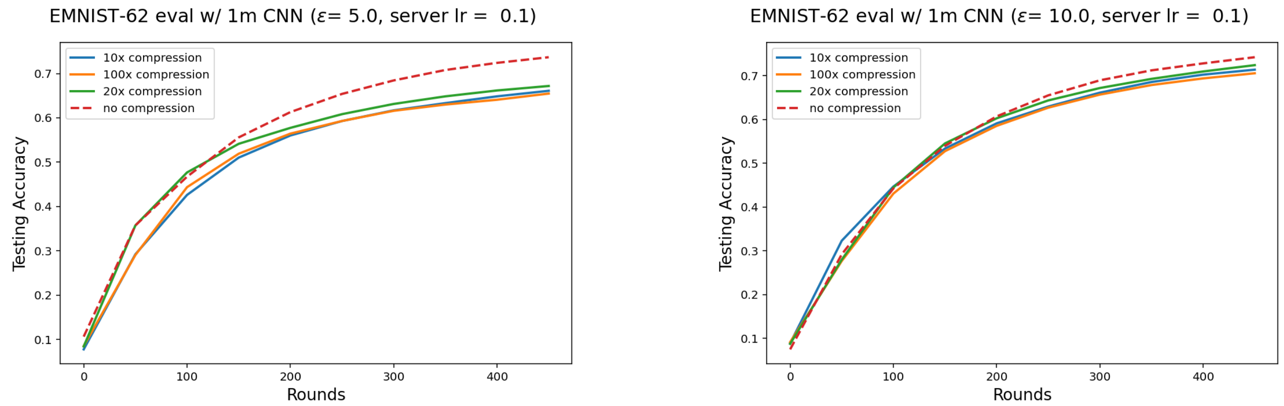

Figure 7. Accuracy of MF and SGMF on EMNIST, with $\delta = 10^{-5}$, clipped norm $\Delta_2 = 1.0$, and server learning rate 0.1.

D. Proofs

D.1. Proof of Theorem 4.1

For any $\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_n$, it holds that

$$
\begin{aligned}
& D_\alpha\left(\boldsymbol{s}_1 \odot \boldsymbol{g}_1 + \sum_{i=2}^n \boldsymbol{s}_i \odot \boldsymbol{g}_i + Z \middle\| \sum_{i=2}^n \boldsymbol{s}_i \odot \boldsymbol{g}_i + Z\right) \\
& \overset{\text{(a)}}{\leq} D_\alpha\left(\boldsymbol{s}_1 \odot \boldsymbol{g}_1 + Z \| Z\right) \\
& \overset{\text{(b)}}{=} \sum_{j=1}^d D_\alpha\left(\boldsymbol{s}_1(j) \cdot \boldsymbol{g}_1(j) + Z_j \| Z_j\right) \\
& = \sum_{j=1}^d D_\alpha\left(\gamma \mathcal{N}(\boldsymbol{g}_1(j), \sigma^2) + (1-\gamma)\mathcal{N}(0,\sigma^2) \middle\| \mathcal{N}(0,\sigma^2)\right),
\end{aligned}
$$

where (a) is due to the data processing inequality, and in (b) holds since $\boldsymbol{s}_1(j)$ and Z_j are independent across $j \in [d]$. Similarly, it holds that

$$
D_\alpha\left(\sum_{i=2}^n \boldsymbol{s}_i \odot \boldsymbol{g}_i + Z \middle\| \boldsymbol{g}_1 + \sum_{i=2}^n \boldsymbol{s}_i \odot \boldsymbol{g}_i + Z\right) \leq \sum_{j=1}^d D_\alpha\left(\mathcal{N}(0,\sigma^2) \middle\| \gamma \mathcal{N}(\boldsymbol{g}_1(j), \sigma^2) + (1-\gamma)\mathcal{N}(0,\sigma^2)\right),
$$

