Bounding Contribution Optimally for Federated Frequency Estimation under User-level Distributed Differential Privacy

Anonymous Author(s)

ABSTRACT

We study how to perform federated frequency estimation under user-level distributed differential privacy, where our goal is to protect the privacy of all entries from any single user under secure aggregation protocols. While many works address the fundamental analytics of private frequency estimation under the central and trusted aggregator assumption, this problem has not been specifically addressed under the federated and user-level distributedly private setting. To achieve this, we first introduce a thresholding frequency estimator satisfying the federated and secure aggregation constraints, and show that there is an optimal quantile yielding almost minimal measurement error. Then, we design methods fully compatible with the above-mentioned constraints, to estimate the quantile privately. To this end, we propose end-to-end federated protocols integrating these approaches and assess their performance via extensive experiments. Our evaluations verify our theoretical findings, and demonstrate the effectiveness of the protocols over existing solutions.

ACM Reference Format:
Anonymous Author(s). 2018. Bounding Contribution Optimally for Federated Frequency Estimation under User-level Distributed Differential Privacy. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Federated/distributed analytics (FA) and learning (FL) have emerged as a new paradigm which enables multiple users or parties to collaborate and perform analytics/learning without sharing their data directly, offering promises of protecting user privacy [29, 43]. Specifically, frequency or histogram estimation, is a fundamental analytical task, where each user holds one or more items/data from domain d, and the server wishes to estimate the global frequency of these items while guaranteeing user privacy. However, inferring private information from users is shown to be possible even under such a distributed setting, indicating that this technology alone does not protect user privacy in a strict sense [34, 54, 65]. Consequently, recent studies combine FA with privacy-enhancing technologies such as cryptographical protocols and differential privacy (DP) to preserve user privacy rigorously. In particular, *secure aggregation* (SecAgg) [12] has become a popular secure multi-party computation solution enhancing the security and privacy of FA. SecAgg requires users to add carefully designed noises to their data such that the server can only obtain the aggregated data (with noises canceled out), but is unable to read any individual data.

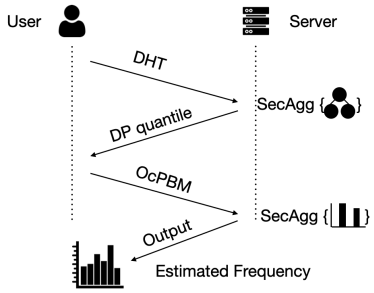


Figure 1: Overview of our two-phase federated frequency estimation protocol. Users first perform Discrete Haar wavelet Transform under distributed differential privacy (DHT) to obtain a differentially private quantile (DP quantile) from the server (Section 4). Using the DP quantile, the users perform Optimal clipped Poisson Binomial mechanism (OcPBM) on their data, which are then aggregated to obtain the desired (estimated) frequencies (Section 3.3). Our proposal satisfies user-level DP, secure aggregation conditions and achieves almost minimal measurement error.

This paper focuses on *federated frequency estimation*, the problem of performing frequency estimation under the distributed and SecAgg constraints, as well as *user-level distributed DP*, referring to techniques provably limiting individual information the server is able to deduce from the aggregated data.

Contributions. More precisely, in this paper, we tackle an unexplored question of how to define protocols to perform **cross-device federated frequency estimation (FFE)** that strictly satisfy *user-level* distributed DP (DDP) and *secure aggregation* (SecAgg) conditions, resulting in *provably private* and *cryptographically secure* FA with low error. The overview of our main proposal is illustrated in Figure 1. Technically, we make the following contributions:

- We introduce clipped Poisson Binomial mechanism, **cPBM**, a thresholding frequency estimator that satisfies user-level DDP and SecAgg constraints. By our careful construction, we show that one can provably optimize the bias-variance error trade-offs by choosing the threshold (to bound the user contribution) to be a certain private measurement's quantile (Section 3), leading to an optimal estimator, **OcPBM**, which has almost minimal frequency measurement error.
- In order to achieve almost minimal error, we design a federated wavelet-decomposition technique to perform the private quantile measurement (**DHT**). It requires only a little portion of the privacy budget, and is fully compatible with federated and

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

SecAgg constraints. These lead us to introducing an end-to-end, secure and user-private two-phase protocol, **TFFE**, in Section 4.
- Furthermore, we consider the extension where user data is processed in multiple rounds, motivated by recent federated applications that analyze large populations and must process user reports in batches. We propose a multi-round protocol, **MFFE**, which utilizes online quantile measurement to efficiently collect frequencies over multiple rounds for FFE.
- In Section 5, we perform extensive experiments to validate our proposals: (a) we verify empirically that our method indeed yields almost minimal measurement error, (b) our quantile estimation proposals are superior than existing baselines, (c) our end-to-end FFE protocols perform better than existing solutions.

In the following, we describe related work, followed by the problem setup and preliminaries in Section 2. We conclude in Section 6.

Notations. We use boldface letters to denote vectors, subscripts to denote which user it belongs to, and normal letters with square brackets to denote the coordinate. That is, if $\mathbf{x}$ is a vector, $\mathbf{x}_i$ is a vector held by user i, and $x_i[j]$ is the j-th coordinate. See also Tab. 4 in Appendix, where we give a summary of notations.

1.1 Related work

Federated Frequency Estimation. Frequency or histogram estimation is a fundamental analytical task [19, 33, 62] with many federated applications: frequencies are utilized to calibrate the classification scores of trained FL models [23], and as a building block of more advanced tasks such as learning [17] and sparse analytics [6].

Differential Privacy and Secure Aggregation. Early works of DP considered the central-DP model, which requires users to trust the server to process raw user data and publish the noisy statistics. However, this is impractical in many industrial settings as well as distibuted settings mainly due to increasingly stringent data privacy protection laws and regulations. In contrary, the *local* model of DP (LDP) asks users to perturb their reports by themselves, requiring least trust assumptions and is appropriate under the distributed setting, but introduces large errors in the published statistics [45].

A recent line of research in DP leverages cryptographic primitives (specifically SecAgg) to establish more practical trust assumptions in distributed settings while maintaining the utility of published statistics [10, 12, 52, 53]. It achieves the best of both worlds by allowing users to add lower amount of local noises without relying on a central trustworthy entity. Such primitives rule out commonly-used (continuous-value) DP mechanism such as Gaussian mechanism, and spur the study of compatible (discrete-value) DP mechanisms, also known as distributed DP, or DDP mechanisms [1, 2, 9, 20, 42].

User-level privacy. We are interested in protecting the privacy of user contributing possibly more than one sample of data per user, i.e., user-level privacy, in contrast to standard DP protecting privacy at the sample/item level. This consideration is inspired by cross-device applications which are rich in data per user or device [4, 35, 47, 49, 50, 56].

To summarize, while privacy-enhancing techniques mentioned above have been studied in depth separately before, it is little known how to integrate them in a full-fledged system to perform federated analytical tasks in a principled way (see also App. A for other related work). We resolve these questions affirmatively by presenting comprehensive studies of sytems achieving private, secure and accurate FFE.

