The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation

Peter Kairouz [1] Ziyu Liu [1] Thomas Steinke [1]

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.

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 (McMahan et al., 2017; Kairouz et al., 2019), cryptographic secure aggregation (SecAgg) to prevent a server from inspecting individual user updates (Bonawitz et al., 2017; Bell et al., 2020), and differentially private stochastic gradient descent (DP-SGD) to train models with provably limited information leakage (Abadi et al., 2016; Tramèr & Boneh, 2020). 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 present some preliminaries in Section 2, summarize our main results in Section 3, and review related works in Section 4. In Section 5, we introduce the distributed discrete Gaussian mechanism, analyze its privacy guarantees, and show how to apply it in federated learning. We present experiments in Section 6 and conclude with a few interesting extensions in Section 7. All formal definitions, proofs, algorithmic details, extensions, and additional experiments are deferred to the supplementary material.

2. Preliminaries

We begin by defining the Rényi divergence of order $\alpha \in (1, \infty)$ of distribution P with respect to distribution Q as

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

We now state the definitions of concentrated DP (Bun & Steinke, 2016) and Rényi DP (Mironov, 2017) and relate these to the standard definition of differential privacy (Dwork et al., 2006b;a).

Definition 1 (Concentrated Differential Privacy)**.** *A ran-*

[1] Google Research. Correspondence to: Peter Kairouz <kairouz@google.com>, Ziyu Liu <klz@google.com>, Thomas Steinke <ddg@thomas-steinke.net>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] See "Distributed DP" paragraph in Section 4 for a definition of this notion of DP and a literature review.

domized 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 $\sup_{\alpha \in (1,\infty)} \frac{1}{\alpha} \mathrm{D}_\alpha\left(M(x) \| M(x')\right) \le \frac{1}{2}\varepsilon^2$.

Definition 2 (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* $\mathrm{D}_\alpha\left(M(x) \| M(x')\right) \le \varepsilon$.

Definition 3 (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*

$$\mathbb{P}\left[M(x) \in E\right] \le e^{\varepsilon} \cdot \mathbb{P}\left[M(x') \in E\right] + \delta \tag{1}$$

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 (Bun & Steinke, 2016; Canonne et al., 2020; Asoodeh et al., 2020) from concentrated DP to approximate DP.

Lemma 4. *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*

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

We adopt *user-level privacy* in this work – i.e., each entry in the input corresponds to *all* the records associated with a single user (McMahan et al., 2018), and thus the privacy guarantee holds with respect to all data belonging to that user. 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 define DP with respect to addition or removing the records of an individual, rather than replacement. 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.

3. 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

$$\bar{z} := \sum_i^n z_i \mod m = \sum_i^n \mathcal{A}_{\text{client}}(x_i) \mod m, \tag{2}$$

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 (Bell et al., 2020; Bonawitz et al., 2016); 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 (Canonne et al., 2020) 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 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

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

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 (Bun & Steinke, 2016; Mironov, 2017; Wang et al., 2019) 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.

