The Skellam Mechanism for Differentially Private Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce the multi-dimensional Skellam mechanism, a discrete differential privacy mechanism based on the difference of two independent Poisson random variables. To quantify its privacy guarantees, we analyze the privacy loss distribution via a numerical evaluation and provide a sharp bound on the Rényi divergence between two shifted Skellam distributions. While useful in both centralized and distributed privacy applications, we investigate how it can be applied in the context of federated learning with secure aggregation under communication constraints. Our theoretical findings and extensive experimental evaluations demonstrate that the Skellam mechanism provides the same privacy-accuracy trade-offs as the continuous Gaussian mechanism, even when the precision is low. More importantly, Skellam is closed under summation and sampling from it only requires sampling from a Poisson distribution – an efficient routine that ships with all machine learning and data analysis software packages. These features, along with its discrete nature and competitive privacy-accuracy trade-offs, make it an attractive alternative to the newly introduced discrete Gaussian mechanism.

1 Introduction

The Gaussian mechanism is the workhorse for a multitude of differentially private learning algorithms [33, 7, 1]. While simple enough for mathematical reasoning and privacy accounting analyses, its continuous nature presents a number of challenges in practice. For example, it cannot be exactly represented on finite computers, making it prone to numerical errors that can break its privacy guarantees [28]. Moreover, it cannot be used in distributed learning settings with cryptographic multi-party computation primitives involving modular arithmetic, such as secure aggregation [9, 8]. To address these shortcomings, the binomial and (distributed) discrete Gaussian mechanisms were recently introduced [15, 2, 12, 19]. Unfortunately, both have their own drawbacks: the privacy loss for the binomial mechanism can be infinite with a non-zero probability, and the discrete Gaussian: (a) is not closed under summation (i.e. sum of discrete Gaussians is not a discrete Gaussian), complicating analysis in distributed settings and leading to a performance worse than continuous Gaussian in the highly distributed, low-noise regime [19]; (b) requires a sampling algorithm that is not shipped with mainstream machine learning or data analysis software packages, making it difficult for engineers to use it in production settings (naïve implementations may lead to catastrophic privacy errors).

Our contributions To overcome the abovementioned limitations, we introduce and analyze the multi-dimensional Skellam mechanism, a mechanism based on adding noise distributed according to the difference of two independent Poisson random variables. The Skellam mechanism is closed under summation (i.e. sums of Skellam random variables is Skellam distributed) and can be sampled from easily – efficient Poisson sampling routines are widely available in numerical software packages.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

To analyze the privacy guarantees of the Skellam mechanism and compare it with other mechanisms, we provide a numerical evaluation of the privacy loss random variable and prove a sharp bound on the Rényi divergence between two shifted Skellam distributions. Our careful analysis shows that for a multi-dimensional query function with ℓ_1 sensitivity Δ_1 and ℓ_2 sensitivity Δ_2, the Skellam mechanism with variance μ achieves $(\alpha, \varepsilon(\alpha))$ Rényi differential privacy (RDP) [29] for $\varepsilon(\alpha) \leq \frac{\alpha\Delta_2}{2\mu} + \min\left(\frac{(2\alpha-1)\Delta_2+6\Delta_1}{4\mu^2}, \frac{3\Delta_1}{2\mu}\right)$ (see Theorem 3.5). This implies that the RDP guarantees of the Skellam mechanism are at most $1 + O(1/\mu)$ worse than those of the Gaussian mechanism.

To analyze the performance of the Skellam mechanism in practice, we consider a differentially private and communication constrained federated learning (FL) setting [20] where the noise is added locally to client updates (d-dimensional discretized gradients) and summed securely via a cryptographic multi-party computation protocol, such as secure aggregation (SecAgg) [8, 9]. We provide an end-to-end algorithm that appropriately discretizes the data and applies the Skellam mechanism followed by modular clipping to bound the range of the data before applying SecAgg.

We demonstrate that on distributed mean estimation and two benchmark FL datasets, Federated EMNIST [11] and Stack Overflow [5], our method can match the performance of the continuous Gaussian baseline under tight privacy and communication budgets, despite using generic RDP amplification via sampling [38] for our approach and the precise RDP analysis for the subsampled Gaussian mechanism [30]. We implement our algorithms in TensorFlow Privacy [25] and TensorFlow Federated [18]. Even though we mostly focus on FL applications, we note that the Skellam mechanism can be applied in other contexts of learning and analytics, including centralized settings.

Related work The scalar version of the Skellam mechanism was first introduced in the context of computational differential privacy from lattice-based cryptography [36]. However, the privacy analysis in [36] relies on providing a direct bound on $\delta(\varepsilon)$, making only advanced composition theorems [22, 17] applicable in learning settings where the mechanism is applied many times. As demonstrated in Figure 1, this leads to poor accuracy-privacy trade-offs. Moreover, we demonstrate in Section 3.1 that extending the direct privacy analysis to the multi-dimensional setting is non-trivial because the worst-case neighboring dataset pair is unknown in this case. Our tight privacy analysis relies on proving a sharp bound on Rényi differential privacy (RDP) guarantees of the Skellam mechanism. These guarantees (almost) match those of the Gaussian mechanism and allow us to use generic RDP amplification via subsampling methods [38].

The closest mechanisms to Skellam are the binomial [2, 15] and the discrete Gaussian mechanisms [12, 19]. The binomial mechanism can (asymptotically) match the continuous Gaussian mechanism (when properly scaled). However, it does not achieve Rényi or concentrated DP [29, 10] and has a privacy loss that can be infinite with a non-zero probability, leading to catastrophic privacy failures. The discrete Gaussian mechanism yields Rényi DP and can be applied to distributed settings [19], but it requires a sampling algorithm that is not available in data analysis software packages. The discrete Gaussian is also not closed under summation and the divergence can be large in highly distributed low-noise settings, which causes privacy degradation.

