The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation

Peter Kairouz* Ziyu Liu† Thomas Steinke‡

Abstract

We consider training models on private data that are distributed across user devices. To ensure privacy, we add on-device noise and use secure aggregation so that only the noisy sum is revealed to the server. We present a comprehensive end-to-end system, which appropriately discretizes the data and adds discrete Gaussian noise before performing secure aggregation. We provide a novel privacy analysis for sums of discrete Gaussians and carefully analyze the effects of data quantization and modular summation arithmetic. Our theoretical guarantees highlight the complex tension between communication, privacy, and accuracy. Our extensive experimental results demonstrate that our solution is essentially able to match the accuracy to central differential privacy with less than 16 bits of precision per value.

Contents

*Google Research kairouz@google.com
†Google Research klz@google.com
‡Google Research, Brain Team ddg@thomas-steinke.net

1 Introduction

Software and service providers rely on increasingly complex data analytics and machine learning models to improve their services. However, training these machine learning models hinges on the availability of large datasets, which are often distributed across user devices and contain sensitive information. The collection of these datasets comes with several privacy risks – can the service provider address issues around consent, transparency, control, breaches, persistence, processing, and release of data? There is thus a strong desire for technologies which systematically address privacy concerns while preserving, to the best extent possible, the utility of the offered services.

To address this need, several privacy-enhancing technologies have been studied and built over the past few years. Prominent examples of such technologies include federated learning (FL) to ensure that raw data never leaves users' devices [MMRHA17; KM+19], cryptographic secure aggregation (SecAgg) to prevent a server from inspecting individual user updates [BIKMMPRSS17; BBGLR20], and differentially private stochastic gradient descent (DP-SGD) to train models with provably limited information leakage [ACGMMTZ16; TB20]. While these technologies have been extremely well studied in a separate fashion, little work has focused on understanding precisely how they can be combined in a rigorous and principled fashion. Towards this end, we present a comprehensive end-to-end system where each client appropriately discretizes their model update and adds discrete Gaussian noise to it before sending it for modular secure summation using SecAgg. This provides the first concrete step towards building a communication-efficient FL system with distributed DP[1] and SecAgg guarantees.

Organization The remainder of the paper is organized as follows. We summarize our main results and review related works in this section. We present the preliminaries in Section 2. In Section 3, we introduce the distributed discrete Gaussian mechanism and analyze its privacy guarantees. In Section 4, we show how $\mathbb{R}$-valued vectors can be efficiently mapped

[1]See "Distributed DP" paragraph in Section 1.2 for a definition of this notion of DP.

to $\mathbb{Z}$-valued vectors and how the distributed discrete Gaussian mechanism can be combined with modulo clipping to obtain noisy vectors in $\mathbb{Z}_m^d$. We present our experimental results in Section 5 and conclude our paper with a few interesting and non-trivial extensions in Section 6.

1.1 Main Results

We start by considering a single round of federated learning in which we are simply summing model update vectors. That is, we have n clients and assume that each client holds a vector $x_i \in \mathbb{R}^d$ and our goal is to privately approximate $\bar{x} := \sum_i^n x_i$. Client i computes $z_i = \mathcal{A}_{\text{client}}(x_i) \in \mathbb{Z}_m^d$; here, $\mathcal{A}_{\text{client}}(\cdot)$ can be thought of as a compression and privatization scheme. Using secure aggregation as a black box,[2] the server observes

$$\bar{z} := \sum_i^n z_i \mod m = \sum_i^n \mathcal{A}_{\text{client}}(x_i) \mod m, \tag{1}$$

and uses $\bar{z}$ to estimate $\mathcal{A}_{\text{server}}(\bar{z}) \approx \bar{x} = \sum_i^n x_i$.

The protocol consists of three parts – the client side $\mathcal{A}_{\text{client}}$, secure aggregation, and the server side $\mathcal{A}_{\text{server}}$. There is already ample work on implementing secure aggregation [BBGLR20; BIKMMPRSS16]; thus we treat SecAgg as a black box which is guaranteed to faithfully compute the modular sum of the inputs, while revealing no further information to a potential privacy adversary. Further discussion of SecAgg and the required trust assumptions is beyond the scope of this work. This allows us to focus on the requirements for $\mathcal{A}_{\text{client}}$ and $\mathcal{A}_{\text{server}}$:

- **Privacy:** The sum $\bar{z} = \sum_i^n \mathcal{A}_{\text{client}}(x_i) \mod m$ must be a differentially private function of the inputs $x_1, \cdots, x_n$. Specifically, adding or removing one client should only change the distribution of the sum slightly. Note that our requirement is weaker than local DP, since we only reveal the sum, rather than the individual responses $z_i = \mathcal{A}_{\text{client}}(x_i)$.

  Privacy is achieved by each client independently adding discrete Gaussian noise [CKS20] to its (appropriately discretized) vector. The sum of independent discrete Gaussians is *not* a discrete Gaussian, but we show that it is *extremely* close for the parameter regime of interest. This is the basis of our differential privacy guarantee, and we believe this result to be of independent interest.

- **Accuracy:** Our goal is to approximate the sum $\mathcal{A}_{\text{server}}(\bar{z}) \approx \bar{x} = \sum_i^n x_i$. For simplicity, we focus on the mean squared error, although our experiments also evaluate the accuracy by aggregating client model updates for federated learning.

  There are three sources of error to consider: (i) the discretization of the x_i vectors from $\mathbb{R}^d$ to $\mathbb{Z}^d$; (ii) the noise added for privacy (which also depends on the norm $\|x_i\|$ and

[2] We will assume the secure aggregation protocol accepts z_i's on $\mathbb{Z}_m^d$ (i.e., length-d integer vectors modulo m) and computes the sum modulo m. Our methods do not depend on the specifics of the implementation of SecAgg.

Algorithm 1 Client Procedure $\mathcal{A}_{\text{client}}$

Input: Private vector $x_i \in \mathbb{R}^d$. {Assume dimension d is a power of 2.}

Parameters: Dimension $d \in \mathbb{N}$; clipping threshold $c > 0$; granularity $\gamma > 0$; modulus $m \in \mathbb{N}$; noise scale $\sigma > 0$; bias $\beta \in [0, 1)$.

Shared/public randomness: Uniformly random sign vector $\xi \in \{-1, +1\}^d$.

Clip and scale vector: $x_i' = \frac{1}{\gamma} \min\left\{1, \frac{c}{\|x_i\|_2}\right\} \cdot x_i \in \mathbb{R}^d$.

Flatten vector: $x_i'' = H_d D_\xi x_i' \in \mathbb{R}^d$ where $H \in \{-1/\sqrt{d}, +1/\sqrt{d}\}^{d \times d}$ is a Walsh-Hadamard matrix satisfying $H^T H = I$ and $D_\xi \in \{-1, 0, +1\}^{d \times d}$ is a diagonal matrix with ξ on the diagonal.

repeat

Let $\tilde{x}_i \in \mathbb{Z}^d$ be a randomized rounding of $x_i'' \in \mathbb{R}^d$. I.e., $\tilde{x}_i$ is a product distribution with $\mathbb{E}[\tilde{x}_i] = x_i''$ and $\|\tilde{x}_i - x_i''\|_\infty < 1$.

until $\|\tilde{x}_i\|_2 \le \min\left\{c/\gamma + \sqrt{d}, \sqrt{c^2/\gamma^2 + \frac{1}{4}d + \sqrt{2\log(1/\beta)} \cdot \left(c/\gamma + \frac{1}{2}\sqrt{d}\right)}\right\}$.

Let $y_i \in \mathbb{Z}^d$ consist of d independent samples from the discrete Gaussian $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2/\gamma^2)$.

Let $z_i = (\tilde{x}_i + y_i) \mod m$.

Output: $z_i \in \mathbb{Z}_m^d$ for the secure aggregation protocol.

how discretization affects this); and (iii) the potential modular wrap-around introduced by SecAgg modular sum. We provide a detailed analysis of all three effects and how they affect one another.

- **Communication and Computation:** It is crucial that our algorithms are efficient, especially the client side, which may be running on a mobile device. Computationally, our algorithms run in time that is nearly linear in the dimension. The communication cost is $O(d \log m)$. While we cannot control the dimension d, we can minimize the number of bits per coordinate, which is $\log m$. However, this introduces a tradeoff between communication and accuracy – larger m means more communication, but we can reduce the probability of a modular wrap around and pick a finer discretization to reduce the rounding error.

We focus our discussion on the simple task of summing vectors. In a realistic federated learning system, there will be many summing rounds as we iteratively update our model. Each round will be one invocation of our protocol. The privacy loss parameters of the larger system can be controlled using the composition and subsampling properties of differential privacy. That is, we can use standard privacy accounting techniques [BS16; Mir17; WBK19] to analyse the more complex system, as long as we have differential privacy guarantees for the basic protocol that is used as a subroutine.

We now present our algorithm in two parts – the client part $\mathcal{A}_{\text{client}}$ in Algorithm 1 and the server part $\mathcal{A}_{\text{server}}$ in Algorithm 2. The two parts are connected by a secure aggregation protocol. We also note that our algorithms may be a subroutine of a larger FL system.

We briefly remark about the parameters of the algorithm: d is the dimension of the inputs x_i and outputs, which we assume is a power of 2 for convenience. The input vectors

Algorithm 2 Server Procedure $\mathcal{A}_{\text{server}}$

Input: Vector $\bar{z} = (\sum_i^n z_i \mod m) \in \mathbb{Z}_m^d$ via secure aggregation.
Parameters: Dimension $d \in \mathbb{N}$; number of clients $n \in \mathbb{N}$; clipping threshold $c > 0$; granularity $\gamma > 0$; modulus $m \in \mathbb{N}$; noise scale $\sigma > 0$; bias $\beta \in [0, 1)$.
Shared/public randomness: Uniformly random sign vector $\xi \in \{-1, +1\}^d$.
Map $\mathbb{Z}_m$ to $\{1 - m/2, 2 - m/2, \cdots, -1, 0, 1 \cdots, m/2 - 1, m/2\}$ so that $\bar{z}$ is mapped to $\bar{z}' \in [-m/2, m/2]^d \cap \mathbb{Z}^d$ (and we have $\bar{z}' \mod m = \bar{z}$).
Output: $y = \gamma D_\xi H_d^T \bar{z}' \in \mathbb{R}^d$. {Goal: $y \approx \bar{x} = \sum_i^n x_i$}

must have their norm clipped for privacy; c controls this tradeoff – larger c will require more noise for privacy (larger σ) and smaller c will distort the vectors more. If $\beta = 0$, then the discretization via randomized rounding is unbiased, but the norm of $\tilde{x}_i$ could be larger; each iteration of the randomized rounding loop succeeds with probability at least $1 - \beta$. The modulus m will determine the communication complexity – z_i requires $d \log_2 m$ bits to represent. The noise scale σ determines the privacy, specifically $\varepsilon \approx c/\sqrt{n}\sigma$. Finally, the granularity γ gives a tradeoff: smaller γ means the randomized rounding introduces less error, but also makes it more likely that the modulo m operation introduces error.

We also remark about some of the techniques used in our system: The first step in Algorithm 1 scales and clips the input vector so that $\|x_i'\|_2 \leq c/\gamma$. The next step performs a unitary rotation/reflection operation $x_i'' = H_d D_\xi x_i'$ [SFKM17]. This operation "flattens" the vector – i.e., $\|x_i''\|_\infty \approx \frac{1}{\sqrt{d}} \|x_i'\|_2$. Flattening ensures that the modular arithmetic does not introduce large distortions due to modular wrap around (i.e., large coordinates of x_i'' will be subject to modular reduction). This flattening operation and the scaling by γ are undone in the last step of Algorithm 2. The x_i'' is randomly rounded to the integer grid in an unbiased manner. That is, each coordinate is independently rounded to one of the two nearest integers. E.g., 42.3 has a 30% probability of being rounded up to 43 and a 70% probability of being rounded down to 42. This may increase the norm – $\|\tilde{x}_i\|_2 \leq \|x_i''\|_2 + \sqrt{d}$. To mitigate this, we perform *conditional* randomized rounding: repeatedly perform independent randomized rounding on x_i'' until $\|\tilde{x}_i\|_2$ is not too big. This introduces a small amount of bias, but, since the noise we add to attain differential privacy must scale with the norm of the discretized vector, reducing the norm reduces the noise variance.

Privacy We now state the privacy of our algorithm.

Theorem 1 (Privacy of Our Algorithm)**.** *Let $c, d, \gamma, \beta, \sigma$ be the parameters of Algorithm 1*

and n the number of trustworthy clients. Define

$$\Delta_2^2 := \min\left\{ \begin{array}{c} c^2 + \frac{\gamma^2 d}{4} + \sqrt{2\log\left(\frac{1}{\beta}\right)} \cdot \gamma \cdot \left(c + \frac{\gamma}{2}\sqrt{d}\right), \\ \left(c + \gamma\sqrt{d}\right)^2 \end{array} \right\}, \tag{2}$$

$$\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2 \frac{\sigma^2}{\gamma^2} \cdot \frac{k}{k+1}}, \tag{3}$$

$$\varepsilon := \min\left\{ \begin{array}{l} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}. \tag{4}$$

Then Algorithm 1 satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy,[3] assuming that secure aggregation only reveals the sum $z = (\sum_i^n z_i \mod m) \in \mathbb{Z}_m^d$ to the privacy adversary.

We remark on the parameters of the theorem: To first approximation, $\varepsilon \approx \frac{c}{\sqrt{n}\sigma}$. This is because the input vectors are clipped to have norm c and then each client adds (discrete) Gaussian noise with variance $\approx \sigma^2$. The noise added to the sum thus has variance $\approx n\sigma^2$. However, there are two additional effects to account for: First, randomized rounding can increase the norm from c to Δ_2 and this becomes the sensitivity bound that we use for the privacy analysis. Second, the sum of n discrete Gaussians is *not* a discrete Gaussian, but it is close; τ bounds the max divergence between the sum of n discrete Gaussians each with scale parameter σ/γ and one discrete Gaussian with scale parameter $\sqrt{n}\sigma/\gamma$.

Note that $\frac{1}{2}\varepsilon^2$-concentrated DP [BS16] is equivalent to satisfying $\left(\alpha, \frac{1}{2}\varepsilon^2\alpha\right)$-Rényi DP [Mir17] simultaneously for all $\alpha > 1$. Concentrated DP can be converted to the more standard approximate differential privacy [CKS20]: For any $\delta > 0$, $\frac{1}{2}\varepsilon^2$-concentrated DP implies $(\varepsilon_{\text{aDP}}(\delta), \delta)$-DP, where

$$\varepsilon_{\text{aDP}}(\delta) = \inf_{\alpha>1} \frac{1}{2}\varepsilon^2\alpha + \frac{\log(1/\alpha\delta)}{\alpha - 1} + \log(1 - 1/\alpha) \leq \frac{1}{2}\varepsilon^2 + \sqrt{2\log(1/\delta)} \cdot \varepsilon.$$

Accuracy Next we turn to the accuracy of the algorithm. We provide both an empirical evaluation and theoretical analysis. We give the following asymptotic guarantee; a more precise guarantee with exact constants can be found in Theorem 36.

Theorem 2 (Accuracy of Our Algorithm)**.** *Let $n, m, d \in \mathbb{N}$ and $c, \varepsilon > 0$ satisfy*

$$m \geq \tilde{O}\left(n + \sqrt{\frac{\varepsilon^2 n^3}{d}} + \frac{\sqrt{d}}{\varepsilon}\right).$$

[3]Note that this is with respect to the addition or removal of an individual, not replacement (which would double the ε parameter). To keep n fixed, we could define addition/removal to simply zero-out the relevant vectors.

Let $\tilde{A}(x) = \mathcal{A}_{\text{server}}\left(\sum_i^n \mathcal{A}_{\text{client}}(x_i) \mod m\right)$ *denote the output of the system given by Algorithms 1 and 2 instantiated with parameters* $\gamma = \tilde{\Theta}\left(\frac{cn}{m\sqrt{d}} + \frac{c}{\varepsilon m}\right)$, $\beta \leq \Theta\left(\frac{1}{n}\right)$, *and* $\sigma = \tilde{\Theta}\left(\frac{c}{\varepsilon\sqrt{n}} + \sqrt{\frac{d}{n}} \cdot \frac{\gamma}{\varepsilon}\right)$. *Then* $\tilde{A}$ *satisfies* $\frac{1}{2}\varepsilon^2$*-concentrated differential privacy and attains the following accuracy. Let* $x_1, \cdots, x_n \in \mathbb{R}^d$ *with* $\|x_i\|_2 \leq c$ *for all* $i \in [n]$. *Then*

$$\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \leq O\left(\frac{c^2 d}{\varepsilon^2}\right). \tag{5}$$

To interpret Theorem 2, note that mean squared error $O\left(\frac{c^2 d}{\varepsilon^2}\right)$ is, up to constants, exactly the error we would expect to attain for differential privacy in the central model. Our analysis attains reasonably sharp constants (at the expense of many lower order terms that we suppress here in the introduction). However, to truly gauge the practicality of our method, we perform an empirical evaluation.

Experiments To investigate the interplay between communication, accuracy, and privacy under our proposed protocol in practice, we empirically evaluate our protocol and compare it to the commonly used centralized continuous Gaussian mechanism on two canonical tasks: distributed mean estimation (DME) and federated learning (FL). For DME, each client holds a vector and the server's goal is to obtain a differentially private mean estimate of the vectors. We show that 16 bits per coordinate are sufficient to nearly match the utility of the Gaussian baseline for regimes of interest. For FL, we show on Federated EMNIST [CDWLKMST18] and Stack Overflow [Aut19] that our approach gives good performance under tight privacy budgets, despite using generic RDP amplification via sampling [ZW19] for our methods and the precise RDP analysis for the subsampled Gaussian mechanism [MTZ19]. We provide an open-source implementation of our methods in TensorFlow Privacy [ATMR19] and TensorFlow Federated [IO19].[4]

1.2 Related Work

Federated Learning Under FL, a set of clients (e.g., mobile devices or institutions) collaboratively train a model under the orchestration of a central server, while keeping training data decentralized [MMRHA17; Bon+19]. It embodies the principles of focused data collection and minimization, and can mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning and data science approaches. FL performs many rounds of interaction between the server and subsets of online clients; for example, each round may consist of computing and aggregating the gradients of the loss for a given set of model weights, which are then updated using the aggregated gradients for the next round. This allows us to focus on the simple task of computing the sum of

[4]Code: `https://github.com/google-research/federated/tree/master/distributed_dp`.

vectors (model updates) held by the clients. We refer the reader to Kairouz, McMahan, et al. [KM+19] for a survey of recent advances and open problems in FL.

While the above features can offer significant practical privacy improvements over centralizing training data, FL offers no formal guarantee of privacy and has to be composed with other privacy technologies to offer strong (worst-case) privacy guarantees. The primary goal of this paper is to show how two such technologies, namely secure aggregation and differential privacy, can be carefully combined with FL to offer strong and quantifiable privacy guarantees.

Secure Aggregation SecAgg is a lightweight instance of cryptographic secure multi-party computation (MPC) that enables clients to submit vector inputs, such that the server learns just an aggregate function of the clients' vectors, typically the sum. In most contexts of FL, single-server SecAgg is achieved via additive masking over a finite group [BBGLR20; BIKMMPRSS16]. To be precise, clients add randomly sampled zero-sum mask vectors by working in the space of integers modulo m and sampling the coordinates of the mask uniformly from $\mathbb{Z}_m$. This process guarantees that each client's masked update is indistinguishable from random values. However, when all the masked updates are summed modulo m by the server, the masks cancel out and the server obtains the exact sum. Observe that in practice, the model updates computed by the clients are real valued vectors whereas SecAgg requires the input vectors to be from $\mathbb{Z}_m$ (i.e., integers modulo m). This discrepancy is typically bridged by clipping the values to a fixed range, say $[-r, r]$, which is then translated and scaled to $\left[0, \frac{m-1}{n}\right]$, and then uniformly quantizing the values in this range to integers in $\{0, 1, \cdots, \lfloor \frac{m-1}{n} \rfloor\}$, where n is the number of clients. This ensures that, up to clipping and quantization, the server computes the exact sum without overflowing (i.e., the sum is in $[0, m-1]$, which is unaffected by the modular arithmetic) [BSKMG19]. In our work, we provide a novel strategy for transforming $\mathbb{R}$-valued vectors into $\mathbb{Z}_m$-valued ones.

Distributed DP While SecAgg prevents the server from inspecting individual client updates, the server is still able to learn the sum of the updates, which itself may leak potentially sensitive information [MSDCS19; CLEKS19; SS19a; DSSUV15; SS19b; NSTPC21; SSSS17]. To address this issue, differential privacy (DP) [DMNS06], and in particular, DP-SGD can be employed [SCS13; BST14; ACGMMTZ16; TB20]. DP is a rigorous measure of information disclosure about individuals participating in computations over centralized or distributed datasets. Over the last decade, an extensive set of techniques has been developed for differentially private data analysis, particularly under the assumption of a centralized setting, where the raw data is collected by a trusted service provider prior to applying perturbations necessary to achieve privacy. This setting is commonly referred to as the central DP setting. More recently, there has been a great interest in the local model of DP [KLNRS11; ESAG04; War65] where the data is perturbed on the client side before it is collected by a service provider.