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: $y \approx \bar{x} = \sum_i^n x_i$}
```

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 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 $\|x_i'\|_2 \le c/\gamma$. The next step performs a unitary rotation/reflection operation $x_i'' = H_d D_\xi x_i'$ (Suresh et al., 2017). 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 – $\|\tilde{x}_i\|_2 \le \|x_i''\|_2 + \sqrt{d}$. 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 5 (Privacy of Our Algorithm)**.** *Let $c, d, \gamma, \beta, \sigma$ be the parameters of Algorithm 1 and n the number of trustworthy clients. Define*

$$\Delta_2^2 := \min\left\{ \begin{array}{c} c^2 + \frac{\gamma^2 d}{4} + \sqrt{2\log\left(\frac{1}{\beta}\right)} \cdot \gamma \cdot \left(c + \frac{\gamma}{2}\sqrt{d}\right), \\ \left(c + \gamma\sqrt{d}\right)^2 \end{array} \right\}, \tag{3}$$

$$\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2 \frac{\sigma^2}{\gamma^2} \cdot \frac{k}{k+1}}, \tag{4}$$

$$\varepsilon := \min\left\{ \begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array} \right\}. \tag{5}$$

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 $\sqrt{n}\sigma/\gamma$.

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 the accompanying supplementary material.

Theorem 6 (Accuracy of Our Algorithm)**.** *Let $n, m, d \in \mathbb{N}$ and $c, \varepsilon > 0$ satisfy*

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

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

$$\mathbb{E}\left[\left\|\tilde{A}(x) - \sum_i^n x_i\right\|_2^2\right] \leq O\left(\frac{c^2 d}{\varepsilon^2}\right). \tag{6}$$

To interpret Theorem 6, 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 (Caldas et al., 2018) and Stack Overflow (Authors, 2019) that our approach gives good performance under tight privacy budgets, despite using generic RDP amplification via sampling (Zhu & Wang, 2019) for our methods and the precise RDP analysis for the subsampled Gaussian mechanism (Mironov et al., 2019). We provide an open-source implementation of our methods in TensorFlow Privacy (Andrew et al., 2019) and TensorFlow Federated (Ingerman & Ostrowski, 2019).[4]

4. 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 (McMahan et al., 2017; Bonawitz et al., 2019a). 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 vectors (model updates) held by the clients. We refer the reader to Kairouz et al. (2019) for a survey of recent

[3] See the supplementary material or Bun & Steinke (2016) for a formal definition. 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.

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

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 (Bell et al., 2020; Bonawitz et al., 2016). 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) (Bonawitz et al., 2019b). 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 (Melis et al., 2019; Carlini et al., 2019; Song & Shmatikov, 2019a; Dwork et al., 2015; Song & Shmatikov, 2019b; Nasr et al., 2021; Shokri et al., 2017). To address this issue, differential privacy (DP) (Dwork et al., 2006b), and in particular, DP-SGD can be employed (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016; Tramèr & Boneh, 2020). 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 (Kasiviswanathan et al., 2011; Evfimievski et al., 2004; Warner, 1965) 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 (Kasiviswanathan et al., 2011; Duchi et al., 2013; Kairouz et al., 2016). 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 (Erlingsson et al., 2019; Bittau et al., 2017; Cheu et al., 2019). For more information on the shuffled model of DP, we refer the reader to Ghazi et al. (2020b; 2021; 2020c;a); Ishai et al. (2006); Balle et al. (2019; 2020); Balcer & Cheu (2020); Balcer et al. (2021); Girgis et al. (2020).

