D^2P-FED: DIFFERENTIALLY PRIVATE FEDERATED LEARNING WITH EFFICIENT COMMUNICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose the **d**iscrete Gaussian based **d**ifferentially **p**rivate **fed**erated learning (D^2P-FED), a unified scheme to achieve both differential privacy (DP) and communication efficiency in federated learning (FL). In particular, compared with the only prior work taking care of both aspects, D^2P-FED provides stronger privacy guarantee, better composability and smaller communication cost. The key idea is to apply the discrete Gaussian noise to the private data transmission. We provide complete analysis of the privacy guarantee, communication cost and convergence rate of D^2P-FED. We evaluated D^2P-FED on INFIMNIST and CIFAR10. The results show that D^2P-FED outperforms the-state-of-the-art by 6.7% to 9.75% in terms of model accuracy while saving one third of the communication cost.

1 INTRODUCTION

Federated learning (FL) is a popular machine learning paradigm that allows a central server to train models over decentralized data sources. In federated learning, each client performs training locally on their data source and only updates the model change to the server, which then updates the global model based on the aggregated local updates. Since the data stays locally, FL can provide better privacy protection than traditional centralized learning. However, FL is facing two main challenges: (1) FL lacks a rigorous privacy guarantee (e.g., differential privacy (DP)) and indeed, it has been shown to be vulnerable to various inference attacks (Nasr et al., 2019; Pustozerova & Mayer); (2) FL incurs considerable communication costs. In many potential applications of FL such as mobile devices, these two challenges are present *simultaneously*.

However, privacy and communication-efficiency have mostly been studied independently in the past. As regards privacy, existing work has applied a gold-standard privacy notion – differential privacy (DP) – to FL, which ensures that the server could hardly determine the participation of each client by observing their updates (Geyer et al., 2017). To achieve DP, each client needs to inject noise to their local updates and as a side effect, the performance of the trained model would inevitably degrade. To improve model utility, secure multiparty computation (SMC) has been used in tandem with DP to reduce noise (Jayaraman et al., 2018; Truex et al., 2019). The key idea is to prevent the server from observing the individual updates, make only the aggregate accessible, and thus transform from local DP to central DP. However, SMC introduces extra communication overhead to each client. There has been extensive research on improving communication efficiency of FL while ignoring the privacy aspect (Tsitsiklis & Luo, 1987; Balcan et al., 2012; Zhang et al., 2013; Arjevani & Shamir, 2015; Chen et al., 2016). However, these communication reduction methods either have incompatible implementations with the existing DP mechanisms or would break the DP guarantees when combined with SMC.

The only existing work that tries to reconcile DP and communication efficiency in FL is cpSGD (Agarwal et al., 2018). The authors leveraged the Binomial mechanism, which adds Binomial noise into local updates to ensure differential privacy. The discrete nature of Binomial noise allows it to be transmitted efficiently. However, cpSGD faces several limitations when applied to real-world applications. Firstly, with Binomial noise, the output of a learning algorithm would have different supports when any client participates or withdraws from learning; as a result, Binomial noise can only guarantee approximate DP where the participation of the client can be completely exposed with nonzero probability. Also, there lacks a tight composition for DP with Binomial noise and the resulting privacy budget skyrockets in a multi-round FL protocol. Hence, the Binomial

mechanism cannot produce a useful model with a reasonable privacy budget on complex tasks. Last but not least, the Binomial mechanism involves several mutually constrained hyper-parameters and the privacy formula is extremely complicated, which makes hyper-parameter tuning a difficult task.

In this paper, we propose the **d**iscrete Gaussian based **d**ifferential **p**rivate **fed**erated learning (D^2P-Fed), an alternative technique to reduce communication costs while maintaining differential privacy in FL. Our key idea is to leverage the discrete Gaussian mechanism in FL, which adds discrete Gaussian noise into client updates. We show that the discrete Gaussian mechanism satisfies Rényi DP which provides better composability. We employ secure aggregation along with the discrete Gaussian mechanism to lower the noise and exhibit the privacy guarantee for this hybrid privacy protection approach. To save the communication cost, we integrate the stochastic quantization and random rotation into the protocol. We then cast FL as a general distributed mean estimation problem and provide the analysis of the utility for the overall protocol. Our theoretical analysis sheds light on the superiority of D^2P-Fed to cpSGD. Our experiments show that D^2P-Fed can lead to state-of-the-art performance in terms of managing the trade-off among privacy, utility, and communication.

2 Related Work

It is well studied how to improve the communication cost in traditional distributed learning settings (Tsitsiklis & Luo (1987); Balcan et al. (2012); Zhang et al. (2013); Arjevani & Shamir (2015); Chen et al. (2016)). However, most of the approaches either require communication between the workers or are designed for specific learning tasks so they cannot be applied directly to general-purpose FL. The most relevant work is Suresh et al. (2017) which proposed to use stochastic quantization to save the communication cost and random rotation to lower mean squared error of the estimated mean. We follow their approach to improve the communication efficiency and model utility of D^2P-Fed. Nevertheless, our work differs from theirs in that we also study how to ensure DP for rotated and quantized data transmission and prove a convergence result for the learning algorithm with both communication cost reduction and privacy protection steps in place.

On the other hand, differentially private FL is undergoing rapid development during the past few years (Geyer et al. (2017); McMahan et al. (2017); Jayaraman et al. (2018)). However, these methods mainly focus on improving utility under a small privacy budget and ignore the issue of communication cost. In particular, we adopt a similar hybrid approach to Truex et al. (2019), which combines SMC with DP for reducing the noise. SMC ensures that the centralized server can only see the aggregated update but not individual ones from clients and as a result, the noise added by each client can be reduced by a factor of the square root of the number of clients participating in one round. The difference of our work from theirs is that we inject discrete Gaussian noise to local updates instead of the continuous Gaussian noise. This allows us to use secure aggregation (Bonawitz et al., 2017) which is much cheaper than threshold homomorphic encryption used by Truex et al. (2019). We further study the interaction between discrete Gaussian noise and the secure aggregation as well as their effects on the learning convergence.