2 Preliminaries

We begin by providing a formal definition for (ε, δ)-Differential Privacy (DP) [16].

Definition 2.1 (Differential Privacy)**.** *For $\varepsilon, \delta \geq 0$, a randomized mechanism M satisfies (ε, δ)-DP if for all neighboring datasets D, D' and all $\mathcal{S}$ in the range of M, we have that*

$$P\left(M(D) \in \mathcal{S}\right) \leq e^{\varepsilon} P\left(M(D') \in \mathcal{S}\right) + \delta,$$

where D and D' are neighboring pairs if they can be obtained from each other by adding or removing all the records that belong to a particular user.

We adopt *user-level differential privacy* – i.e., D and D' are neighboring pairs if one of them can be obtained from the other by adding or removing *all* the records associated with a single user [26]. 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 also make use of Rényi Differential Privacy (RDP) [29] which allows for tight privacy accounting.

Definition 2.2 (Rényi Differential Privacy)**.** *A mechanism M satisfies (α, ε)-RDP if for any two neighboring datasets D, D', we have that*

$$D_\alpha(M(D), M(D')) \leq \varepsilon,$$

where $D_\alpha(P, Q)$ is the Rényi Divergence between P and Q and is given by

$$D_\alpha(P, Q) \triangleq \frac{1}{\alpha - 1} \log \left(\mathbb{E}_{x \sim Q} \left[\left(\frac{P(x)}{Q(x)} \right)^\alpha \right] \right) = \frac{1}{\alpha - 1} \log \left(\mathbb{E}_{x \sim P} \left[\left(\frac{P(x)}{Q(x)} \right)^{\alpha - 1} \right] \right).$$

A closely related privacy notion is concentrated DP (CDP) [10]. In fact, $\frac{1}{2}\varepsilon^2$-CDP is equivalent to simultaneously satisfying an infinite family of RDP guarantees, namely $(\alpha, \frac{1}{2}\varepsilon^2\alpha)$-Rényi differential privacy for all $\alpha \in (1, \infty)$. The following conversion lemma from [10, 12, 4] relates RDP to (ε, δ)-DP.

Lemma 2.3. *If M satisfies (α, ε)-RDP, then, for any $\delta > 0$, M satisfies $(\varepsilon_{DP}(\delta), \delta)$-DP, where*

$$\varepsilon_{DP}(\delta) = \inf_{\alpha > 1} \varepsilon + \frac{\log(1/\alpha\delta)}{\alpha - 1} + \log(1 - 1/\alpha).$$

For any query function f, we define the Δ_p sensitivity as $\max_{D,D'} \|f(D) - f(D')\|_p$, where D and D' are neighboring pairs differing by adding or removing all the records from a particular user.

3 The Skellam Mechanism

We begin by presenting the definition of the Skellam distribution, which is the basis of the Skellam Mechanism for releasing integer ranged multi-dimensional queries.

Definition 3.1 (Skellam Distribution)**.** *The multidimensional Skellam distribution over $\mathbb{Z}^d$ with mean $\Delta \in \mathbb{Z}^d$ and variance μ denoted as $\mathrm{Sk}_{\Delta,\mu}(X)$ is given with each coordinate X_i distributed independently as*

$$X_i \sim \mathrm{Sk}_{\Delta_i,\mu}(X_i) \triangleq e^{-\mu} I_{k-\Delta_i}(\mu),$$

for $k \in \mathbb{Z}$. Here, $I_\nu(x)$ is the modified Bessel function of the first kind. A key property of Skellam random variables which motivates their use in DP is that they are closed under summation, i.e. let $X_1 \sim \mathrm{Sk}_{\Delta_1,\mu_1}$ and $X_2 \sim \mathrm{Sk}_{\Delta_2,\mu_2}$ then $X_1 + X_2 \sim \mathrm{Sk}_{\Delta_1+\Delta_2,\mu_1+\mu_2}$. This follows from the fact that a Skellam random variable X can be obtained by taking the difference between two independent Poisson random variables with means μ.[1] We are now ready to introduce the Skellam Mechanism.

Definition 3.2 (The Skellam Mechanism)**.** *Given an integer-valued query $f(D) \in \mathbb{Z}^d$, the Skellam Mechanism is given by*

$$f(D) + Z \text{ where } X \sim \mathrm{Sk}_{0,\mu},$$

and the total ℓ_2 error of the mechanism is bounded by $\mathbb{E}\left[\|\mathrm{Sk}_{0,\mu}(f(D)) - f(D)\|_2^2\right] \leq d\mu$.

The Skellam mechanism was first introduced in [36] for the scalar case. As our goal is to apply the Skellam mechanism in the learning context, we have to address the following challenges. (1) *Tight privacy compositions:* Learning algorithms are iterative in nature and require the application of the DP mechanism many times (often > 1000). The current direct approximate DP analysis in [36] can be combined with advanced composition (AC) theorems [22, 17] but that leads to poor privacy-accuracy trade-offs (see Fig 1).

(2) *Privacy analysis for multi-dimensional queries:* In learning algorithms, the differentially private queries are multi-dimensional (where the dimension equals the number of model parameters, typically $\geq 10^6$). Using composition theorems lead to poor accuracy-privacy trade-offs and a direct extension of approximate DP guarantee [36] for the multi-dimensional case leads to a strong dependence on ℓ_1 sensitivity which is prohibitively large in high dimensions. (3) *Data discretization:* The gradients are naturally continuous vectors but we would like to apply an integer based mechanism. This requires properly discretizing the data while making sure that the norm of the vectors (sensitivity of the query) is preserved. We will tackle challenges (1) and (2) in the remainder of this section and leave (3) for the next section.