2 PROBLEM SETUP AND PRELIMINARIES

Problem setup. Consider each of n users holding a finite number of items from a discrete domain $[d]$, where $d \in \mathbb{N}$. We write this succinctly using $\mathbf{x}_i \in \mathbb{R}^d$, where $\mathbf{x}_i = (x_i[1], ..., x_i[d])$, and $x_i[j]$ is the number of item j held by user i.

We consider federated frequency estimation under (user-level) DDP and SecAgg. That is, (a) the server does not collect the raw personal data from users directly (federated setting). (b) The intermediate and resulting output of our protocols, i.e., the estimated frequencies, must satisfy user-level DP. (c) The server is limited to receiving and performing computation using only the aggregation of per-user vectors.

Our goal is to estimate the true population-level frequency $\bar{\mathbf{x}} \equiv \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ under these constraints. Let $\hat{\bar{\mathbf{x}}}$ be the corresponding estimated frequency (let $\hat{x}_i[j]$ denote the collected value of $x_i[j]$). We quantify the error of estimation via the expected l_2 distance between these two quantities:

$$\mathbb{E}\|\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}\|_2 = \frac{1}{n}\mathbb{E}\left[\sqrt{\sum_{j=1}^{d}\left(\sum_{i=1}^{n} x_i[j] - \hat{x}_i[j]\right)^2}\right] \tag{1}$$

where the expectation is taken over the randomization due to our DP algorithm. Our protocol aims to minimize this error. In the following, we give a brief review of DP, SecAgg, and user-level distributed DP.

Differential privacy. First, let $\mathbf{D} = (\mathbf{x}_i)_{i=1}^n$ be the dataset of users considered above. We say that a dataset $\mathbf{D}'$ is a adjacent dataset of $\mathbf{D}$ when they differ in at most one individual data.

Definition 1 (Differential Privacy [25, 26])**.** Given $\epsilon \geq 0$ and $\delta \in [0, 1]$, a randomization mechanism, $\mathcal{M} : \mathcal{D}^n \to \mathcal{S}$ with domain $\mathcal{D}^n$ and range $\mathcal{S}$ satisfies (ϵ, δ)-differential privacy (DP) if for any two adjacent datasets $\mathbf{D}, \mathbf{D}' \in \mathcal{D}^n$ with n users and for any subset of outputs $S \subseteq \mathcal{S}$, the following holds:

$$\Pr[\mathcal{M}(\mathbf{D}) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(\mathbf{D}') \in S] + \delta. \tag{2}$$

We emphasize that under this definition, user-level DP is being considered, instead of item-level DP, where the adjacent datasets instead differ in a single instance, e.g., $x_i[j]$ in our case. We next introduce Rényi differential privacy, the main notion used here.

Definition 2 (Rényi Differential Privacy (RDP) [55])**.** A randomization mechanism $\mathcal{M} : \mathcal{D}^n \to \mathcal{S}$ is ϵ-Rényi differential privacy of order $\lambda \in (1, \infty)$ (or (λ, ϵ)-RDP), if for any adjacent databases $\mathbf{D}, \mathbf{D}' \in \mathcal{D}^n$, the Rényi divergence of order λ between $\mathcal{M}(\mathbf{D})$ and $\mathcal{M}(\mathbf{D}')$ is upper-bounded by ϵ: $D_\lambda(\mathcal{M}(\mathbf{D})||\mathcal{M}(\mathbf{D}')) \leq \epsilon$, where $D_\lambda(P||Q) = \frac{1}{\lambda-1}\log\left[\mathbb{E}_Q\left(\frac{P}{Q}\right)^\lambda\right]$.

Two mechanisms satisfying (λ, ϵ_1),(λ, ϵ_2)-RDP respectively, under composition, satisfy $(\lambda, \epsilon_1 + \epsilon_2)$-RDP, allowing for clean privacy composition [13, 28, 55]. The RDP notion can be converted to the conventional approximate DP notion; see App. C.1.

SecAgg/SecSum. We are mainly concerned with the secure aggregation (SecAgg) framework, composed of three components: (1)

Algorithm 1 Main user protocol for FFE

Inputs: Data $\mathbf{x}$. Tight frame U. Parameters θ, C, m.
Output: Data encoded on a finite group $\mathbb{Z}_M^D$
Kashin's representation: Calculate the Kashin's representation, $\mathbf{y}$, of $\mathbf{x}$
cPBM: Apply Alg. 3 with parameters θ, C, m on $\mathbf{y}$ to output $\mathbf{Z}$
Return: $(Z[k])_{k=1}^D$

a client-side local encoder, $\mathcal{A}$, (2) a secure protocol, $\mathcal{P}$ and (3) a untrusted analyzer, or central server, $\mathcal{S}$. At each communication round, the protocol proceeds as follows: Each user first encodes her sensitive report $\mathbf{x}_i$ using $\mathcal{A}$: $Z_i = \mathcal{A}(\mathbf{x}_i)$. Then, the reports are processed by $\mathcal{P}$, of which the output is relayed to $\mathcal{S}$. $\mathcal{P}$ ensures that the output seen by the $\mathcal{S}$ (on which no trust assumption is put) satisfies certain properties. We mainly focus on the secure summation (SecSum) protocols , where each individual report $\mathbf{x}$ is encrypted to $\mathbf{Z} \in \mathbb{Z}_M^D$, $D, M \in \mathbb{N}$ a finite additive group, but only the modular summation of reports, $\sum_i \mathbf{Z}_i \mod M$, is exposed to $\mathcal{S}$. Finally, $\mathcal{S}$ decodes the received reports to obtain an estimate of $\sum_i \mathbf{x}_i$. Given that the implementation of SecSum is well-studied [10, 12], we treat $\mathcal{P}$ as a black box guaranteed to compute the modular sum faithfully without security or privacy leakage concerns. [1]

User-level Distributed Differential Privacy. $\mathcal{P}$ alone does not guarantee DP because aggregated quantities could still leak sensitive information. On the other hand, adding noises at the side of $\mathcal{S}$ requires one to fully trust $\mathcal{S}$, not favorable in practice. *Distributed differential privacy* (DDP) is a privacy model that aims to achieve near-central-DP guarantees without a trustworthy server. This is done by adding client-side perturbations (which alone are too small to provide meaningful DP guarantees) through $\mathcal{A}$ to the reports, and using SecAgg to achieve meaningful DP guarantees from the view of $\mathcal{S}$. Our interest is in *user-level DDP*, where all items held by any single user is encoded, bounded, and perturbed appropriately by $\mathcal{A}$, and are simultaneously guaranteed with DDP. The most common way of bounding user contribution is via l_2-norm clipping: $\mathbf{x} \leftarrow \mathbf{x} \cdot \min(\frac{C}{\|\mathbf{x}\|_2}, 1)$ where $C \in \mathbb{R}^+$. By clipping to a certain C, each user can only contribute $\mathbf{x}$ with l_2-norm at most C, irrespective of the number of items she holds.