We identify cpSGD (Agarwal et al. (2018)) as the most comparable work to D^2P-Fed. Just like D^2P-Fed, cpSGD aims to improve both the communication cost and the utility under rigorous privacy guarantee. However, cpSGD suffers from three main defects discussed in Section 1. This paper proposes to use the discrete Gaussian mechanism to mitigate these issues in cpSGD.

3 Background and Notation

In this section, we provide an overview of FL and DP and establish the notation system. We use bold lower-case letters (*e.g.* **a**,**b**,**c**) to denote vectors, and bold upper-case letters (*e.g.* **A**, **B**, **C**) for matrices. We denote $1 \cdots n$ by $[n]$.

FL Overview. In a FL system, there are one server and n clients $\mathcal{C}_i, i \in [n]$. The server holds a global model of dimension m. Each client holds (IID or non-IID) samples drawn from some unknown distribution $\mathcal{D}$. The goal is to learn the global model $\boldsymbol{w} \in \mathbb{R}^d$ that minimizes some loss function $\mathcal{L}(\boldsymbol{w}, \mathcal{D})$. To achieve this, the system runs a T-round FL protocol. The server initializes the global model with $\boldsymbol{w}_0$. In round $t \in [T]$, the server randomly sub-samples γn clients from $[n]$ with sub-sampling rate γ and broadcasts the global model $\boldsymbol{w}_{t-1}$ to the chosen clients. Each chosen client $\mathcal{C}_i$ then runs the local optimizers (*e.g.* SGD, Adam, and RMSprop), computes the difference

between the locally optimized model $\boldsymbol{w}_t^{(i)}$ and the global model $\boldsymbol{w}_{t-1}$: $\boldsymbol{g}_t^{(i)} = \boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{t-1}$, and uploads $\boldsymbol{g}_t^{(i)}$ to the server. The server takes the average of the differences and update the global model $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{1}{k}\sum \boldsymbol{g}_t^{(i)}$.

Communication in FL. The clients in FL are often edge devices, where the upload bandwidth is fairly limited; therefore, communication efficiency is of uttermost importance to FL. Let π denote a communication protocol. We denote the per-round communication cost as $\mathcal{C}(\pi, \boldsymbol{g}^{[n]})$. To lower the communication cost, the difference vectors are typically compressed before sent to the server. The compression would degrade model performance and we measure the performance loss via the mean squared error. Specifically, letting $\bar{\boldsymbol{g}}$ denote the actual mean of difference vectors $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}^{(i)}$ and $\tilde{\boldsymbol{g}}$ denote the server's estimated mean of difference vectors using some protocol such as D^2P-FED, we could measure the performance loss by $\mathcal{E}(\pi, \boldsymbol{\delta}^{[n]}) = \mathbb{E}[\|\tilde{\boldsymbol{g}} - \bar{\boldsymbol{g}}\|^2]$, i.e., the mean squared error between the estimated and the actual mean. This mean squared error is directly related to the convergence rate of FL (Agarwal et al., 2018).

Threat Model & Differential Privacy. We assume that the server is honest-but-curious. Namely, the server will follow the protocol honestly under the law enforcement or reputation pressure, but is curious to learn the client-side data from the legitimate client-side messages. In the FL context, the server wants to get information about the client-side data by studying the local updates received without deviating from the protocol.

The above attack, widely known as the inference attack (Shokri et al., 2017; Yeom et al., 2018; Nasr et al., 2019), can be effectively mitigated using a canonical privacy notation namely differential privacy (DP). Intuitively, DP, in the context of ML, ensures that the trained model is nearly the same regardless of the participation of any arbitrary client.

Definition 1 ((ϵ, δ)-DP)**.** *A randomized algorithm $f : \mathcal{D} \to \mathcal{R}$ is (ϵ, δ)-differentially private if for every pair of neighboring datasets D and D' that differs only by one datapoint, and every possible (measurable) output set E the following inequality holds: $P[f(D) \subseteq E] \leq e^{\epsilon} P[f(D') \subseteq E] + \delta$.*

(ϵ, δ)-DP has been used as a privacy notion in most of the existing works of privacy-preserving FL. However, in this paper, we consider a generalization of DP, Rényi differential privacy (RDP), which is strictly stronger than (ϵ, δ)-DP and allows tighter analysis for compositing multiple mechanisms. This second point is particularly appealing, as FL mostly comprises multiple rounds yet the existing works suffer from skyrocketing privacy budgets for multi-round learning.

Definition 2 ((α, ϵ)-RDP)**.** *For two probability distributions P and Q with the same support, the Rényi divergence of order $\alpha > 1$ is defined by $D_\alpha(P\|Q) \triangleq \frac{1}{\alpha-1}\log \mathbb{E}_{x\sim Q}(\frac{P(x)}{Q(x)})^\alpha$. A randomized mechanism $f : \mathcal{D} \to \mathcal{R}$ is (α, ϵ)-RDP, if for any neighboring datasets $D, D' \in \mathcal{D}$ it holds that $D_\alpha(f(D)\|f(D')) \leq \epsilon$.*

The intuition behind RDP is the same as other variants of differential privacy: "Similar inputs should yield similar output distributions," and the similarity is measured by the Rényi divergence under RDP. RDP can also be converted to (ϵ, δ)-DP using the following transformation.

Lemma 1 (RDP-DP conversion (Mironov (2017)))**.** *If $\mathcal{M}$ obeys (α, ϵ)-RDP, then $\mathcal{M}$ obeys $(\epsilon + \log(1/\delta)/(\alpha - 1), \delta)$-DP for all $0 < \delta < 1$.*

RDP enjoys an operationally convenient and quantitatively accurate way of tracking cumulative privacy loss when compositing multiple mechanisms (Lemma 2) or being combined with subsampling (Wang et al., 2018). As a result, RDP is particularly suitable for the context of ML.