3.1 Tight Numerical Accounting via Privacy Loss Distributions

[1] We only consider the symmetric version of Skellam, but it is often more generally defined as the difference of independent Poisson random variables with different variances.

We begin by defining the notion of privacy loss distributions (PLDs).

Definition 3.3 (Privacy Loss Distribution)**.** *For a multi-dimensional discrete privacy mechanism M and neighboring datasets D, D', for any $x \in \mathbb{Z}^d$, we define $f(x) = \log\left(\frac{M(D)=x}{M(D')=x}\right)$. The privacy loss random variable of M at (D, D') is $Z_{D,D'} = f(M(D))$ [17]. The privacy loss distribution (PLD) of M, denoted by $\mathrm{PLD}_{D,D'}$, is the distribution of $Z_{D,D'}$.*

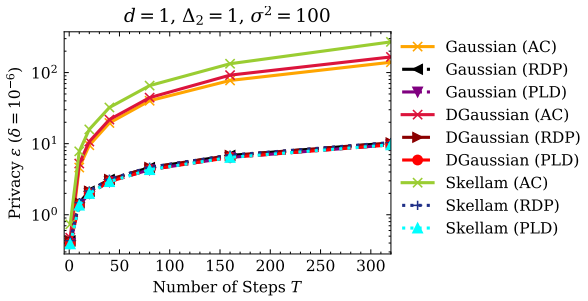

Figure 1: Comparing privacy compositions across various mechanisms and accounting methods.

The PLD of a mechanism M can be used to characterize its (ε, δ)-DP guarantees.

Lemma 3.4. *A mechanism M is (ε, δ)-DP if and only if the following holds for all neighboring datasets D, D' where $[x]_+ = \max(0, x)$,*

$$\delta \geq \mathbb{E}_{Z \sim \mathrm{PLD}_{D,D'}} \left[1 - e^{\varepsilon - Z}\right]_+ . \tag{3.1}$$

When a mechanism M is applied T times on a dataset, the overall PLD of the composed mechanism at (D, D') is the T-fold convolution of $\mathrm{PLD}_{D,D'}$ [17]. Since discrete convolutions can be computed efficiently using fast Fourier transforms (FFTs) and the expectation in Eq. (3.1) can be numerically approximated, PLDs are attractive for tight numerical accounting [24, 27, 14]. Applying the above to the Skellam mechanism, a direct calculation shows that

$$Z_{D,D'} = \sum_{i=1}^{d} \log\left(\frac{I_{X_i - f(D)_i}(\mu)}{I_{X_i - f(D')_i}(\mu)}\right),$$

where the X_i's are independent and distributed according to $\mathrm{Sk}_{0,\mu}$. When $d = 1$, it suffices to look at $Z = \log(I_{X-\Delta}(\mu)/I_X(\mu))$, where $\Delta = \max_{D,D'} |f(D) - f(D')|$ and $X \sim \mathrm{Sk}_{0,\mu}$. Since X has a discrete and symmetric probability distribution and that log is monotonic, the distribution of Z can be easily characterized. This gives us a tight numerical accountant for the Skellam mechanism in the scalar case, which we use to compare it with both the Gaussian and Discrete Gaussian mechanisms. Fig 1 shows this comparison, highlighting the competitiveness of the Skellam mechanism and the problem of combining the direct analysis of [36] with advanced composition (AC) theorems. When $d > 1$, there are combinatorially many $Z_{D,D'}$'s that need to be considered, even when the ℓ_2 sensitivity of $f(D)$ is bounded. The discrete Gaussian mechanism faces a similar issue (see Theorem 15 of [12]). To provide a tight privacy analysis in the multi-dimensional case, we follow the approach of Rényi DP under which we provide a tight bound in the following subsection.

3.2 Tight Accounting via Rényi Differential Privacy

The following theorem states our main theoretical result in the paper, providing a relatively sharp bound on the RDP properties for the Skellam machanism.

Theorem 3.5. *For $\alpha \in \mathbb{Z}$, $\alpha > 1$ and sensitivity $\Delta \in \mathbb{Z}$, the Skellam Mechanism is (α, ε)-RDP with*

$$\varepsilon(\alpha) \leq \frac{\alpha \Delta^2}{2\mu} + \min\left(\frac{(2\alpha - 1)\Delta^2 + 6\Delta}{4\mu^2}, \frac{3\Delta}{2\mu}\right), \tag{3.2}$$

To remind the reader in comparison, the Gaussian mechanism is (α, ε)-RDP with $\varepsilon(\alpha) = \frac{\alpha \Delta^2}{2\mu}$. The bound we provide is at most $1 + O(1/\mu)$ worse than the bound for the Gaussian, which is negligible for all practical choices of μ, especially as the privacy requirements increase.[2] Next we show a simple corollary which follows via the independent composition of RDP across dimensions.

Corollary 3.6. *The multi-dimensional Skellam Mechanism is (α, ε)-RDP with*

$$\varepsilon(\alpha) = \frac{\alpha \Delta_2^2}{2\mu} + \min\left(\frac{(2\alpha - 1)\Delta_2^2 + 6\Delta_1}{4\mu^2}, \frac{3\Delta_1}{2\mu}\right). \tag{3.3}$$

[2] The restriction that α needs to be an integer is a technical one owing to known bounds on Bessel functions, in practice as we show, this restriction has a negligible effect.

3.2.1 Proof Overview for Theorem 3.5

In this subsection, we provide the proof of Theorem 3.5 assuming a technical bound on the ratios of Bessel functions presented as Lemma 3.7, which may be of independent interest and is the core of our analysis. We provide a proof overview for Lemma 3.7, deferring the full proof to the appendix.