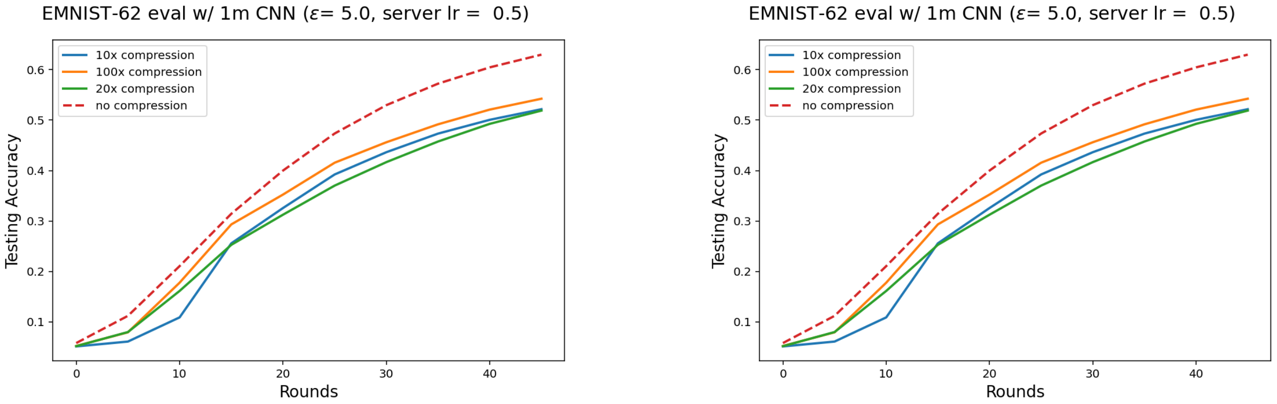

Figure 8. Accuracy of MF and SGMF on EMNIST, with $\delta = 10^{-5}$, clipped norm $\Delta_2 = 0.03$, and server learning rate 0.5.

For notational simplicity, let us define $\kappa_j \triangleq \boldsymbol{g}_1(j)$. Notice that $\boldsymbol{g}_1 \in \mathbb{S}^{d-1}$ implies $\|\kappa\|_2 \leq 1$.

Then for each $j \in [d]$, by Corollary 7 of Mironov et al. (2019),

$$\begin{aligned}
&\max\left(D_\alpha\left(\gamma\mathcal{N}(x_{1j},\sigma^2)+(1-\gamma)\mathcal{N}(0,\sigma^2)\middle\|\mathcal{N}(0,\sigma^2)\right), D_\alpha\left(\mathcal{N}(0,\sigma^2)\middle\|\gamma\mathcal{N}(x_{1j},\sigma^2)+(1-\gamma)\mathcal{N}(0,\sigma^2)\right)\right)\\
&= D_\alpha\left(\gamma\mathcal{N}(x_{1j},\sigma^2)+(1-\gamma)\mathcal{N}(0,\sigma^2)\middle\|\mathcal{N}(0,\sigma^2)\right)\\
&= \frac{1}{\alpha-1}\log\left(\mathbb{E}_{X\sim q}\left[\left((1-\gamma)+\gamma\frac{p}{q}(X)\right)^\alpha\right]\right)
\end{aligned}$$

where p is a density function of $\mathcal{N}(\kappa_j, \sigma^2)$ and q is a density function of $\mathcal{N}(0, \sigma^2)$.

For any integer α, we have

$$\begin{aligned}
&\frac{1}{\alpha-1}\log\left(\mathbb{E}_{X\sim q}\left[\left((1-\gamma)+\gamma\frac{p}{q}(X)\right)^\alpha\right]\right)\\
&= \frac{1}{\alpha-1}\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\mathbb{E}_{X\sim q}\left[\exp\left(\ell\left(-\frac{1}{2\sigma^2}\right)\left((X-\kappa_j)^2-X^2\right)\right)\right]\right)\\
&= \frac{1}{\alpha-1}\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\mathbb{E}_{X\sim q}\left[\exp\left(-\frac{\ell}{2\sigma^2}(\kappa_j^2-2\kappa_j X)\right)\right]\right)\\
&= \frac{1}{\alpha-1}\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\exp\left(\frac{-\ell\kappa_j^2}{2\sigma^2}\right)\mathbb{E}_{X\sim q}\left[\exp\left(\frac{\kappa_j\ell}{\sigma^2}X\right)\right]\right)\\
&\overset{\text{(a)}}{=} \frac{1}{\alpha-1}\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\exp\left(\frac{-\ell\kappa_j^2}{2\sigma^2}\right)\exp\left(\left(\frac{\kappa_j\ell}{\sigma^2}\right)^2\frac{1}{2}\sigma^2\right)\right)\\
&= \frac{1}{\alpha-1}\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\exp\left(\frac{(\ell^2-\ell)\kappa_j^2}{2\sigma^2}\right)\right)
\end{aligned}$$

where (a) is due to the generating function of normal distribution.