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 (Goryczka et al., 2013; Truex et al., 2019; Valovich & Alda, 2017). 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 (Agarwal et al., 2018), which also serves as an inspiration for much of our work. cpSGD uses a distributed version of the binomial mechanism (Dwork et al., 2006a) 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 (Mironov, 2017; Bun & Steinke, 2016) 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 (Canonne et al., 2020), 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 (Wang et al., 2021). 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.

5. Distributed Discrete Gaussian

We will use the discrete Gaussian (Canonne et al., 2020) as the basis of our privacy guarantee.

Definition 7 (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*

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

The discrete Gaussian has many of the desirable properties of the continuous Gaussian (Canonne et al., 2020), including the fact that it can be used to provide differential privacy.

Theorem 8 (Privacy of the Discrete Gaussian)**.** *Let $\sigma > 0$ and $\mu, \mu' \in \mathbb{Z}$. Then, for all $\alpha > 1$,*

$$\mathrm{D}_\alpha\left(\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2) \middle\| \mathcal{N}_{\mathbb{Z}}(\mu', \sigma^2)\right) \leq \alpha \cdot \frac{(\mu - \mu')^2}{2\sigma^2}, \qquad (7)$$

where $\mathrm{D}_\alpha(P\|Q)$ is the Rényi divergence of order α.

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 et al. (2020).

Theorem 9 (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*

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

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).

Theorem 9 can easily be extended to sums of more than two discrete Gaussians by the triangle inequality and to the multivariate setting by composition. Combining with Theorem 8 yields our privacy result:

Proposition 10 (Privacy for Sums of Multidimensional Discrete Gaussians)**.** *Let $\sigma \geq \frac{1}{2}$. 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 $Z_n = \sum_i^n X_i \in \mathbb{Z}^d$. Let $\tau := 10 \cdot \sum_{k=1}^{n-1} e^{-2\pi^2\sigma^2 \frac{k}{k+1}}$. 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*

$$\varepsilon = \min\left\{\begin{array}{c} \sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \\ \sqrt{\frac{\Delta_2^2}{n\sigma^2} + 2\frac{\Delta_1}{\sqrt{n}\sigma} \cdot \tau + \tau^2 d}, \\ \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d} \end{array}\right\}. \qquad (8)$$

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

Lemma 11 (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 $t \in \mathbb{R}$, $\mathbb{E}\left[e^{tX}\right] \leq e^{t^2\sigma^2/2}$.*

5.1. Applying Distributed Discrete Gaussian

Proposition 10 provides the basis of our distributed privacy guarantee – each user device adds a small amount of discrete Gaussian noise to their data so that the sum is protected. However, in order to apply this result, we must first discretize the data and, in order to use secure aggregation, we must ensure that the modular arithmetic does not introduce too much error. We now briefly describe how these steps and the analysis work. For details, formal statements, and proofs, see the supplementary material.

(Conditional) Randomized Rounding Algorithm 1 rounds $x_i'' \in \mathbb{R}^d$ to $\tilde{x}_i \in \mathbb{Z}^d$. Each coordinate is randomly and independently rounded up or down so that $\mathbb{E}[\tilde{x}_i] = x_i''$. This has many desirable properties; in particular, it is unbiased. However, the norm can increase: We have $\|\tilde{x}_i - x_i''\|_\infty < 1$ and, hence, $\|\tilde{x}_i\|_2^2 \leq \left(\|x_i''\|_2 + \sqrt{d}\right)^2$. The amount of noise we add to ensure differential privacy must scale with the sensitivity of the sum, which is determined by the norm, so we want a sharp bound.

On average, the norm increases less than this: $\mathbb{E}\left[\|\tilde{x}_i\|_2^2\right] \leq \|x_i''\|_2^2 + \frac{d}{4}$. Furthermore, we can show subgaussian high probability bounds using standard concentration of measure techniques. In order to exploit this, Algorithm 1 performs *conditional* randomized rounding. That is, if the norm of $\tilde{x}_i$ is too large, it simply re-runs the randomized rounding procedure. This introduces a small amount of bias, but allows us to keep the privacy noise to a minimum.