On a macroscopic level, our proof structure largely mimics the RDP proof for the Gaussian mechanism. The main object of our interest is to bound the following quantity, defined for any X, Δ, α:

$$\Phi_{X,\alpha,\Delta}(\mu) \triangleq \log\left(\frac{I_{X-\Delta}(\mu)}{I_{X-\alpha\Delta}(\mu)}\left(\frac{I_{X-\Delta}(\mu)}{I_X(\mu)}\right)^{\alpha-1}\right). \tag{3.4}$$

The following lemma states our main bound on this quantity.

Lemma 3.7. *For any $X, \alpha \in \mathbb{N}$, with $, \alpha > 1$ and $\Delta \in \mathbb{Z}$, we have that for all $\mu \geq 0$*

$$\Phi_{X,\alpha,\Delta}(\mu) \leq \frac{\alpha(\alpha-1)\Delta^2}{2\mu} + \min\left(\frac{(2\alpha-1)(\alpha-1)\Delta^2}{4\mu^2} + \frac{3(\alpha-1)|\Delta|}{2\mu^2}, \frac{3(\alpha-1)|\Delta|}{2\mu}\right)$$

Note that in contrast if we consider the analogous notion of Φ for the Gaussian mechanism (replacing $I_X(\mu)$ with the Gaussian density $e^{-X^2/2\mu}$), we readily get the bound $\frac{\alpha(\alpha-1)\Delta^2}{2\mu}$, which is the same as our bound up to lower order terms. We now provide the proof of Theorem 3.5.

Proof of Theorem 3.5. By the definition of RDP, we need to bound the following for any Δ, $\alpha \geq 1$,

$$\frac{\log\left(\sum_{X=-\infty}^{\infty} \mathrm{Sk}_{\Delta,\mu}(X)\left(\frac{\mathrm{Sk}_{\Delta,\mu}(X)}{\mathrm{Sk}_{0,\mu}(X)}\right)^{\alpha-1}\right)}{\alpha-1} = \frac{\log\left(\sum_{X=-\infty}^{\infty} e^{-\mu} I_{X-\Delta}(\mu)\left(\frac{I_{X-\Delta}(\mu)}{I_X(\mu)}\right)^{\alpha-1}\right)}{\alpha-1}$$

Now consider the following calculations:

$$\begin{aligned}
\log\left(\sum_{X=-\infty}^{\infty} \frac{I_{X-\Delta}(\mu)}{e^{\mu}}\left(\frac{I_{X-\Delta}(\mu)}{I_X(\mu)}\right)^{\alpha-1}\right) &= \log\left(\sum_{X=-\infty}^{\infty} \frac{I_{X-\alpha\Delta}(\mu)}{e^{\mu}} e^{\Phi_{X,\alpha,\Delta(\mu)}}\right) \\
&\leq \log\left(\sum_{X=-\infty}^{\infty} e^{-\mu} I_{X-\alpha\Delta}(\mu)\right) + \max_{X\in\mathbb{Z}} \Phi_{X,\alpha,\Delta(\mu)} \\
&\leq \frac{\alpha(\alpha-1)\Delta^2}{2\mu} + \min\left(\frac{(2\alpha-1)(\alpha-1)\Delta^2}{4\mu^2} + \frac{3(\alpha-1)|\Delta|}{2\mu^2}, \frac{3(\alpha-1)|\Delta|}{2\mu}\right),
\end{aligned}$$

where the inequality follows from Lemma 3.7. □

We now provide an overview for the proof of Lemma 3.7 highlighting the crux of the argument. As a first step we collect some known facts regarding regarding Bessel functions. It is known that for $x \geq 0$ and $\nu \in \mathbb{Z}$, $\nu \geq 0$, $I_\nu(x)$ is a decreasing function in ν, $I_{-\nu}(x) = I_\nu(x)$ and $\frac{I_{\nu-1}(\mu)}{I_\nu(x)}$ is an increasing function in ν [34]. A succession of works consider bounding the ratio of successive Bessel functions, $I_{\nu-1}(x)/I_\nu(x)$ which is a natural quantity to considering the objective in Lemma 3.7. We use the following very tight characterization for this recently proved in [32, Theorem 5].

Lemma 3.8. *For any $\nu \geq 1/2, x \geq 0$ define the following function we have that*

$$\operatorname{arcsinh}(\delta_0(\nu,x)) \leq \log(I_{\nu-1}(x)) - \log(I_\nu(x)) \leq \operatorname{arcsinh}(\delta_2(\nu,x))$$

where $\delta_\alpha(\nu,x)$ is defined as $\delta_\alpha(\nu,x) \triangleq \frac{\nu-1/2}{x} + \frac{\nu+(\alpha-1)/2}{2x\sqrt{(\nu+(\alpha-1)/2)^2+x^2}}$.

Standard bounds such as those appearing in [3, 36] lead to the following conclusion:

$$\operatorname{arcsinh}\left((\nu-1/2)/x\right) \leq \log(I_{\nu-1}(x)) - \log(I_\nu(x)) \leq \operatorname{arcsinh}\left(\nu/x\right).$$

While the above bound is significantly easier to work with, it leads to an RDP guarantee of Gaussian RDP + $O(\frac{\Delta}{\mu})$. In high dimensions this manifests as $O(\frac{\Delta_1}{\mu})$ and overall leads to a constant multiplicative factor over the Gaussian. On the other hand we prove a Gaussian RDP + $o_\mu(1)$ bound. Our proof of Lemma 3.7 splits into various cases depending on the signs of the quantities involved. We show the derivation for a single case below and defer the full proof to the appendix.