As a result, summing $j \in [d]$ yields

$$\varepsilon^*(\alpha) \leq \max_{\kappa:\|\kappa\|_2\leq\Delta_2,\|\kappa\|_\infty\leq\Delta_\infty}\sum_j \frac{1}{\alpha-1}\underbrace{\log\left(\sum_{\ell=0}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^\ell\exp\left(\frac{(\ell^2-\ell)\kappa_j^2}{2\sigma^2}\right)\right)}_{\triangleq f(\kappa_j^2)}$$

First, observe that (1) $\kappa^2 \mapsto f(\kappa^2)$ is increasing and convex (since it is log-sum-exp), and (2) $f(0) = 0$. Next, define $\kappa_1^* \geq \kappa_2^* \geq \cdots \geq \kappa_d^*$ as the unique sequence such that

- $\kappa_j^* = \Delta_\infty$ for any $j \leq \frac{\Delta_2^2}{\Delta_\infty^2}$;
- $\kappa_j^* = 0$ for any $j > \frac{\Delta_2^2}{\Delta_\infty^2} + 1$;
- $\sum_j (\kappa_j^*)^2 = \Delta_2^2$.

Then, it is obvious that $\left(\kappa_1^*\right)^2, \left(\kappa_2^*\right)^2, \cdots, \left(\kappa_d^*\right)^2$ is a majorization[5] of any $\kappa_1^2 \geq \kappa_2^2 \geq \cdots \geq \kappa_d^2$ such that $\sum_{j=1}^d \kappa_j^2 = \Delta_2$ and $\max_{j\in[d]} \kappa_j^2 \leq \Delta_\infty^2$. Applying Karamata's inequality[6] yields

$$\begin{aligned}
&\max_{\kappa: \|\kappa\|_2 \leq \Delta_2, \|\kappa\|_\infty \leq \Delta_\infty} \sum_j \frac{1}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\kappa_j^2}{2\sigma^2}\right)\right)\\
&= \sum_j \frac{1}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)(\kappa_j^*)^2}{2\sigma^2}\right)\right)\\
&= \frac{\lfloor \Delta_2^2/\Delta_\infty^2 \rfloor}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\Delta_\infty^2}{2\sigma^2}\right)\right)\\
&\quad + \frac{1}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)(\Delta_2^2 - \Delta_\infty^2 \cdot \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor)}{2\sigma^2}\right)\right)\\
&\leq \frac{\Delta_2^2/\Delta_\infty^2}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\Delta_\infty^2}{2\sigma^2}\right)\right),
\end{aligned}$$

where the last inequality holds due to the convexity and the following Jensen's inequality:

$$\begin{aligned}
&\frac{1}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)(\Delta_2^2 - \Delta_\infty^2 \cdot \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor)}{2\sigma^2}\right)\right)\\
&= \frac{1}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\left(\Delta_\infty^2\left(\Delta_2^2/\Delta_\infty^2 - \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor\right)\right)}{2\sigma^2}\right)\right)\\
&\leq \frac{1-\left(\Delta_2^2/\Delta_\infty^2 - \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor\right)}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\Delta_\infty^2 \cdot 0}{2\sigma^2}\right)\right)\\
&\quad + \frac{\Delta_2^2/\Delta_\infty^2 - \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\Delta_\infty^2 \cdot 1}{2\sigma^2}\right)\right)\\
&= \frac{\Delta_2^2/\Delta_\infty^2 - \lfloor \Delta_2^2/\Delta_\infty^2 \rfloor}{\alpha-1} \log\left(\sum_{\ell=0}^{\alpha} \binom{\alpha}{\ell} (1-\gamma)^{\alpha-\ell} \gamma^\ell \exp\left(\frac{(\ell^2-\ell)\Delta_\infty^2}{2\sigma^2}\right)\right).
\end{aligned}$$