Local DP avoids the need for a fully trusted aggregator. However, it is now well-established that local DP usually leads to a steep hit in accuracy [KLNRS11; DJW13;

KBR16]. In order to recover some of the utility of central DP, without having to rely on a fully trusted central server, an emerging set of models of DP, often referred to as distributed DP, can be used. Under distributed DP, clients employ a cryptographic protocol (e.g., SecAgg) to simulate some of the benefits of a trusted central party. Clients first compute minimal application-specific reports, perturb these slightly, and then execute the aggregation protocol. The untrusted server then only has access to the aggregated reports, with the aggregated perturbations. The noise added by individual clients is typically insufficient for a meaningful local DP guarantee on its own. However, after aggregation, the aggregated noise is sufficient for a meaningful DP guarantee, under the security assumptions necessary for the cryptographic protocol.

FL with SecAgg and Distributed DP Despite the recent surge of interest in distributed DP, much of the work in this space focuses on the shuffled model of DP where a trusted third party (or a trusted execution environment) shuffles the noisy client updates before forwarding them to the server [EFMRTT19; BEMMRLRKTS17; CSUZZ19]. For more information on the shuffled model of DP, we refer the reader to Ghazi, Kumar, Manurangsi, and Pagh [GKMP20], Ghazi, Golowich, Kumar, Pagh, and Velingker [GGKPV21], Ghazi, Manurangsi, Pagh, and Velingker [GMPV20], Ghazi, Golowich, Kumar, Manurangsi, Pagh, and Velingker [GGKMPV20], Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS06], Balle, Bell, Gascón, and Nissim [BBGN19; BBGN20], Balcer and Cheu [BC20], Balcer, Cheu, Joseph, and Mao [BCJM21], and Girgis, Data, Diggavi, Kairouz, and Suresh [GDDKS20].

The combination of SecAgg and distributed DP in the context of communication-efficient FL is far less studied. For instance, the majority of existing works ignore the finite precision and modular summation arithmetic associated with secure aggregation [GXS13; TBASLZZ19; VA17]. This is especially problematic at low SecAgg bit-widths (e.g., in practical FL settings where communication efficiency is critical).

The closest work to ours is cpSGD [ASYKM18], which also serves as an inspiration for much of our work. cpSGD uses a distributed version of the binomial mechanism [DKMMN06] to achieve distributed DP. When properly scaled, the binomial mechanism can (asymptotically) match the continuous Gaussian mechanism. However, there are several important differences between our work and cpSGD. First, the binomial mechanism does not achieve Rényi or concentrated DP [Mir17; BS16] and hence we cannot combine it with state-of-the-art composition and subsampling results, which is a significant barrier if we wish to build a larger FL system. The binomial mechanism is analyzed via approximate DP; in other words, the privacy loss for the binomial mechanism can be infinite with a non-zero probability. We avoid this issue by basing our privacy guarantee on the discrete Gaussian mechanism [CKS20], which also matches the performance of the continuous Gaussian and yields clean concentrated DP guarantees that are suitable for sharp composition and subsampling analysis. cpSGD also does not consider the impact of modular arithmetic, which makes it harder to combine with secure aggregation.

Previous attempts at achieving DP using a distributed version of the discrete Gaussian mechanism have either inaccurately glossed over the fact that the sum of discrete Gaussians

is not a discrete Gaussian, or assumed that all clients secretly share a seed that is used to generate the same discrete Gaussian instance, which is problematic because a single honest-but-curious client can fully break the privacy guarantees [WJS21]. We provide a careful privacy analysis for sums of discrete Gaussians. Our privacy guarantees degrade gracefully as a function of the fraction of malicious (or dropped out) clients.

2 Preliminaries

We begin by defining the Rényi divergences, which we use throughout to quantify privacy.

Definition 3 (Rényi divergences)**.** *Let P and Q be probability distributions on some common domain Ω. Assume that P is absolutely continuous with respect to Q so that the Radon-Nikodym derivative $P(x)/Q(x)$ is well-defined for $x \in \Omega$.*[5]

For $\alpha \in (1, \infty)$, we define the Rényi divergence of order α of P with respect to Q as

$$\mathrm{D}_\alpha\left(P \| Q\right) := \frac{1}{\alpha - 1} \log \mathop{\mathbb{E}}_{X \leftarrow P}\left[\left(\frac{P(X)}{Q(X)}\right)^{\alpha-1}\right]. \tag{6}$$

We also define

$$\mathrm{D}_1\left(P \| Q\right) := \mathop{\mathbb{E}}_{X \leftarrow P}\left[\log\left(\frac{P(X)}{Q(X)}\right)\right] = \lim_{\alpha \to 1} \mathrm{D}_\alpha\left(P \| Q\right), \tag{7}$$

$$\mathrm{D}_\infty\left(P \| Q\right) := \sup_{x \in \Omega} \log\left(\frac{P(x)}{Q(x)}\right) = \lim_{\alpha \to \infty} \mathrm{D}_\alpha\left(P \| Q\right), \tag{8}$$

$$\mathrm{D}_{\pm\infty}\left(P \| Q\right) := \sup_{x \in \Omega} \left|\log\left(\frac{P(x)}{Q(x)}\right)\right| = \max\{\mathrm{D}_\infty\left(P \| Q\right), \mathrm{D}_\infty\left(Q \| P\right)\}, \tag{9}$$

$$\mathrm{D}_*\left(P \| Q\right) := \sup_{\alpha \in (1,\infty)} \frac{1}{\alpha} \mathrm{D}_\alpha\left(P \| Q\right). \tag{10}$$

We will abuse this notation by considering the divergence between random variables when we mean the divergence between their respective distributions.

We now state some properties of the Rényi divergences; proofs and further properties can be found in the literature [BS16; BS19].

Lemma 4. *Let P, Q, R be probability distributions such that P is absolutely continuous with respect to Q and Q is absolutely continuous with respect to R. Then the following hold.*

- ***Gaussian divergence:*** *For all $\mu, \mu' \in \mathbb{R}$ and all $\sigma > 0$, $\mathrm{D}_*\left(\mathcal{N}(\mu, \sigma^2) \| \mathcal{N}(\mu', \sigma^2)\right) = \frac{(\mu - \mu')^2}{2\sigma^2}$.*

[5] If P is *not* absolutely continuous with respect to Q, then we define all of these divergences to be infinity. The Radon-Nikodym derivative is only unique up to measure zero events. In the definition of $\mathrm{D}_\infty\left(P \| Q\right)$ and $\mathrm{D}_{\pm\infty}\left(P \| Q\right)$, we ignore zero probability events (i.e., we take the essential supremum). That is, we assume the Radon-Nikodym derivative is chosen to minimize these quantities.

- ***Conversion from max divergence:*** $\mathrm{D}_*(P\|Q) \leq \min\{\mathrm{D}_\infty(P\|Q), \frac{1}{2}(\mathrm{D}_{\pm\infty}(P\|Q))^2\}$.
- ***Triangle inequality:*** $\mathrm{D}_*(P\|R) \leq \left(\sqrt{\mathrm{D}_*(P\|Q)} + \sqrt{\mathrm{D}_*(Q\|R)}\right)^2$ *and* $\mathrm{D}_\alpha(P\|R) \leq \min\{\mathrm{D}_\alpha(P\|Q) + \mathrm{D}_\infty(Q\|R), \mathrm{D}_\infty(P\|Q) + \mathrm{D}_\alpha(Q\|R)\}$ *for all* $\alpha \in [1,\infty] \cup \{*\}$.
- ***Product distributions (non-adaptive composition):*** *If* $P = P_1 \times P_2$ *is a product distribution and* $Q = Q_1 \times Q_2$ *is a corresponding product distriution, then* $\mathrm{D}_\alpha(P\|Q) = \mathrm{D}_\alpha(P_1\|Q_1) + \mathrm{D}_\alpha(P_2\|Q_2)$ *for all* $\alpha \in [1,\infty] \cup \{*\}$.
- ***Postprocessing (data processing inequality):*** $0 \leq \mathrm{D}_\alpha(f(P)\|f(Q)) \leq \mathrm{D}_\alpha(P\|Q)$ *for all* $\alpha \in [1,\infty] \cup \{*\}$ *and all* f*, where* $f(P)$ *denotes the distribution obtained by applying some function* f *to a sample from the distribution* P*. This also holds if* f *is an independently randomized function.*
- ***Monotonicity:*** $\mathrm{D}_\alpha(P\|Q) \leq \mathrm{D}_{\alpha'}(P\|Q)$ *whenever* $1 \leq \alpha \leq \alpha' \leq \infty$.
- ***(Quasi)convexity:*** *If* P' *is a distribution on the same space as* P *and* Q' *is a distribution on the same space as* Q *and* P' *is absolutely continuous with respect to* Q'*, then*

$$\mathrm{D}_1(tP + (1-t)P'\|tQ + (1-t)Q') \leq t \cdot \mathrm{D}_1(P\|Q) + (1-t) \cdot \mathrm{D}_1(P'\|Q')$$

and, for $\alpha \in (1,\infty)$*,*

$$\begin{aligned}\mathrm{D}_\alpha(tP + (1-t)P'\|tQ + (1-t)Q') &\leq \frac{\log\left(t \cdot e^{(\alpha-1)\mathrm{D}_\alpha(P\|Q)} + (1-t) \cdot e^{(\alpha-1)\mathrm{D}_\alpha(P'\|Q')}\right)}{\alpha - 1} \\ &\leq \max\{\mathrm{D}_\alpha(P\|Q), \mathrm{D}_\alpha(P'\|Q')\},\end{aligned}$$

where $tP + (1-t)P'$ *denotes the convex combination of distributions.*

Now we can state the definitions of concentrated differential privacy [BS16] and Rényi differential privacy [Mir17] and relate these to the standard definition of differential privacy [DMNS06; DKMMN06]. We adopt *user-level privacy* – i.e., each entry in the input corresponds to *all* the records associated with a single person [MRTZ18]. Thus the differential privacy distributional similarity guarantee holds with respect to adding or removing all of the data belonging to a single person. This is stronger than the commonly-used notion of item level privacy where, if a user contributes multiple records, only the addition or removal of one record is protected.

We choose to define differential privacy with respect to adding or removing the records of an individual, rather than replacing the records. Since replacement can be achieved by a combination of an addition and a removal, group privacy (a.k.a. the triangle inequality) implies a differential privacy guarantee for replacement; however, the privacy parameter will be doubled. We define $\mathcal{X}^* = \bigcup_{n=0}^{\infty} \mathcal{X}^n$ to be the set of varying-size inputs from $\mathcal{X}$.

Definition 5 (Concentrated Differential Privacy)**.** *A randomized algorithm $M : \mathcal{X}^* \to \mathcal{Y}$ satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy iff, for all $x, x' \in \mathcal{X}^*$ differing by the addition or removal of a single user's records, we have $\mathrm{D}_*\left(M(x) \| M(x')\right) \leq \frac{1}{2}\varepsilon^2$.*

Definition 6 (Rényi Differential Privacy)**.** *A randomized algorithm $M : \mathcal{X}^* \to \mathcal{Y}$ satisfies (α, ε)-Rényi differential privacy iff, for all $x, x' \in \mathcal{X}^*$ differing by the addition or removal of a single user's records, we have $\mathrm{D}_\alpha\left(M(x) \| M(x')\right) \leq \varepsilon$.*

Definition 7 (Differential Privacy)**.** *A randomized algorithm $M : \mathcal{X}^* \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy iff, for all $x, x' \in \mathcal{X}^*$ differing by the addition or removal of a single user's records, we have*

$$\mathbb{P}\left[M(x) \in E\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[M(x') \in E\right] + \delta \tag{11}$$

for all events $E \subset \mathcal{Y}$. We refer to $(\varepsilon, 0)$-differential privacy as pure differential privacy or pointwise differential privacy and we refer to (ε, δ)-differential privacy with $\delta > 0$ as approximate differential privacy.

We remark that $(\varepsilon, 0)$-DP is equivalent to (∞, ε)-DP. Similarly, $\frac{1}{2}\varepsilon^2$-concentrated DP is equivalent to satisfying $(\alpha, \frac{1}{2}\varepsilon^2\alpha)$-Rényi DP simultaneously for all $\alpha \in (1, \infty)$.

In addition we have the following conversion lemma [BS16; CKS20; ALCKS20] from concentrated DP to approximate DP.

Lemma 8. *If M satisfies $(\varepsilon, 0)$-differential privacy, then it satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy. If M satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy, then, for any $\delta > 0$, M satisfies $(\varepsilon_{aDP}(\delta), \delta)$-differential privacy, where*

$$\varepsilon_{aDP}(\delta) = \inf_{\alpha > 1} \frac{1}{2}\varepsilon^2\alpha + \frac{\log(1/\alpha\delta)}{\alpha - 1} + \log(1 - 1/\alpha) \leq \varepsilon \cdot \left(\sqrt{2\log(1/\delta)} + \varepsilon/2\right).$$

3 Distributed Discrete Gaussian

We will use the discrete Gaussian [CKS20] as the basis of our privacy guarantee.

Definition 9 (Discrete Gaussian)**.** *The discrete Gaussian with scale parameter $\sigma > 0$ and location parameter $\mu \in \mathbb{Z}$ is a probability distribution supported on the integers $\mathbb{Z}$ denoted by $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$ and defined by*

$$\forall x \in \mathbb{Z} \qquad \mathop{\mathbb{P}}_{X \leftarrow \mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)}\left[X = x\right] = \frac{\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)}{\sum_{y \in \mathbb{Z}} \exp\left(\frac{-(y-\mu)^2}{2\sigma^2}\right)}.$$

The discrete Gaussian has many of the desirable properties of the continuous Gaussian [CKS20], including the fact that it can be used to provide differential privacy.

Theorem 10 (Privacy of the Discrete Gaussian)**.** *Let $\sigma > 0$ and $\mu, \mu' \in \mathbb{Z}$. Then*

$$\mathrm{D}_*\left(\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2) \middle\| \mathcal{N}_{\mathbb{Z}}(\mu', \sigma^2)\right) = \frac{(\mu - \mu')^2}{2\sigma^2}. \tag{12}$$

Unlike the continuous Gaussian, the sum/convolution of two independent discrete Gaussians is *not* a discrete Gaussian. However, we show that, for reasonable parameter settings, it is very close to one. The following result is a simpler version of Theorem 4.6 of Genise, Micciancio, Peikert, and Walter [GMPW20].

Theorem 11 (Convolution of two Discrete Gaussians)**.** *Let $\sigma, \tau \geq \frac{1}{2}$. Let $X \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ and $Y \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \tau^2)$ be independent. Let $Z = X + Y$. Let $W \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \sigma^2 + \tau^2)$. Then*

$$\mathrm{D}_{\pm\infty}(Z \| W) = \sup_{z \in \mathbb{Z}} \left| \log\left(\frac{\mathbb{P}[Z = z]}{\mathbb{P}[W = z]}\right)\right| \leq 5 \cdot e^{-2\pi^2/(1/\sigma^2 + 1/\tau^2)}. \tag{13}$$

The bound of the theorem is surprisingly strong; if $\sigma^2 = \tau^2 = 3$, then the bound is $\leq 10^{-12}$, which should suffice for most applications. Furthermore, closeness in max divergence is the strongest measure of closeness that we could hope for (rather than, say, total variation distance).

Proof. For all $z \in \mathbb{Z}$,

$$\begin{aligned}
\mathbb{P}[Z = z] &= \sum_{x \in \mathbb{Z}} \mathbb{P}[X = x] \cdot \mathbb{P}[Y = z - x] \\
&= \sum_{x \in \mathbb{Z}} \frac{e^{-x^2/2\sigma^2}}{\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}} \frac{e^{-(x-z)^2/2\tau^2}}{\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}} \\
&= \frac{\sum_{x \in \mathbb{Z}} \exp\left(-\frac{(\tau^2+\sigma^2)x^2 - 2\sigma^2 xz + \sigma^2 z^2}{2\sigma^2\tau^2}\right)}{\left(\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}\right) \cdot \left(\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}\right)} \\
&= \frac{\sum_{x \in \mathbb{Z}} \exp\left(-\frac{x^2 - 2\frac{\sigma^2}{\tau^2+\sigma^2}xz + \frac{\sigma^2}{\tau^2+\sigma^2}z^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right)}{\left(\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}\right) \cdot \left(\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}\right)} \\
&= \frac{\sum_{x \in \mathbb{Z}} \exp\left(-\frac{\left(x - \frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2 - \left(\frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2 + \frac{\sigma^2}{\tau^2+\sigma^2}z^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right)}{\left(\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}\right) \cdot \left(\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}\right)} \\
&= \exp\left(\frac{\left(\frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2 - \frac{\sigma^2}{\tau^2+\sigma^2}z^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right) \frac{\sum_{x \in \mathbb{Z}} \exp\left(-\frac{\left(x - \frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right)}{\left(\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}\right) \cdot \left(\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}\right)} \\
&= \exp\left(\frac{-z^2}{2(\tau^2+\sigma^2)}\right) \frac{\sum_{x \in \mathbb{Z}} \exp\left(-\frac{\left(x - \frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right)}{\left(\sum_{u \in \mathbb{Z}} e^{-u^2/2\sigma^2}\right) \cdot \left(\sum_{v \in \mathbb{Z}} e^{-v^2/2\tau^2}\right)}.
\end{aligned}$$

The $\exp\left(\frac{-z^2}{2(\tau^2+\sigma^2)}\right)$ term is exactly what we want – it is, up to scaling, $\mathbb{P}[W = z]$. The denominator is a constant (i.e., it does not depend on z), which means we do not need to worry about it. The troublesome term is

$$\sum_{x\in\mathbb{Z}} \exp\left(-\frac{\left(x-\frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2}{2\frac{\sigma^2\tau^2}{\tau^2+\sigma^2}}\right) = \sum_{x\in\mathbb{Z}} \exp\left(-\frac{\left(x-\frac{\sigma^2}{\tau^2+\sigma^2}z\right)^2}{2}\left(\frac{1}{\sigma^2}+\frac{1}{\tau^2}\right)\right)$$
$$= \sum_{x\in\mathbb{Z}} f_{(1/\sigma^2+1/\tau^2)^{-1}}\left(x-\frac{\sigma^2}{\tau^2+\sigma^2}z\right),$$

where $f_{\rho^2}(x) := \exp\left(\frac{-x^2}{2\rho^2}\right)$. Now we apply the Poisson summation formula using the Fourier transform $\hat{f}_{\rho^2}(y) = \sqrt{2\pi\rho^2}\cdot\exp(-2\pi^2\rho^2y^2)$. For $t, \rho \in \mathbb{R}$, we have

$$\begin{aligned} g_{\rho^2}(t) := \sum_{x\in\mathbb{Z}} f_{\rho^2}(x-t) &= \sum_{y\in\mathbb{Z}} \hat{f}_{\rho^2}(y)\cdot e^{-2\pi\sqrt{-1}yt} \\ &= \sqrt{2\pi\rho^2}\sum_{y\in\mathbb{Z}} e^{-2\pi^2\rho^2y^2}\cdot e^{-2\pi\sqrt{-1}yt} \\ &= \sqrt{2\pi\rho^2}\sum_{y\in\mathbb{Z}} e^{-2\pi^2\rho^2y^2}\cdot\cos(2\pi yt) \\ &= \sqrt{2\pi\rho^2}\left(1+2\sum_{n=1}^{\infty} e^{-2\pi^2\rho^2n^2}\cdot\cos(2\pi nt)\right). \end{aligned}$$

Note that $g_{\rho^2}(t) \le g_{\rho^2}(0)$ for all $t, \rho \in \mathbb{R}$ [CKS20, Lemma 6], since $\cos(2\pi nt) \le 1 = \cos(2\pi n0)$ for all n and t. Our goal is to prove a lower bound on $g_{\rho^2}(t)/g_{\rho^2}(0)$, which follows from the following bound:

$$\begin{aligned} g_{\rho^2}(0) - g_{\rho^2}(t) &= \sqrt{2\pi\rho^2}\cdot 2\sum_{n=1}^{\infty} e^{-2\pi^2\rho^2n^2}\cdot(1-\cos(2\pi nt)) \\ &\le \sqrt{2\pi\rho^2}\cdot 2\sum_{n=1}^{\infty} e^{-2\pi^2\rho^2n^2}\cdot 2 \\ &= 4\sqrt{2\pi\rho^2}\cdot e^{-2\pi^2\rho^2}\cdot\sum_{n=1}^{\infty} e^{-2\pi^2\rho^2(n^2-1)} \\ &\le 4\sqrt{2\pi\rho^2}\cdot e^{-2\pi^2\rho^2}\cdot\sum_{n=1}^{\infty} e^{-6\pi^2\rho^2(n-1)} \quad (n^2-1=(n+1)(n-1)\ge 3(n-1).) \\ &= 4\sqrt{2\pi\rho^2}\cdot e^{-2\pi^2\rho^2}\cdot\frac{1}{1-e^{-6\pi^2\rho^2}} \\ &\le 4\frac{e^{-2\pi^2\rho^2}}{1-e^{-6\pi^2\rho^2}}\cdot g_{\rho^2}(0). \qquad (g_{\rho^2}(0)\ge\sqrt{2\pi\rho^2}.) \end{aligned}$$

Thus we obtain the bound

$$1 - 4\frac{e^{-2\pi^2\rho^2}}{1-e^{-6\pi^2\rho^2}} \le \frac{g_{\rho^2}(t)}{g_{\rho^2}(0)} \le 1.$$

For any $z \in \mathbb{Z}$, we have

$$\begin{aligned}\frac{\mathbb{P}[Z=z]}{\mathbb{P}[W=z]} &= \frac{g_{(1/\sigma^2+1/\tau^2)^{-1}}(0)\cdot\sum_{w\in\mathbb{Z}} e^{-w^2/2(\sigma^2+\tau^2)}}{\left(\sum_{u\in\mathbb{Z}} e^{-u^2/2\sigma^2}\right)\cdot\left(\sum_{v\in\mathbb{Z}} e^{-v^2/2\tau^2}\right)} \cdot \frac{g_{(1/\sigma^2+1/\tau^2)^{-1}}\left(\frac{\sigma^2}{\sigma^2+\tau^2}z\right)}{g_{(1/\sigma^2+1/\tau^2)^{-1}}(0)} \\ &= c(\sigma^2,\tau^2)\cdot\frac{g_{(1/\sigma^2+1/\tau^2)^{-1}}\left(\frac{\sigma^2}{\sigma^2+\tau^2}z\right)}{g_{(1/\sigma^2+1/\tau^2)^{-1}}(0)} \\ &\in \left[c(\sigma^2,\tau^2)\cdot\left(1-4\frac{e^{-2\pi^2/(1/\sigma^2+1/\tau^2)}}{1-e^{-6\pi^2/(1/\sigma^2+1/\tau^2)}}\right), c(\sigma^2,\tau^2)\right].\end{aligned}$$

Note that this interval is independent of z. Here $c(\sigma^2,\tau^2)$ is an appropriate constant.