Proof of Lemma 3.7 in the case $X \geq \alpha\Delta$, $\Delta \geq 0$. Replacing $Y = X - \alpha\delta$ we get that

$$\begin{aligned}
\Phi_{X,\alpha,\Delta}(\mu) &= \log\left(\frac{I_{Y+(\alpha-1)\Delta}(\mu)}{I_Y(\mu)}\left(\frac{I_{Y+(\alpha-1)\Delta}(\mu)}{I_{Y+\alpha\Delta}(\mu)}\right)^{\alpha-1}\right) \\
&= \sum_{j=0}^{\alpha-2}\left(\sum_{i=Y+j\Delta+1}^{Y+j\Delta+\Delta}\left(\log\left(\frac{I_{i-1+(\alpha-1-j)}(\mu)}{I_{i+(\alpha-1-j)\Delta}(\mu)}\right) - \log\left(\frac{I_{i-1}(\mu)}{I_i(\mu)}\right)\right)\right) \\
&\leq \sum_{j=0}^{\alpha-2}\left(\sum_{i=Y+j\Delta+1}^{Y+j\Delta+\Delta}\left(\delta_2(i+(\alpha-1-j)\Delta,\mu) - \delta_0(i,\mu)\right)\right) \\
&\leq \frac{\alpha(\alpha-1)\Delta^2}{2\mu} + \min\left(\frac{\alpha(\alpha-1)\Delta^2 + 2(\alpha-1)\Delta}{4\mu^2}, \frac{(\alpha-1)\Delta}{2\mu}\right),
\end{aligned}$$

where the first inequality follows from Lemma 3.8 and the fact that for all $0 \leq x \leq y$, $\operatorname{arcsinh}(y) - \operatorname{arcsinh}(x) \leq y - x$ and the second inequality follows from Lemma B.1 (provided in the appendix):

$$\delta_2(\nu_1, x) - \delta_0(\nu_2, x) \leq \frac{\nu_1 - \nu_2}{x} + \frac{1}{2x}\min\left(\frac{\nu_1 - \nu_2 + 1}{x}, 1\right).$$

□

4 Applying the Skellam Mechanism to Federated Learning

With a sharp RDP analysis for the multi-dimensional Skellam mechanism presented in the previous section, we are now ready to apply it to differentially private federated learning. We first outline the general problem setting and then describe our approach under central and distributed DP models.

Problem setting At a high-level, we consider the distributed mean estimation problem. There are n clients each holding a vector x_i in $\mathbb{R}^d$ such that for all i, the vector norm is bounded as $\|x_i\|_2 \leq c$ for some $c \geq 0$. We denote the set of vectors as $\mathcal{X} = \{x_i\}_{i=1}^n$, and the aim is for each client to communicate the vectors x_i to a central server which then aggregates them as $\widehat{x} = \frac{1}{n}\sum_i x_i$ for an external analyst. In federated learning, the client vectors x_i are the model gradients or model deltas (typically $d \geq 10^6$) after training on the clients' local datasets, and this procedure can be repeated for many rounds ($T > 1000$). A large d and T thus necessitate accounting methods that provide tight privacy compositions for high-dimensional queries.

We are primarily concerned with three metrics for this procedure and their trade-offs: (1) *Privacy*: the mean $\widehat{x}$ should be differentially private with a reasonably small (ε, δ); (2) *Error*: we wish to minimize the expected ℓ_2 error; and (3) *Communication*: we wish to minimize the average number of bits communicated per coordinate. Characterizing this trade-off is an important research problem. For example, it has been recently shown [37] that without formal privacy guarantees, the client training data could still be revealed by the model updates x_i; on the other hand, applying differential privacy [35] to these updates can degrade the final utility.

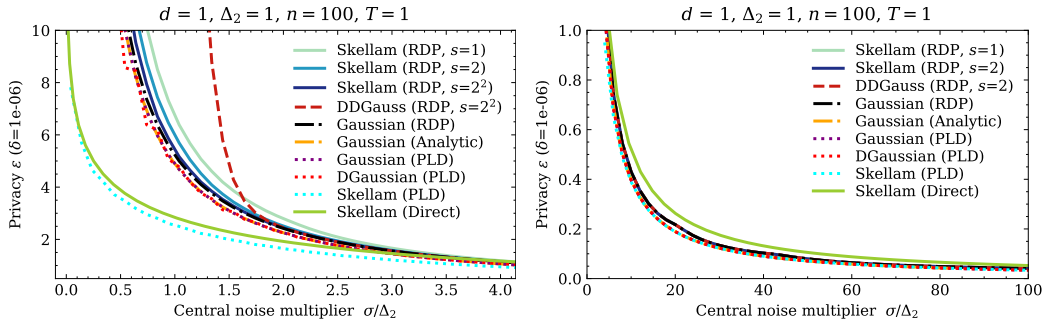

Figure 2: Benchmarking the Skellam mechanism on sensitivity-1 queries under various accounting methods. RDP: Rényi DP. PLD: privacy loss distributions. Gaussian (Analytic): [6]. DGaussian: central discrete Gaussian. DDGauss: distributed discrete Gaussian. s denotes the scaling applied to both Δ and σ. **Left**: $\varepsilon <= 10$. **Right**: $\varepsilon <= 1$.