3 OUR METHOD

We describe our method of performing FFE through collaboration between users and server. At the user side, our proposed protocol (Alg. 1) consists of the following steps:

(1) **Data projection**: Each user project $\mathbf{x}_i$ to its Kashin's representation, which bounds the l_∞ of $\mathbf{x}_i$ while maintaining its l_2 geometry.
(2) **Bounding contribution**: User applies the clipped Poisson binomial mechanism with threshold C to the Kashin's representation, and sends it out to be processed by the SecSum protocol.

The server decodes the SecSum of user reports following Alg. 2. In the following, we first describe these steps in detail, and subsequently study how to bound user contribution optimally.

[1] Hereafter, we use the terms SecSum/SecAgg interchangeably.

Algorithm 2 Main server protocol for FFE

Inputs: SecSum of user data $(\sum_{i=1}^n Z_i[k])_{k=1}^D$. Tight frame U. Parameters θ, C, m.
Output: Decoded mean of user data
Rescale: $\hat{\tilde{\mathbf{y}}} \leftarrow \frac{C}{mn\theta}(\sum_{i=1}^n \mathbf{Z}_i - \frac{mn}{2}\mathbf{1})$
Decode: $\hat{\tilde{\mathbf{x}}} \leftarrow U\hat{\tilde{\mathbf{y}}}$
Return: $\hat{\tilde{\mathbf{x}}}$

Algorithm 3 Clipped Poisson binomial mechanism (cPBM)

Inputs: Representation $\mathbf{y}$. Parameters θ, C, m
Output: Data encoded on a finite group $\mathbb{Z}_M^D$
Clip: $\mathbf{y} \leftarrow \mathbf{y} \cdot \min(\frac{C}{\|\mathbf{y}\|_2}, 1)$
for $k \in [D]$ **do**
  $p[k] = \frac{\theta}{C} y[k] + \frac{1}{2}$
  Sample $Z[k] \sim \text{Binom}(m, p[k])$
Return: $(Z[k])_{k=1}^D$

3.1 Data projection

We utilize the Kashin's representation for data projection:

Definition 3 (Kashin's representation [44]). For a set of vectors $(\mathbf{u}_k)_{k=1}^D$, the Kashin's representation of $\mathbf{x} \in \mathbb{R}^d$ at level K is the set of coefficients $y[k]$ satisfying the following: $\mathbf{x} = \sum_{k=1}^D y[k]\mathbf{u}_k$, $\max_k |y[k]| \le \frac{K}{\sqrt{D}}\|\mathbf{x}\|_2$.

One may think of it as a generalization of orthogonal bases where one uses a redundant system of vectors $(\mathbf{u}_k)_{k=1}^D, \mathbf{u}_k \in \mathbb{R}^d$, to represent $\mathbf{x} \in \mathbb{R}^d$, where $D > d$. This is called a tight frame when for any $\mathbf{x}$, the Parseval's identity is satisfied: $\|\mathbf{x}\|_2 = \sum_{k=1}^D |\langle \mathbf{x}, \mathbf{u}_k \rangle|^2$. We utilize this representation to spread out the information contained in $\mathbf{x}$ across several coefficients of $\mathbf{u}_k$'s, to bound their l_∞-norm (such that we can apply DP noises independently to each coordinate with well-defined sensitivity). It also satisfies the following.

Lemma 1 (Uncertainty principle [51]). There exists a tight frame $U = [\mathbf{u}_1, ..., \mathbf{u}_D]$ such that $D = \Theta(d)$ and $K = O(1)$.

This means that one can represent any $\mathbf{x}$ with $\mathbf{y} = (y[1], ..., y[D])$ that is in the range $[-\frac{K}{\sqrt{D}}\|\mathbf{x}\|_2, \frac{K}{\sqrt{D}}\|\mathbf{x}\|_2]$. Conversely, one can decode y to recover $\mathbf{x}$ via $\mathbf{x} = U\mathbf{y}$. Details of computing these values are available in App. B.

3.2 Clipped Poisson Binomial Mechanism

Now each of the coordinates of $\mathbf{y}$ has bounded l_∞. We next encode each coordinate of $\mathbf{y}$ to values on a finite group that is DP and compatible with SecSum. We propose a variant of the Poisson binomial mechanism [20] which we call clipped Poisson binomial mechansim (**cPBM**), as shown in Alg. 3.

In particular, for each coordinate of $\mathbf{y}$, we (1) clip its l_2 norm to C, (2) encode $y[k]$ to $p[k] = \frac{\theta}{C}\text{clip}(y[k]) + \frac{1}{2}$, (3) sample from the binomial distribution, $Z[k] \sim \text{Binom}(m, p[k])$ and send $Z[k]$ to the server. Here, $\theta \in [0, 1/4]$ and $m \in \mathbb{N}$ are parameters that govern the privacy-utility trade-offs. m is also the number of bits sent out by each user. The clipping operation bounds user contribution, and we use clipping/bounding user contribution interchangeably from

now onwards. Note that for SecSum operating on a finite group of size M, when summing n quantities of size m each, we avoid integer overflows by choosing parameters such that $M = nm$. Thus, each user sends $\log_2(m) + \log_2(n)$ bits per dimension to the server.

Utility of cPBM. Using properties of binomial distribution, it is easy to show that the server receiving $\sum_i Z[k]$ can make an unbiased estimate of the mean and variance of $y[k]$ via

$$\hat{y}[k] = \frac{C}{mn\theta}\left(\sum_i Z[k] - \frac{mn}{2}\right),\ \mathrm{Var}(\hat{y}[k]) \le \frac{C^2}{4nm\theta^2}. \tag{3}$$

Privacy of cPBM. The SecSum of the outputs, $\sum_i Z[k]$, of Alg. 3 satisfies RDP, which is given below (proof in App. C.2).

Lemma 2 (cPBM RDP bounds)**.** The SecSum of the cPBM outputs of n users is differentially private, and its RDP of order λ bounded by $\epsilon(\lambda) \le \frac{md}{\lambda-1}\log\max(\Pi_1, \Pi_2)$, where $\Pi_1 = \sum_{m=0}^{n'+1}\mathrm{Bin}(m, n', \frac{1}{2}+\theta)e^{\lambda\xi_m}$, $\Pi_2 = \sum_{m=0}^{n'+1}\mathrm{Bin}(m, n', \frac{1}{2}-\theta)(\frac{1}{2}+\theta)(\frac{m}{n'-m+1}\frac{1+2\theta}{1-2\theta} + \frac{1-2\theta}{1+2\theta})e^{-\lambda\mu_m}$; $\xi_m = \log(\frac{1+2\theta}{1-2\theta}\frac{n'+1-m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{m}{n'+1})$, $\mu_m = \log(\frac{1+2\theta}{1-2\theta}\frac{m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{n'+1-m}{n'+1})$, and $n' = \lceil\frac{n-1}{2}\rceil$, $\mathrm{Bin}(m, n, p) = \binom{n}{m}p^m(1-p)^{n-m}$. Asymptotically, $\epsilon(\lambda) = \Theta(md\theta^2\lambda/n)$.

Compared to [20], where only asymptotic bounds are given, we re-derive the RDP of cPBM to give an explicit and computable bound with all the exact constants in the above Lemma.