The interval must contain 1, since Z and W are both probability distributions. Thus $c(\sigma^2,\tau^2) \ge 1$ and $c(\sigma^2,\tau^2)\cdot\left(1-4\frac{e^{-2\pi^2/(1/\sigma^2+1/\tau^2)}}{1-e^{-6\pi^2/(1/\sigma^2+1/\tau^2)}}\right) \le 1$, whence, for all $z \in \mathbb{Z}$,

$$1-4\frac{e^{-2\pi^2/(1/\sigma^2+1/\tau^2)}}{1-e^{-6\pi^2/(1/\sigma^2+1/\tau^2)}} \le \frac{\mathbb{P}[Z=z]}{\mathbb{P}[W=z]} \le \frac{1}{1-4\frac{e^{-2\pi^2/(1/\sigma^2+1/\tau^2)}}{1-e^{-6\pi^2/(1/\sigma^2+1/\tau^2)}}}$$

and

$$\left|\log\left(\frac{\mathbb{P}[Z=z]}{\mathbb{P}[W=z]}\right)\right| \le -\log\left(1-4\frac{e^{-2\pi^2/(1/\sigma^2+1/\tau^2)}}{1-e^{-6\pi^2/(1/\sigma^2+1/\tau^2)}}\right) \le 5\cdot e^{-2\pi^2/(1/\sigma^2+1/\tau^2)},$$

as long as $1/\sigma^2 + 1/\tau^2 \le 8$. □

Theorem 11 can easily be extended to sums of more than two discrete Gaussians by induction:

Corollary 12 (Convolution of Many Discrete Gaussians)**.** *Let $\sigma \ge \frac{1}{2}$. Let $X_i \leftarrow \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)$ independently for each i. Let $Z_n = \sum_i^n X_i$. Let $W_n \leftarrow \mathcal{N}_{\mathbb{Z}}(0, n\cdot\sigma^2)$. Then*

$$\mathrm{D}_{\pm\infty}(Z_n\|W_n) = \sup_{z\in\mathbb{Z}}\left|\log\left(\frac{\mathbb{P}[Z_n=z]}{\mathbb{P}[W_n=z]}\right)\right| \le 5\cdot\sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}} \le 5(n-1)e^{-\pi^2\sigma^2}. \tag{14}$$

Proof. Let $\tilde{Z}_n = W_{n-1} + X_n$. (Note that $Z_n = Z_{n-1} + X_n$.) By the triangle inequality and postprocessing,

$$\begin{aligned}\mathrm{D}_{\pm\infty}(Z_n\|W_n) &\le \mathrm{D}_{\pm\infty}\left(Z_n\middle\|\tilde{Z}_n\right) + \mathrm{D}_{\pm\infty}\left(\tilde{Z}_n\middle\|W_n\right) \\ &\le \mathrm{D}_{\pm\infty}(Z_{n-1}\|W_{n-1}) + \mathrm{D}_{\pm\infty}\left(\tilde{Z}_n\middle\|W_n\right).\end{aligned}$$

By Theorem 11,

$$\mathrm{D}_{\pm\infty}\left(\tilde{Z}_n \middle\| W_n\right) \le 5 \cdot e^{-2\pi^2/(1/\sigma^2+1/(n-1)\sigma^2)} = 5 \cdot e^{-2\pi^2\sigma^2(n-1)/n}.$$

The result now follows by induction; the base case $n = 1$ is trivial. □

We can now use the triangle inequality to combine our convolution closeness results with the privacy guarantee of a single discrete Gaussian to obtain a privacy guarantee for sums of discrete Gaussians:

Proposition 13 (Privacy for Sums of Discrete Gaussians)**.** *Let $\sigma \ge \frac{1}{2}$. Let $X_i \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ independently for each i. Let $Z_n = \sum_i^n X_i$. Then, for all $\Delta \in \mathbb{Z}$ and all $\alpha \in [1, \infty)$,*

$$\mathrm{D}_\alpha\left(Z_n \| Z_n + \Delta\right) \le \min\left\{ \frac{\alpha\Delta^2}{2n\sigma^2} + 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}}, \frac{\alpha}{2} \cdot \left(\frac{|\Delta|}{\sqrt{n}\sigma} + 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}} \right)^2 \right\}. \tag{15}$$

That is, an algorithm M that adds Z_n to a sensitivity-Δ query satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy for $\varepsilon = \min\left\{ \sqrt{\frac{\Delta^2}{n\sigma^2} + 5 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}}}, \frac{|\Delta|}{\sqrt{n}\sigma} + 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}} \right\}$.

To make the above bound concrete, if $\sigma = \Delta = 1$ and $n = 10^4$, then $\varepsilon < 0.02$.

Proof. Let $W \leftarrow \mathcal{N}_{\mathbb{Z}}(0, n \cdot \sigma^2)$. By the triangle inequality,

$$\mathrm{D}_\alpha\left(Z_n \| Z_n + \Delta\right) \le \min\left\{ \begin{array}{c} \mathrm{D}_\infty\left(Z_n \| W\right) + \mathrm{D}_\alpha\left(W \| W + \Delta\right) + \mathrm{D}_\infty\left(W + \Delta \| Z_n + \Delta\right), \\ \alpha \cdot \left(\sqrt{\mathrm{D}_*\left(Z_n \| W\right)} + \sqrt{\mathrm{D}_*\left(W \| W + \Delta\right)} + \sqrt{\mathrm{D}_*\left(W + \Delta \| Z_n + \Delta\right)} \right)^2 \end{array} \right\}.$$

By Theorem 10, $\mathrm{D}_*\left(W \| W + \Delta\right) \le \frac{\Delta^2}{2n\sigma^2}$. By Corollary 12,

$$\mathrm{D}_{\pm\infty}\left(Z_n \| W\right) = \mathrm{D}_{\pm\infty}\left(W + \Delta \| Z_n + \Delta\right) \le 5 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2\frac{k}{k+1}}.$$

By Lemma 4, $\mathrm{D}_*\left(Z_n \| W\right) \le \frac{1}{2}(\mathrm{D}_{\pm\infty}\left(Z_n \| W\right))^2$. Combining yields the result. □

Finally, we extend Proposition 13 to the multidimensional setting using the composition property:

Proposition 14 (Privacy for Sums of Multidimensional Discrete Gaussians)**.** *Let $\sigma \ge \frac{1}{2}$. Let $X_{i,j} \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ independently for each i and j. Let $X_i = (X_{i,1}, \cdots, X_{i,d}) \in \mathbb{Z}^d$. Let $Z_n = \sum_i^n X_i \in \mathbb{Z}^d$. Then, for all $\Delta \in \mathbb{Z}^d$ and all $\alpha \in [1, \infty)$,*

$$\mathrm{D}_\alpha\left(Z_n \| Z_n + \Delta\right) \le \min\left\{ \begin{array}{c} \frac{\alpha\|\Delta\|_2^2}{2n\sigma^2} + \tau \cdot d, \\ \frac{\alpha}{2} \cdot \left(\frac{\|\Delta\|_2^2}{n\sigma^2} + 2\frac{\|\Delta\|_1}{\sqrt{n}\sigma} \cdot \tau + \tau^2 \cdot d \right), \\ \frac{\alpha}{2} \cdot \left(\frac{\|\Delta\|_2}{\sqrt{n}\sigma} + \tau \cdot \sqrt{d} \right)^2 \end{array} \right\}, \tag{16}$$

where $\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2 \frac{k}{k+1}}$. *An algorithm* M *that adds* Z_n *to a query with* ℓ_p *sensitivity* Δ_p *satisfies* $\frac{1}{2}\varepsilon^2$*-concentrated differential privacy for*

$$\varepsilon = \min \left\{ \begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \sqrt{\frac{\Delta_2^2}{n\sigma^2} + 2\frac{\Delta_1}{\sqrt{n}\sigma} \cdot \tau + \tau^2 d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}. \tag{17}$$

Proof. This follows from Proposition 13 and summing over coordinates. Note that before summing we expand

$$\sum_i \left(\frac{|\Delta_i|}{\sqrt{n}\sigma} + 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2 \frac{k}{k+1}} \right)^2 = \sum_i \frac{\Delta_i^2}{n\sigma^2} + 2\frac{|\Delta_i|}{\sqrt{n}\sigma}\tau + \tau^2 = \frac{\|\Delta\|_2^2}{n\sigma^2} + 2\frac{\|\Delta\|_1}{\sqrt{n}\sigma}\tau + \tau^2 \cdot d.$$

To obtain the third expression we apply the bound $\|\Delta\|_1 \leq \sqrt{d} \cdot \|\Delta\|_2$ and complete the square again. □

Finally, we state a utility bound for the discrete Gaussian.

Lemma 15 (Utility of the Discrete Gaussian)**.** *Let* $X \leftarrow \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$*. Then* $\mathbb{E}[X] = 0$ *and* $\mathsf{Var}[X] = \mathbb{E}[X^2] < \sigma^2$*. For all* $t \in \mathbb{R}$*,* $\mathbb{E}\left[e^{tX}\right] \leq e^{t^2\sigma^2/2}$*.*

4 Theoretical Utility Analysis

We now delve into the accuracy analysis of our algorithm. There are three sources of error that we must account for: (i) discretization via (conditional) randomized rounding, (ii) the noise added for privacy (which depends on the norm of the *discretized* vector), and (iii) the modular clipping operation. We address these concerns one at a time.

4.1 Randomized Rounding

In order to apply discrete noise, we must first round the input vectors to the discrete grid. We must analyze the error (both bias and variance) that this introduces, and also ensure that it doesn't increase the sensitivity too much. That is, the rounded vector may have larger norm than the original vector, and we must control this.

We begin by defining the randomized rounding operation:

Definition 16 (Randomized Rounding)**.** *Let* $\gamma > 0$ *and* $d \in \mathbb{N}$*. Define* $R_\gamma : \mathbb{R}^d \to \gamma\mathbb{Z}^d$ *(where* $\gamma\mathbb{Z}^d := \{(\gamma z_1, \gamma z_2, \cdots, \gamma z_d) : z_1, \cdots, z_d \in \mathbb{Z}\} \subset \mathbb{R}^d$*) as follows. For* $x \in [0, \gamma]^d$*,* $R_\gamma(x)$ *is a product distribution on* $\{0, \gamma\}^d$ *with mean* x*; that is, indepdently for each* $i \in [d]$*, we have* $\mathbb{P}[R_\gamma(x)_i = 0] = 1 - x_i/\gamma$ *and* $\mathbb{P}[R_\gamma(x)_i = \gamma] = x_i/\gamma$*. In general, for* $x \in \mathbb{R}^d$*, we have* $R_\gamma(x) = \gamma\lfloor x/\gamma \rfloor + R_\gamma(x - \gamma\lfloor x/\gamma \rfloor)$*; here* $\gamma\lfloor x/\gamma \rfloor \in \gamma\mathbb{Z}^d$ *is the point* x *rounded down coordinate-wise to the grid.*

We first look at how randomized rounding impacts the norm. It is easy to show that

$$\mathbb{P}\left[\|R_\gamma(x) - x\|_p \le \gamma \cdot d^{1/p}\right] = 1 \tag{18}$$

for all $p \in [1, \infty]$. This bound may be sufficient for many purposes, but, if we relax the probability 1 requirement, we can do better (by constant factors), as demonstrated by the following lemma.

Lemma 17 (Norm of Randomized Rounding)**.** *Let $\gamma > 0$ and $x \in \mathbb{R}^d$. Let R_γ be as in Definition 16. Then $\mathbb{E}[R_\gamma(x)] = x$, $\mathbb{E}[\|R_\gamma(x)\|_1] = \|x\|_1$, and*

$$\mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] = \|x\|_2^2 + \gamma\|y\|_1 - \|y\|_2^2 \le \|x\|_2^2 + \frac{1}{4}\gamma^2 d, \tag{19}$$

where $y := x - \gamma\lfloor x/\gamma \rfloor \in [0, \gamma]^d$. Furthermore, for any $\beta \in (0, 1)$, we have

$$\mathbb{P}\left[\|R_\gamma(x)\|_2^2 \le \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] + \sqrt{2\log(1/\beta)} \cdot \gamma \cdot \left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right)\right] \ge 1 - \beta \tag{20}$$

and

$$\mathbb{P}\left[\|R_\gamma(x)\|_1 \le \|x\|_1 + \gamma \cdot \sqrt{\frac{1}{2} d \log(1/\beta)}\right] \ge 1 - \beta. \tag{21}$$

Proof. Define $\underline{x}, \overline{x} \in \gamma\mathbb{Z}^d$ by $\underline{x}_i = \gamma\lfloor x_i/\gamma \rfloor$ and $\overline{x}_i = \underline{x}_i + \gamma$. Then $y = x - \underline{x} \in [0, \gamma]^d$.

Fix some $i \in [d]$. By definition, $\mathbb{P}[R_\gamma(x)_i = \underline{x}_i] = 1 - y_i/\gamma$ and $\mathbb{P}[R_\gamma(x)_i = \underline{x}_i + \gamma] = y_i/\gamma$. Thus $\mathbb{E}[R_\gamma(x)_i] = \underline{x}_i + y_i = x_i$, $\mathbb{E}[|R_\gamma(x)_i|] = |x_i|$, and

$$\mathbb{E}\left[R_\gamma(x)_i^2\right] = (1 - y_i/\gamma)\underline{x}_i^2 + (y_i/\gamma)(\underline{x}_i + \gamma)^2 = (\underline{x}_i + y_i)^2 + \gamma y_i - y_i^2 = x_i^2 + \gamma y_i - y_i^2.$$

Note that $\gamma y_i - y_i^2 \le \gamma(\gamma/2) - (\gamma/2)^2 = \gamma^2/4$ for all values of $y_i \in [0, \gamma]$. Summing over $i \in [d]$ gives $\mathbb{E}[R_\gamma(x)] = x$, $\mathbb{E}[\|R_\gamma(x)\|_1] = \|x\|_1$, and $\mathbb{E}[\|R_\gamma(x)\|_2^2] = \|x\|_2^2 + \gamma\|y\|_1 + \|y\|_2^2 \le \|x\|_2^2 + \gamma^2 d/4$, as required.

Fix some $i \in [d]$ and $t, \lambda \geq 0$. By Hoeffding's lemma,

$$\begin{aligned}
\mathbb{E}\left[\exp(t \cdot R_\gamma(x)_i^2)\right] &= (1 - y_i/\gamma) \cdot e^{t\underline{x}_i^2} + (y_i/\gamma) \cdot e^{t \cdot \overline{x}_i^2} \\
&\leq \exp\left(t \cdot \mathbb{E}\left[R_\gamma(x)_i^2\right] + \frac{t^2}{8}(\overline{x}_i^2 - \underline{x}_i^2)^2\right) \\
&= \exp\left(t \cdot \mathbb{E}\left[R_\gamma(x)_i^2\right] + \frac{t^2}{8}(2\gamma \underline{x}_i + \gamma^2)^2\right) \\
&\leq \exp\left(t \cdot \mathbb{E}\left[R_\gamma(x)_i^2\right] + \frac{t^2\gamma^2}{2}\left(x_i^2 + \gamma|x_i| + \frac{1}{4}\gamma^2\right)\right), \\
\mathbb{E}\left[\exp(t \cdot \|R_\gamma(x)\|_2^2)\right] &= \prod_i^d \mathbb{E}\left[\exp(t \cdot R_\gamma(x)_i^2)\right] \\
&\leq \exp\left(t \cdot \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] + \frac{t^2\gamma^2}{2}\left(\|x\|_2^2 + \gamma\|x\|_1 + \frac{1}{4}\gamma^2 d\right)\right) \\
&\leq \exp\left(t \cdot \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] + \frac{t^2\gamma^2}{2}\left(\|x\|_2^2 + \gamma\sqrt{d}\|x\|_2 + \frac{1}{4}\gamma^2 d\right)\right) \\
&= \exp\left(t \cdot \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] + \frac{t^2\gamma^2}{2}\left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right)^2\right), \\
\mathbb{P}\left[\|R_\gamma(x)\|_2^2 \geq \lambda\right] &= \mathbb{P}\left[\exp(t \cdot (\|R_\gamma(x)\|_2^2 - \lambda)) \geq 1\right] \\
&\leq \mathbb{E}\left[\exp(t \cdot (\|R_\gamma(x)\|_2^2 - \lambda))\right] \\
&\leq \exp\left(t \cdot \left(\mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] - \lambda\right) + \frac{t^2\gamma^2}{2}\left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right)^2\right).
\end{aligned}$$

Setting $t = \frac{\lambda - \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right]}{\gamma^2\left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right)^2}$ and $\lambda = \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] + \sqrt{2\gamma^2(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d})^2 \log(1/\beta)}$ gives

$$\mathbb{P}\left[\|R_\gamma(x)\|_2^2 \geq \lambda\right] \leq \exp\left(\frac{-(\lambda - \mathbb{E}\left[\|R_\gamma(x)\|_2^2\right])^2}{2\gamma^2\left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right)^2}\right) \leq \beta.$$

Fix some $i \in [d]$ and $t, \lambda \geq 0$. Assume, without loss of generality, that $x_i \geq 0$. By Hoeffding's lemma,

$$\begin{aligned}
\mathbb{E}\left[\exp\left(t \cdot |R_\gamma(x_i)|\right)\right] &= (1 - y_i/\gamma) \cdot e^{t \cdot \underline{x}_i} + (y_i/\gamma) \cdot e^{t \cdot \overline{x}_i} \\
&\leq \exp\left(t \cdot x_i + \frac{t^2\gamma^2}{8}\right), \\
\mathbb{E}\left[\exp(t \cdot \|R_\gamma(x)\|_1)\right] &\leq \exp\left(t \cdot \|x\|_1 + \frac{t^2\gamma^2}{8} \cdot d\right), \\
\mathbb{P}\left[\|R_\gamma(x)\|_1 \geq \lambda\right] &\leq \exp\left(t \cdot \|x\|_1 + \frac{t^2\gamma^2}{8} \cdot d - t \cdot \lambda\right).
\end{aligned}$$

Setting $t = 4\frac{\lambda - \|x\|_1}{\gamma^2 d}$ and $\lambda = \|x\|_1 + \gamma\sqrt{\frac{1}{2}d\log(1/\beta)}$ gives

$$\mathbb{P}\left[\|R_\gamma(x)\|_1 \geq \lambda\right] \leq \exp\left(-2\frac{(\lambda - \|x\|_1)^2}{\gamma^2 d}\right) \leq \beta.$$

□

Remark 18. *The expectation and high probability bounds of Lemma 17 are only a constant factor better than the worst-case bound* (18)*. Namely, Lemma 17 gives the bound*

$$\mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] = \|x\|_2^2 + \gamma\|y\|_1 - \|y\|_2^2 \leq \|x\|_2^2 + \frac{1}{4}\gamma^2 d,$$

whereas the worst case bound is

$$\|R_\gamma(x)\|_2^2 \leq \left(\|x\|_2 + \gamma\sqrt{d}\right)^2 \leq 2\|x\|_2^2 + 2\gamma^2 d.$$

Nevertheless, constant factor improvements matter in practical systems. However, hopefully, γ is sufficiently small that the increase in norm from randomized rounding is entirely negligible, even if we apply the worst-case bound.

Remark 19. *In Lemma 17, the inequality $\mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] = \|x\|_2^2 + \gamma\|y\|_1 - \|y\|_2^2 \leq \|x\|_2^2 + \frac{1}{4}\gamma^2 d$ is tight when $y = x - \gamma\lfloor x/\gamma \rfloor$ has all entries being $\gamma/2$. However, if $Y \in [0,\gamma]^d$ is uniformly random, then $\mathbb{E}\left[\gamma\|Y\|_1 - \|Y\|_2^2\right] = \frac{1}{6}\gamma^2 d$. Consequently, if we were to round not to $\gamma\mathbb{Z}^d$ but to a randomly translated grid (i.e., $\gamma\mathbb{Z}^d + U$ for a uniformly random $U \in [0,\gamma]^d$), then this error term can be reduced; the shift U can also be released without compromising privacy. We do not explore this direction further.*

Lemma 17 shows that, with high probability, randomized rounding will not increase the norm too much. We could use this directly as the basis of a privacy guarantee – the probability of the norm being too large would correspond to some kind of privacy failure probability. Instead what we will do is, if the norm is too large, we simply fix that – namely, by resampling the randomized rounding procedure. That is, instead of accepting a small probability of privacy failing, we accept a small probability of inaccuracy.