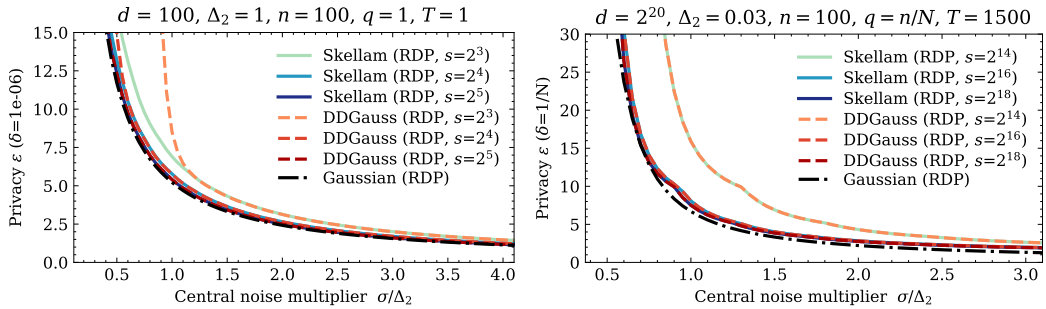

Figure 3: Comparing Skellam and Distributed Discrete Gaussian (DDGauss) on real-valued queries, stochastically rounded with $\beta = \exp(-1/2)$. s is the scaling applied to both σ and Δ_2, which reduces the rounding error and the norm inflation. q is the sampling rate. **Left:** A simple setting with $\Delta_2 = 1$. **Right:** A realistic FL setting for training CNNs on Federated EMNIST (see Section 5).

Skellam for central DP The central DP model refers to adding Skellam noise onto the non-private aggregate $\widehat{x}$ before releasing it to the external analyst. One important consideration is that the model updates in FL are continuous in nature, while Skellam is a discrete probability distribution. This means the client updates must first be appropriately discretized, e.g., via uniform quantization (which involves scaling the inputs by a factor $s \sim 2^b$ for some bit-width b followed by stochastic rounding for unbiased estimates).[3] Note that this allows us to re-parameterize the variance of the added Skellam noise as $s^2\mu$, giving the following simple corollary based on Cor. 3.6:

Corollary 4.1 (Scaled Skellam Mechanism)**.** *With a scaling factor $s \in \mathbb{R}$, the multi-dimensional Skellam Mechanism is (α, ε)-RDP with*

$$\varepsilon(\alpha) = \frac{\alpha\Delta_2^2}{2\mu} + \min\left(\frac{(2\alpha-1)\Delta_2^2}{4s^2\mu^2} + \frac{3\Delta_1}{2s^3\mu^2}, \frac{3\Delta_1}{2s\mu}\right) \tag{4.1}$$

As s increases, the RDP of scaled Skellam rapidly approaches that of Gaussian as the second term above approaches 0, suggesting that under practical regimes with moderate compression bit-width, Skellam should perform competitively compared to Gaussian. Another aspect worth noting is that rounding vector coordinates from reals to integers can inflate the ℓ_2-sensitivity Δ_2, and thus more noise is required for the same privacy. To this end, we leverage the *conditional* rounding procedure introduced in [19] to obtain a bounded norm on the scaled and rounded client vector:

Proposition 4.2 (Norm of stochastically rounded vector)**.** *Let $\tilde{x}$ be a stochastic rounding of vector $x \in \mathbb{R}^d$ to the integer grid. Then, for $\beta \in (0, 1)$, we have*

$$\mathbb{P}\left[\|\tilde{x}\|_2^2 \leq \|x\|_2^2 + d/4 + \sqrt{2\log(1/\beta)} \cdot \left(\|x\|_2 + \sqrt{d}/2\right)\right] \geq 1 - \beta. \tag{4.2}$$

The conditional rounding procedure is thus defined as retrying the stochastic rounding on x_i until $\|\tilde{x}_i\|_2^2$ is within the probabilistic bound above. We can then add Skellam noise to the aggregate $\sum_i \tilde{x}_i$ according to the inflated sensitivity $\tilde{\Delta}_2$, before undoing the quantization (unscaling). Note that a larger scaling s before rounding reduces the norm inflation and the extra noise needed (Fig 3 right).

Skellam for distributed DP with secure aggregation A stronger notion of privacy in FL can be obtained via the distributed DP model [19] that leverages secure aggregation (SecAgg [9]). The fact that the Skellam distribution is closed under summation allows us to easily extend from central DP to distributed DP. Under this model, the client vectors are quantized as in central DP model, but the Skellam noise is now added locally with variance μ/n. Then, the noisy client updates are summed via SecAgg (b bits per coordinate for field size 2^b) which only reveals the noisy aggregate to the server. While the local noise might be insufficient for local DP guarantees, the aggregated noise at the server provides privacy and utility comparable to the central DP model, thus removing trust away from the central aggregator. Note that the modulo operations introduced by SecAgg does not impact privacy as it can be viewed as a post-processing of an already differentially private query.

We remark on several properties of the distributed Skellam compared to the recently proposed distributed discrete Gaussian (DDG [19]): (1) DDG is not closed under summation, and the divergence

[3] Example of stochastic rounding: 42.3 has 0.7 and 0.3 probability to be rounded to 42 and 43, respectively. Other discretization schemes are possible; we do not explore this direction further in this work.

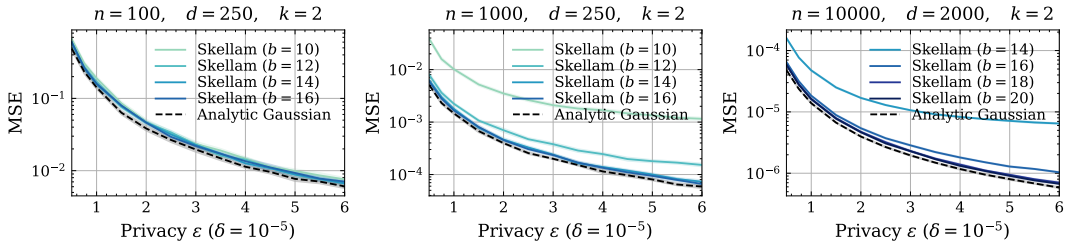

Figure 4: Distributed mean estimation with the distributed Skellam mechanism.

between discrete Gaussians can lead to notable privacy degradation in settings such as federated analytics with large n and small local noises σ (see also left of Fig. 2 and Fig. 3). While scaling mitigates this issue, it requires additional bit-width and makes Skellam attractive under tight communication constraints; (2) Sampling from Skellam only requires sampling from Poisson, for which efficient implementations are widely available. This makes distributed Skellam easily deployable even to mobile and edge devices; (3) In practice where $s \gg 1$ (dictated by bit-width b), both Skellam (cf. Cor. 4.1) and DDG (with an exponentially small divergence) quickly approaches Gaussian under RDP, and any differences will be negligible (Fig. 3).