In contrast, cpSGD (Agarwal et al., 2018) performs unconditional randomized rounding and the sensitivity (i.e., the norm of the discretized vector) is only bounded with high probability. In their analysis, this failure probability is added to the δ of approximate (ε, δ)-differential privacy.

To control the norm, we also preprocess: The first step in Algorithm 1, $x_i' = \frac{1}{\gamma} \min\left\{1, \frac{c}{\|x_i\|_2}\right\} \cdot x_i$, clips the norm of the input vector $x_i \in \mathbb{R}^d$ if its norm exceeds c and then scales it by $1/\gamma$; thus $\|x_i'\|_2 \leq c/\gamma$. (The final step of Algorithm 2 undoes this scaling.) Scaling the input up by $1/\gamma$ is equivalent to rounding to a finer grid $\gamma\mathbb{Z}^d$, which correspondingly reduces the error introduced by the conditional randomized rounding. However, smaller γ also increases the error introduced by the modular arithmetic, which we consider next. Thus we must carefully choose the parameter γ to minimize the sum of these two sources of error.

Modular Clipping The server receives $\bar{z} = \sum_i^n z_i \mod m$, where z_i is the sum of the discretized data vector $\tilde{x}_i$ and the discrete Gaussian noise y_i. The modular arithmetic here is an undesirable side effect of secure aggregation with limited precision. Note that the modular arithmetic does *not* compromise privacy, as it can simply be treated as a postprocessing. If one of the coordinates of the data plus noise falls outside $[-m/2, m/2]$, then there will be modular "wrap around" to bring it back inside this range. This can introduce substantial error and, hence, we wish to prevent this case from arising. That is, we wish to ensure that $\|\sum_i^n \tilde{x}_i + y_i\|_\infty < m/2$ with high probability.

The bad case for modular clipping is when $\tilde{x}_i \approx x_i''$ is concentrated on one coordinate. As in previous work (Suresh et al., 2017), Algorithm 1 avoids this with the pre-processing $x_i'' = H_d D_\xi x_i'$ (which Algorithm 2 inverts at the end). Here D_ξ is a random diagonal sign matrix and H_d represents the Walsh-Hadamard transform.[5] The salient properties of H_d are that (i) it is unitary – i.e., $H_d^T H_d = I$, (ii) it has small entries – i.e., $H_d \in [-1/\sqrt{d}, 1/\sqrt{d}]^{d \times d}$, and (iii) the matrix-vector multiplication $H_d D_\xi x_i'$ can be computed with $O(d \log d)$ operations.

To understand why this unitary operation controls the infinity norm, consider the case where x_i' is concentrated on one coordinate. Then, up to a sign and scaling, x_i'' is just a column of H_d, which has small entries, as required. In this case, we don't even need the random diagonal sign matrix D_ξ. In the more general case, the random signs ensure that each entry of x_i'' is subgaussian. Namely,

$$\forall t \in \mathbb{R}\, \forall j \in [d] \quad \mathbb{E}\left[e^{t \cdot (x_i'')_j}\right] \leq e^{t^2 \|x_i'\|_2^2 / 2d} \leq e^{t^2 c^2 / 2\gamma^2 d}.$$

The distortion from the conditional randomized rounding is also subgaussian – $\mathbb{E}\left[e^{t \cdot (\tilde{x}_i - x_i'')_j}\right] \leq (1-\beta)^{-1} \cdot e^{t^2/8}$ – by Hoeffding's lemma; the $1 - \beta$ comes from the fact that conditional randomized rounding conditions on an event with probability $\geq 1 - \beta$. The discrete Gaussian noise is also subgaussian (Lemma 11) and, since these subgaussian bounds are independent, for all $j \in [d]$ and all $t \in \mathbb{R}$,

$$\mathbb{E}\left[e^{t \cdot (\tilde{x}_i + y_i)_j}\right] \leq (1-\beta)^{-1} \cdot e^{t^2 c^2 / 2\gamma^2 d + t^2/8 + t^2 \sigma^2 / 2\gamma^2}.$$

Summing over the n independent clients and using a union bound gives, for all $t \in \mathbb{R}$,

$$\mathbb{E}\left[e^{t \cdot \left\|\sum_i^n \tilde{x}_i + y_i\right\|_\infty}\right] \leq 2d \cdot (1-\beta)^{-n} \cdot e^{nt^2\left(\frac{c^2}{2\gamma^2 d} + \frac{1}{8} + \frac{\sigma^2}{2\gamma^2}\right)}.$$

By Markov's inequality, for an appropriate $t > 0$, we have