Definition 20 (Conditional Randomized Rounding)**.** *Let $\gamma > 0$ and $d \in \mathbb{N}$ and $G \subset \mathbb{R}^d$. Define $R_\gamma^G : \mathbb{R}^d \to \gamma\mathbb{Z}^d \cap G$ to be R_γ conditioned on the output being in G. That is, $\mathbb{P}\left[R_\gamma^G(x) = y\right] = \mathbb{P}\left[R_\gamma(x) = y\right]/\mathbb{P}\left[R_\gamma(x) \in G\right]$ for all $y \in \gamma\mathbb{Z}^d \cap G$, where R_γ is as in Definition 16.*

To implement conditional randomized rounding, we simply re-run $R_\gamma(x)$ again and again until it generates a point in G and then we output that. The expected number of times we must run the randomized rounding to get a point in G is $\mathbb{P}\left[R_\gamma(x) \in G\right]^{-1}$. Thus it is important to ensure that $\mathbb{P}\left[R_\gamma(x) \in G\right]$ is not too small.

Now we give a lemma that bounds the error of conditional randomized rounding.

Lemma 21 (Error of Conditional Randomized Rounding)**.** *Let $\gamma > 0$, $x \in \mathbb{R}^d$, and $G \subset \mathbb{R}^d$. Let $1 - \beta = \mathbb{P}\left[R_\gamma(x) \in G\right] > 0$. Then*

$$\left\|\mathbb{E}\left[R_\gamma^G(x)\right] - x\right\|_2 \leq \frac{\beta \cdot \gamma \cdot \sqrt{d}}{1-\beta} \tag{22}$$

and

$$\mathbb{E}\left[\left\|R_\gamma^G(x) - \mathbb{E}\left[R_\gamma^G(x)\right]\right\|_2^2\right] \leq \mathbb{E}\left[\left\|R_\gamma^G(x) - x\right\|_2^2\right] \leq \frac{\gamma^2 d}{4(1-\beta)}. \tag{23}$$

For all $t \in \mathbb{R}^d$,

$$\mathbb{E}\left[\exp(\langle t, R_\gamma^G(x) - x\rangle)\right] \leq \frac{\exp\left(\frac{\gamma^2}{8} \cdot \|t\|_2^2\right)}{1-\beta}. \tag{24}$$

Proof. For an arbitrary random variable X and nontrivial event E, we have

$$\mathbb{E}\left[X\right] = \mathbb{E}\left[X|E\right]\mathbb{P}\left[E\right] + \mathbb{E}\left[X|\overline{E}\right]\mathbb{P}\left[\overline{E}\right], \tag{25}$$

which rearranges to give

$$\mathbb{E}\left[X|E\right] = \frac{\mathbb{E}\left[X\right] - \mathbb{E}\left[X|\overline{E}\right]\mathbb{P}\left[\overline{E}\right]}{\mathbb{P}\left[E\right]} \tag{26}$$

and

$$\mathbb{E}\left[X|E\right] - \mathbb{E}\left[X\right] = \frac{\mathbb{P}\left[\overline{E}\right]}{\mathbb{P}\left[E\right]}\left(\mathbb{E}\left[X\right] - \mathbb{E}\left[X|\overline{E}\right]\right). \tag{27}$$

Thus

$$\begin{aligned}
\left\|\mathbb{E}\left[R_\gamma^G(x)\right] - x\right\|_2 &= \left\|\mathbb{E}\left[R_\gamma(x)|R_\gamma(x) \in G\right] - \mathbb{E}\left[R_\gamma(x)\right]\right\|_2 \\
&= \frac{\mathbb{P}\left[R_\gamma(x) \notin G\right]}{\mathbb{P}\left[R_\gamma(x) \in G\right]}\left\|\mathbb{E}\left[R_\gamma(x)\right] - \mathbb{E}\left[R_\gamma(x)|R_\gamma(x) \notin G\right]\right\|_2 \\
&= \frac{\beta}{1-\beta}\|x - x^*\|_2,
\end{aligned}$$

where $x^* \in \gamma\lfloor x/\gamma \rfloor + [0,\gamma]^d$ and, hence, $\|x - x^*\|_2 \leq \gamma \cdot \sqrt{d}$.

Next, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|R_\gamma^G(x) - x\right\|_2^2\right] &= \frac{\mathbb{E}\left[\|R_\gamma(x) - x\|_2^2\right] - \mathbb{E}\left[\|R_\gamma(x) - x\|_2^2 | R_\gamma(x) \notin G\right]\mathbb{P}\left[R_\gamma(x) \notin G\right]}{\mathbb{P}\left[R_\gamma(x) \in G\right]} \\
&\leq \frac{\mathbb{E}\left[\|R_\gamma(x) - x\|_2^2\right]}{\mathbb{P}\left[R_\gamma(x) \in G\right]} \\
&= \frac{\gamma\|y\|_1 - \|y\|_2^2}{1-\beta} && \text{(Lemma 17)} \\
&\leq \frac{\gamma^2 d}{4(1-\beta)}.
\end{aligned}$$

Note that the apply the bias variance decomposition: Since $\mathbb{E}\left[R_\gamma(x)\right] = x$, we have $\mathbb{E}\left[\|R_\gamma(x)\|_2^2\right] = \mathbb{E}\left[\|R_\gamma(x) - x\|_2^2\right] + \|x\|_2^2$. Similarly,

$$\mathbb{E}\left[\left\|R_\gamma^G(x) - x\right\|_2^2\right] = \mathbb{E}\left[\left\|R_\gamma^G(x) - \mathbb{E}\left[R_\gamma^G(x)\right]\right\|_2^2\right] + \left\|\mathbb{E}\left[R_\gamma^G(x)\right] - x\right\|_2^2 \geq \mathbb{E}\left[\left\|R_\gamma^G(x) - \mathbb{E}\left[R_\gamma^G(x)\right]\right\|_2^2\right].$$

By Hoeffding's lemma, since $R_\gamma(x) \in \gamma\lfloor x/\gamma \rfloor + \{0,\gamma\}^d$ and is a product distribution with mean x, we have

$$\forall t \in \mathbb{R}^d \qquad \mathbb{E}\left[\exp(\langle t, R_\gamma(x) - x\rangle)\right] \leq \exp\left(\frac{\gamma^2}{8} \cdot \|t\|_2^2\right). \tag{28}$$

Thus

$$\forall t \in \mathbb{R}^d \qquad \mathbb{E}\left[\exp(\langle t, R_\gamma^G(x) - x\rangle)\right] \leq \frac{\mathbb{E}\left[\exp(\langle t, R_\gamma(x) - x\rangle)\right]}{\mathbb{P}\left[R_\gamma(x) \in G\right]} \leq \frac{\exp\left(\frac{\gamma^2}{8} \cdot \|t\|_2^2\right)}{1-\beta}.$$

□

We summarize the results of this section. First we give a proposition for a single instance of randomized rounding (this combines Lemma 17 and 21).

Proposition 22 (Properties of Randomized Rounding)**.** *Let $\beta \in [0,1)$, $\gamma > 0$, and $x \in \mathbb{R}^d$. Let*

$$\Delta_2^2 := \min\left\{ \begin{array}{c} \|x\|_2^2 + \frac{1}{4}\gamma^2 d + \sqrt{2\log(1/\beta)} \cdot \gamma \cdot \left(\|x\|_2 + \frac{1}{2}\gamma\sqrt{d}\right), \\ \left(\|x\|_2 + \gamma\sqrt{d}\right)^2 \end{array} \right\} \tag{29}$$

and $G := \left\{y \in \mathbb{R}^d : \|y\|_2^2 \leq \Delta_2^2\right\}$. Let $R_\gamma(x)$ and $R_\gamma^G(x)$ be as in Definitions 16 and 20.

Then $\mathbb{P}\left[R_\gamma(x) \in G\right] \geq 1 - \beta$ and, consequently, the following hold.

$$\left\|\mathbb{E}\left[R_\gamma^G(x)\right] - x\right\|_2 \leq \frac{\beta \cdot \gamma \cdot \sqrt{d}}{1-\beta}. \tag{30}$$

$$\mathbb{E}\left[\left\|R_\gamma^G(x) - \mathbb{E}\left[R_\gamma^G(x)\right]\right\|_2^2\right] \leq \frac{\gamma^2 d}{4(1-\beta)}. \tag{31}$$

$$\forall t \in \mathbb{R}^d \qquad \mathbb{E}\left[\exp(\langle t, R_\gamma^G(x) - x\rangle)\right] \leq \frac{\exp\left(\frac{\gamma^2}{8} \cdot \|t\|_2^2\right)}{1-\beta}. \tag{32}$$

$$\mathbb{P}\left[\|R_\gamma^G(x)\|_2^2 \leq \Delta_2^2\right] = 1 = \mathbb{P}\left[R_\gamma^G(x) \in \gamma\mathbb{Z}^d\right]. \tag{33}$$

Now we give a proposition for sums of randomized roundings.

Proposition 23 (Randomized Rounding & Sums)**.** *Let $\beta \in [0,1)$, $\gamma > 0$, and $x_1, \cdots, x_n \in \mathbb{R}^d$. Suppose $\|x_i\|_2 \leq c$ for all $i \in [n]$. Let*

$$\Delta_2^2 := \min\left\{ \begin{array}{c} c^2 + \frac{1}{4}\gamma^2 d + \sqrt{2\log(1/\beta)} \cdot \gamma \cdot \left(c + \frac{1}{2}\gamma\sqrt{d}\right), \\ \left(c + \gamma\sqrt{d}\right)^2 \end{array} \right\} \tag{34}$$

and $G := \left\{y \in \mathbb{R}^d : \|y\|_2^2 \le \Delta_2^2\right\}$. *Let* R_γ^G *be as in Definition 20.*

Then the following hold.

$$\left\| \mathbb{E}\left[\sum_i^n R_\gamma^G(x_i)\right] - \sum_i^n x_i \right\|_2 \le \frac{\beta \cdot \gamma \cdot \sqrt{d} \cdot n}{1-\beta}. \tag{35}$$

$$\mathbb{E}\left[\left\| \sum_i^n R_\gamma^G(x_i) - \mathbb{E}\left[\sum_i^n R_\gamma^G(x_i)\right] \right\|_2^2\right] \le \frac{\gamma^2 \cdot d \cdot n}{4(1-\beta)}. \tag{36}$$

$$\mathbb{E}\left[\left\| \sum_i^n R_\gamma^G(x_i) - \sum_i^n x_i \right\|_2^2\right] \le \frac{\gamma^2 \cdot d \cdot n}{4(1-\beta)} + \left(\frac{\beta}{1-\beta}\gamma\sqrt{d}n\right)^2. \tag{37}$$

$$\forall t \in \mathbb{R}^d \qquad \mathbb{E}\left[\exp\left(\left\langle t, \sum_i^n R_\gamma^G(x_i) - \sum_i^n x_i \right\rangle\right)\right] \le \frac{\exp\left(\frac{\gamma^2}{8} \cdot \|t\|_2^2 \cdot n\right)}{(1-\beta)^n}. \tag{38}$$

$$\mathbb{P}\left[\forall i \quad \|R_\gamma^G(x_i)\|_2^2 \le \Delta_2^2\right] = 1 = \mathbb{P}\left[\forall i \quad R_\gamma^G(x_i) \in \gamma\mathbb{Z}^d\right]. \tag{39}$$

Remark 24. *Proposition 23 provides some guidance on how to set the parameter* β. *We have the mean squared error bound* (37)

$$\mathbb{E}\left[\left\| \sum_i^n R_\gamma^G(x_i) - \sum_i^n x_i \right\|_2^2\right] \le \frac{\gamma^2 \cdot d \cdot n \cdot (1 - \beta + 4\beta^2 n)}{4(1-\beta)^2}.$$

If we set $\beta \approx 1/\sqrt{n}$, *then the bias and variance terms in the bound are of the same order. Setting* β *too small would needlessly increase the sensitivity* Δ_2. *And we see that there is little value in setting* $\beta \ll 1/\sqrt{n}$. *So the theory suggests setting* $\beta \approx 1/\sqrt{n}$.

However, we emphasize that this is a worst-case upper bound on the error and it is likely that, in practice, the error would likely be considerably less. Thus it is justifiable to set β *to be considerably larger – e.g.,* $\beta = e^{-1/2}$ *– and simply hope for the best in terms of accuracy.*

Remark 25. *Proposition 23 covers the error introduced by discretization. Obviously, reducing the granularity* γ *will reduce the discretization error. However, this comes at a cost in communication, so the choice of this parameter will need to be carefully made.*

In Section 3 we have covered the noise that is injected to preserve privacy and in Section 4.1 we have covered the error introduced by discretizing the data. Now we state a result that combines these.

Proposition 26 (Randomized Rounding + Discrete Gaussian)**.** *Let* $\beta \in [0, 1)$, $\sigma^2 \ge \frac{1}{2}\gamma > 0$,

and $c > 0$. Let

$$\Delta_2^2 := \min\left\{ \begin{array}{c} c^2 + \frac{1}{4}\gamma^2 d + \sqrt{2\log(1/\beta)} \cdot \gamma \cdot \left(c + \frac{1}{2}\gamma\sqrt{d}\right), \\ \left(c + \gamma\sqrt{d}\right)^2 \end{array} \right\}, \tag{40}$$

$$G := \left\{y \in \mathbb{R}^d : \|y\|_2^2 \le \Delta_2^2\right\}, \tag{41}$$

$$\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2 \frac{\sigma^2}{\gamma^2} \cdot \frac{k}{k+1}}, \tag{42}$$

$$\varepsilon := \min\left\{ \begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}. \tag{43}$$

Let R_γ^G be as in Definition 20. Define a randomized algorithm $A : (\mathbb{R}^d)^n \to \gamma\mathbb{Z}^d$ by[6]

$$A(x) = \sum_i^n R_\gamma^G\left(\min\left\{1, \frac{c}{\|x_i\|_2}\right\} \cdot x_i\right) + \gamma \cdot Y_i, \tag{44}$$

where $Y_1, \cdots, Y_n \in \mathbb{Z}^d$ are independent random vectors with each entry drawn independently from $\mathcal{N}_\mathbb{Z}(0, \sigma^2/\gamma^2)$.

Then A satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy.[7]

Let $x_1, \cdots, x_n \in \mathbb{R}^d$ with $\|x_i\|_2 \le c$ for all $i \in [n]$. Then the following hold.

$$\left\|\mathbb{E}\left[A(x)\right] - \sum_i^n x_i\right\|_2 \le \frac{\beta \cdot \gamma \cdot \sqrt{d} \cdot n}{1-\beta}. \tag{45}$$

$$\mathbb{E}\left[\|A(x) - \mathbb{E}\left[A(x)\right]\|_2^2\right] \le \frac{\gamma^2 \cdot d \cdot n}{4(1-\beta)} + n \cdot d \cdot \sigma^2. \tag{46}$$

$$\mathbb{E}\left[\left\|A(x) - \sum_i^n x_i\right\|_2^2\right] \le \frac{\gamma^2 \cdot d \cdot n}{4(1-\beta)} + \left(\frac{\beta}{1-\beta}\gamma\sqrt{d}n\right)^2 + n \cdot d \cdot \sigma^2 \tag{47}$$

$$\forall t \in \mathbb{R}^d \quad \mathbb{E}\left[\exp\left(\left\langle t, A(x) - \sum_i^n x_i\right\rangle\right)\right] \le \frac{\exp\left(\left(\frac{\gamma^2}{8} + \frac{\sigma^2}{2}\right) \cdot \|t\|_2^2 \cdot n\right)}{(1-\beta)^n}. \tag{48}$$

4.2 Flattening

It is possible that the inputs x_i and the sum $\bar{x} = \sum_i^n x_i$ are very heavily concentrated on one coordinate. This is a bad case, as the modular clipping will create a very large error,

[6]Note that $\min\left\{1, \frac{c}{\|x_i\|_2}\right\} \cdot x_i$ is simply x_i with it's norm clipped to c.

[7]Not that this is with respect to the addition or removal of an element, not replacement. To keep n fixed, we would need addition/removal to be defined to simply zero-out the relevant vectors.

unless we use a very large modulus. To avoid this problem we will "flatten" the inputs as a pre-processing step (which is inverted by the server at the end of the protocol).

Specifically, our goal is to pre-process the inputs $x_1, \cdots, x_n$ so that $\sum_i x_i \in [a,b]^d$ and then at the end we can undo the pre-processing to obtain the original value. Here $[a,b]$ is the range where modular arithmetic does not cause errors.

To flatten the vectors we will transform the data points by multiplying them by a (random) matrix $U \in \mathbb{R}^{d \times d}$; at the end we multiply by U^{-1} to undo this operation. We will take U to be a unitary matrix so that $U^{-1} = U^T$.[8]

Remark 27. *The flattening matrix U is* shared randomness *– that is, the server and all the clients must have access to this matrix. Fortunately, the differential privacy guarantee does not depend on this randomness remaining hidden; thus U can be published, and we do not need to worry about the privacy adversary having access to it.*

The specific property that we want the flattening matrix U to satisfy is that, for all $x \in \mathbb{R}^d$ and all $i \in [d]$, the value $(Ux)_i \in \mathbb{R}$ has a subgaussian distribution (determined by the randomness of U) with variance proxy $O(\|x\|_2^2/d)$.

There are many possibilities for this flattening transformation. A natural option is for U to be a random unitary matrix or rotation matrix. This would attain the desired property:

Lemma 28. *Let $x \in \mathbb{R}^d$ be fixed. Let $U \in \mathbb{R}^{d \times d}$ be a uniformly random unitary matrix or rotation matrix, so that Ux is a uniformly random vector with norm $\|x\|_2$. Then $\mathbb{E}\left[e^{t(Ux)_i}\right] \le e^{t^2\|x\|_2^2/2d}$ for all $t \in \mathbb{R}$ and all $i \in [d]$.*

Proof. Let $Y = ((Ux)_i/\|x\|_2 + 1)/2 \in [0,1]$. Then Y follows a $\mathsf{Beta}((d-1)/2, (d-1)/2)$ distribution [whu14]. Thus $\mathbb{E}\left[e^{t(Y-1/2)}\right] \le e^{t^2/8}$ for all $t \in \mathbb{R}$ [MA+17]. The result follows. □

Unfortunately, a random rotation or unitary matrix has several downsides: Generating a random unitary or rotation matrix is itself a non-trivial task [Mez06]. Even storing such a matrix in memory and performing the required matrix-vector multiplication could be prohibitive – it would take $\Theta(d^2)$ time and space where we seek algorithms that run in $\tilde{O}(d)$ time and space.

Instead our approach is to first randomize the signs of the entries of x and then multiply by a matrix with small entries. This attains the desired guarantee:

Lemma 29. *Let $H \in [-\sqrt{\rho/d}, \sqrt{\rho/d}]^{d \times d}$ be a fixed unitary matrix. Let $D \in \{0, -1, +1\}^{d \times d}$ be a diagonal matrix with random signs on the diagonal. Fix $x \in \mathbb{R}^d$ and $i \in [d]$. Let $Y = (HDx)_i \in \mathbb{R}$. Then $\mathbb{E}\left[e^{tY}\right] \le e^{t^2\|x\|_2^2\rho/2d}$ for all $t \in \mathbb{R}$.*

Proof. We have $Y = (HDx)_i = \sum_j^d H_{i,j}D_{j,j}x_j$ and, by independence, $\mathbb{E}\left[e^{tY}\right] = \prod_j^d \mathbb{E}\left[e^{tH_{i,j}D_{j,j}x_j}\right]$. Since $H_{i,j} \in [-\sqrt{\rho/d}, \sqrt{\rho/d}]$ and $D_{j,j} \in \{-1, +1\}$ is uniformly random, we have $\mathbb{E}\left[e^{tH_{i,j}D_{j,j}x_j}\right] \le e^{t^2(\sqrt{\rho/d})^2x_j^2/2}$ by Hoeffding's lemma. □

[8] We take U to be a square matrix, but it is in general possible to also increase the dimension during this pre-processing step. We also could extend beyond unitary matrices to invertible (and well-conditioned) matrices.

Generating the required random signs is easy. (We can use a pseudorandom generator and a small shared random seed.) We just need a unitary matrix $H \in [-\sqrt{\rho/d}, \sqrt{\rho/d}]^{d\times d}$ with ρ as small as possible.[9] And we want H to be easy to work with – specifically, we want efficient algorithms to compute the matrix-vector product Hx and its inverse $H^T x$ for arbitrary $x \in \mathbb{R}^d$.

Walsh-Hadamard matrices are ideal for H (after scaling appropriately). They attain the optimal $\rho = 1$ and the fast Walsh-Hadamard transform can compute the matrix-vector products in $O(d \log d)$ operations. This is what we use in our experiments. Formally, the Walsh-Hadamard matrices are defined recursively as follows:

$$H_{2^0} = (1), \qquad \forall k \geq 0 \quad H_{2^{k+1}} = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{pmatrix} \in \left\{ \frac{-1}{\sqrt{2^{k+1}}}, \frac{1}{\sqrt{2^{k+1}}} \right\}^{2^{k+1}\times 2^{k+1}}. \quad (49)$$

The only downside of Walsh-Hadamard matrices is that they require the dimension d to be a power of 2. We can pad the input vectors with zeros to ensure this. However, in the worst case, padding may nearly double the dimension d, which correspondingly slows down our algorithm. (E.g., if $d = 2^k + 1$, then we must pad to dimension $d = 2^{k+1}$.)

To avoid or reduce padding, there are several solutions:

- The first, and simplest, solution is to use the discrete cosine transform as the matrix $H'_d \in \mathbb{R}^{d\times d}$, which is defined by

$$\forall d \in \mathbb{N} \;\; \forall i, j \in [d] \qquad H'_d(i,j) = \sqrt{\frac{2}{d}} \cdot \cos\left(\frac{\pi}{d} \cdot i \cdot (j - \frac{1}{2})\right). \quad (50)$$

  Such matrices exist for any dimension d and the required matrix-vector products can still be computed in $O(d \log d)$ time. However, they attain a slightly suboptimal sub-gaussian flatness parameter of $\rho = 2$.