5 Empirical Evaluation

In this section, we empirically evaluate the Skellam mechanism on two sets of experiments: distributed mean estimation and federated learning. In both cases, we focus on the distributed DP model, but note that the Skellam mechanism can be easily adapted to the central DP setting as discussed in the earlier section. Unless otherwise stated, we use RDP accounting for all experiments due to the high-dimensional data and the ease of composition (Section 3).

Note that we also introduce a random orthogonal transformation [23, 2, 19] before discretizing the client vectors (which can be reverted after the aggregation); this makes the vector coordinates sub-Gaussian and helps spread the magnitudes of the vector coordinates across all dimensions, thus reducing the rounding error and potential wrap-around from SecAgg modulo operations. Moreover, by approximating the sub-exponential tail of the Skellam distribution as sub-Gaussian, we can derive a heuristic for choosing s based on a bound on the aggregated signal variance $\tilde{\sigma}^2 \leq c^2 n^2/d + n/(4s^2) + \mu$ [19]. We choose s such that $2k\tilde{\sigma}$ are bounded within the SecAgg field size 2^b, and we fix $k = 3$ unless otherwise stated. To obtain Δ_1 for Skellam RDP, we note that $\Delta_1 \leq \Delta_2 \cdot \min(\sqrt{d}, \Delta_2)$ since $\Delta_1 \leq \Delta_2^2$ for integers (after rounding) and $\Delta_1 \leq \sqrt{d}\Delta_2$ otherwise.

Algorithm 1 (in the Appendix) summarizes the aggregation procedure for the distributed Skellam mechanism, as well as the parameters used for the experiments. In summary, we have an ℓ_2 clip norm $c > 0$; bit-width b; target central noise variance $\mu > 0$; number of clients n; signal bound multiplier $k > 0$; and rounding bias $\beta \in [0, 1)$. We fix $\beta = \exp(-1/2)$ for all experiments. Note that the per-coordinate bit-width b is for the aggregate values as it determines the field size of SecAgg. For federated learning, we also consider the number of rounds T and the total number of clients N (thus the uniform sampling ratio $q = n/N$ at every round). Our experiments are implemented in Python, TensorFlow Privacy [25], and TensorFlow Federated [18], and code will be open-sourced soon. See also Appendix for additional results and more details on the experimental setup.

5.1 Distributed Mean Estimation (DME)

We first consider DME as the generalization of (single round) FL. We randomly generate n client vectors $X = \{x_i\}_{i=1}^n$ from the d-dimensional L2 sphere with radius $c = 10$, and compute the true mean $\widehat{x} = \frac{1}{n}\sum_i^n x_i$. We then apply the aggregation procedure described in Algorithm 1 (in the Appendix) to compute a private mean estimate $\bar{x}$. For a strong baseline, we use the analytic Gaussian mechanism [6] with tight accounting (see also Figure 2). We plot the MSE as $\|\widehat{x} - \bar{x}\|_2^2/d$ with 95% confidence interval over 10 dataset initializations across different values of b, d, and n on Figure 4. Results demonstrate that Skellam can match Gaussian even with $n = 10000$ clients as long as the bit-width is sufficient.

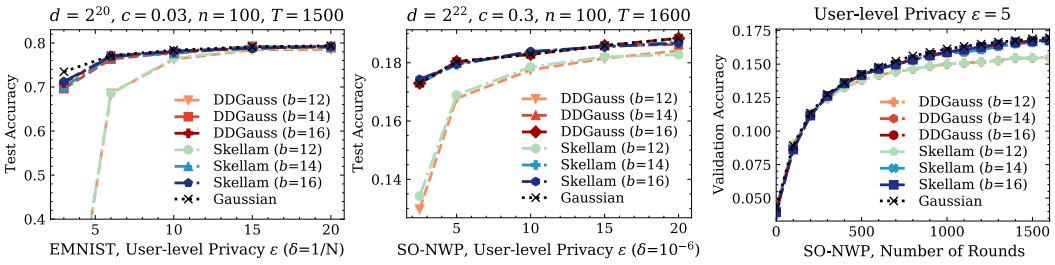

Figure 5: Federated learning with the (distributed) Skellam mechanism. DDGauss: Distributed Discrete Gaussian [19]. **Left / Middle**: Test accuracy on EMNIST / Stack Overflow NWP across different ε and b. **Right**: Validation accuracy on SO-NWP with $\varepsilon = 5$.