3.3 Bounding contribution optimally (OcPBM)

Building on the tools developed in previous subsections, we introduce **OcPBM**, our method to bound user contribution optimally. We first characterize the expected l_2-error (Eq. 1) of cPBM.

Theorem 1 (cPBM's expected l_2-error)**.** Let $\hat{\bar{\mathbf{x}}}$ be the decoded SecSum of user reports following the protocols of Alg. 1 and 2. The expected l_2-error of cPBM is at most

$$\mathbb{E}\|\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}\|_2 \le \sqrt{d}\cdot\mathcal{E}(C;\mathbf{D}) \tag{4}$$

where $\mathcal{E}(C;\mathbf{D}) := \frac{1}{n}\sum_{i=1}^n \max(0, \|\mathbf{y}_i\|_2 - C) + C\sqrt{\frac{D}{4mn\theta^2}}$ with $\mathbf{y}_i$ being $\mathbf{x}_i$'s Kashin's representation.

The proof can be found in App. C. In essence, we first show that the expected l_2-error of $\mathbf{x}$ equals to that of Kashin's representation up to a multiplicative constant of $\sqrt{d}$. Then, we perform bias-variance decomposition on the expected l_2-error to arrive at the required statement.

Theorem 1 indicates that choosing a C too small induces a large bias, while a C too large causes a large variance. Choosing an optimal C can hence lead to optimal bias-variance trade-off which minimizes the expected l_2-error. It can be seen from the definition of $\mathcal{E}(C;D)$ that it is a function convex in C. Thus, one can find the optimal C by taking the derivative of $\mathcal{E}(C;D)$ to be zero:

$$0 = -\frac{1}{n}\sum_{i:\|\mathbf{y}_i\|_2 > C} 1 + \sqrt{\frac{D}{4mn\theta^2}} \tag{5}$$

Thus, *the optimal C is the $1 - \sqrt{\frac{D}{4mn\theta^2}}$-th quantile of $\|\mathbf{y}_i\|_2$.*

Since the optimal C is data ($\|y\|_2$) dependent, it has to be estimated privately to avoid privacy leakage. We resolve this issue in the next Section.

Remark on clipping threshold and privacy budget. One may wonder why the clipping threshold can be chosen according to Eq. 5 without affecting the privacy budget ϵ. This is because in our formulation, ϵ does not depend explicitly on the threshold C (Lemma 2), in the same way that the privacy budget of the Gaussian mechanism does not depend on the clipping threshold: $(\epsilon, \delta) = (\sqrt{2\log(1.25/\delta)/\sigma}, \delta)$ when the noise scale is $C\sigma$. This does not contradict the common fact that larger C introduces larger "noises"; from Eq. 3, we see that the variance of the private output is proportional to C^2; larger C indeed leads to larger uncertainty in the estimation of the output.

Remark on novelty. Our insight leading to optimal bias-variance trade-off lies in our careful algorithmic design: *by bounding user contribution in the representation space and applying a suitable DP mechanism, we can solve a convex function for an optimal threshold with theoretical guarantees.* Naively, one could bound user contribution by clipping $\mathbf{x}$ directly and using well-known discrete DP mechanisms (e.g., discrete Gaussian [42]) to achieve user-level DP under SecSum constraints. This however does not work well as clipped $\mathbf{x}$ encoded and added with discrete (Gaussian) noises leads to a complicated non-convex expression of the l_2-error that is hard to optimize (e.g., Eq. 63 of [42]). Experiments to be presented in the subsequent sections also verify our approach's effectiveness.

4 FEDERATED QUANTILE ESTIMATION

Quantile estimation is well studied under the central-DP model [36, 60, 61]. These algorithms utilize sub-routines such as sorting that require the full view of data not possible under the distributed settings. Recently, [57] studied this problem under the federated setting by essentially quantizing per-user values to per-user (flat or hierarchical) histogram, performing randomization, and aggregating them using SecSum.

Here, we propose an alternative quantile estimation solution based on wavelet decomposition techniques. Our contribution here is showing that it performs even better, and how this technique can be adapted to be compatible with DDP constraints.

4.1 Setting and Baselines

Recall that our aim is to obtain the quantile of of $\|\mathbf{y}_i\|_2$, which is a private statistics to be calculated from each user i holding $\|\mathbf{y}_i\|_2$.

Flat histograms. Consider a one-dimensional (non-empty) histogram of range B with 2^b bins. [2] In this approach, each user discretizes $\|\mathbf{y}\|_2$ into her local histogram: $\|\mathbf{y}\|_2 \in \{0,1\}^{2^b}$. Then, the user applies DP noises (e.g., cPBM introduced earlier) to each bin of the histogram, and finally sends the histogram to the SecSum protocol [57]. The quantile is estimated from the aggregated histogram by finding the bin that minimizes the difference between the estimated and target quantile. Denote the bin size by Δ. This approach yields error depending linearly on the range of the histogram, B:

Proposition 1. The variance of the estimated quantile of the flat-histogram approach is at most $\mathrm{Var}_{\mathrm{flat}}(\hat{\gamma}) \le \frac{B}{4\Delta mn^2\theta^2}$ when the underlying DP mechanism is cPBM (proof in App. C.4).

[2] The number of bins can be any integer; making it equal to the power of 2 is simply for later convenience.

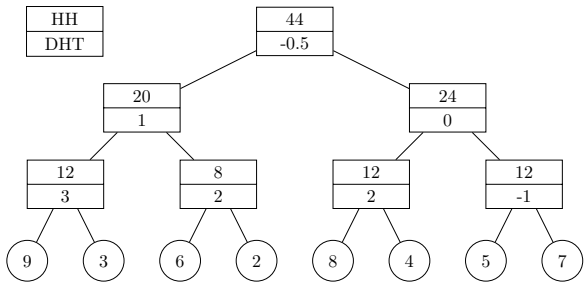


Figure 2: An example of using a binary tree to perform quantile/range estimation. The leaves (circles) represent the histogram bin counts. The upper part of the internal nodes (rectangles) represents the hierarchial histograms (HH) approach, which maintains the sum of all leaves below it. The lower part represents the Discrete Haar wavelet transform (DHT) approach, which maintains the Haar coefficient calculated using Eq. 6. To perform federated quantile estimation, each user adds DP noises to her own binary tree, and sends it to the secure aggregation protocols. See text for details.

Hierarchical histograms (HH). [57] also considered an alternative approach where each user maintains, on top of the single bins as in the flat histogram approach, aggregated counts of the bins in the form of a binary tree, consisting of nodes where each of them is the sum of all leaves below that node. See Fig. 2 for an example.

As before, the user applies DP noises to all the nodes and leaves (bins) and send the histogram to the SecSum protocol. The main strength of HH comes from using counts from upper nodes to estimate the quantile, resulting in less noisy estimation. In fact, asymptotically, the error has logarithmic dependency with respect to B (proof in App. C.5):

Proposition 2. The variance of the estimated quantile of the HH approach is at most $\mathrm{Var}_{\mathrm{HH}}(\hat{\gamma}) \leq \frac{\log_2(B/\Delta)}{4mn^2\theta^2}$ when the underlying DP mechanism is cPBM.