Lemma 2 (Adaptive composition of RDP (Mironov (2017)))**.** *If (randomized) mechanism $\mathcal{M}_1$ obeys (α, ϵ_1)-RDP, and $\mathcal{M}_2$ obeys (α, ϵ_2)-RDP, then their composition obeys $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

4 Discrete Gaussian Mechanism

In this section, we present the discrete Gaussian mechanism and establish its privacy guarantee. We first introduce discrete Gaussian distribution.

Definition 3 (Discrete Gaussian Distribution)**.** *Discrete Gaussian is a probability distribution on a discrete additive subgroup $\mathbb{L}$ parameterized by σ. For a discrete Gaussian distribution $N_{\mathbb{L}}(\sigma)$ and $x \in \mathbb{L}$, the probability mass on x is proportional to $e^{-\pi x^2/(2\pi\sigma^2)}$.*

Discrete Gaussian mechanism works by adding noise drawn from discrete Gaussian distribution. Canonne et al. (2020) proved concentrated DP for the discrete Gaussian mechanism. However, there lacks tight privacy amplification and composition theorem for concentrated DP. To address, we turn to RDP and provide the first RDP analysis for the discrete Gaussian mechanism. The proof is delayed to Appendix A due to space limitation.

Theorem 1 (RDP for discrete Gaussian mechanism)**.** *If f has sensitivity 1, then the discrete Gaussian mechanism: $f(\cdot) + N_{\mathbb{L}}(\sigma)$ satisfies $(\alpha, \alpha/(2\sigma^2))$-RDP.*

Under RDP, discrete Gaussian exhibits tight privacy amplification under sub-sampling (Wang et al., 2018). This suits FL well since a subset of clients is sub-sampled to upload the updates in each round.

Theorem 2 (Privacy amplification for discrete Gaussian mechanism)**.** *If a discrete Gaussian mechanism is $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP, then augmented with subsampling (without replacement), the privacy guarantee is amplified to (1) $(\alpha, \mathcal{O}(\frac{\alpha\gamma^2}{\sigma^2}))$ in the high privacy regime; or (2) $(\alpha, \mathcal{O}(\alpha\gamma^2 e^{\frac{1}{\sigma^2}}))$ in the low privacy regime.*

Besides, RDP enables discrete Gaussian mechanism to be composed tightly with analytical moments accountant (Wang et al., 2018), which saves a huge amount of privacy budget in a multi-round FL.

Theorem 3 (Composition for discrete Gaussian mechanism)**.** *If a discrete Gaussian mechanism is $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP, then the sequential composition of T such mechanisms yield $(\alpha, \frac{T\alpha}{2\sigma^2})$-RDP guarantee. If we convert all the RDP guarantees back to (ϵ, δ)-DP, the growth of ϵ under the same δ is asymptotically $\mathcal{O}(\sqrt{T})$.*

Note that both privacy amplification and composition are given in an asymptotic form for the clarity of presentation. For tight bound, we refer the readers to Theorem 27 and Section 3.3 in Wang et al. (2018).

5 D^{2}P-FED: ALGORITHM AND PRIVACY ANALYSIS

In this section, we formally present D^{2}P-FED and provide rigorous privacy analysis.

5.1 ALGORITHM

Algorithm 1 provides the pseudocode for D^{2}P-FED. It follows the general FL pipeline which iteratively performs the following steps: (1) the server broadcasts the global model to a subset of clients; (2) the selected clients train the global model on their local data and upload the resulting model difference; and (3) the server aggregates the model differences uploaded by the clients and updates the global model. Grounded on the general FL pipeline, D^{2}P-FED introduces some additional steps at the client side as follows to improve communication efficiency and privacy.

Stochastic Quantization & Random Rotation (line 11-14). To lower the communication cost, the clients stochastically quantize the values in the update vectors to some discrete domain. Compared with real number encoding which costs 32 or 64 bits per dimension, the quantized value only requires $\log_2(k)$ bits per dimension. On the other hand, quantization lowers the fidelity of the update vector and thus leads to some error in estimating the mean of gradients. To lower the estimation error, the clients apply randomized rotation to the updates before quantization as proposed by McMahan et al. (2016). The details are discussed in Section 6.1.

Discrete Gaussian Mechanism (line 15). We apply the discrete Gaussian mechanism to ensure DP. To determine the noise magnitude to be added, we need to bound the ℓ_2-sensitivity of the gradient aggregate. Without the quantization and random rotation, one could clip the individual gradient update and consequently the ℓ_2-sensitivity is just the clipping threshold. However, the inclusion of compression steps makes the analysis of ℓ_2-sensitivity more sophisticated. We provide the analysis of sensitivity and RDP guarantees for the entire algorithm in Section 5.2. Each client samples the noise from the discrete Gaussian distribution. In contrast to prior work where each client adds independent noise, we require the clients to share the same random seed, generate the same noise and add an average share of the noise in each round. This is because the sum of multiple independent

discrete Gaussians is no longer a discrete Gaussian. Note that the same random seed is also required for communication reduction in secure aggregation so we can conveniently reuse it here without introducing more further overhead. The noise magnitude is set to ensure that the aggregate noise from all clients provides the global DP.

Algorithm 1: D^{2}P-FED Protocol.

Input: Support Lattice: $\mathbb{L} = \frac{2\boldsymbol{g}^{\max}}{k-1} \cdot \mathbb{Z}$, Noise Scale: σ, Rotation Matrix: $\boldsymbol{R}$, Random Seed: s

$\phi_q(x) = \frac{2\boldsymbol{g}^{\max}}{k-1}((\frac{k-1}{2\boldsymbol{g}^{\max}}x + \frac{q-1}{2}) \mod q - \frac{q-1}{2})$, q is odd.