5.2 Federated Learning

Setup We evaluate on two public federated datasets with real-world characteristics: Federated EMNIST character recognition [13], and Stack Overflow next word prediction (SO-NWP [5]). EMNIST is an image classification dataset that consists of 671,585 training images grouped into $N = 3400$ clients, and models are evaluated based on classification accuracy. SO-NWP is a text dataset that consists of 135,818,730 training sentences grouped into $N = 342477$ clients, and the goal is to predict the next word given the preceding words in a sentence. In particular, we emphasize that both datasets have natural client heterogeneity that are representative of real-world FL problems: the partition in EMNIST is based on the writer of the handwritten digits, and sentences in SO-NWP are grouped by the corresponding Stack Overflow user. We train a small CNN with model size $\bar{d} < 2^{20}$ for EMNIST and use the LSTM model architecture defined in [31] for SO-NWP. See also Appendix and [31] for more details on models and datasets. The hyperparamters for our experiments mainly follow settings of [19] and [21], and we limit our hyperparameter tuning. For EMNIST, we fix $c = 0.03$, $n = 100$, $T = 1500$, client learning rate $\eta_{\text{client}} = 10^{-1.5}$, server learning rate $\eta_{\text{server}} = 1$, and client batch size $m = 20$. For SO-NWP, we fix $c = 0.3$, $n = 100$, $T = 1600$, $\eta_{\text{client}} = 0.5$, and $m = 16$; following [21], we sweep $\eta_{\text{server}} \in \{0.3, 1\}$ and limit max examples per client to 256. For both datasets, clients train for 1 epoch on their local datasets, and the client updates are weighted uniformly (as opposed to weighting by number of examples).

Results Figure 5 summarizes the FL experiments. For EMNIST, we report the average test accuracy across the last 100 rounds. For SO-NWP, we report the top-1 accuracy (without padding, out-of-vocab, or begining/end-of-sentence tokens) on the test set as well as on the validation set (averaged very 100 rounds) for $\varepsilon = 5$. The results indicate that Skellam performs as good as DDG and Gaussian despite relying on generic RDP amplification via sampling [38] (cf. Fig. 3). This bears significant practical relevance given the advantages of Skellam over DDG in real-world deployments.

6 Discussion & Conclusion

We have introduced the multi-dimensional Skellam mechanism for federated learning. We analyzed the Skellam mechanism through the lens of approximate DP, privacy loss distributions, and Rényi divergences, and derived a sharp RDP bound that enables Skellam to match Gaussian and discrete Gaussian in practical settings as demonstrated by our large-scale experiments. Since Skellam is closed under summation, it easily extends from the central DP model to distributed DP where it represents an attractive alternative to Discrete Gaussian due to the ease of sampling. Some interesting future work includes: (1) our scalar PLD analysis for Skellam suggests room for improvements on our multi-dimensional analysis particularly via a complete PLD characterization, and (2) our results on FL may be further improved via a targeted analysis for RDP amplification via sampling akin to [30]. Overall, this work is situated within the active area of private machine learning and aims at making ML more trustworthy. One potential negative impact is that our method could be (deliberately or inadvertently) misused – such as sampling the wrong noise or using a minuscule scaling factor – to provide non-existent privacy guarantees for real users' data. We nevertheless believe our results have positive impact as they facilitate the deployment of differential privacy in practice.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpSGD: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[3] Donald E Amos. Computation of modified bessel functions and their ratios. *Mathematics of Computation*, 28(125):239–251, 1974.

[4] 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)*, pages 920–925, 2020.

[5] The TensorFlow Federated Authors. Tensorflow federated stack overflow dataset. 2019.

[6] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR, 2018.

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

[8] James Bell, K. A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. Cryptology ePrint Archive, Report 2020/704, 2020. `https://eprint.iacr.org/2020/704`.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[10] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[11] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[12] Clément Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In *NeurIPS*, 2020.

[13] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[14] Google Differential Privacy Team. Privacy loss distributions. `https://github.com/google/differential-privacy/blob/master/accounting/docs/Privacy_Loss_Distributions.pdf`. 2020.

[15] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[17] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[18] Alex Ingerman and Krzys Ostrowski. Introducing tensorflow federated. 2019.

[19] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*. PMLR, 2021.

[20] Peter Kairouz, McMahan, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[21] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. *arXiv preprint arXiv:2103.00039*, 2021.

[22] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[23] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[24] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 2560–2569. PMLR, 2020.

[25] H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

[26] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *ICLR*, 2018.

[27] Sebastian Meiser and Esfandiar Mohammadi. Tight on budget? tight bounds for r-fold approximate differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 247–264, 2018.

[28] Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 650–661, 2012.

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

[30] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[31] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[32] Diego Ruiz-Antolín and Javier Segura. A new type of sharp bounds for ratios of modified bessel functions. *Journal of Mathematical Analysis and Applications*, 443(2):1232–1246, 2016.

[33] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[34] VR Thiruvenkatachar and TS Nanjundiah. Inequalities concerning bessel functions and orthogonal polynomials. In *Proceedings of the Indian Academy of Sciences-Section A*, volume 33, page 373. Springer, 1951.

[35] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*, 2020.

[36] Filipp Valovich and Francesco Alda. Computational differential privacy from lattice-based cryptography. In *International Conference on Number-Theoretic Methods in Cryptology*, pages 121–141. Springer, 2017.

[37] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. *arXiv preprint arXiv:2104.07586*, 2021.

[38] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642. PMLR, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The claims made in the abstract and introduction are supported by both theory (Section 3) and experiments (Section 5).
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Confidence intervals were reported for DME experiments over different random seeds. Error bars were not reported for FL experiments due to computational costs. The EMNIST test accuracy and SO-NWP validation accuracy are however averaged across many rounds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.
   (b) Did you mention the license of the assets? [Yes] See Appendix. In summary, EMNIST uses the BSD 2-Clause License while SO-NWP uses the CC BY-SA 3.0 License.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets are publicly curated by TensorFlow.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets are publicly curated by TensorFlow.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] This work did not use crowdsourcing or conduct research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] This work did not use crowdsourcing or conduct research with human subjects.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] This work did not use crowdsourcing or conduct research with human subjects.