4.2 Discrete Haar Wavelet Transform (DHT)

Although the HH approach has lower asymptotic error as indicated by Proposition 2, [57] finds that in practice, the flat histogram approach performs better. One reason is that the HH approach does not fully utilize the binary-tree structure, leading to redundancy: we output the value of each node, which should sum to the value of the parent node (output independently by the HH algorithm). While some studies aim to reduce this redundancy to improve estimation via postprocessing [38, 39, 46], we explore an alternative representation of the binary-tree structure where no such redundancy occurs: the Discrete Haar Wavelet Transform. Here, pairwise averaging and differencing of child nodes are performed, instead of performing summation as in the HH approach. The resulting node values/coefficients are independent to each other, and therefore have no redundancy [63].

More technically, as before, consider a binary tree over $\mathcal{B}$ with 2^b leaves corresponding to each of the bins. Suppose l is the level of the node v in the binary tree, starting at zero from the leaves. The Haar wavelet coefficient, or simply Haar coefficient $H_l[v]$ for

Algorithm 4 User-side DHT protocol

1: **Inputs:** Data $\mathbf{x}_i$. Parameters B, b, θ', m'
2: **Output:** Haar coefficients
3: **Step 1:** Calculate Kashin's representation's norm, $\|\mathbf{y}_i\|_2$
4: **Step 2:** Discretize $\|\mathbf{y}_i\|_2$ to histogram (range B, bin size 2^b)
5: **Step 3:** Apply DHT (Eq. 6) to histogram to yield H_l's
6: **Step 4:** Apply cPBM to H_l's with parameters θ', m' to yield $\hat{H}_l$
7: **Return:** $\hat{H}_l$

Algorithm 5 Server-side DHT protocol

1: **Inputs:** SecSum of user data $\hat{H}_l$, Target quantile γ^*, bin size Δ
2: **Output:** Estimated quantile of $\|\mathbf{y}_i\|_2$
3: **Step 1:** Decode SecSum of $\hat{H}_l$ with Eq. 3 to yield $\hat{H}_l$
4: **Step 2:** Decode $\hat{H}_l$ with Eq. 7 to yield $\hat{c}$
5: **Step 3:** Obtain $\hat{v} = \mathrm{argmin}_{v\in\text{leaves}} |\frac{\hat{c}[1:v]}{n} - \gamma^*|$
6: **Return:** $\Delta \cdot \hat{v}$ as the estimated quantile of $\|\mathbf{y}_i\|_2$

node v at level l is defined as

$$H_l[v] = (a_L - a_R)/2^l, \tag{6}$$

with a_L (a_R) being the total counts of the leaves in the left (right) subtree of v. Defining the base coefficient H_0 as the sum of the counts of all leaves, we can decode the leaf count (or the count of bin $v, v \in [2^b]$ of the histogram), $c[v]$, from the Haar coefficients as $c[v] = H_0 + \sum_l H_l[v] \cdot g_l[v]$, where $g_l[v]$ is +1 (−1) if v is in the left (right) subtree of the l-level ancestor node. See Fig. 2.

Federated quantile estimation via DHT. We adapt DHT to federated quantile estimation with the following user protocol. Each user first discretizes her $\|\mathbf{y}_i\|_2$ into a histogram of range B with 2^b ($b \in \mathbb{N}$) bins. We rescale the Haar coefficient to be $H_l[v] = a_L - a_R$ such that each user only has a non-zero leaf on her binary tree, and a non-zero Haar coefficient of bounded value, ±1 at each level. The user then applies cPBM to all nodes of the binary tree with $C = 1$, and subsequently send them to the SecSum protocols; see Alg. 4. The server finally obtains a summation of the binary trees, obtaining the estimated coefficients, $\hat{H}_l$'s, as an unbiased estimator of H_l's. The leaf count can be decoded as follows:

$$c[v] = H_0 + \sum_l \frac{\hat{H}_l[v]}{2^l} \cdot g_l[v], \tag{7}$$

and be used to estimate a quantile via Eq. 7 by summing up $c[v]$'s from the first leaf node to a given range and dividing it by the number of user (Alg. 5).

Utility. Observe that for any internal node of which all *or* none of its child leaves is a subset of the given range, its contribution to Eq. 7 is exactly zero. Non-zero contributing terms are ranges with leaf nodes that "cut" through the subtree of an internal node. Roughly speaking, this implies that the number of noisy Haar coefficients that contributes to quantile estimation is logarithmic with respect to the range.

Let us quantify the variance of the estimated quantile more precisely. Let $\hat{c}[1:v]$ to be the estimated sum of counts up to v-th leaf node. Also let $s_L[w]$ ($s_R[w]$) be the number of leaf nodes within the range in the left (right) subtrees of node w. Then, we can write

Algorithm 6 Two-phase FFE protocol (TFFE)

1: **Inputs:** User data $\mathbf{x}_i$. Parameters B, b, U, θ, θ', m, m'
2: **Output:** Estimated frequencies
3: **Step 1:** User applies DHT (Alg. 4) and sends the Haar coefficients to the SeSum protocol
4: **Step 2:** Server obtains the SecSum of the Haar coefficients, and returns the estimated DP quantiles, $\hat{C}$ (Alg. 5)
5: **Step 3:** User runs OcPBM with $\hat{C}$ and sends it to the SecSum protocol (Alg. 1)
6: **Step 4:** Server obtains $\hat{\tilde{\mathbf{x}}}$ with Alg. 2
7: **Return:** $\hat{\tilde{\mathbf{x}}}$

$\hat{c}[1:v] = vH_0 + \sum_w (s_L[w] - s_R[w]) \frac{\hat{H}_l[w]}{2^l}$. The threshold that corresponds to the optimal quantile γ^* is v that minimizes $|\frac{\hat{c}[1:v]}{n} - \gamma^*|$. With this observation, we have the following Proposition quantifying the utility of the DHT protocol (proof in App. C.6).

Proposition 3 (Utility of DHT)**.** The variance for federated quantile measurement using the DHT protocol is at most $\mathrm{Var}(\hat{\gamma}) \leq \frac{\log_2(B/\Delta)}{16m'n^2\theta'^2}$, with $\Delta =: B/2^b$ the bin size, m', θ' parameters of cPBM.

Other advantages. The design of our protocol has several key advantages, in addition to the logarithmic dependency on the range B and non-redundant properties: (a) The Haar coefficients of the entire population can be conveniently reconstructed from individual Haar coefficients by summation, satisfying SecSum requirements. (b) The entry of the binary tree is symmetric around zero, i.e. it is bounded to be ± 1 (such that there is no positive bias, unlike common histogram approaches that have entry of either 0 or 1); it is convenient to apply DP mechanism like cPBM which is also symmetric around zero without any modification.