- The second solution is to use more general Hadamard matrices. It is conjectured that for all $d \in \{1, 2\} \cup \{4\ell : \ell \in \mathbb{N}\}$ there exist unitary matrices H_d in $\{-1/\sqrt{d}, 1/\sqrt{d}\}^{d\times d}$. This conjecture would allow us to do very little padding (adding at most three extra coordinates to reach the next multiple of 4), but we do not have a proof of this conjecture, much less efficient algorithms for computing with these matrices.

  Fortunately, there are explicit constructions of Hadamard matrices of many sizes which also allow efficient matrix-vector computations. By considering sizes other than powers of 2, we can significantly reduce the required amount of padding.

  For example, we can generalize the Kronecker product construction (49) to dimension

[9] A lower bound of $\rho \geq 1$ applies as H is unitary – $H^T H = I$, so $1 = (H^T H)_{1,1} = \sum_i^d H_{i,1}^2 \leq d(\sqrt{\rho/d})^2 = \rho$.

$d = 12 \cdot 2^k$ for integers $k \geq 0$:

$$H_{12\cdot 2^k} = \frac{1}{\sqrt{12}} \begin{pmatrix} H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} \\ H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} \\ H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} \\ H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} \\ H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} \\ H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} \\ H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} \\ H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} \\ H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} \\ H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} \\ H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} \\ H_{2^k} & -H_{2^k} & H_{2^k} & -H_{2^k} & -H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} & H_{2^k} & -H_{2^k} & H_{2^k} & H_{2^k} \end{pmatrix}. \tag{51}$$

The addition of this construction alone is sufficient to reduce the worst case for padding from a factor of 2 to a factor of 1.5 – now $2^k + 1$ can be padded to $12 \cdot 2^{k-3} = 1.5 \cdot 2^k$. The other desirable properties of the Hadamard matrices are also retained.

- A third solution is to move from the reals to complex numbers and use the discrete Fourier transform. Our real vector of length d can be encoded as a complex vector of length $d/2$ (two real entries become the real and imaginary components of one complex entry). Instead of D being a diagonal matrix with random signs, the diagonal entries are $e^{\sqrt{-1}\cdot\theta}$ for a uniformly random $\theta \in [0, 2\pi)$. (In fact, it suffices to have $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ uniform. This only requires one bit of shared randomness per coordinate.[10]) Then H is the discrete Fourier transform matrix. This gives us a complex vector of length $d/2$ that can be decoded back to a real vector of length d. This transformation is unitary and linear and attains the optimal subgaussian flatness constant $\rho = 1$ (i.e., matches the guarantee of a random rotation or unitary matrix from Lemma 28). The only requirement is that the dimension must be even – i.e., we must pad at most one zero.

  If we wish to avoid thinking about complex numbers, the complex numbers can be replaced with 2×2 rotation matrices. That is

$$H''_{2d} = \frac{1}{\sqrt{d}} \begin{pmatrix} W^0 & W^0 & W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & W^2 & W^3 & \cdots & W^d \\ W^0 & W^2 & W^4 & W^6 & \cdots & W^{2d} \\ W^0 & W^3 & W^6 & W^9 & \cdots & W^{3d} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ W^0 & W^d & W^{2d} & W^{3d} & \cdots & W^{d^2} \end{pmatrix} \in \mathbb{R}^{2d \times 2d}, \tag{52}$$

[10] Note that $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ corresponds to $e^{i\theta} \in \{1, i, -1, -i\}$ and, in Equation 53, to $R_\theta \in \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right\}$.

where $W = \begin{pmatrix} \cos(2\pi/d) & -\sin(2\pi/d) \\ \sin(2\pi/d) & \cos(2\pi/d) \end{pmatrix} \in \mathbb{R}^{2\times 2}$, and

$$D_\Theta = \begin{pmatrix} R_{\theta_1} & 0 & 0 & \cdots & 0 \\ 0 & R_{\theta_2} & 0 & \cdots & 0 \\ 0 & 0 & R_{\theta_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & R_{\theta_d} \end{pmatrix} \in \mathbb{R}^{2d\times 2d}, \tag{53}$$

where $\Theta \in [0, 2\pi)^d$ and $R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \in \mathbb{R}^{2\times 2}$.

Proposition 30. *Let $d \in \mathbb{N}$ be even and let Θ be uniformly random in either $[0, 2\pi)^{d/2}$ or $\{0, \pi/2, \pi, 3\pi/2\}^{d/2}$. Define $U = H''_d D_\Theta \in \mathbb{R}^{d\times d}$ where H''_d and D_Θ are given in Equations 52 and 53. Then U is unitary and $\mathbb{E}\left[\exp(t(Ux)_i)\right] \le \exp(t^2\|x\|_2^2/2d)$ for all $x \in \mathbb{R}^d$, all $i \in [d]$, and all $t \in \mathbb{R}$.*

Proof. The fact that U is unitary follows from the fact that both H''_d and D_Θ are. Since $R_{\theta_i} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \in \mathbb{R}^{2\times 2}$ is unitary, this implies D_Θ is unitary. The matrix H''_d is a block matrix with the block in row $i \in [d/2]$ and column $j \in [d/2]$ being $W^{(i-1)(j-1)}$ for $W = R_{4\pi/d}$. Then $H''_d(H''_d)^T$ is also a block matrix. The block in row $i \in [d/2]$ and column $j \in [d/2]$ is

$$\begin{aligned} (H''_d(H''_d)^T)_{2i-1:2i,2j-1:2j} &= \frac{2}{d}\sum_{k=1}^{d/2} W^{(i-1)(k-1)}(W^{(k-1)(j-1)})^T \\ &= \frac{2}{d}\sum_{k=1}^{d/2} W^{(i-1)(k-1)-(k-1)(j-1)} \\ &= \frac{2}{d}\begin{cases} \frac{d}{2}I & \text{if } i = j \\ 0 & \text{if } i \ne j \end{cases}. \end{aligned}$$

This follows from the fact that W is unitary (i.e., $W^T = W^{-1}$), the fact that $W^{d/2} = I$, and the fact that, for any $\ell \in \mathbb{Z}$ that is *not* a multiple of $d/2$, $W^\ell - I$ is nonsingular and, hence, $\sum_{k=1}^{d/2} W^{\ell\cdot(k-1)} = 0$.

Fix $x \in \mathbb{R}^d$ and $i \in [d/2]$. Now, by the properties of rotation matrices,

$$\begin{aligned} \begin{pmatrix} (Ux)_{2i-1} \\ (Ux)_{2i} \end{pmatrix} &= \sqrt{\frac{2}{d}}\sum_{j=1}^{d/2} W^{(i-1)(j-1)} R_{\theta_j} \begin{pmatrix} x_{2i-1} \\ x_{2i} \end{pmatrix} \\ &= \sqrt{\frac{2}{d}}\sum_{j=1}^{d/2} R_{4\pi(i-1)(j-1)/d+\theta_j+\tilde{\theta}_i} \begin{pmatrix} \sqrt{x_{2i-1}^2 + x_{2i}^2} \\ 0 \end{pmatrix}, \end{aligned}$$

where $\tilde{\theta}_i \in [0, 2\pi)$ is such that $\cos\tilde{\theta}_i = \frac{x_{2i-1}}{\sqrt{x_{2i-1}^2+x_{2i}^2}}$ and $\sin\tilde{\theta}_i = \frac{x_{2i}}{\sqrt{x_{2i-1}^2+x_{2i}^2}}$. For $j \in [d/2]$, define $\hat{\theta}_{i,j} = 4\pi(i-1)(j-1)/d + \tilde{\theta}_i$. Then $(Ux)_{2i-1} = \sqrt{\frac{2}{d}} \sum_{j=1}^{d/2} \sqrt{x_{2i-1}^2 + x_{2i}^2} \cdot \cos(\theta_j + \hat{\theta}_{i,j})$ and $(Ux)_{2i} = \sqrt{\frac{2}{d}} \sum_{j=1}^{d/2} \sqrt{x_{2i-1}^2 + x_{2i}^2} \cdot \sin(\theta_j + \hat{\theta}_{i,j})$. Fix $t \in \mathbb{R}$. If $\Theta \in [0, 2\pi)^{d/2}$ follows a product distribution, then

$$\mathop{\mathbb{E}}_{\Theta}[\exp(t(Ux)_{2i-1})] = \prod_{j=1}^{d/2} \mathop{\mathbb{E}}_{\theta_j}\left[\exp\left(t\sqrt{\frac{2}{d}(x_{2i-1}^2 + x_{2i}^2)}\cos(\theta_j + \hat{\theta}_{i,j})\right)\right] \tag{54}$$

and, similarly,

$$\mathop{\mathbb{E}}_{\Theta}[\exp(t(Ux)_{2i})] = \prod_{j=1}^{d/2} \mathop{\mathbb{E}}_{\theta_j}\left[\exp\left(t\sqrt{\frac{2}{d}(x_{2i-1}^2 + x_{2i}^2)}\cos(\theta_j + \hat{\theta}_{i,j} - \pi/2)\right)\right], \tag{55}$$

as $\sin(\psi) = \cos(\psi - \pi/2)$ for all $\psi \in \mathbb{R}$. If we can show that $\mathop{\mathbb{E}}_{\theta_j}[\exp(\lambda \cdot \cos(\theta_j + \psi)] \leq \exp(\lambda^2/4)$ for all $\lambda, \psi \in \mathbb{R}$ and all j, then we are done. Lemma 31 covers the case where θ_j is uniform on $\{0, \pi/2, \pi, 3\pi/2\}$ and Lemma 32 covers the case where it is uniform on $[0, 2\pi)$. Then the right sides of both Equations 54 and 55 become

$$\begin{aligned}
&\leq \prod_{j=1}^{d/2} \exp\left(\left(t\sqrt{\frac{2}{d}(x_{2i-1}^2 + x_{2i}^2)}\right)^2/4\right) \\
&= \exp\left(\sum_{j=1}^{d/2} t^2 \frac{2}{d}(x_{2i-1}^2 + x_{2i}^2)\frac{1}{4}\right) \\
&= \exp(t^2\|x\|_2^2/2d).
\end{aligned}$$

□

Lemma 31. *Let $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ be uniformly random. Let $\lambda, \psi \in \mathbb{R}$ be fixed. Then $\mathop{\mathbb{E}}_{\theta}[\exp(\lambda \cdot \cos(\theta + \psi)] \leq \exp(\lambda^2/4)$.*

Proof. Let $x = \lambda\cos\psi$ and $y = \lambda\cos(\psi + \pi/2)$. Then $\lambda\cos(\psi + \pi) = -x$ and $\lambda\cos(\psi + 3\pi/2) = -y$. Note $x^2 + y^2 = \lambda^2$. Thus

$$\begin{aligned}
\mathop{\mathbb{E}}_{\theta}[\exp(\lambda \cdot \cos(\theta + \psi)] &= \frac{e^x + e^{-x} + e^y + e^{-y}}{4} \\
&= 1 + \sum_{k=1}^{\infty} \frac{x^{2k} + y^{2k}}{2 \cdot (2k)!} \\
&\leq 1 + \sum_{k=1}^{\infty} \frac{\lambda^{2k}}{4^k \cdot k!} \\
&= \exp(\lambda^2/4).
\end{aligned}$$

The inequality follows from the fact that $x^{2k} + y^{2k} \leq (x^2 + y^2)^k = (\lambda^2 \cos^2 \psi + \lambda^2 \sin^2 \psi)^k = \lambda^{2k}$ and $2 \cdot (2k)! \geq 4^k \cdot k!$ for all integers $k \geq 1$. The last fact can be easily verified by induction: For $k = 1$ both sides are equal to 4. Moving from k to $k+1$ multiplies the right side by $4(k+1)$ and the left side by $(2k+1)(2k+2) = 4(k+1)(k+1/2) > 4(k+1)$. □

Lemma 32. *Let $\theta \in [0, 2\pi)$ be uniformly random. Let $\lambda, \psi \in \mathbb{R}$ be fixed. Then $\mathop{\mathbb{E}}_{\theta}[\exp(\lambda \cdot \cos(\theta + \psi)] \leq \exp(\lambda^2/4)$.*

Proof. Since θ is uniform on $[0, 2\pi)$ and cos is a periodic function with period 2π, the distribution of $\cos(\theta+\psi)$ is the same as that of $\cos\theta$, so we may ignore ψ. Also the distribution is symmetric (i.e., the distribution of $-\cos\theta = \cos(\theta+\pi)$ is the same as that of $\cos\theta$). Thus $\mathbb{E}\left[\cos^k \theta\right] = 0$ for all odd k. We also have $\mathbb{E}\left[\cos^2 \theta\right] = \mathbb{E}\left[\frac{1+\cos(2\theta)}{2}\right] = \frac{1}{2}$. Integration by parts yields the recurrence $\mathop{\mathbb{E}}_{\theta}\left[\cos^k \theta\right] = \frac{k-1}{k}\mathop{\mathbb{E}}_{\theta}\left[\cos^{k-2} \theta\right]$ for $k \geq 2$. This yields $\mathop{\mathbb{E}}_{\theta}\left[\cos^{2k} \theta\right] = \frac{(2k-1)!}{2^{2k-1} \cdot k! \cdot (k-1)!}$ for all integers $k \geq 1$. Thus

$$\begin{aligned}
\mathop{\mathbb{E}}_{\theta}[\exp(\lambda \cdot \cos\theta)] &= 1 + \sum_{k=1}^{\infty} \frac{\lambda^{2k}}{(2k)!} \mathop{\mathbb{E}}_{\theta}\left[\cos^{2k} \theta\right] \\
&= 1 + \sum_{k=1}^{\infty} \frac{\lambda^{2k} \cdot (2k-1)!}{(2k)! \cdot 2^{2k-1} \cdot k! \cdot (k-1)!} \\
&= 1 + \sum_{k=1}^{\infty} \frac{(\lambda^2/4)^k}{k!} \cdot \frac{1}{k!} \\
&\leq 1 + \sum_{k=1}^{\infty} \frac{(\lambda^2/4)^k}{k!} \\
&= \exp(\lambda^2/4).
\end{aligned}$$

□

We emphasize that the discrete fourier transform (i.e., matrix-vector multiplications with H''_{2d} from Equation 52) can be computed in $O(d \log d)$ operations for *any* d – not just powers of 2. Although the exact efficiency (i.e., constants) depends on d [Wik21].

4.3 Modular Clipping

In this section we cover third and final source of error – modular arithmetic. This is introduced by the secure aggregation procedure.

We first define the modular clipping operation in a convenient form for real numbers.

Definition 33. *For $a < b$, define $M_{[a,b]} : \mathbb{R} \to [a, b]$ by $M(x) = x + (b-a) \cdot n$ where $n \in \mathbb{Z}$ is chosen so that $x + (b-a) \cdot n \in [a, b]$. (Ties are broken arbitrarily.) We also define $M_{[a,b]}(x) = (M_{[a,b]}(x_1), M_{[a,b]}(x_2), \cdots, M_{[a,b]}(x_d)) \in [a, b]^d$ for $x \in \mathbb{R}^d$.*

Modular arithmetic is usually performed over $\mathbb{Z}_m$, which we equate with the set of integers $\{0, 1, 2, \cdots, m-1\}$. Our algorithm maps real values to this set of integers. However, for our analysis, it will be convenient to imagine the modular clipping operation taking place directly over the real numbers. (This is completely equivalent to the usual view, but has the advantage of allowing us to perform the analysis over a centered interval $[-r, r]$, rather than $[0, m]$.) Specifically, our algorithm can be thought of as performing modular arithmetic over $\{\gamma(1-m/2), \gamma(2-m/2), \cdots, -\gamma, 0, \gamma, \cdots, \gamma(m/2-1), \gamma(m/2)\}$ instead of $\mathbb{Z}_m$. This operation is denoted as $M_{[-m\gamma/2, m\gamma/2]}$.

Note that definition 33 does not specify whether $M_{[a,b]}(a) = a$ or $M_{[a,b]}(a) = b$ (and likewise for $M_{[a,b]}(b)$). Thus our analysis does not depend on how this choice is made.

A key property of the modular operation is that it is homomorphic:

$$\forall a < b \;\; \forall x, y \in \mathbb{R} \qquad M_{[a,b]}(x+y) = M_{[a,b]}(M_{[a,b]}(x) + M_{[a,b]}(y)). \tag{56}$$

Our goal is to analyze $M_{[a,b]}(A(x))$, where A is as in Proposition 26. The modular clipping arises from the secure aggregation step, which works over a finite group. Note that A discretizes the values (although this is not crucial for this part of the analysis).

We want to ensure that $M_{[a,b]}(A(x)) \approx x$. We have already established that $A(x) \approx x$ and our goal is now to analyze the modular clipping operation. If $A(x) \in [a, b]^d$, then $M_{[a,b]}(A(x)) = A(x)$ and we are in good shape; thus our analysis centers on ensuring that this is the case.

We will use the fact that the flattening operation, as well as the randomized rounding and noise addition, result in each coordinate being a centered subgaussian random variable. This allows us to bound the probability of straying outside $[a, b]$.

First we present a technical lemma.

Lemma 34. *Let $r > 0$ and $x \in \mathbb{R}$. Then*

$$|M_{[-r,r]}(x) - x| \le 2r \cdot \left(\exp\left(\frac{1}{2}t \cdot \left(\frac{x}{r} - 1\right)\right) + \exp\left(\frac{1}{2}t \cdot \left(\frac{-x}{r} - 1\right)\right)\right)$$

and

$$(M_{[-r,r]}(x) - x)^2 \le 4r^2 \cdot \left(\exp\left(t \cdot \left(\frac{x}{r} - 1\right)\right) + \exp\left(t \cdot \left(\frac{-x}{r} - 1\right)\right)\right)$$

for all $t \ge \log 2$.

Proof. We have $|M_{[-r,r]}(x) - x| \le 2r \cdot \left\lfloor \frac{|x|+r}{2r} \right\rfloor$ for all $x \in \mathbb{R}$. We also have $\lfloor x \rfloor \le e^{t \cdot (x-1)}$ for all $x \in \mathbb{R}$ and all $t \ge \log 2$. (For $x < 1$, we have $\lfloor x \rfloor \le 0 \le e^{t \cdot (x-1)}$. For $1 \le x < 2$, we have $\lfloor x \rfloor = e^0 \le e^{t(x-1)}$. We have $e^{t(1-1)} = 1$ and $e^{t(2-1)} \ge 2$; since $e^{t(x-1)}$ is convex, this implies $\lfloor x \rfloor \le x \le e^{t(x-1)}$ for $x \ge 2$.) Thus

$$|M_{[-r,r]}(x) - x| \le 2r \cdot \exp\left(t \cdot \frac{|x| - r}{2r}\right) \le 2r \cdot \left(\exp\left(t \cdot \frac{x-r}{2r}\right) + \exp\left(t \cdot \frac{-x-r}{2r}\right)\right)$$

and, hence,

$$(M_{[-r,r]}(x) - x)^2 \leq 4r^2 \cdot \exp\left(t \cdot \frac{|x| - r}{r}\right) \leq 4r^2 \cdot \exp\left(t \cdot \frac{x - r}{r}\right) + 4r^2 \cdot \exp\left(t \cdot \frac{-x - r}{r}\right)$$

for all $t \geq \log 2$ and $x \in \mathbb{R}$, as required. □

Now we have our bound on the error of modular clipping:

Proposition 35 (Error of Modular Clipping)**.** *Let $a < b$ and $\omega, \sigma > 0$ satisfy $\sigma \leq (b-a)/2$. Let $X \in \mathbb{R}$ satisfy $\mathbb{E}\left[e^{tX}\right] \leq \omega \cdot e^{t^2\sigma^2/2}$ for all $t \in \mathbb{R}$. Then*

$$\mathbb{E}\left[\left|M_{[a,b]}(X) - X\right|\right] \leq (b-a) \cdot \omega \cdot e^{-(b-a)^2/8\sigma^2} \cdot \left(e^{\frac{a^2-b^2}{4\sigma^2}} + e^{\frac{b^2-a^2}{4\sigma^2}}\right)$$

and

$$\mathbb{E}\left[\left(M_{[a,b]}(X) - X\right)^2\right] \leq (b-a)^2 \cdot \omega \cdot e^{-(b-a)^2/8\sigma^2} \cdot \left(e^{\frac{a^2-b^2}{4\sigma^2}} + e^{\frac{b^2-a^2}{4\sigma^2}}\right).$$

Proof. First we center: Let $c = (a+b)/2$ and $r = (b-a)/2$. Then $M_{[a,b]}(x) = M_{[-r,r]}(x-c)+c$ and $|M_{[a,b]}(x) - x| = |M_{[-r,r]}(x-c) - (x-c)|$ for all $x \in \mathbb{R}$. Let $X' = X - c$.

By Lemma 34, for all $t \geq \log 2$,

$$\begin{aligned}
\mathbb{E}\left[\left|M_{[a,b]}(X) - X\right|\right] &= \mathbb{E}\left[\left|M_{[-r,r]}(X') - X'\right|\right] \\
&\leq 2r \cdot \mathbb{E}\left[\exp\left(\frac{1}{2}t \cdot \left(\frac{X'}{r} - 1\right)\right) + \exp\left(\frac{1}{2}t \cdot \left(\frac{-X'}{r} - 1\right)\right)\right] \\
&\leq 2r \cdot \omega \cdot e^{t^2\sigma^2/8r^2 - t/2} \cdot \left(e^{-tc/2r} + e^{tc/2r}\right) \\
&= (b-a) \cdot \omega \cdot e^{t^2\sigma^2/2(b-a)^2 - t/2} \cdot \left(e^{-\frac{t}{2}\cdot\frac{a+b}{b-a}} + e^{\frac{t}{2}\cdot\frac{a+b}{b-a}}\right).
\end{aligned}$$

Set $t = (b-a)^2/2\sigma^2 \geq \log 2$ to obtain the first part of the result.