1 **for** $t \leftarrow [T]$ **do**
2   **Server:**
3   Sub-sample a subset of clients $S \subset [n], |S| = \gamma n$ and broadcast $\boldsymbol{w}_{t-1}$ and $\boldsymbol{g}^{\max}$ to S
4   **Client:**
5   **foreach** *client* $i \in S$ **do** Send $\boldsymbol{u}_{ij}$ to j, $\boldsymbol{u}_{ij} \sim Unif(\mathbb{L}_q^m), \mathbb{L}_q := \{x \in \mathbb{L}, |x| \leq \frac{q-1}{2}\}$
6   **Client:**
7   **foreach** *client* $i \in S$ **do**
8   Train the model $\boldsymbol{w}_t^{(i)}$ with $\boldsymbol{w}_t$ as initialization
9       $\boldsymbol{g}_t^{(i)} = \mathbf{w}_t^{(i)} - \mathbf{w}_{t-1}$ /* **Compute the difference** */
10       $\boldsymbol{g}_{t,clipped}^{(i)} = \boldsymbol{g}_t^{(i)} / \max(1, \frac{\|\boldsymbol{g}_t^{(i)}\|_2}{D})$ /* **Clip the difference** */
11       $\boldsymbol{g}_{t,rotated}^{(i)} = \boldsymbol{R} \times \boldsymbol{g}_{t,clipped}^{(i)}$ /* **Random Rotation** */
12   let $\boldsymbol{b}[r] := -\boldsymbol{g}^{\max} + \frac{2r\boldsymbol{g}^{\max}}{k-1}$ for every $r \in [0, k)$ /* **Quantize** */
13   **for** $j \in d, \boldsymbol{b}[r] \leq \tilde{\boldsymbol{g}}_{t,quantized}^{(i)}[j] \leq \boldsymbol{b}[r+1]$ **do**
14         $\tilde{\boldsymbol{g}}_{t,quantized}^{(i)}[j] = \begin{cases} \boldsymbol{b}[r] & \text{w.p. } \dfrac{\boldsymbol{g}_{t,rotated}^{(i)}[j] - \boldsymbol{b}[r]}{\boldsymbol{b}[r+1] - \boldsymbol{b}[r]} \\ \boldsymbol{b}[r+1] & \text{o.w.} \end{cases}$
15       $\tilde{\boldsymbol{g}}_{t,dp}^{(i)} = \tilde{\boldsymbol{g}}_{t,quantized}^{(i)} + \frac{\boldsymbol{\nu}_i}{\gamma n}, \boldsymbol{\nu}_i \overset{s}{\sim} N_{\mathbb{L}}^m(\sigma)$ /* **Discrete Gaussian** */
16       $\tilde{\boldsymbol{g}}_t^{(i)} = \phi_q(\phi_q(\tilde{\boldsymbol{g}}_{t,dp}^{(i)}) + \sum_{j \neq i, j \in S} \boldsymbol{u}_{ij} - \sum_{j \neq i, j \in S} \boldsymbol{u}_{ji})$ /* **Mask** */
17   Send $\tilde{\boldsymbol{g}}_t^{(i)}$ to the server
18   **Server:**
19     $\tilde{\boldsymbol{g}}_t = \frac{1}{k} \sum_{i \in S} \tilde{\boldsymbol{g}}_t^{(i)}$ /* **Aggregate** */
20     $\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \tilde{\boldsymbol{g}}_t$

Secure aggregation. (line 5,16,19) To reduce the noise magnitude for ensuring DP, we hide clients' individual updates from the central server and only allow it to see the aggregated updates via the technique of secure aggregation. If the individual updates were available to the central server, they should also be protected with the same privacy guarantee as the averaged update; in this case, the required noise scales up with $\mathcal{O}(\gamma n)$. On the other hand, if the central server can only access the aggregated updates, then the required noise is $\mathcal{O}(1)$. Hence, secure aggregation of local updates can lead to significant noise reduction. However, there exists a challenge for integrating secure aggregation with discrete Gaussian mechanism: discrete Gaussian variables have infinite support, thereby incompatible with secure aggregation which operates on finite field. In Section 5.2, we propose to address this challenge by mapping the noised vector to a cyclic additive group and then applying secure aggregation and show that the RDP guarantees are preserved under the mapping.

5.2 PRIVACY ANALYSIS

Before applying discrete Gaussian mechanism to D^{2}P-FED, we need to figure out how to calibrate the added noise. In differential privacy, the calibration is guided by *sensitivity* of the function as defined below.

Definition 4 (ℓ_2-sensitivity)**.** *Given a function $f : \mathcal{D} \to \mathbb{R}$ and two neighboring datasets D and D', the ℓ_2-sensitivity of f: Δ_f is defined as $\Delta_f = \max_{D,D'} \|f(D) - f(D')\|_2$.*

In DP for deep learning, the traditional way to bound ℓ_2-sensitivity is to clip the update vector. However, quantization will further influence the sensitivity after clipping. We provide the sensitivity after quantization as below. The proof is delayed to Appendix B due to space limitation.

Theorem 4. *If we clip the l_2 norm of $\boldsymbol{g}$ to D, and quantize it to $k = \sqrt{d} + 1$ levels, then the ℓ_2 sensitivity of the difference is $4D$.*

Given Theorem 1 and Theorem 4, we provide the RDP bound for D^{2}P-FED.