4.3 Federated Frequency Estimation Protocols

Equipped with the quantile estimation techniques, we propose an end-to-end protocol, extending the single-round data aggregation setting to a two-round scheme that first have all users estimate the quantile of $\|\mathbf{y}\|_2$ privately, before performing FFE by aggregating all users' frequencies with the estimated quantile. Alg. 6 outlines our two-phase Federated Frequency Estimation protocol, **TFFE**: we first let each user quantize her $\|\mathbf{y}\|_2$ into a per-user histogram, and process the histogram with DHT before aggregating it via SecSum to estimate the quantile of $\|\mathbf{y}\|_2$. Then, we perform the estimation using Alg. 1 to obtain the desired frequency. The overall privacy guarantees are given as follows (proof in App. C.7):

Proposition 4 (Privacy guarantees of TFFE)**.** Let m, θ (m', θ') be the parameters of the underlying cPBM applied to federated quantile (frequency) estimation in Alg. 6. Then, Alg. 6 satisfies $\epsilon(\lambda)$-RDP where $\epsilon(\lambda) = \Theta(md\theta^2\lambda/n + m'(2^b - 1)\theta'^2\lambda/n)$.

Extension to multi-round Federated Frequency Estimation (MFF). Here, we consider briefly the situation where not all users are not available at the same time, and aggregating all user data requires multiple rounds of communication, motivated due to the following practical reasons: (a) the population is too large that the server has to process the data in batches due to communication and compute constraints. (b) Users (e.g., mobile devices) may not be available at the same time to participate in FA as they may face battery outage, network disconnection, etc.. Under such a constraint, it may be too inefficient communication-wise to gather all user reports synchronously twice for quantile and frequency estimation. We henceforth propose to estimate the quantile or threshold in an *online* and private fashion, while performing frequency estimation *simultaneously*.

This can be done by, at communication round r, requiring the users, along with their data clipped with the online value of C, $C^{(r)}$, to send a private indicator of whether her $\|\mathbf{y}_i\|_2$ is larger than $C^{(r)}$: $\mathbb{I}_{\|y_i^{(r)}\|_2 > C^{(r)}}$. The private indicator can be privatized with cPBM, and aggregated via SecSum to estimate the current quantile, $\hat{\gamma}^{(r)}$. Consequently, it can be used to update C geometrically to move it closer to the desired γ^*: $C^{(r+1)} \leftarrow C^{(r)} \exp(-\eta(\hat{\gamma}^{(r)} - \gamma^*))$, where η is the learning rate parameter. Frequency is then estimated with improved values of C at each round, and the overall frequency is obtained at the end by aggregating results from all round. Details of the protocol, the validity of Theorem 1 for the multi-round scenario, and the privacy guarantees can be found in App. D.

5 EXPERIMENTS

We provide empirical studies of our method and protocols in this Section. [3] We first give an in-depth analysis of each part of our proposed protocol to demonstrate that our theoretical findings are well supported empirically. Then, we investigate the performance of our protocol compared to existing approaches. Due to space constraints, we relegate some additional results and details to App. E.

Common setup. Throughout our studies, we use both synthetic and real-world datasets:

- Synthetic data: We generate both i.i.d. and heterogenous data following [50] : For i.i.d., each user samples 100 samples from $\mathbf{p}$, a discrete distribution over $[d]$ with probability proportional to $1/j$ for $j \in [d]$. For heterogenous dataset, we sample n κ's from $\mathrm{Dir}(2)$. Each user i draws $100\kappa_i$ Poisson samples from $\mathbf{p}_i \sim \mathrm{Dir}(\mathbf{p}/2)$ where $\mathbf{p}$ is the same distribution defined for the i.i.d. dataset.
- Real-world data: Two real-world datasets are used: **Foursquare:** We use the long-term (April 2012 to September 2013) global-scale data of users checking in to various locations [64]. Each user is assigned an anonymous ID, and the check-in data is a two-column table containing anonymous ID and location ID. The location IDs can be divided into different categories (arts, coffee, restaurant, etc.). We link each user ID to its check-in categories to build the user histogram. A histogram with d bins is a histogram with the top-d most checked-in categories globally. **Sentiment140:** We use a dataset of user tweets [37]. For each user's tweets, we parse them to words and treat the words as items for the histogram. A histogram with d bins is a histogram with the top-d most used words over all users.

5.1 Analyzing each component of our protocol

We first give an in-depth empirical analysis of various proposals and arguments given in the previous Sections.

[3] Code: https://anonymous.4open.science/r/fedfreqest-anonymous-DCF5

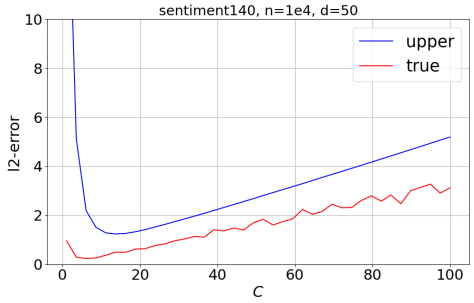


Figure 3: Our thoretically derived bound (blue) indeed upper-bounds the true l_2-error (red) when varying threshold C.

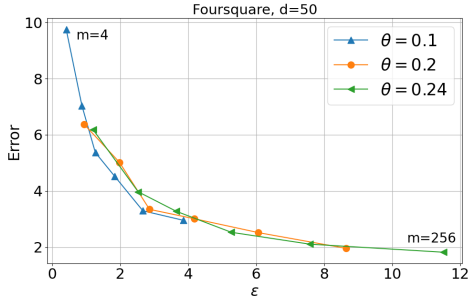


Figure 4: l_2-error versus privacy budget for various parameters of cPBM.

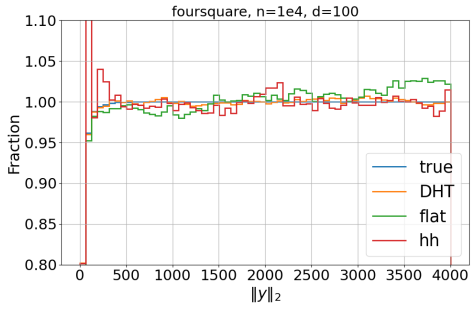


Figure 5: Federated quantile estimation, comparing our DHT approach (orange) with flat-histogram (green), hierarchical histograms (red), and the un-noised histogram (blue). The orange line almost overlaps with the blue one, demonstrating excellent agreement with the true quantile.

Our bound in Theorem 1 indeed upper-bounds the true error. We first empirically verify Theorem 1, which provides an upper bound on the l_2-error as a function of C. To do this, we plot the true l_2-error and the upper bound, setting $n = 10^4, d = 50, \theta = 0.2, m = 30$, running the experiments three times and taking the average, in Figure 3. It can be seen that our upper bounds on the expected l_2-error is indeed larger than the true ones. Similar conclusions apply to other datasets; see App. E.

i.i.d. dataset	C^*	0.9	0.92	0.94	0.96	0.98
default	1.160	1.217	**1.011**	1.141	1.109	1.205
$d = 1000$	3.168	**3.300**	3.391	3.460	3.506	3.603
$m = 10$	1.909	1.919	2.003	**1.904**	2.248	2.146
$n = 10^5$	0.334	0.425	0.412	0.366	**0.358**	0.387

Table 1: l_2-error of OcPBM (C set to C^*) and cPBM with C set to various quantile values. Bold values are fixed quantiles that yield minimal errors. There is no single fixed quantile that minimizes the l_2-error for all settings.