By Lemma 34, for all $t \geq \log 2$,

$$\begin{aligned}
\mathbb{E}\left[\left(M_{[a,b]}(X) - X\right)^2\right] &= \mathbb{E}\left[\left(M_{[-r,r]}(X') - X'\right)^2\right] \\
&\leq 4r^2 \cdot \mathbb{E}\left[\exp\left(t \cdot \left(\frac{X'}{r} - 1\right)\right) + \exp\left(t \cdot \left(\frac{-X'}{r} - 1\right)\right)\right] \\
&\leq 4r^2 \cdot \omega \cdot e^{t^2\sigma^2/2r^2 - t} \cdot \left(e^{-tc/r} + e^{tc/r}\right) \\
&= (b-a)^2 \cdot \omega \cdot e^{2t^2\sigma^2/(b-a)^2 - t} \cdot \left(e^{-t\frac{a+b}{b-a}} + e^{t\frac{a+b}{b-a}}\right).
\end{aligned}$$

Set $t = (b-a)^2/4\sigma^2 \geq \log 2$ to obtain the second part of the result. □

4.4 Putting Everything Together

We have now analyzed the three sources of error – randomized rounding, privacy-preserving noise, and modular arithmetic. It remains to combine these results. This yields our main result:

Theorem 36 (Main Theoretical Result)**.** *Let $\beta \in [0,1)$, $\sigma^2 \geq \frac{1}{2}\gamma > 0$, and $c > 0$. Let $n, d \in \mathbb{N}$ and $\rho \geq 1$. Let $U \in \mathbb{R}^{d\times d}$ be a random unitary matrix such that*

$$\forall x \in \mathbb{R}^d \quad \forall i \in [d] \quad \forall t \in \mathbb{R} \qquad \mathbb{E}\left[\exp(t(Ux)_i)\right] \leq \exp(t^2\rho\|x\|_2^2/2d).$$

Let

$$\Delta_2^2 := \min\left\{ \begin{array}{c} c^2 + \frac{1}{4}\gamma^2 d + \sqrt{2\log(1/\beta)}\cdot\gamma\cdot\left(c + \frac{1}{2}\gamma\sqrt{d}\right), \\ \left(c+\gamma\sqrt{d}\right)^2 \end{array} \right\}, \tag{57}$$

$$G := \left\{y \in \mathbb{R}^d : \|y\|_2^2 \leq \Delta_2^2\right\}, \tag{58}$$

$$\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2 \frac{\sigma^2}{\gamma^2}\cdot\frac{k}{k+1}}, \tag{59}$$

$$\varepsilon := \min\left\{ \begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}. \tag{60}$$

Let R_γ^G be as in Definition 20. Let $r > 0$ and let $M_{[-r,r]}$ be as in Definition 33. Define a randomized algorithm $\tilde{A} : (\mathbb{R}^d)^n \to \gamma\mathbb{Z}^d$ by

$$\tilde{A}(x) = U^T M_{[-r,r]}\left(\sum_i^n R_\gamma^G\left(\min\left\{1, \frac{c}{\|x_i\|_2}\right\}\cdot Ux_i\right) + \gamma\cdot Y_i\right), \tag{61}$$

where $Y_1, \cdots, Y_n \in \mathbb{Z}^d$ are independent random vectors with each entry drawn independently from $\mathcal{N}_\mathbb{Z}(0, \sigma^2/\gamma^2)$.

Then $\tilde{A}$ satisfies $\frac{1}{2}\varepsilon^2$-concentrated differential privacy.[11]

Let $x_1, \cdots, x_n \in \mathbb{R}^d$ with $\|x_i\|_2 \leq c$ for all $i \in [n]$. Let

$$\hat{\sigma}^2(x) := \frac{\rho}{d}\left\|\sum_i^n x_i\right\|_2^2 + \left(\frac{\gamma^2}{4} + \sigma^2\right)\cdot n \leq \frac{\rho}{d}c^2n^2 + \left(\frac{\gamma^2}{4} + \sigma^2\right)\cdot n \tag{62}$$

If $\hat{\sigma}^2(x) \leq r^2$, then

$$\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \leq \frac{d\cdot n}{1-\beta}\cdot\left(\frac{2\sqrt{2}\cdot r\cdot e^{-r^2/4\hat{\sigma}^2(x)}}{\sqrt{n\cdot(1-\beta)^{n-1}}} + \sqrt{\frac{\gamma^2}{4} + \frac{\beta^2\gamma^2 n}{1-\beta} + (1-\beta)\sigma^2}\right)^2. \tag{63}$$

[11] Note that this is with respect to the addition or removal of an element, not replacement. To keep n fixed, we would need addition/removal to be defined to simply zero-out the relevant vectors.

There is a lot to unpack in Theorem 36. Let us work through the parameters:

- n is the number of individuals and d is the dimension of the data.
- c is the bound on 2-norm of the individual data vectors.
- γ is the granularity of the discretization – we round data vectors to the integer grid $\gamma\mathbb{Z}^d$.
- r is the range for the modular clipping – our final sum ends up being clipped to $\gamma\mathbb{Z}^d \cap [-r, r]^d$. The secure aggregation does modular arithmetic over a group of size $m = 2r/\gamma$ (note r should be a multiple of γ). This ratio determines the communication complexity.
- σ^2 is the variance of the individual discrete Gaussian noise that we add; the sum will have variance $n\sigma^2$. This determines the privacy; specifically $\varepsilon \approx \frac{c}{\sqrt{n}\sigma}$ and we attain $\frac{1}{2}\varepsilon^2$-concentrated differential privacy.
- β is a parameter that controls the conditional randomized rounding. $\beta = 0$ yields unconditional randomized rounding, and larger β entails more aggressive conditioning. It will be helpful to think of $\beta = \sqrt{\gamma/n}$; although, in practice, slightly larger β may be preferable.
- ρ measures how good the flattening matrix U is (cf. Lemma 29). Think of $\rho = 1$ or at most $\rho \leq 2$.
- The other parameters – Δ_2, G, τ, $\hat{\sigma}$ – are not important, as they are determined by the previous parameters. Δ_2 and G determine how much the conditional randomized rounding can increase the norm (initially the norm is c). τ quantifies how far the sum of discrete Gaussians is from just a single discrete Gaussian and how this affects the differential privacy guarantee. The ratio $\hat{\sigma}/r$ measures how much error the modular clipping contributes. $\hat{\sigma}$ is determined by other parameters, but note that $\|\sum_i^n x_i\| \leq \sum_i^n \|x_i\| \leq cn$ may be a loose upper bound, in which case, the clipping error is less.

Now we look at the error bound. If we assume $\beta \leq 1/\sqrt{n}$ and $\hat{\sigma}^2(x) \leq r^2/4\log(r\sqrt{n}/\gamma^2)$, then the guarantee (63) is simply

$$\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \leq O\left(dn\left(\sigma^2 + \gamma^2\right)\right).$$

The first term is roughly the cost of privacy – $dn\sigma^2 \approx d\frac{c^2}{\varepsilon^2}$. The second term, $dn\gamma^2$, is the cost of randomized rounding and modular clipping. (We have assumed β and $\hat{\sigma}$ are sufficiently small to avoid any additional terms.)

Proof of Theorem 36. The differential privacy guarantee follows from the postprocessing property of differential privacy and Proposition 26 (which, in turn, applies Proposition 14).

Now we turn to the utility analysis. Let A be as in Proposition 26. Then $\tilde{A}(x) = U^T M_{[-r,r]}(A(Ux))$, where $Ux = (Ux_1, Ux_2, \cdots, Ux_n)$. Proposition 26 gives us the following guarantees.

$$\left\| \mathbb{E}\left[A(Ux)\right] - U\sum_i^n x_i \right\|_2 \leq \frac{\beta\cdot\gamma\cdot\sqrt{d}\cdot n}{1-\beta}.$$

$$\mathbb{E}\left[\left\|A(Ux) - \mathbb{E}\left[A(Ux)\right]\right\|_2^2\right] \leq \frac{\gamma^2\cdot d\cdot n}{4(1-\beta)} + n\cdot d\cdot\sigma^2.$$

$$\mathbb{E}\left[\left\|A(Ux) - U\sum_i^n x_i\right\|_2^2\right] \leq \frac{\gamma^2\cdot d\cdot n}{4(1-\beta)} + \left(\frac{\beta}{1-\beta}\gamma\sqrt{d}n\right)^2 + n\cdot d\cdot\sigma^2$$

$$\forall t\in\mathbb{R}\ \forall j\in[d]\quad \mathbb{E}\left[\exp\left(t\cdot\left(A(Ux) - U\sum_i^n x_i\right)_j\right)\right] \leq \frac{\exp\left(\left(\frac{\gamma^2}{8}+\frac{\sigma^2}{2}\right)\cdot t^2\cdot n\right)}{(1-\beta)^n}.$$

By our assumption on U (and independence) we have

$$\forall t\in\mathbb{R}\ \forall j\in[d]\quad \mathbb{E}\left[\exp\left(t\cdot\left(A(Ux)\right)_j\right)\right] \leq \exp\left(\frac{t^2\rho}{2d}\left\|\sum_i^n x_i\right\|_2^2\right)\cdot\frac{\exp\left(\left(\frac{\gamma^2}{8}+\frac{\sigma^2}{2}\right)\cdot t^2\cdot n\right)}{(1-\beta)^n}.$$

Recall $\hat{\sigma}^2(x) = \frac{\rho}{d}\left\|\sum_i^n x_i\right\|_2^2 + \left(\frac{\gamma^2}{4}+\sigma^2\right)\cdot n$. By Proposition 35, for all $j\in[d]$,

$$\mathbb{E}\left[\left|M_{[a,b]}(A(Ux))_j - A(Ux)_j\right|\right] \leq (b-a)\cdot\frac{1}{(1-\beta)^n}\cdot e^{-(b-a)^2/8\hat{\sigma}^2(x)}\cdot\left(e^{\frac{a^2-b^2}{4\hat{\sigma}^2}} + e^{\frac{b^2-a^2}{4\hat{\sigma}^2}}\right)$$

and

$$\mathbb{E}\left[\left(M_{[a,b]}(A(Ux))_j - A(Ux)_j\right)^2\right] \leq (b-a)^2\cdot\frac{1}{(1-\beta)^n}\cdot e^{-(b-a)^2/8\hat{\sigma}^2(x)}\cdot\left(e^{\frac{a^2-b^2}{4\hat{\sigma}^2}} + e^{\frac{b^2-a^2}{4\hat{\sigma}^2}}\right),$$

where $a=-r$ and $b=r$ here. Summing over $j\in[d]$ yields

$$\mathbb{E}\left[\left\|M_{[-r,r]}(A(Ux)) - A(Ux)\right\|_2^2\right] \leq 4r^2\cdot\frac{d}{(1-\beta)^n}\cdot e^{-r^2/2\hat{\sigma}^2(x)}\cdot 2.$$

For all $u, v \in \mathbb{R}$, we have $(u+v)^2 = \inf_{\lambda>0}(1+\lambda)u^2 + (1+1/\lambda)v^2$. Thus

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\left(U^T M_{[-r,r]}(A(Ux)) - U^T A(Ux)\right) + \left(U^T A(Ux) - U^T \sum_i^n Ux_i\right)\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\left(M_{[-r,r]}(A(Ux)) - A(Ux)\right) + \left(A(Ux) - \sum_i^n Ux_i\right)\right\|_2^2\right] \\
&\leq \inf_{\lambda>0}(1+\lambda)\mathbb{E}\left[\left\|M_{[-r,r]}(A(Ux)) - A(Ux)\right\|_2^2\right] + (1+1/\lambda)\mathbb{E}\left[\left\|A(Ux) - \sum_i^n Ux_i\right\|_2^2\right] \\
&\leq \inf_{\lambda>0}(1+\lambda)\cdot 4r^2 \cdot \frac{d}{(1-\beta)^n}\cdot e^{-r^2/2\hat{\sigma}^2(x)}\cdot 2 + (1+1/\lambda)\cdot\left(\frac{\gamma^2\cdot d\cdot n}{4(1-\beta)} + \left(\frac{\beta}{1-\beta}\gamma\sqrt{dn}\right)^2 + n\cdot d\cdot\sigma^2\right) \\
&= \left(\sqrt{8r^2\cdot\frac{d}{(1-\beta)^n}\cdot e^{-r^2/2\hat{\sigma}^2(x)}} + \sqrt{\frac{\gamma^2\cdot d\cdot n}{4(1-\beta)} + \left(\frac{\beta}{1-\beta}\gamma\sqrt{dn}\right)^2 + n\cdot d\cdot\sigma^2}\right)^2 \\
&= \frac{d\cdot n}{1-\beta}\cdot\left(\frac{2\sqrt{2}\cdot r\cdot e^{-r^2/4\hat{\sigma}^2(x)}}{\sqrt{n(1-\beta)^{n-1}}} + \sqrt{\frac{\gamma^2}{4} + \frac{\beta^2\gamma^2 n}{1-\beta} + (1-\beta)\sigma^2}\right)^2.
\end{aligned}
$$

□

We gave an asymptotic version of Theorem 36 in the introduction.

Finally, we analyse how to set the parameters to obtain this bound. Note that we do not attempt to optimize constants here at all.

Proof of Theorem 2. Theorem 36 gives the following parameters.

$$\begin{aligned}
\Delta_2^2 &:= \min\left\{ \begin{array}{c} c^2 + \frac{1}{4}\gamma^2 d + \sqrt{2\log(1/\beta)}\cdot\gamma\cdot\left(c + \frac{1}{2}\gamma\sqrt{d}\right), \\ \left(c+\gamma\sqrt{d}\right)^2 \end{array} \right\}, \\
\tau &:= 10\cdot\sum_{k=1}^{n-1} e^{-2\pi^2\frac{\sigma^2}{\gamma^2}\cdot\frac{k}{k+1}}, \\
\varepsilon &:= \min\left\{ \begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}, \\
\hat{\sigma}^2(x) &:= \frac{\rho}{d}\left\|\sum_i^n x_i\right\|_2^2 + \left(\frac{\gamma^2}{4} + \sigma^2\right)\cdot n \\
&\le \frac{\rho}{d}c^2n^2 + \left(\frac{\gamma^2}{4} + \sigma^2\right)\cdot n, \\
\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] &\le \frac{d\cdot n}{1-\beta}\cdot\left(\frac{2\sqrt{2}\cdot r\cdot e^{-r^2/4\hat{\sigma}^2(x)}}{\sqrt{n(1-\beta)^{n-1}}} + \sqrt{\frac{\gamma^2}{4} + \frac{\beta^2\gamma^3 2n}{1-\beta} + (1-\beta)\sigma^2}\right)^2.
\end{aligned}$$

Note that $r = \frac{1}{2}\gamma m$. All we must do is verify that setting the parameters as specified in Theorem 2 yields $\frac{1}{2}\varepsilon^2$-concentrated DP and the desired accuracy. First,

$$\begin{aligned}
\varepsilon^2 &\le \frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d \\
&\le \frac{(c+\gamma\sqrt{d})^2}{n\sigma^2} + 5nde^{-\pi^2(\sigma/\gamma)^2} \\
&\le \frac{2c^2}{n\sigma^2} + \frac{2d}{n(\sigma/\gamma)^2} + 5nde^{-\pi^2(\sigma/\gamma)^2}.
\end{aligned}$$

Thus the privacy requirement is satisfied as long as $\sigma \ge 2c/\varepsilon\sqrt{n}$ and $(\sigma/\gamma)^2 \ge 8d/\varepsilon^2 n$, and $5nde^{\pi^2(\sigma/\gamma)^2} \le \varepsilon^2/4$. So we can set

$$\sigma = \max\left\{\frac{2c}{\varepsilon\sqrt{n}}, \frac{\gamma\sqrt{8d}}{\varepsilon\sqrt{n}}, \frac{\gamma}{\pi^2}\log\left(\frac{20nd}{\varepsilon^2}\right)\right\} = \tilde{\Theta}\left(\frac{c}{\varepsilon\sqrt{n}} + \sqrt{\frac{d}{n}}\cdot\frac{\gamma}{\varepsilon}\right).$$

We set $\beta = \min\{1/n, 1/2\} = \Theta\left(\frac{1}{n}\right)$.

Next

$$\begin{aligned}
\hat{\sigma}^2 &\leq \frac{\rho}{d}c^2n^2 + \left(\frac{\gamma^2}{4} + \sigma^2\right)\cdot n, \\
&\leq \frac{c^2n^2}{d} + \gamma^2 n + \sigma^2 n \\
&\leq O\left(\frac{c^2n^2}{d} + \gamma^2 n + \frac{c^2}{\varepsilon^2} + \frac{\gamma^2 d}{\varepsilon^2} + \gamma^2 n\log^2\left(\frac{nd}{\varepsilon^2}\right)\right) \\
&\leq O\left(\frac{c^2n^2}{d} + \frac{c^2}{\varepsilon^2}\right) + \gamma^2\cdot O\left(n + \frac{d}{\varepsilon^2} + n\log^2\left(\frac{nd}{\varepsilon^2}\right)\right).
\end{aligned}$$

Now we work out the asymptotics of the accuracy guarantee:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \\
&\leq \frac{d\cdot n}{1-\beta}\cdot\left(\frac{2\sqrt{2}\cdot r\cdot e^{-r^2/4\hat{\sigma}^2(x)}}{\sqrt{n(1-\beta)^{n-1}}} + \sqrt{\frac{\gamma^2}{4} + \frac{\beta^2\gamma^2 n}{1-\beta} + (1-\beta)\sigma^2}\right)^2 \\
&\leq O\left(nd\left(\frac{re^{-r^2/4\hat{\sigma}^2}}{\sqrt{n}} + \sqrt{\gamma^2+\gamma^2+\sigma^2}\right)^2\right) \\
&\leq O\left(nd\left(\frac{r^2}{n}e^{-r^2/2\hat{\sigma}^2} + \gamma^2 + \sigma^2\right)\right) \\
&\leq O\left(nd\left(\frac{\gamma^2m^2}{n}\exp\left(\frac{-\gamma^2m^2}{8\hat{\sigma}^2}\right) + \gamma^2 + \frac{c^2}{\varepsilon^2 n} + \frac{\gamma^2 d}{\varepsilon^2 n} + \gamma^2\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right) \\
&= O\left(\frac{c^2d}{\varepsilon^2} + \gamma^2 nd\left(\frac{m^2}{n}\exp\left(\frac{-\gamma^2m^2}{8\hat{\sigma}^2}\right) + 1 + \frac{d}{\varepsilon^2 n} + \log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right).
\end{aligned}$$

Now we wish to set γ so that $\frac{m^2}{n}\exp\left(\frac{-\gamma^2m^2}{8\hat{\sigma}^2}\right) \leq 1$ – i.e., $\gamma \geq \frac{\hat{\sigma}}{m}\sqrt{8\log(1+m^2/n)}$. However, simply setting $\gamma = \frac{\hat{\sigma}^*}{m}\sqrt{8\log(1+m^2/n)}$, where $\hat{\sigma}^*$ is our upper bound on $\hat{\sigma}$, is cyclic, because our bound on $\hat{\sigma}$ depends on γ. Fortunately, we can resolve this as long as the coefficient in this cycle is $\leq 1-\Omega(1)$. That coefficient is $O\left(n + \frac{d}{\varepsilon^2} + n\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\cdot\frac{\log(1+m^2/n)}{m^2}$. A sufficient condition for this is

$$m^2 \geq O\left(n + \frac{d}{\varepsilon^2} + n\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\cdot\log(1+m^2/n) = \tilde{O}\left(n + d/\varepsilon^2\right).$$

Thus we can set

$$\gamma^2 = O\left(\frac{c^2n^2}{d} + \frac{c^2}{\varepsilon^2}\right)\cdot\frac{\log(1+m^2/n)}{m^2}$$

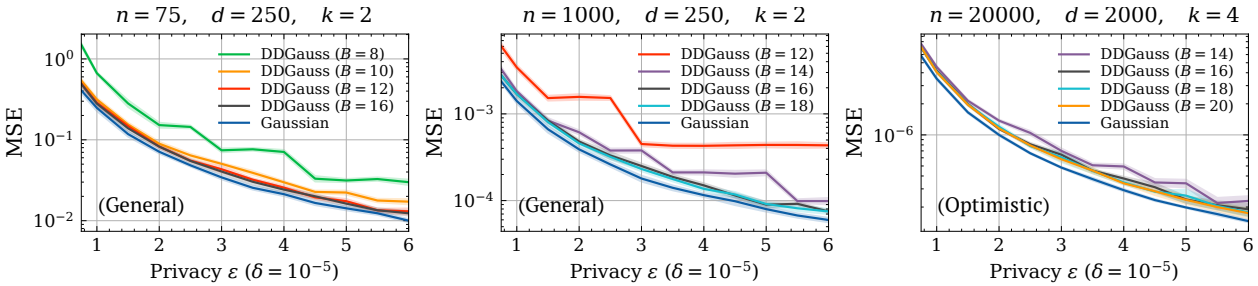

Figure 1: Distributed mean estimation with the distributed discrete Gaussian. n: number of clients. d: vector dimension. k: number of stddevs of $\sum_i^n \tilde{x}_i + y_i$ to bound. B: per-coordinate bit-width. General/Optimistic: assumes $\|\sum_i^n x_i\|_2 \leq cn$ or $\leq c\sqrt{n}$ for choosing γ.

and we obtain

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \\
&\leq O\left(\frac{c^2 d}{\varepsilon^2} + \gamma^2 nd\left(1 + 1 + \frac{d}{\varepsilon^2 n} + \log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right) \\
&\leq O\left(\frac{c^2 d}{\varepsilon^2} + \left(\frac{c^2 n^2}{d} + \frac{c^2}{\varepsilon^2}\right) \cdot \frac{\log(1 + m^2/n)}{m^2} nd\left(1 + \frac{d}{\varepsilon^2 n} + \log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right) \\
&\leq O\left(\frac{c^2 d}{\varepsilon^2} + \frac{c^2 d}{\varepsilon^2}\left(\frac{\varepsilon^2 n^2}{d} + 1\right) \cdot \frac{\log(1 + m^2/n)}{m^2}\left(n + \frac{d}{\varepsilon^2} + n\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right) \\
&\leq O\left(\frac{c^2 d}{\varepsilon^2}\left(1 + \frac{\log(1 + m^2/n)}{m^2} \cdot \left(n^2 + \frac{d}{\varepsilon^2} + \left(\frac{\varepsilon^2 n^3}{d} + n\right)\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right)\right).
\end{aligned}
$$

Thus, if

$$
m^2 \geq O\left(\log(1 + m^2/n) \cdot \left(n^2 + \frac{d}{\varepsilon^2} + \left(\frac{\varepsilon^2 n^3}{d} + n\right)\log^2\left(\frac{nd}{\varepsilon^2}\right)\right)\right) = \tilde{O}\left(n^2 + \frac{\varepsilon^2 n^3}{d} + \frac{d}{\varepsilon^2}\right),
$$

then the mean squared error is $O(c^2 d/\varepsilon^2)$, as required. □

5 Experiments

We empirically evaluate the distributed discrete Gaussian mechanism (DDGauss) on two tasks: distributed mean estimation (DME) and federated learning (FL). Our goal is to demonstrate that the utility of DDGauss matches that of the continuous Gaussian mechanism under the same privacy guarantees when given sufficient communication budget. For both tasks, the top-level parameters include the number of participating clients n, the ℓ_2 norm bound for the client vectors c, the dimension d, the privacy budget ε, and the bit-width B which determines the modulo field size $m = 2^B$. For FL, we also consider the number of rounds T and the total number of clients N from which we randomly sample n clients in each round. We fix the conditional rounding bias to $\beta = e^{-1/2}$ unless otherwise stated.