$$\mathbb{P}\left[\left\|\sum_i^n \tilde{x}_i + y_i\right\|_\infty \geq \frac{m}{2}\right] \leq \frac{\mathbb{E}\left[e^{t \cdot \left\|\sum_i^n \tilde{x}_i + y_i\right\|_\infty}\right]}{e^{tm/2}} \tag{9}$$

$$\leq \frac{2d}{(1-\beta)^n} \cdot \exp\left(\frac{-m^2\gamma^2}{n \cdot \left(\frac{8c^2}{d} + 2\gamma^2 + 8\sigma^2\right)}\right). \tag{10}$$

Thus, if $m \geq O\left(\frac{1}{\gamma}\sqrt{n \cdot \left(\frac{c^2}{d} + \gamma^2 + \sigma^2\right) \cdot (\beta n + \log d)}\right)$, then the modular clipping is unlikely to cause any error. This is, roughly, the analysis underlying Theorem 6's guarantee.

6. 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 bitwidth 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

[5] We assume d is a power of 2, as the Walsh-Hadamard transform is otherwise not defined. We can always pad the input vectors with zeros to ensure this. We discuss ways to reduce the need for padding in the supplementary material.

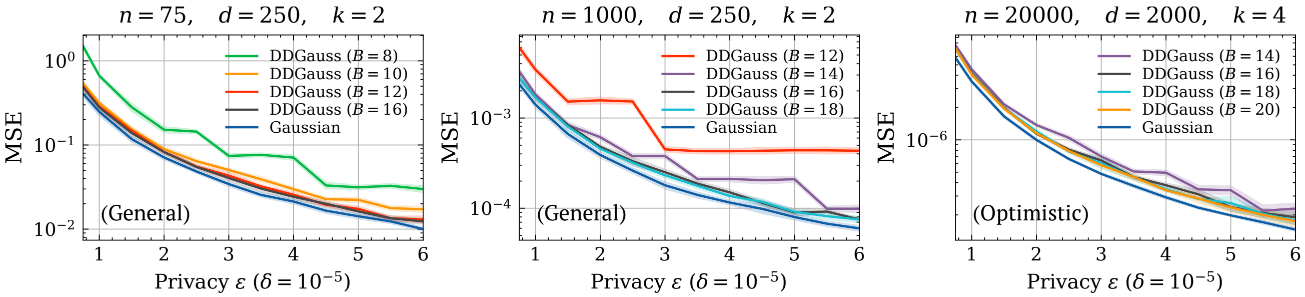

Figure 1. Distributed mean estimation with the distributed discrete Gaussian. n: number of clients. d: vector dimension. k: number of stddevs of $\sum_i^n \tilde{x}_i + y_i$ to bound. B: per-coordinate bit-width. General/Optimistic: assumes $\left\|\sum_i^n x_i\right\|_2 \leq cn$ or $\leq c\sqrt{n}$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

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 field size 2^B where $\hat{\sigma}^2 = c^2n^2/d + \left(\gamma^2/4 + \sigma^2\right) \cdot n$. Here, k captures the trade-off between the errors from quantization and modular clipping: 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. See the supplementary material for additional results and full details on experimental setup.

6.1. 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 (Balle & Wang, 2018) 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.[6] The first two plots assume a general norm bound $\left\|\sum_i^n x_i\right\|_2 \leq cn$ when choosing γ (generally applicable to FL applications), while the third plot assumes an optimistic bound $\left\|\sum_i^n x_i\right\|_2 \leq c\sqrt{n}$ as x_i's are sampled uniformly randomly on the ℓ_2 sphere. Results indicate that DDGauss achieves a good communication-utility trade-off and matches the Gaussian with sufficient bit-widths.

6.2. Federated Learning

We evaluate on two realistic FL tasks: Federated EMNIST (Cohen et al., 2017; Caldas et al., 2018) and Stack Overflow Next Word Prediction (SO-NWP, Authors (2019)). We defer additional results and full details on datasets, models, and setup to the supplementary material.

Datasets EMNIST is an image dataset with hand-written digits/letters over 62 classes grouped into N = 3400 clients by their writer. Stack Overflow is a text dataset based on questions/answers from stackoverflow.com with sentences grouped by the N = 342477 users. These datasets differ from those commonly used in related work (e.g. MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky et al., 2009)) in that they are substantially larger and involve *user-level* (instead of example-level) DP with natural 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. See also Reddi et al. (2020) for more details.

Models For EMNIST, we train a small convolutional net similar to the model defined in Reddi et al. (2020) but with model size d slightly under 2^{20} parameters to reduce padding from the Walsh-Hadamard transform. For SO-NWP, we use the architecture from Reddi et al. (2020).

Setup For both tasks, we train with federated averaging with server momentum of 0.9 (McMahan et al., 2017; Hsu et al., 2019). In each round, we uniformly sample $n = 100$ clients without replacement following Andrew et al. (2019) and 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 with replacement across rounds. For EMNIST and SO-NWP respectively, we set the number of rounds T to 1500 and 1600, c to 0.03 and 0.3, client learning rate η_c to 0.032 and 0.5, and client batch size to 20 and 16. Server LR η_s is set to 1 for EMNIST and selected from a small grid $\{0.3, 1\}$ for SO-NWP. 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). The privacy guarantees ε we report rely on privacy amplification via sampling (Kasiviswanathan et al., 2011; Bassily et al., 2014; Abadi et al., 2016), which is necessary to obtain reasonable privacy-accuracy tradeoffs