Our selection of quantile is robust and yields almost minimal measurement error. We study how sensitive the l_2-error is with respect to the value of the quantile for various settings of data and hyperparameters. To do this, we use the i.i.d. dataset, and let the default setting to be $n = 10^4, d = 50, m = 30, \theta = 0.2$, varying n, d, m for other settings. Then, we measure the l_2-error obtained from cPBM with fixed quantile values. These are to be compared with OcPBM, where the optimal quantile is chosen adaptively.

The results are in Table 1. We see that the optimal and fixed quantile varies across different settings, meaning that there is no single fixed quantile that can achieve empirically minimal error for all settings. Our results also show that the the error is quantile-sensitive. However, the error obtained from OcPBM is either optimal or remains close to the empirically optimal and fixed quantile, at most 1.15 times larger. This shows that OcPBM can provably choose the optimal quantile adaptively based on the properties of data and hyperparameters, which also means that OcPBM is *robust* across datasets and hyperparameters, always achieving close-to-minimal error.

Hyperparameter selection. We next study how the parameters of cPBM (θ and m) affect utility and privacy. We plot the l_2-error (with optimal C) for $m = 4, ..., 256$ and different values of θ in Figure 4 . Note that θ cannot be too small; otherwise no optimal solution for Eq. 5 would exist. We find that when fixing ϵ, different θ's do not affect the utility much, but changes the range of ϵ given a fixed range of m's. As we are interested in ϵ in the $O(1)$ range, $\theta = 0.2$ is most suitable, and for the rest of the paper, we set $\theta = 0.2$.

Federated quantile estimation. We now compare the proposed DHT protocol with the baseline approaches (flat histogram and HH), outlined in Section 4 empirically. We note that one must choose a suitable range of the histogram, B to compare these approaches. By noticing that $\|\mathbf{y}\|_2 \le K\|\mathbf{x}\|_2$ from Def. A, and $\|\mathbf{x}\|_2 \le \|\mathbf{x}\|_1$, one can set the range of B to be K times the maximum number of items held by any user, $\max_i \|\mathbf{x}_i\|_1$. [4]

Figure 5 compares the quantile function (cumulative distribution function) of the non-DP (un-noised) histogram, DHT, the (DP) flat histogram, and HH. We see that the DHT histogram traces the non-DP histogram better than the flat histogram and HH approaches. Particularly, the quantile in consideration (Eq. 5) is close to 1 for large n; other approaches fluctuate around 1 and cannot give accurate estimation for quantiles we are interested in. See App. E for other figures and a more quantitative comparison.

[4]We consider the maximum number of items held by any user to be a known non-private value; in practice, commercial devices typically have default upper limits or can set limits based on their storage capacity.

Dataset	d	CDP	LDP	Best C^*	**OcPBM**	C_{median}	`Normalize`	`Sample(one)`	`Sample(0.25)`
i.i.d.	50	0.04 ± 0.00	20.82 ± 9.58	1.21 ± 0.07	**1.16 ± 0.18**	1.49 ± 0.16	27.3 ± 0.00	28.2 ± 0.00	26.9 ± 0.02
	100	0.04 ± 0.00	17.09 ± 11.41	6.02 ± 0.24	**6.29 ± 0.52**	9.19 ± 0.22	24.4 ± 0.00	24.7 ± 0.00	23.79 ± 0.01
hetero	50	0.35 ± 0.03	43.04 ± 18.12	4.93 ± 0.35	**5.31 ± 0.64**	10.38 ± 0.13	28.14 ± 0.00	28.40 ± 0.00	28.62 ± 0.12
	100	0.31 ± 0.03	51.24 ± 33.84	5.86 ± 0.55	**5.35 ± 0.08**	19.5 ± 0.02	24.4 ± 0.00	24.7 ± 0.00	24.85 ± 0.04
Foursquare	50	0.49 ± 0.03	27.29 ± 12.1	3.92 ± 0.36	**4.47 ± 0.25**	6.62 ± 0.10	17.46 ± 0.00	17.76 ± 0.00	17.8 ± 0.23
	100	0.48 ± 0.02	36.4 ± 24.64	4.71 ± 0.22	**5.90 ± 0.50**	6.42 ± 0.00	17.69 ± 0.00	18.00 ± 0.00	18.43 ± 0.24
Sentiment140	50	0.02 ± 0.0	1.84 ± 0.61	0.28 ± 0.03	**0.29 ± 0.04**	0.41 ± 0.01	1.00 ± 0.00	1.19 ± 0.00	0.37 ± 0.03
	100	0.03 ± 0.0	2.39 ± 1.48	0.32 ± 0.02	**0.36 ± 0.02**	0.39 ± 0.01	1.05 ± 0.00	1.25 ± 0.00	0.48 ± 0.01

Table 2: l_2-errors of two-round federated frequency estimation. Mean (standard deviation) over 5 runs.

Dataset	d	Best C^*	**OcPBM**	C_{small}	C_{median}
i.i.d.	50	0.54 ± 0.04	**0.65 ± 0.08**	0.94 ± 0.10	3.47 ± 0.11
	100	0.7 ± 0.04	**0.65 ± 0.05**	1.3 ± 0.06	2.93 ± 0.07
hetero	50	2.67 ± 0.20	**2.85 ± 0.26**	4.97 ± 0.20	10.45 ± 0.08
	100	3.22 ± 0.17	**3.26 ± 0.09**	5.92 ± 0.07	8.81 ± 0.07
Foursquare	50	2.37 ± 0.22	**3.26 ± 0.09**	5.92 ± 0.07	8.81 ± 0.07
	100	2.92 ± 0.27	**2.93 ± 0.20**	4.85 ± 0.14	6.66 ± 0.06
Sentiment140	50	0.15 ± 0.01	**0.14 ± 0.01**	0.23 ± 0.01	0.38 ± 0.00
	100	0.19 ± 0.01	**0.19 ± 0.01**	0.27 ± 0.01	0.35 ± 0.01

Table 3: l_2-errors of multi-round federated frequency estimation. Mean (standard deviation) over 5 runs.

5.2 End-to-end comparisons

Here, we first study the empirical performance of our two-phase FFE protocol (TFFE, Alg. 6) and compares it with existing baselines. Additionally, we evaluate the multi-round FFE (MFFE) protocol as studied at the end of Section 4.3.

Baselines. Let us consider existing solutions of bounding user contribution for performing FFE.