Theorem 5 (RDP for D^{2}P-FED)**.** *Given the clipping bound D, the noise scale σ,* D^{2}P-FED *follows* $(\alpha, \frac{8\alpha D^2}{\sigma^2})$*-RDP.*

Theorem 5 is built on the assumption that the centralized server only has access to the summed updates but not the individual ones. If the centralized server has access to individual updates, the noise has to scale up γn times to maintain the same privacy guarantee which severely hinders the model accuracy. To consolidate the assumption, we leverage a cryptographic technique, secure aggregation (Bonawitz et al., 2017) which guarantees that the centralized server can only see the aggregated result. The basic intuition is to mask the inputs with random values canceling out in pairs. However, since discrete Gaussian has infinite support, we cannot directly apply random masks to it. To reconcile secure aggregation with discrete Gaussian, we propose to project the involved values into a quotient group after shifting and then apply the random masks as shown in line 16 in Algorithm 1. According to the post-processing theorem of RDP (Mironov (2017)), the result still follows rigorous Rényi differential privacy as proved in Appendix C. Note that we consider a simplified version of the full secure aggregation protocol (Bonawitz et al. (2017)) in Algorithm 1 and omit many interesting details such as the generation of the random masks and how to deal with dropout. We deem this to be enough to clarify the idea behind the reconciliation. The complete version of secure aggregation can be reconciled using exactly the same trick.

Theorem 6 (Informal)**.** *Distributed discrete Gaussian mechanism with secure aggregation obeys the same RDP guarantee as vanilla global discrete Gaussian mechanism with the same parameters.*

6 COMMUNICATION PROTOCOL & UTILITY ANALYSIS

In this section, we present our communication protocol in detail and discuss the communication cost and the estimation error of D^{2}P-FED with direct comparison to cpSGD. The drastic improvement of D^{2}P-FED mainly comes from the tight composition of discrete Gaussian mechanism compared to binomial mechanism in cpSGD.

6.1 COMMUNICATION PROTOCOL

As the first step, we leverage stochastic k-level quantization proposed by McMahan et al. (2016) to lower the communication cost as described in line 13-14 in Algorithm 1. If we denote vanilla stochastic k-level quantization with π_k, then we successfully reduce the per-round communication cost $\mathcal{C}(\pi_k, \boldsymbol{g}^{[n]}) = n \cdot (d\lceil \log_2 k \rceil + \tilde{\mathcal{O}}(1))$.

However, stochastic quantization sacrifices some accuracy for communication efficiency. Concretely, $\mathcal{E}(\pi_k, \boldsymbol{g}^{[n]}) = \mathcal{O}(\frac{d}{n} \cdot \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{g}^{(i)}\|_2^2)$. Since d is tens of thousands to hundreds of thousands in federated learning, the estimation error of the mean is too large. Thus to reduce the estimation error, we randomly rotate the difference vector (McMahan et al., 2016) as the second step. The key intuition is that the MSE of stochastic uniform quantization is $\mathcal{O}(\frac{d}{n}(\boldsymbol{g}^{\max})^2)$. With random rotation, we can limit $\boldsymbol{g}^{\max}$ to $\sqrt{\frac{\log d}{d}}$ w.h.p. so the MSE will be improved to $\mathcal{O}(\frac{\log d}{n})$. Agarwal et al. (2018) also leverages random rotation to reduce MSE. However, in their setting, random rotation intrinsically harms their privacy guarantee because the ℓ_∞-sensitivity might increase with rotation. A natural advantage of discrete Gaussian is that the privacy guarantee only depends on ℓ_2-sensitivity which is an invariant under rotation. Thus, random rotation does not harm our privacy guarantee at all. We omit the details here and refer the interested readers to McMahan et al. (2016). We denote the protocol using quantization and random rotation with $\pi_k^{(rot)}$. We know that $\mathcal{C}(\pi_k^{(rot)}, \boldsymbol{g}^{[n]})$ remains the same while the MSE error is reduced as $\mathcal{E}(\pi_k^{(rot)}, \boldsymbol{g}^{[n]}) = \mathcal{O}(\frac{\log d}{n} \cdot \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{g}^{(i)}\|_2^2)$.

6.2 CONVERGENCE RATE OF D^{2}P-FED

In this section, we relate the convergence rate with mean squared error using Corollary 1 and analyze the mean squared error of mean estimation in D^{2}P-FED. Note that we assume each client executes one iteration in each round so $\boldsymbol{g}$ equals to the gradient or belongs to the sub-gradients.

Corollary 1 (Ghadimi & Lan (2013))**.** *Let $F(\boldsymbol{w}) = \mathcal{L}(\boldsymbol{w}, \mathcal{D})$ for some given distribution $\mathcal{D}$. Let $F(\boldsymbol{w})$ be L-smooth and $\forall \boldsymbol{w}, \|\nabla F(\boldsymbol{w})\| \leq \rho$. Let $\boldsymbol{w}^0$ satisfy $F(\boldsymbol{w}^0) - F(\boldsymbol{w}^*) \leq \rho_F$. Then after T rounds*

$$\mathbb{E}_{t\sim(Unif(T))}[\|\nabla F(\boldsymbol{w}_t)\|_2^2] \leq \frac{2\rho_F L}{T} + \frac{2\sqrt{2}\lambda\sqrt{L\rho_F}}{\sqrt{T}} + \rho B$$

, where $\lambda^2 = \max_{1\leq t\leq T} 2\mathbb{E}[\|\boldsymbol{g}(\boldsymbol{w}_t) - \nabla F(\boldsymbol{w}_t)\|_2^2] + 2\max_{1\leq t\leq T} \mathbb{E}_q[\|\boldsymbol{g}(\boldsymbol{w}_t) - \tilde{\boldsymbol{g}}(\boldsymbol{w}_t)\|_2^2]$, and $B = \max_{1\leq t\leq T} \|\boldsymbol{g}(\boldsymbol{w}_t) - \tilde{\boldsymbol{g}}(\boldsymbol{w}_t)\|$.

As corollary 1 indicates, for a given gradient bound, the convergence rate approximately grows with the reciprocal of MSE. Thus, we analyze D^{2}P-FED's MSE and obtain the following theorem. The proof is delayed to Appendix D due to space limitation.