[6] 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.

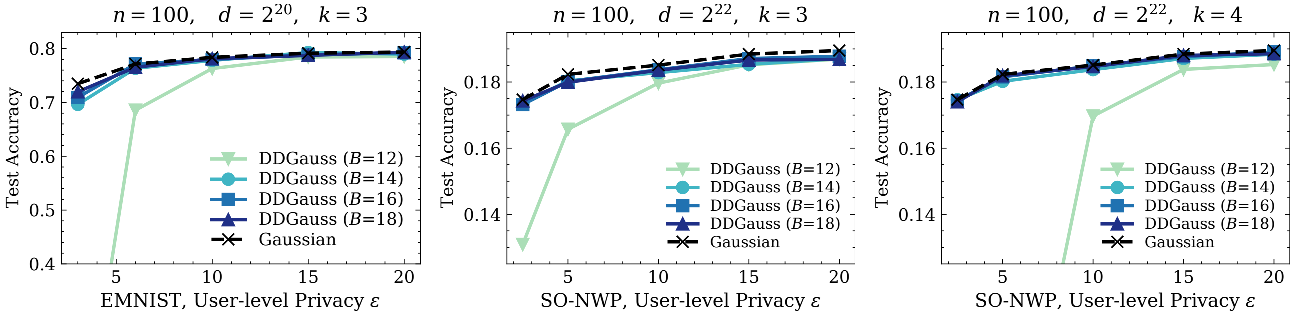

Figure 2. Test accuracies on EMNIST (averaged over last 100 rounds) and SO-NWP. $\delta = 1/N$ for EMNIST and 10^{-6} for SO-NWP.

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 Mironov et al. (2019) for Gaussian and the generic upper bound from Zhu & Wang (2019) for DDGauss (we do not explore a precise analysis in this work), which could lead to more noise being added for DDGauss to achieve the same privacy as Gaussian.

Results Figure 2 shows the test accuracies on EMNIST and SO-NWP respectively. Overall, with more communication and privacy budget, DDGauss achieves a better utility both relative to Gaussian and in absolute performance, and it can match Gaussian as long as B is sufficient. Note also the error trade-off between modular clipping and quantization: on SO-NWP, $k = 3$ allows $B = 12$ to match higher bit-widths when noise is small, but it introduces a slight accuracy gap to Gaussian; setting $k = 4$ allows DDGauss to close the gap, but it leads to worse performance at $B = 12$.

Figure 3 scales up to $n = 1000$ clients on SO-NWP (similar to production settings described in Hard et al. (2018); Ramaswamy et al. (2019)) and shows the validation accuracies during training across different noise multipliers.[7] We set $c = 1$ and $\eta_s = 1$ for $z \approx 0.3$ and $z \approx 0.5$ and $\eta_s = 3$ otherwise. $z \approx 0.07$ gives a target test accuracy of around 25.2% while other noise levels give ε of 10 and 234 respectively. Results indicate that DDGauss can match the continuous Gaussian in large-scale, production-like settings.

7. 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.

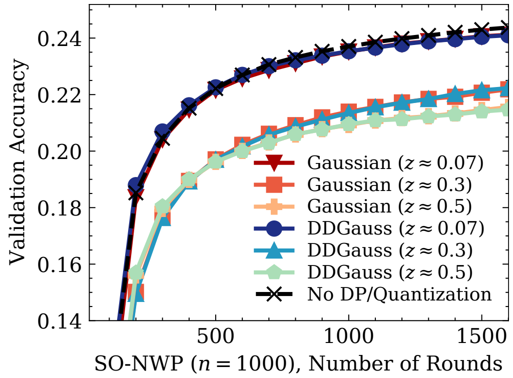

Figure 3. Validation accuracies on SO-NWP (averaged every 100 rounds) with $n = 1000$ and $B = 18$. z is the noise multiplier.

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.

8. Acknowledgments

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

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

References

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

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

Andrew, G., Thakkar, O., McMahan, H. B., and Ramaswamy, S. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

Asoodeh, S., Liao, J., Calmon, F. P., Kosut, O., and Sankar, L. A better bound gives a hundred rounds: Enhanced privacy guarantees via f-divergences. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 920–925, 2020. doi: 10.1109/ISIT44484.2020.9174015.

Authors, T. T. F. Tensorflow federated stack overflow dataset. 2019. URL https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data.

Balcer, V. and Cheu, A. Separating local & shuffled differential privacy via histograms. In *ITC*, pp. 1:1–1:14, 2020.

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

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

Balle, B., Bell, J., Gascón, A., and Nissim, K. The privacy blanket of the shuffle model. In *CRYPTO*, pp. 638–667, 2019.

Balle, B., Bell, J., Gascón, A., and Nissim, K. Private summation in the multi-message shuffle model. pp. 657–676, 2020. doi: 10.1145/3372297.3417242. URL https://doi.org/10.1145/3372297.3417242.

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

Bell, J., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. Secure single-server aggregation with (poly)logarithmic overhead. Cryptology ePrint Archive, Report 2020/704, 2020. https://eprint.iacr.org/2020/704.

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

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

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C. M., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design. In *SysML 2019*, 2019a. URL https://arxiv.org/abs/1902.01046.

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