To select the granularity parameter γ, we carefully balance the errors from randomized rounding and modular clipping. From the earlier sections, we know that each entry of $\sum_i^n \tilde{x}_i + y_i$ is subgaussian with known constants. Thus, for a fixed B, we can choose γ to ensure that the modular clipping range includes k standard deviations of $\sum_i^n \tilde{x}_i + y_i$. Specifically, the heuristic is to select γ such that $2k\hat{\sigma}$ is bounded within the modulo field size 2^B where $\hat{\sigma}^2 = \frac{c^2 n^2}{d} + \left(\frac{\gamma^2}{4} + \sigma^2\right) \cdot n$. Here, k captures the trade-off between the errors from quantization and modular clipping and should be application-dependent. A small k leads to a small γ and thus less error from rounding but more error from modular clipping; a large k means modular clipping happens rarely but at a cost of more rounding error.

5.1 Distributed Mean Estimation

In this experiment, n clients each hold a d-dimensional vector x_i uniformly randomly sampled from the ℓ_2 sphere with radius $c = 10$. We compute the ground truth mean vector $\bar{x} = \frac{1}{n}\sum_i^n x_i$ as well as the differentially private mean estimates $\widehat{x}$ across different mechanisms and communication/privacy budgets. We use the analytic Gaussian mechanism [BW18] as the strong baseline. Figure 1 shows the mean MSE $\|\bar{x} - \widehat{x}\|_2^2/d$ with 95% confidence interval over 10 random dataset initializations.[12] The first two plots assume a general norm bound $\|\sum_i^n x_i\|_2 \leq cn$ when choosing γ (generally applicable to FL applications), while the third plot assumes an optimistic bound $\|\sum_i^n x_i\|_2 \leq c\sqrt{n}$ as x_i's are sampled uniformly randomly on the ℓ_2 sphere. Note that the bit-width B applies to each coordinate of the quantized and noisy aggregate. Results indicate that DDGauss achieves a good communication-utility trade-off and matches the Gaussian with sufficient bit-widths.

In Figure 2, we additionally investigate the trade-off between quantization errors and modular clipping errors by trying different values of k. Here, we use the optimistic norm bound on the vector sum as the general norm bound could be loose (thus γ would be chosen conservatively such that modular wrap-around rarely happens). At $k = 2$, the effect of modular clipping is now evident (the gap between DDGauss and Gaussian). With increasingly larger k (larger γ), we incur more quantization errors (thus worse low bit-width performance) but less modular wrapping and can close the utility gap to Gaussian at high bit-widths.

5.2 Federated Learning

We evaluate on three public FL benchmarks: Federated EMNIST [CATVS17; CDWLKMST18], Stack Overflow Tag Prediction (SO-TP, [Aut19]), and Stack Overflow Next Word Prediction (SO-NWP, [Aut19]).

[12] The kinks on the low bit-width curves are due to the TensorFlow implementation of the discrete Gaussian sampler taking integer noise scales; to preserve privacy, noise scales are rounded up as $\lceil \sigma/\gamma \rceil$ in all experiments.

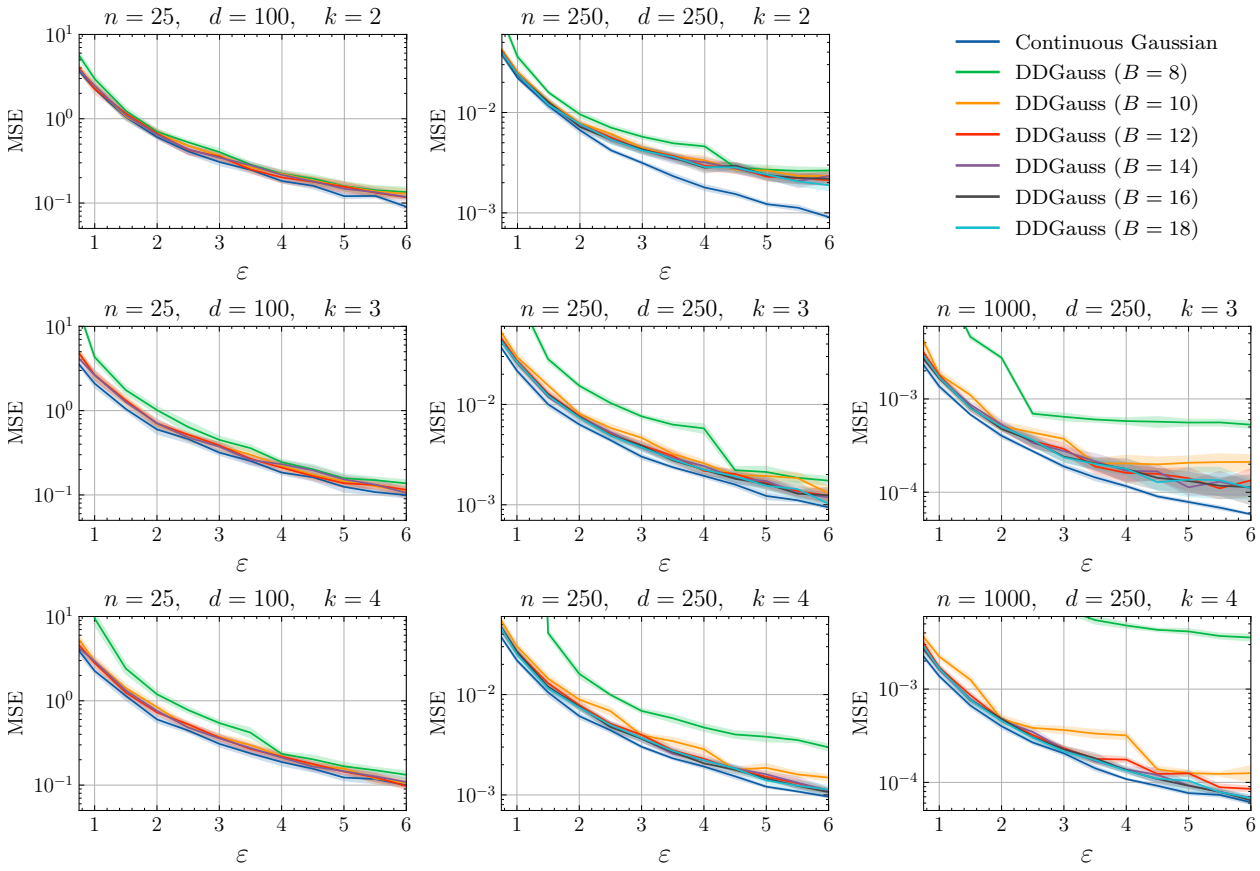

Figure 2: Distributed mean estimation with the distributed discrete Gaussian, assuming an optimistic bound $\|\sum_i^n x_i\|_2 \leq c\sqrt{n}$ as vectors are sampled uniformly randomly from a sphere. The first, second, and third row uses $k = 2$, $k = 3$, and $k = 4$, respectively. $\delta = 10^{-5}$.

5.2.1 Datasets

Federated EMNIST is an image classification dataset containing 671,585 training handwritten digit/letter images over 62 classes grouped into $N = 3400$ clients by their writer. Stack Overflow is a large-scale text dataset based on the question answering site Stack Overflow. It contains over 10^8 training sentences extracted from the site grouped by the $N = 342477$ users, and each sentence has associated metadata such as tags. The task of SO-TP involves predicting the tags of a given sentence, while the task of SO-NWP involves predicting the next words given the preceding words in a sentence. For more details on the datasets and tasks, we refer the reader to [RCZGRKKM20]. We note that these datasets differ from those commonly used in related work (e.g. MNIST [LCB10] and CIFAR-10 [Kri+09]) in that they are substantially larger, more challenging, and involve *user-level* (instead of example-level) DP with real-world client heterogeneity and label/size imbalance. Obtaining a small ε on EMNIST is also harder due to the relatively large sampling rate $q = n/N$ needed for stable convergence under noising.

5.2.2 Models

For EMNIST, We train a small convolutional net with two 3×3 conv layers with 32/64 channels followed by two fully connected layers with 128/62 output units; a 2×2 max pooling layer and two dropout layers with drop rate 0.25/0.5 are added after the first 3 trainable layers, respectively. The total number of parameters is $d = 1018174$, which is slightly under 2^{20} to avoid excessive zero padding for the Walsh-Hadamard transform. For SO-TP, we follow [RCZGRKKM20] and train a simple logistic regression model for tag prediction. The vocabulary size is limited to 10000 for word tokens and 500 for tags, and each sentence is represented as a bag-of-words vector. The resulting model size is $d = 5000500$. For SO-NWP, we use the LSTM architecture defined in [RCZGRKKM20] directly, which has a model size of $d = 4050748$ parameters.

5.2.3 Setup

For all benchmarks, we used the standard dataset split provided by TensorFlow. For EMNIST, the dataset is split into training and test set and performance is reported on the test set. For Stack Overflow (SO-TP and SO-NWP), the dataset is split into training, validation, and test sets. Validation accuracies and test accuracies are reported on the validation and test sets respectively. Note that using the dataset splits from TensorFlow is standard practice as in previous work (e.g. [RCZGRKKM20; MRTZ18; ASGXR21]), and it allows our results to be comparable in similar settings. Note also that validation techniques such as k-fold validation can incur additional privacy costs.

We adopt most hyperparameters from previous work [RCZGRKKM20; ATMR19; KMSTTX21]. For all tasks, we train with federated averaging with server momentum of 0.9 [MMRHA17; HQB19]. In each round, we uniformly sample $n = 100$ clients for EMNIST and SO-NWP following [ATMR19] and $n = 60$ clients for SO-TP due to memory limit. We train 1 epoch over clients' local datasets. Each client's model updates are weighted uniformly (instead of by their number of samples) to maintain privacy. Clients are sampled without replacement within each round, and with replacement across rounds. For EMNIST, SO-TP, and SO-NWP respectively, we set the number of rounds T to 1500, 1500, and 1600, c to 0.03, 2.0, and 0.3, client learning rate η_c to 0.032, 316, and 0.5, and client batch size to 20, 100, and 16. Server LR η_s is set to 1 for EMNIST, 0.56 for SO-TP, and selected from a small grid $\{0.3, 1\}$ for SO-NWP and reports the best performance. Tuning is limited to c (to tradeoff between the bias from clipping and the noise from privacy) and η_s (to match the selected c and n). For SO-NWP, we limit the max number of examples per client to 256.

The reported privacy guarantees ε rely on privacy amplification via sampling [KLNRS11; BST14; ACGMMTZ16], which is necessary to obtain reasonable privacy-accuracy tradeoffs in differentially private deep learning. This assumes that the identities of the users sampled in every round are hidden from the adversary. This does not hold for the entity initiating connection with the clients (typically the server running the FL protocol) but is applicable to participating clients and the analysts that have requested the model. We adopt the tight amplification bound from [MTZ19] for the Gaussian baseline and use the generic upper

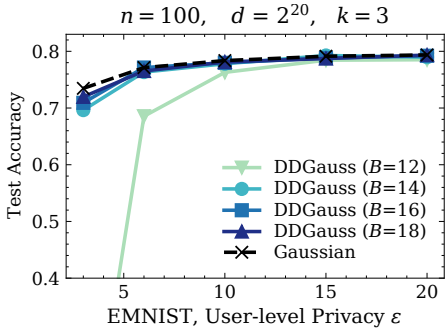

Figure 3: Summary of test accuracies (averaged over last 100 rounds) on EMNIST at $k = 3$ across different values of ε and B. d is the padded model size. $\delta = 1/N$.

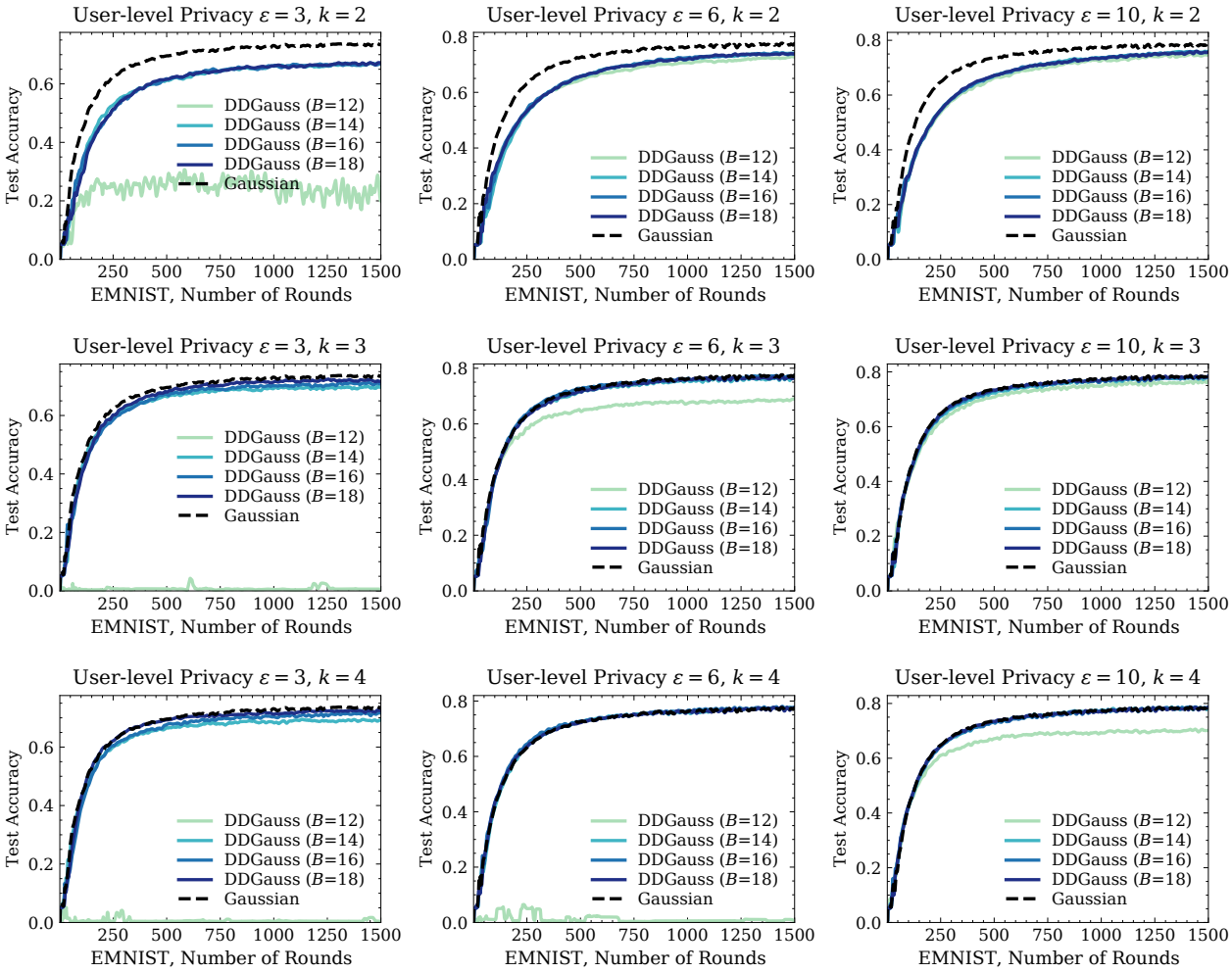

Figure 4: Test accuracies during training for EMNIST across different ε, B, and k. $\delta = 1/N$.

bound from [ZW19] for DDGauss (we do not explore a precise analysis in this work). The generic amplification upper bound could lead to more noise being added for DDGauss to

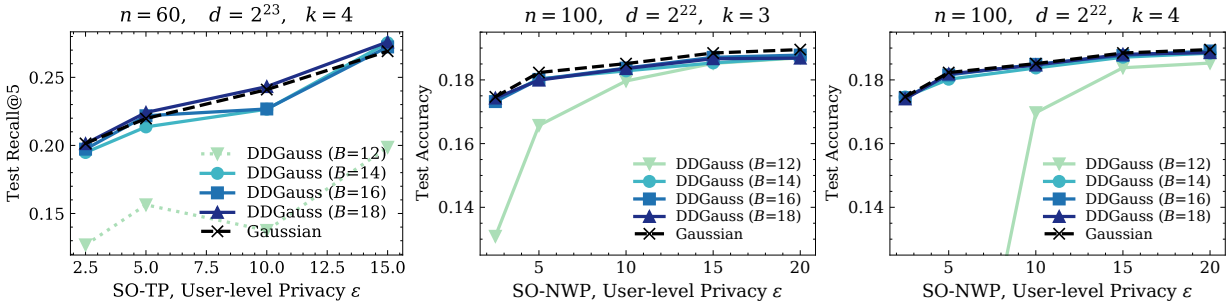

Figure 5: Summary of test performance on Stack Overflow. **Left:** Tag Prediction with logistic regression. Note that all runs of $B = 12$ except at $\varepsilon = 15$ did not converge. **Middle and Right:** Next Word Prediction with $k = 3$ and $k = 4$, respectively. d is the padded model size. $\delta = 1/N$ for SO-TP and $\delta = 10^{-6}$ for SO-NWP.

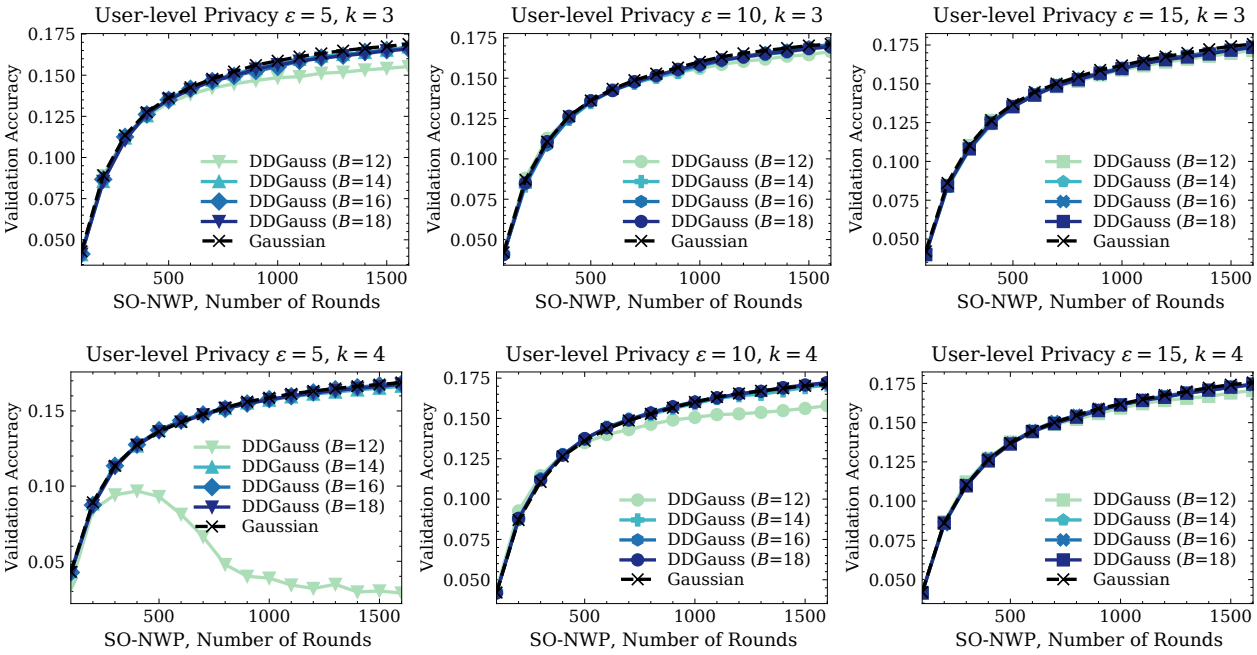

Figure 6: Validation accuracies during training for SO-NWP across different ε, B, and k.

achieve the same privacy as Gaussian.