This establishes the theorem. □

D.2. Proof of Lemma 5.2

To upper bound $D_\alpha\left(P_{\mathbf{P}|\mathbf{G}} \middle\| P_{\mathbf{P}|\mathbf{H}}\right)$, observe that for any coordinate $i \in [d]$, $\mathbf{P}_i \triangleq (\boldsymbol{p}_i^{(1)}, ..., \boldsymbol{p}_i^{(T)})$ depends solely on $\boldsymbol{g}_i^{(1)}$, $\mathbf{s}_i^{(1)}$ and $\left(\mathbf{Z}_i^{(1)}, ..., \mathbf{Z}_i^{(T)}\right)$. Therefore,

$$\begin{aligned}
D_\alpha\left(P_{\mathbf{P}|\mathbf{G}} \middle\| P_{\mathbf{P}|\mathbf{H}}\right) &= \sum_{i=1}^d D_\alpha\left(P_{\mathbf{P}_i|\mathbf{G}_i} \middle\| P_{\mathbf{P}_i|\mathbf{H}_i}\right)\\
&= \sum_{i=1}^d D_\alpha\left(P_{\mathbf{p}_i^{(1)},...,\mathbf{p}_i^{(T)}|\mathbf{G}_i} \middle\| P_{\mathbf{p}_i^{(1)},...,\mathbf{p}_i^{(T)}|\mathbf{H}_i}\right).
\end{aligned}$$

[5] See https://en.wikipedia.org/wiki/Karamata%27s_inequality for a definition of "majorization".

[6] https://en.wikipedia.org/wiki/Karamata%27s_inequality

Then, we claim that releasing $\left\{\boldsymbol{p}_i^{(t)} = \mathbf{M}_{t1}\left(\boldsymbol{g}_i^{(1)} \cdot \boldsymbol{S}_i^{(1)}\right) + \boldsymbol{Z}_i^{(t)}, t \in [T]\right\}$ is indeed an instance of (non-adaptive) subsampled Gaussian mechanism. By writing it in a vector form

$$\begin{bmatrix} \boldsymbol{p}_i^{(1)} \\ \boldsymbol{p}_i^{(2)} \\ \vdots \\ \boldsymbol{p}_i^{(T)} \end{bmatrix} = \boldsymbol{g}_i^{(1)} \cdot \boldsymbol{S}_i^{(1)} \cdot \begin{bmatrix} \mathbf{M}_{11} \\ \mathbf{M}_{21} \\ \vdots \\ \mathbf{M}_{T1} \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_i^{(1)} \\ \mathbf{Z}_i^{(2)} \\ \vdots \\ \mathbf{Z}_i^{(T)} \end{bmatrix}, \tag{6}$$

it becomes clear as a subsampled Gaussian mechanism with sensitivity $\xi\left(\mathbf{M}\right) \cdot \left|\boldsymbol{g}_i^{(1)}\right|$. Since $\mathbf{M} = \mathbf{Q} \cdot \mathbf{C}$ and that $\mathbf{Q}$ is orthonormal, we have $\xi\left(\mathbf{M}\right) = \xi\left(\mathbf{C}\right)$. Also, by the geometrical assumption of data matrix $\mathbf{G}$, it holds that $\sum_{i=1}^{d}\left|\boldsymbol{g}_i^{(1)}\right|^2 \leq \Delta_2$ and $\left|\boldsymbol{g}_i^{(1)}\right| \leq \Delta_\infty$ for all i. Summing across $i \in [d]$ and applying Theorem 4.1 yield

$$D_\alpha\left(P_{\mathbf{P}|\mathbf{G}} \middle\| P_{\mathbf{P}|\mathbf{H}}\right) \leq \frac{\kappa_2^2/\kappa_\infty^2}{\alpha - 1}\log\left((1-\gamma)^{\alpha-1}\left(\gamma(\alpha-1)+1\right) + \sum_{\ell=1}^{\alpha}\binom{\alpha}{\ell}(1-\gamma)^{\alpha-\ell}\gamma^{\ell}e^{(\ell-1)\alpha\frac{\kappa_\infty^2}{2\sigma^2}}\right), \tag{7}$$

establishing the desired result.