Theorem 7. *If we choose $\sigma \geq 1/\sqrt{2\pi}$, the mean squared error is*

$$\mathcal{E}(\pi^{(rot)}_{k,q,N_{\mathbb{L}}(\sigma^2)}, \boldsymbol{g}^{[n]}) \leq (1-\frac{1}{1+3e^{-2\pi^2\sigma^2}}(1-\Phi(nq)))\cdot\frac{4d(\boldsymbol{g}^{\max})^2}{n(k-1)^2}(\frac{1}{4}+\frac{\sigma^2}{\gamma^2 n^2})+(1-\Phi(n(q-k-1)))\cdot q^2$$

where Φ is the cumulative distribution function (CDF) of the standard normal distribution.

Remark. If we choose $\boldsymbol{g}^{\max}$ as $\mathcal{O}(D\sqrt{\frac{\log(d)}{d}})$ following cpSGD, then the MSE is bounded by $\mathcal{O}(\frac{\sigma^2\log(d)}{n(k-1)^2})$ which is on the same order of cpSGD. If some values in the updates exceed $\boldsymbol{g}^{\max}$ under the above choice, the values should be clipped after random rotation.

6.3 COMMUNICATION COST OF D^{2}P-FED

First, we provide the trivial per-round communication cost which is exactly the same as cpSGD.

Theorem 8. *The per-round communication cost of* D^{2}P-FED *is*

$$\mathcal{C}(\pi^{(rot)}_{k,q,N_{\mathbb{L}}(\sigma^2)}, \boldsymbol{g}^{[n]}) = n\cdot(d\log(nq+1) + \tilde{\mathcal{O}}(1))$$

.

Now let's compare the number of rounds in cpSGD and D^{2}P-FED qualitatively. Note that during the following discussion, we usually omit δ for the ease of clarification. For cpSGD, the known tightest bound is by the combination of standard privacy amplification (Balle et al., 2018) and advanced composition (Dwork et al., 2010). Concretely, if a mechanism costs ϵ privacy budget, then after composed with sub-sampling the privacy budget is reduced to $\mathcal{O}(\gamma\epsilon)$ where γ is the sub-sampling rate. If the mechanism is composed sequentially T times, the privacy budget grows to $\mathcal{O}(\sqrt{T\log(1/\delta)}\epsilon)$. Thus, the total privacy budget of cpSGD is $\mathcal{O}(\gamma\sqrt{T\log(1/\delta)}\epsilon)$. On the other hand, D^{2}P-FED provides a total privacy budget of $\mathcal{O}(\gamma\sqrt{T}\epsilon)$, saving a factor of $\sqrt{\log(1/\delta)}$. Since δ is typically very small, the saving is quite significant. If the privacy budgets for the two protocols are the same, then D^{2}P-FED can use noise with $\mathcal{O}(\sqrt{\log(1/\delta)})$ smaller scale than cpSGD in each round. This will lead to a $\mathcal{O}(\sqrt{\log(1/\delta)})$-time faster convergence. Then for a given gradient bound, D^{2}P-FED can reach it with $\mathcal{O}(\log(1/\delta))$-time fewer rounds and thus save $\mathcal{O}(\log(1/\delta))$-time communication cost.

Remark. Both D^{2}P-FED and cpSGD intrinsically require secure aggregation to establish their privacy guarantee. Agarwal et al. (2018) did not discuss the issue explicitly. As pointed out in Bonawitz et al. (2017), once combined with secure aggregation, each field has to expand at least γn times (γn is the number of chosen clients) to prevent overflow of the sum.

7 EVALUATION

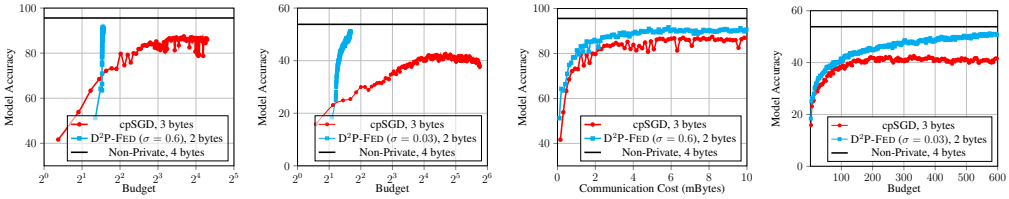
We would like to answer the following three questions using empirical evaluation: (1) How D^{2}P-FED performs in a multi-round federated learning compared to cpSGD under either the same privacy

guarantee or the same communication cost? (2) How different choices of hyper-parameters affect the performance of D^{2}P-FED? (3) Does D^{2}P-FED work under heterogeneous data distribution? Due to space limitation, we present our main results for (1) in this section and defer the results for (2) and (3) to Appendix E.

7.1 EXPERIMENT SETUP

To answer the above questions, we evaluated D^{2}P-FED and cpSGD on INFIMNIST (Bottou (2007)) and CIFAR10 (Krizhevsky et al. (2009)). We sampled 10M hand-written digits from INFIMNIST and randomly split the data among 100K clients. In each round, 100 clients are randomly chosen to upload their difference vectors to train a three-layer MLP. For CIFAR10, we select 10 out of 2000 clients in each round to train a 2-layer convolutional network. All RDP bounds are converted to (ϵ, δ)-DP for the ease of comparison and the total δ is set to $1e{-}5$ for all experiments.

7.2 MODEL ACCURACY VS. PRIVACY BUDGET



INFIMNIST CIFAR10 INFIMNIST CIFAR10

(a) Privacy Budget vs. Model Accuracy. (b) Communication Cost vs. Model Accuracy.

Figure 1: D^{2}P-FED vs. cpSGD.