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

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

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

Canonne, C., Kamath, G., and Steinke, T. The discrete gaussian for differential privacy. In *NeurIPS*, 2020. URL https://arxiv.org/abs/2004.00010.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX}*

Security Symposium ({USENIX} Security 19), pp. 267–284, 2019.

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

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

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

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

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

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

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

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

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

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. Pure differentially private summation from anonymous messages. In *1st Conference on Information-Theoretic Cryptography (ITC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020a.

Ghazi, B., Kumar, R., Manurangsi, P., and Pagh, R. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pp. 3505–3514, 2020b.

Ghazi, B., Manurangsi, P., Pagh, R., and Velingker, A. Private aggregation from fewer anonymous messages. In *Eurocrypt*, 2020c.

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. On the power of multiple anonymous messages. In *Eurocrypt*, 2021. To appear.

Girgis, A. M., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of federated learning: Privacy, communication and accuracy trade-offs. *arXiv preprint arXiv:2008.07180*, 2020.

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

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

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

Ingerman, A. and Ostrowski, K. Introducing tensorflow federated. 2019. URL https://medium.com/tensorflow/introducing-tensorflow-federated-a4147aa20041.

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

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

Kairouz, P., McMahan, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, June 2011. ISSN 0097-5397. doi: 10.1137/090756090. URL http://dx.doi.org/10.1137/090756090.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

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

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *ICLR*, 2018.

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

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

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

Nasr, M., Song, S., Thakurta, A., Papernot, N., and Carlini, N. Adversary instantiation: Lower bounds for differentially private machine learning. *arXiv preprint arXiv:2101.04535*, 2021.

Ramaswamy, S., Mathews, R., Rao, K., and Beaufays, F. Federated learning for emoji prediction in a mobile keyboard. *arXiv:1906.04329*, 2019.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

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

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

Song, C. and Shmatikov, V. Overlearning reveals sensitive attributes. *arXiv preprint arXiv:1905.11742*, 2019b.

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

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

Tramèr, F. and Boneh, D. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*, 2020.

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

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

Wang, L., Jia, R., and Song, D. D2p-fed: Differentially private federated learning with efficient communication. *arXiv preprint arXiv:2006.13039*, 2021.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. Subsampled renyi differential privacy and analytical moments accountant. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1226–1235. PMLR, 2019. URL `http://proceedings.mlr.press/v89/wang19b.html`.

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

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