5.2.4 Results

For EMNIST, Figure 3 summarizes the test accuracies across different values of ε and B for $k = 3$, and Figure 4 shows the accuracies during training. For Stack Overflow, Figure 5 summarizes the test performance on SO-TP (recall@5) and SO-NWP (accuracy), and Figure 6 shows the validation accuracies on SO-NWP.

Overall, with more communication bits (B) and privacy budget (ε), DDGauss achieves a better utility both relative to the Gaussian baseline and in absolute performance, and it

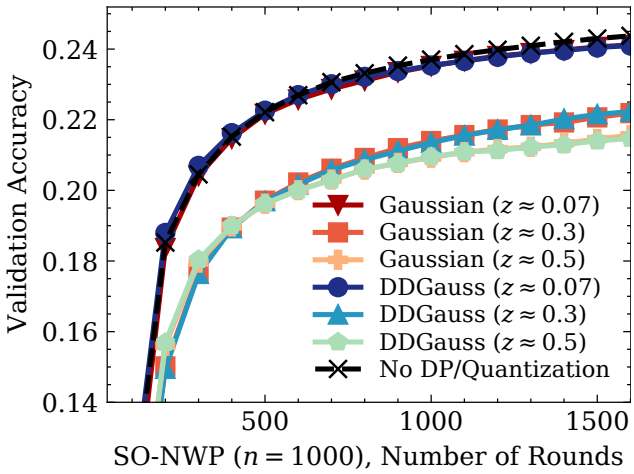

Figure 7: Validation accuracies on SO-NWP (averaged every 100 rounds) with $n = 1000$ and $B = 18$. z is the approximate noise multiplier.

can essentially match Gaussian as long as B is sufficient.

In particular, note from Figure 4 and 5 that a small k can be sub-optimal for learning as the cost of modular wrap-around is more pronounced than quantization errors; using a larger k allows DDGauss to match the Gaussian baseline at the expense of worse low bit-width performance (as γ is larger). Note also that for EMNIST (Figure 4), there is a slight performance gap between Gaussian and DDGauss in the extreme setting with $\varepsilon = 3$ and $k = 4$. We believe this minor mismatch, on top of the errors from rounding and modular clipping, is due to the use of the generic upper bound for privacy amplification via subsampling as discussed earlier in this section.

5.3 Additional Results

Large-Scale Training We additionally consider scaling up the SO-NWP experiments to $n = 1000$ clients per round (similar to production settings described in [HRMRBAEKR18; MRTZ18; RMRB19]), and we show the validation accuracies during training across different noise multipliers[13] in Figure 7. We set $c = 1$ and $\eta_s = 1$ for $z \approx 0.3$ and $z \approx 0.5$, and we set $\eta_s = 3$ otherwise. $z \approx 0.07$ gives a target test accuracy of around 25.2% (utility-first approach to limit performance degradation from DP [KMSTTX21]) while $z \approx 0.5$ and $z \approx 0.3$ give ε of 10 and 234 respectively. The results bear significant practical relevance as they indicate that as long as DDGauss is parameterized properly, it can perform as good as the continuous Gaussian in real-world settings (with large n, large model size d, and natural client heterogeneity from Stack Overflow).

[13] $z = \widehat{\sigma}/c$ where $\widehat{\sigma}$ is the equivalent central noise standard deviation ($\sqrt{n}\sigma$ for DDGauss). The values of z are aligned on privacy budgets and thus z is in fact slightly larger for DDGauss compared to Gaussian due to effects of rounding, generic amplification, etc.

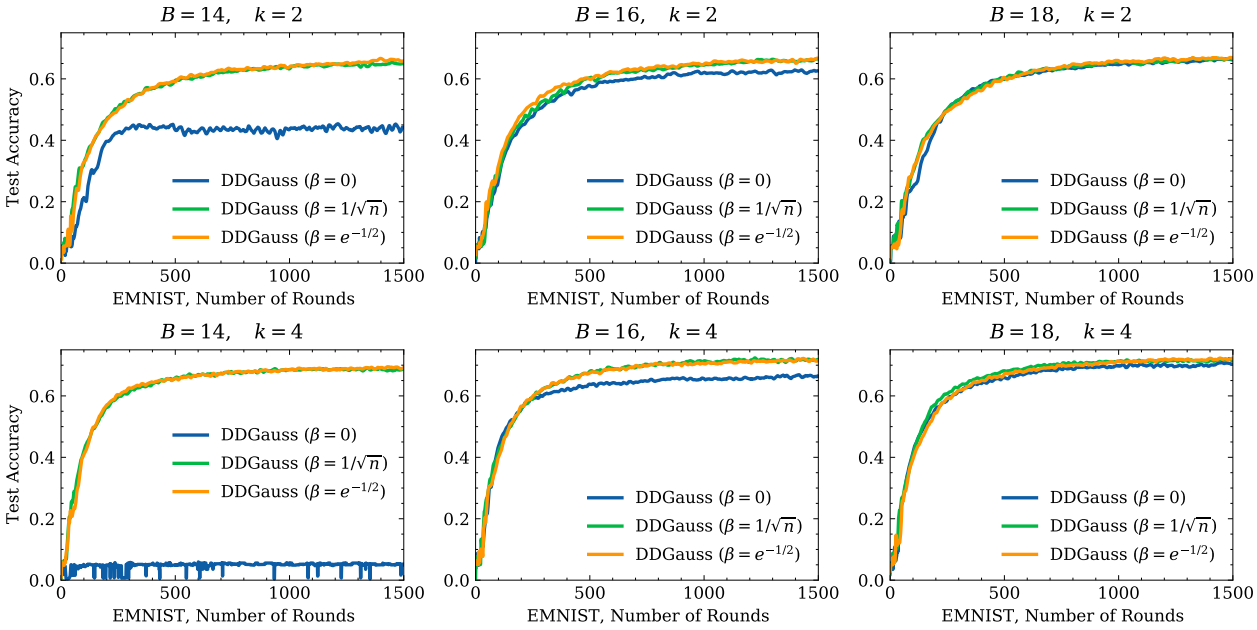

Figure 8: Effects of β on Federated EMNIST. The number of clients per round is $n = 100$ and the user-level privacy budget is fixed at $\varepsilon = 3$. $\delta = 1/N$.

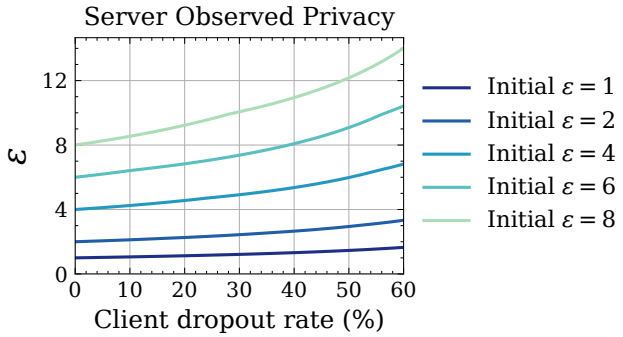

Figure 9: Effects of client dropouts on the server observed privacy. $\delta = 10^{-5}$.

Effects of β Recall from Section 4.1 that the hyperparameter β controls the growth of the client vector norm from conditional randomized rounding. Here, we are interested to know how β and the bias and variance it introduces influence the communication-utility trade-off in practice. Figure 8 shows the results on Federated EMNIST with $\beta \in \{0, \frac{1}{\sqrt{n}}, e^{-1/2}\}$ across $B \in \{14, 16, 18\}$ and $k \in \{2, 4\}$ with user-level privacy budget fixed at $\varepsilon = 3$; other parameters follow those described earlier.[14] We note that when the communication budget is tight (i.e. large k and small B, where the "room" for larger norm and noise variance is limited), the bounded norm growth from conditional rounding can be pivotal to model learning and convergence. When the communication budget is sufficient (i.e. small k and large B, where we can afford unconditional rounding), the bias introduced by $\beta > 0$ have

[14] $\beta = 0$ leads to unconditional rounding, in which case we use the worst case bound $\Delta_2 \leq \|x\|_2 + \gamma\sqrt{d}$.

insignificant impact on the model utility (e.g. $\beta = e^{-1/2} \approx 0.607$ and $\beta = 1/\sqrt{n} = 0.1$ give similar performance and convergence speed).

Privacy Degradation from Client Dropouts Figure 9 shows the privacy degradation as observed by the server if a certain percentage of the clients drops out during aggregation (thus there will be missing local noise shares). Note that for the external analyst, the server can always add the missing shares of noise onto the aggregate to prevent this degradation. Note also that the values of the parameters $(\gamma, \beta, B, c, d, k, n, T)$ does not affect the degradation as they influence each other to arrive at the same initial ε. The sampling rate q is also fixed at 1.0 as subsampling does not apply from the server's perspective.

6 Concluding Remarks

We have presented an complete end-to-end protocol for federated learning with distributed DP and secure aggregation. Our solution relies on efficiently flattening and discretizing the client model updates before adding discrete Gaussian noise and applying secure aggregation. A significant advantage of this approach is that it allows an untrusted server to perform complex learning tasks on decentralized and privacy-sensitive data while achieving the accuracy of a trusted server. Our theoretical guarantees highlight the complex tension between communication, privacy, and accuracy. Our experimental results demonstrate that our solution is essentially able to match the accuracy of central differential privacy with 16 or fewer bits of precision per value.

Several questions remain to be addressed, including (a) tightening the generic RDP amplification via sampling results or conducting a precise analysis of the subsampled distributed discrete Gaussian mechanism, (b) exploring the use of a discrete Fourier transform or other methods instead of the Walsh-Hadamard transform to avoid having to pad by (up to) $d-1$ zeros, (c) developing private self-tuning algorithms that learn how to optimally set the parameters of the algorithm on the fly, and (d) proving a lower bound on m that either confirms that the distributed discrete Gaussian's $m \geq \tilde{O}\left(n + \sqrt{\frac{\varepsilon^2 n^3}{d}} + \frac{\sqrt{d}}{\varepsilon}\right)$ is order optimal or suggests the existence of a better mechanism.

7 Acknowledgments

We thank Naman Agarwal and Kallista Bonawitz for helpful discussions and comments on drafts of this paper.

References

[ACGMMTZ16] 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*. 2016, pp. 308–318.

[ALCKS20] S. Asoodeh, J. Liao, F. P. Calmon, O. Kosut, and L. Sankar. "A Better Bound Gives a Hundred Rounds: Enhanced Privacy Guarantees via f-Divergences". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020, pp. 920–925.

[ASGXR21] M. Al-Shedivat, J. Gillenwater, E. Xing, and A. Rostamizadeh. "Federated Learning via Posterior Averaging: A New Perspective and Practical Algorithms". In: *ICLR*. 2021.

[ASYKM18] 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*. 2018, pp. 7564–7575.

[ATMR19] G. Andrew, O. Thakkar, H. B. McMahan, and S. Ramaswamy. "Differentially private learning with adaptive clipping". In: *arXiv preprint arXiv:1905.03871* (2019).

[Aut19] T. T. F. Authors. "TensorFlow Federated Stack Overflow dataset". In: (2019). URL: `https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data`.

[BBGLR20] J. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova. *Secure Single-Server Aggregation with (Poly)Logarithmic Overhead*. Cryptology ePrint Archive, Report 2020/704. `https://eprint.iacr.org/2020/704`. 2020.

[BBGN19] B. Balle, J. Bell, A. Gascón, and K. Nissim. "The Privacy Blanket of the Shuffle Model". In: *CRYPTO*. 2019, pp. 638–667.

[BBGN20] B. Balle, J. Bell, A. Gascón, and K. Nissim. "Private Summation in the Multi-Message Shuffle Model". In: (2020), pp. 657–676. URL: `https://doi.org/10.1145/3372297.3417242`.

[BC20] V. Balcer and A. Cheu. "Separating Local & Shuffled Differential Privacy via Histograms". In: *ITC*. 2020, 1:1–1:14.

[BCJM21] V. Balcer, A. Cheu, M. Joseph, and J. Mao. "Connecting robust shuffle privacy and pan-privacy". In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2021, pp. 2384–2403.

[BEMMRLRKTS17] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. "Prochlo: Strong Privacy for Analytics in the Crowd". In: *Proceedings of the Symposium on Operating Systems Principles (SOSP)*. 2017, pp. 441–459. URL: https://arxiv.org/abs/1710.00901.

[BIKMMPRSS16] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. "Practical secure aggregation for federated learning on user-held data". In: *arXiv preprint arXiv:1611.04482* (2016).

[BIKMMPRSS17] K. A. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. "Practical secure aggregation for privacy-preserving machine learning". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 1175–1191.

[Bon+19] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. "Towards Federated Learning at Scale: System Design". In: *SysML 2019*. 2019. URL: https://arxiv.org/abs/1902.01046.

[BS16] M. Bun and T. Steinke. "Concentrated differential privacy: Simplifications, extensions, and lower bounds". In: *Theory of Cryptography Conference*. Springer. 2016, pp. 635–658. URL: https://arxiv.org/abs/1605.02065.

[BS19] M. Bun and T. Steinke. "Average-case averages: Private algorithms for smooth sensitivity and mean estimation". In: *Advances in Neural Information Processing Systems*. 2019, pp. 181–191.

[BSKMG19] K. Bonawitz, F. Salehi, J. Konečnỳ, B. McMahan, and M. Gruteser. "Federated learning with autotuned communication-efficient secure aggregation". In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2019, pp. 1222–1226.

[BST14] 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*. IEEE. 2014, pp. 464–473.

[BW18] B. Balle and Y.-X. Wang. "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 394–403.

[CATVS17] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. "EMNIST: Extending MNIST to handwritten letters". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[CDWLKMST18] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. "Leaf: A benchmark for federated settings". In: *arXiv preprint arXiv:1812.01097* (2018).

[CKS20] C. Canonne, G. Kamath, and T. Steinke. "The Discrete Gaussian for Differential Privacy". In: *NeurIPS*. 2020. URL: `https://arxiv.org/abs/2004.00010`.

[CLEKS19] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. "The secret sharer: Evaluating and testing unintended memorization in neural networks". In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 2019, pp. 267–284.

[CSUZZ19] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. "Distributed differential privacy via shuffling". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2019, pp. 375–403.

[DJW13] 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*. IEEE. 2013, pp. 429–438.

[DKMMN06] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. "Our data, ourselves: Privacy via distributed noise generation". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2006, pp. 486–503.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[DSSUV15] C. Dwork, A. Smith, T. Steinke, J. Ullman, and S. Vadhan. "Robust traceability from trace amounts". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 650–669.

[EFMRTT19] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. "Amplification by shuffling: From local to central differential privacy via anonymity". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2468–2479.

[ESAG04] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. "Privacy preserving mining of association rules". In: *Information Systems* 29.4 (2004), pp. 343–364.

[GDDKS20] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh. "Shuffled Model of Federated Learning: Privacy, Communication and Accuracy Trade-offs". In: *arXiv preprint arXiv:2008.07180* (2020).

[GGKMPV20] B. Ghazi, N. Golowich, R. Kumar, P. Manurangsi, R. Pagh, and A. Velingker. "Pure Differentially Private Summation from Anonymous Messages". In: *1st Conference on Information-Theoretic Cryptography (ITC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2020.

[GGKPV21] B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. "On the Power of Multiple Anonymous Messages". In: *Eurocrypt*. To appear. 2021.

[GKMP20] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh. "Private Counting from Anonymous Messages: Near-Optimal Accuracy with Vanishing Communication Overhead". In: *ICML*. 2020, pp. 3505–3514.

[GMPV20] B. Ghazi, P. Manurangsi, R. Pagh, and A. Velingker. "Private Aggregation from Fewer Anonymous Messages". In: *Eurocrypt*. 2020.

[GMPW20] N. Genise, D. Micciancio, C. Peikert, and M. Walter. "Improved discrete Gaussian and subgaussian analysis for lattice cryptography". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2020, pp. 623–651.

[GXS13] S. Goryczka, L. Xiong, and V. Sunderam. "Secure multiparty aggregation with differential privacy: A comparative study". In: *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. 2013, pp. 155–163.

[HQB19] T.-M. H. Hsu, H. Qi, and M. Brown. "Measuring the effects of non-identical data distribution for federated visual classification". In: *arXiv preprint arXiv:1909.06335* (2019).

[HRMRBAEKR18] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. "Federated learning for mobile keyboard prediction". In: *arXiv:1811.03604* (2018).

[IKOS06] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. "Cryptography from anonymity". In: *FOCS*. 2006, pp. 239–248.

[IO19] A. Ingerman and K. Ostrowski. "Introducing TensorFlow Federated". In: (2019). URL: `https://medium.com/tensorflow/introducing-tensorflow-federated-a4147aa20041`.

[KBR16] P. Kairouz, K. Bonawitz, and D. Ramage. "Discrete distribution estimation under local privacy". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2436–2444.

[KLNRS11] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. "What Can We Learn Privately?" In: *SIAM J. Comput.* 40.3 (June 2011), pp. 793–826. ISSN: 0097-5397. URL: http://dx.doi.org/10.1137/090756090.

[KM+19] P. Kairouz, McMahan, et al. "Advances and open problems in federated learning". In: *arXiv preprint arXiv:1912.04977* (2019).

[KMSTTX21] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu. "Practical and Private (Deep) Learning without Sampling or Shuffling". In: *arXiv preprint arXiv:2103.00039* (2021).

[Kri+09] A. Krizhevsky et al. "Learning multiple layers of features from tiny images". In: (2009).

[LCB10] Y. LeCun, C. Cortes, and C. Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[MA+17] O. Marchal, J. Arbel, et al. "On the sub-Gaussianity of the Beta and Dirichlet distributions". In: *Electronic Communications in Probability* 22 (2017). URL: https://arxiv.org/abs/1705.00048.

[Mez06] F. Mezzadri. "How to generate random matrices from the classical compact groups". In: *arXiv preprint math-ph/0609050* (2006). URL: https://arxiv.org/abs/math-ph/0609050.

[Mir17] I. Mironov. "R'enyi differential privacy". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[MMRHA17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Artificial Intelligence and Statistics*. 2017, pp. 1273–1282.

[MRTZ18] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. "Learning Differentially Private Recurrent Language Models". In: *ICLR*. 2018.

[MSDCS19] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. "Exploiting unintended feature leakage in collaborative learning". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 691–706.

[MTZ19] I. Mironov, K. Talwar, and L. Zhang. "R\'enyi Differential Privacy of the Sampled Gaussian Mechanism". In: *arXiv preprint arXiv:1908.10530* (2019).

[NSTPC21] M. Nasr, S. Song, A. Thakurta, N. Papernot, and N. Carlini. "Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning". In: *arXiv preprint arXiv:2101.04535* (2021).

[RCZGRKKM20] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan. "Adaptive federated optimization". In: *arXiv preprint arXiv:2003.00295* (2020).

[RMRB19] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays. "Federated Learning for Emoji Prediction in a Mobile Keyboard". In: *arXiv:1906.04329* (2019).

[SCS13] S. Song, K. Chaudhuri, and A. D. Sarwate. "Stochastic gradient descent with differentially private updates". In: *2013 IEEE Global Conference on Signal and Information Processing*. IEEE. 2013, pp. 245–248.

[SFKM17] A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan. "Distributed mean estimation with limited communication". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3329–3337.

[SS19a] C. Song and V. Shmatikov. "Auditing data provenance in text-generation models". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 196–206.

[SS19b] C. Song and V. Shmatikov. "Overlearning reveals sensitive attributes". In: *arXiv preprint arXiv:1905.11742* (2019).

[SSSS17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. "Membership inference attacks against machine learning models". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 3–18.

[TB20] F. Tramèr and D. Boneh. "Differentially Private Learning Needs Better Features (or Much More Data)". In: *arXiv preprint arXiv:2011.11660* (2020).

[TBASLZZ19] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou. "A hybrid approach to privacy-preserving federated learning". In: *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. 2019, pp. 1–11.

[VA17] F. Valovich and F. Alda. "Computational differential privacy from lattice-based cryptography". In: *International Conference on Number-Theoretic Methods in Cryptology*. Springer. 2017, pp. 121–141.

[War65] S. L. Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.

[WBK19] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan. "Subsampled Renyi Differential Privacy and Analytical Moments Accountant". In: *Proceedings of Machine Learning Research*. Ed. by K. Chaudhuri and M. Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1226–1235. URL: http://proceedings.mlr.press/v89/wang19b.html.

[whu14] whuber. *Distribution of scalar products of two random unit vectors in D dimensions*. Cross Validated Stack Exchange. 2014. eprint: https://stats.stackexchange.com/q/85977. URL: https://stats.stackexchange.com/q/85977.

[Wik21] Wikipedia. *Fast Fourier Transform*. https://en.wikipedia.org/wiki/Fast_Fourier_transform#Other_FFT_algorithms. Feb. 2021.

[WJS21] L. Wang, R. Jia, and D. Song. "D2P-Fed: Differentially Private Federated Learning With Efficient Communication". In: *arXiv preprint arXiv:2006.13039* (2021).

[ZW19] Y. Zhu and Y.-X. Wang. "Poission subsampled rényi differential privacy". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7634–7642.