To answer the first question, we studied model accuracy under the same privacy budget as shown in Figure 1a. Compared with cpSGD, D^{2}P-FED achieves 6.7% higher model accuracy on INFIMNIST and 9.75% higher model accuracy on CIFAR10 after convergence. As expected, D^{2}P-FED composes far tighter under sub-sampling as the lines are much sharper than those of cpSGD. Consequently, D^{2}P-FED converges at a smaller privacy budget than cpSGD as well. Although in Figure 1a cpSGD has better accuracy in the high privacy region, it is not necessarily the case but depends on the scale of the discrete Gaussian noise, as studied in Section E.1. Note that the results for cpSGD in Figure 1 is different from the results in Figure 2 in the original paper (Agarwal et al., 2018). The reason is that in the original paper, they do not sub-sample clients in each round but assign each client to exactly one round beforehand to avoid composition which cpSGD cannot handle well. However, the scheme is far from practical in the real world due to the dynamic nature of the clients.

7.3 MODEL ACCURACY VS. COMMUNICATION COST

To answer the second question, we also studied the model accuracy under the same communication cost. As shown in Figure 1b, D^{2}P-FED consistently achieves better model accuracy under the same communication cost on both INFIMNIST and CIFAR10. The main reason is that the tight composition property allows D^{2}P-FED to use smaller per-feature communication cost while still achieving better accuracy. As a concrete instance, D^{2}P-FED with 50% compression rate can achieve better accuracy than cpSGD with 25% compression rate. cpSGD with 50% compression rate either leads to an unacceptable privacy budget or does not converge.

8 CONCLUSION

In this work, we developed D^{2}P-FED to achieve both differential privacy and communication efficiency in the context of federated learning. By applying the discrete Gaussian mechanism to the private data transmission, D^{2}P-FED provides stronger privacy guarantee, better composability and smaller communication cost than the only prior work, cpSGD, both theoretically and empirically.

REFERENCES

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*, pp. 7564–7575, 2018.

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pp. 1756–1764, 2015.

Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pp. 26–1, 2012.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pp. 6277–6287, 2018.

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*, pp. 1175–1191, 2017.

Leon Bottou. The infinite mnist dataset. 2007.

Clément Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Jiecao Chen, He Sun, David Woodruff, and Qin Zhang. Communication-optimal distributed clustering. In *Advances in Neural Information Processing Systems*, pp. 3727–3735, 2016.

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

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

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

Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pp. 6343–6354, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

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

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739–753. IEEE, 2019.

Anastasia Pustozerova and Rudolf Mayer. Information leaks in federated learning.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International Conference on Machine Learning*, pp. 3329–3337, 2017.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1–11, 2019.

John N Tsitsiklis and Zhi-Quan Luo. Communication complexity of convex optimization. *Journal of Complexity*, 3(3):231–243, 1987.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled r\'enyi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*, 2018.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282. IEEE, 2018.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. Statistical model aggregation via parameter matching. In *Advances in Neural Information Processing Systems*, pp. 10956–10966, 2019a.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. *arXiv preprint arXiv:1905.12022*, 2019b.

Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pp. 2328–2336, 2013.

A Proof for Theorem 1

The proof is straightforward as shown below.

Proof.

$$
\begin{aligned}
&D_\alpha(N(0,\sigma^2)\|N(\mu,\sigma^2)) \\
&\quad= \frac{1}{\alpha-1}\log\sum_{\mathbb{L}}\frac{1}{\sum_{\mathbb{L}}\exp(-(x-\mu)^2/(2\sigma^2))}\exp(-\alpha x^2/(2\sigma^2)\cdot\exp(-(1-\alpha)(x-\mu)^2/(2\sigma^2)) \\
&\quad= \frac{1}{\alpha-1}\log\frac{1}{\sum_{\mathbb{L}}\exp(-(x-\mu)^2/(2\sigma^2))}\sum_{\mathbb{L}}\exp((-x^2+2(1-\alpha)\mu x-(1-\alpha)\mu^2)/(2\sigma^2)) \\
&\quad= \frac{1}{\alpha-1}\log\{\exp((\alpha^2-\alpha)\mu^2/(2\sigma^2))\} \\
&\quad= \alpha\mu^2/(2\sigma^2)
\end{aligned}
$$

□

B Proof for Theorem 4

Proof. The l_2 sensitivity of δ is naturally bounded by $2D$. The rotation does not change the sensitivity. The k-level quantization might expand the space as shown in Figure 2. An upper bound on the radius of the red circle is $D+\sqrt{d}\frac{D}{k-1}$. When we take $k=\sqrt{d}-1$, it reduces to $2D$. Thus, the upper bound on the sensitivity is $4D$. □

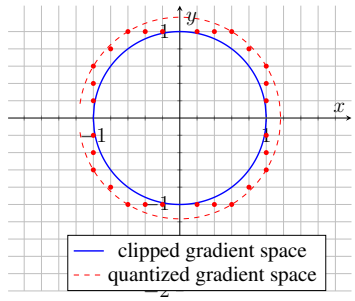


Figure 2: Sensitivity after Quantization.

C Proof for Theorem 6

We first prove the following lemma.

Lemma 3.

$$\sum \phi_q(x) = \phi_q(\sum x)$$

Proof.

$$\begin{aligned}\sum \phi_q(x) &= \sum \frac{2\boldsymbol{g}^{\max}}{k-1}((\frac{k-1}{2\boldsymbol{g}^{\max}}x + \frac{q-1}{2}) \mod q - \frac{q-1}{2}) \\ &= \frac{2\boldsymbol{g}^{\max}}{k-1}((\frac{k-1}{2\boldsymbol{g}^{\max}}\sum x + \frac{q-1}{2}) \mod q - \frac{q-1}{2}) \\ &= \phi_q(\sum x)\end{aligned}$$

□

Then we prove Theorem 6 as follows.

Proof. Given Lemma 3,

$$\begin{aligned}\tilde{\boldsymbol{g}}_t &= \frac{1}{k}\sum_{i\in S}\tilde{\boldsymbol{g}}_t^{(i)} \\ &= \frac{1}{k}\sum_{i\in S}\phi_q(\phi_q(\tilde{\boldsymbol{g}}_{t,dp}^{(i)}) + \sum_{j\neq i, j\in S}\boldsymbol{u}_{ij} - \sum_{j\neq i, j\in S}\boldsymbol{u}_{ji}) \\ &= \frac{1}{k}\phi_q(\sum_{i\in S}(\phi_q(\tilde{\boldsymbol{g}}_{t,dp}^{(i)}) + \sum_{j\neq i, j\in S}\boldsymbol{u}_{ij} - \sum_{j\neq i, j\in S}\boldsymbol{u}_{ji})) \\ &= \frac{1}{k}\phi_q(\sum_{i\in S}\tilde{\boldsymbol{g}}_{t,dp}^{(i)})\end{aligned}.$$

The expression $\sum_{i\in S}\tilde{\boldsymbol{g}}_{t,dp}^{(i)}$ forms a centralized discrete Gaussian mechanism and according to the post-processing theorem, the same RDP guarantee still holds. □

D Proof for Theorem 7

Proof Sketch. Before starting the proof, we introduce two lemmas for the proof.

Lemma 4 (Proposition 19 from Canonne et al. (2020))**.** *For all $\sigma \in \mathbb{R}$ with $\sigma > 0$,*

$$\mathbb{V}[N_{\mathbb{Z}}(0,\sigma^2)] \leq \sigma^2(1 - \frac{4\pi^2\sigma^2}{e^{4\pi^2\sigma^2-1}}) < \sigma^2$$

Moreover, if $\sigma^2 \leq \frac{1}{3}$, then $\mathbb{V}[N_{\mathbb{Z}}(0,\sigma^2)] \leq 3\cdot e^{-\frac{1}{2\sigma^2}}$

Lemma 5 (Proposition 23 from Canonne et al. (2020))**.** *For all $m \in \mathbb{Z}$ with $m \geq 1$ and all $\sigma \in \mathbb{R}$ with $\sigma > 0$,*

$$\mathbb{P}_{X\sim N_{\mathbb{Z}}(0,\sigma^2)}[X \geq m] \leq \mathbb{P}_{X\sim N(0,\sigma^2)}[X \geq m-1].$$

Moreover, if $\sigma \geq 1/\sqrt{2\pi}$, we have

$$\mathbb{P}_{X\sim N_{\mathbb{Z}}(0,\sigma^2)}[X \geq m] \geq \frac{1}{1+3e^{-2\pi^2\sigma^2}}\mathbb{P}_{X\sim N(0,\sigma^2)}[X \geq m]$$

Now we start our proof. MSE can be rewritten in the following format.

$$\mathbb{E}[\|\hat{\bar{X}} - \bar{X}\|_2^2|] = \frac{1}{n^2}\sum_{j=1}^{d}\sum_{i=1}^{n}\mathbb{E}[(\hat{\bar{X}}_i(j) - X_i(j))^2]$$

For each $\star = \mathbb{E}[(\hat{\bar{X}}_i(j) - X_i(j))^2]$, we need to consider two cases. If no overflow happens, due to Lemma 4

$$\star_{\neg o} \leq \mathbb{E}[(\frac{2X^{\max}}{k-1})^2(\mathbb{V}(Ber(p_i(j))) + \mathbb{V}(N_{\mathbb{L}}(\sigma)))] \leq \frac{4(X^{\max})^2}{(k-1)^2}(\frac{1}{4} + \frac{\sigma^2}{\gamma^2 n^2})$$

If overflow happens, trivially we have $\star_o \leq k^2$. Thus, we have

$$\star = \mathbb{P}[\neg o]\cdot\star_{\neg o} + \mathbb{P}[o]\cdot\star_o \leq \mathbb{P}_{X\sim N_{\mathbb{L}}(\sigma^2)}[X \leq q]\cdot\star_{\neg o} + \mathbb{P}_{X\sim N_{\mathbb{L}}(\sigma^2)}[X \geq q-k]\cdot\star_o \quad (1)$$

With Lemma 5, we can bound the probabilities and get the final MSE result in the theorem.

□

E Other Evaluation

E.1 Influence of Noise Scale

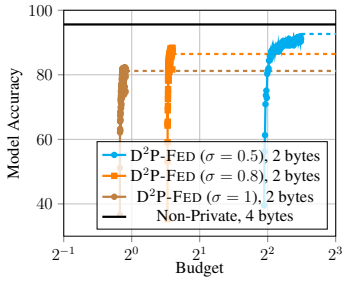


Figure 3: D^{2}P-Fed under different σ.

The hyper-parameter of the most vital interest in D^{2}P-Fed is the scale of the noise. To understand the effect of the noise scale, we evaluated D^{2}P-Fed on INFIMNIST, with 3 different choices of noise scale as shown in Figure 3. It is no surprise that the higher the noise scale, the smaller the privacy budget and the lower the model accuracy. This also illustrates the claim we have in Section 7.2 that D^{2}P-Fed can also have relatively good performance in the high privacy region at the cost of the model accuracy.

E.2 Influence of Heterogeneous Data Distribution

It is well known that data is sometimes heterogeneously distributed among clients in a federated learning system. Thus, to better understand D^{2}P-Fed's behavior under heterogeneous data distribution, we simulated heterogenenous data distribution by distributing the INFIMNIST data according to the classes and evaluated D^{2}P-Fed on these clients. The results are shown in Figure 4 and we can see that under heterogeneous data distribution the model accuracy drops by more than 10%. This complies with the previous empirical results and there have been a line of researches focusing on addressing the issue (Yurochkin et al., 2019a;b; Wang et al., 2020). Although orthogonal to this paper, we deem it as an interesting open problem how to integrate these works with D^{2}P-Fed.

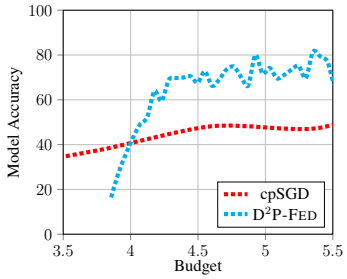


Figure 4: D^{2}P-FED under homogeneous/heterogeneous distribution.