- `Normalize`: One common approach in the literature, e.g., [6], is to simply normalize user contribution to one: $\mathbf{x}_i \leftarrow \mathbf{x}_i / \|\mathbf{x}_i\|_2$.
- `Sample`: Another conventional method, considered in, e.g., [15], is to sample a fixed number of instances from users and add noises proportional to the number of items sent for privatization. We consider sampling a single item (`Sample(one)`), and at most 25% (`Sample(0.25)`) of the maximum number of items, $\max_i \|\mathbf{x}_i\|_1$, from each user. In App. E, we consider more variations.
- C_{median}: Furthermore, we consider a heuristic baseline of clipping user contribution by its median, C_{median} [5].
- `CDP`: We consider the central-DP setting, where there is no federated or secure aggregation constraints on performing frequency estimation. We use the Gaussian mechanism (compared under the same values of ϵ, δ) for quantile estimation and frequency estimation [40].
- `LDP`: We also make comparisons with the local DP model [22, 30]. Again, we use the Gaussian mechanism for privatization but with the noise multiplier set to satisfy local (ϵ, δ)-DP instead.

We set $m = 30$ for Alg. 1, $b = 6$ (2^6 bins), $\theta' = \theta$ and $m' = 3$ for DHT. This means that the privacy budget is split to around 10:1 between performing frequency estimation and quantile estimation. The total privacy spent is $\epsilon = 2.8$ (4) for $d = 50$ ($d = 100$) and $\delta = 10^{-4}$. Table 2 shows the full evaluation results. We see that non-quantile-clipping methods have significantly larger errors, suggesting room for improvement in more advanced applications of FFE presented in, e.g., [6, 15]. Our method, private $\hat{C}$, on the other hand, has error very close to the optimal C^*, basically outperforming all federated baselines (including additional baselines in App. E). Note also that the CDP baseline has at least one order-of-magnitude less error, but it requires different sets of (stronger) trust assumption. Conversely, the LDP model has a much larger error as expected. Our DDP model strikes a balance between these two approaches.

Multi-round Federated Frequency Estimation (MFFE). We also evaluate the performance of our multi-round FFE protocol discussed at the end of Section 4. The baselines in consideration are online quantile estimation but with the target quantile set to match (1) C_{median}, the median, (2) the per-round optimal quantile, C_{small}, i.e., the $1 - \sqrt{\frac{D}{4mN\theta^2}}$-th quantile of $\|\mathbf{y}\|_2$, where N is the per-round number of user (the motivation of this scenario is discussed in App. D). Furthermore, we consider the best-case scenario with the threshold to be the optimal one, C^*, set non-privately by hindsight. We see that our method, private $\hat{C}$, has error very close to the optimal C^*, and outperforms other baselines. See App. D for more details related to the empirical analysis of MFFE.

Additional results. Here, we summarize additional results included in App. E, not included in the main paper due to space constraints. We include detailed results of experiments presented above, considering more variations of datasets, hyperparameters, and other factors. We also explore DP mechanisms other than the proposed cPBM. Additionally, we conduct a detailed study of the multi-round protocol in App. E.1, examining the convergence speed of online quantile estimation, learning rate settings, and related aspects. Finally, we compare the two-phase and multi-round protocols under the same communication and privacy constraints, demonstrating that TFFE generally achieves better privacy-utility trade-offs. Therefore, we recommend using TFFE when practical issues (e.g., user unavailability) do not hinder the aggregation of all user reports in one round.

6 CONCLUSION

In this paper, we have presented end-to-end federated frequency estimation systems, performing threshold optimization and frequency estimation with discrete DP mechanisms, that strictly satisfy user DDP and secure aggregation constraints while achieving nearly minimal measurement error. We have also provided empirical evidence showing that our protocols are practical and effective.

REFERENCES

[1] Naman Agarwal, Peter Kairouz, and Ziyu Liu. 2021. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems* 34 (2021), 5052–5064.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. 2018. cpSGD: Communication-efficient and differentially-private distributed SGD. *Advances in Neural Information Processing Systems* 31 (2018).

[3] Mehrdad Aliasgari, Marina Blanton, Yihua Zhang, and Aaron Steele. 2012. Secure computation on floating point numbers. *Cryptology ePrint Archive* (2012).

[4] Kareem Amin, Alex Kulesza, Andres Munoz, and Sergei Vassilvtiskii. 2019. Bounding user contributions: A bias-variance trade-off in differential privacy. In *International Conference on Machine Learning*. PMLR, 263–271.

[5] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems* 34 (2021), 17455–17466.

[6] Eugene Bagdasaryan, Peter Kairouz, Stefan Mellem, Adrià Gascón, Kallista Bonawitz, Deborah Estrin, and Marco Gruteser. 2021. Towards sparse federated analytics: Location heatmaps under distributed differential privacy with secure aggregation. *arXiv preprint arXiv:2111.02356* (2021).

[7] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. 2020. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2496–2506.

[8] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The privacy blanket of the shuffle model. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer, 638–667.

[9] Ergute Bao, Yizheng Zhu, Xiaokui Xiao, Yin Yang, Beng Chin Ooi, Benjamin Hong Meng Tan, and Khin Mi Mi Aung. 2022. Skellam mixture mechanism: a novel approach to federated learning with differential privacy. *arXiv preprint arXiv:2212.04371* (2022).

[10] James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. 2020. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1253–1269.

[11] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. 2017. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*. 441–459.

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

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

[14] Clément L Canonne, Gautam Kamath, and Thomas Steinke. 2020. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems* 33 (2020), 15676–15688.

[15] Karan Chadha, Junye Chen, John Duchi, Vitaly Feldman, Hanieh Hashemi, Omid Javidbakht, Audra McMillan, and Kunal Talwar. 2023. Differentially Private Heavy Hitter Detection using Federated Analytics. *arXiv preprint arXiv:2307.11749* (2023).

[16] T-H Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)* 14, 3 (2011), 1–24.

[17] Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. 2019. Federated learning of n-gram language models. *arXiv preprint arXiv:1910.03432* (2019).

[18] Wei-Ning Chen, Peter Kairouz, and Ayfer Ozgur. 2020. Breaking the communication-privacy-accuracy trilemma. *Advances in Neural Information Processing Systems* 33 (2020), 3312–3324.

[19] Wei-Ning Chen, Ayfer Ozgur, Graham Cormode, and Akash Bharadwaj. 2023. The communication cost of security and privacy in federated frequency estimation. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4247–4274.

[20] Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. 2022. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning*. PMLR, 3490–3506.

[21] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 375–403.

[22] Graham Cormode, Samuel Maddock, and Carsten Maple. 2021. Frequency estimation under local differential privacy [experiments, analysis and benchmarks]. *arXiv preprint arXiv:2103.16640* (2021).

[23] Graham Cormode and Igor Markov. 2022. Federated calibration and evaluation of binary classifiers. *arXiv preprint arXiv:2210.12526* (2022).

[24] Rachel Cummings, Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Mean estimation with user-level privacy under data heterogeneity. *Advances in Neural Information Processing Systems* 35 (2022), 29139–29151.

[25] Cynthia Dwork. 2006. Differential Privacy, In 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006). 4052, 1–12. https://www.microsoft.com/en-us/research/publication/differential-privacy/

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

[27] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 715–724.

[28] Cynthia Dwork and Guy N Rothblum. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887* (2016).

[29] Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Shanshan Han, Shantanu Sharma, Chaoyang He, Sharad Mehrotra, Salman Avestimehr, et al. 2023. Federated analytics: A survey. *APSIPA Transactions on Signal and Information Processing* 12, 1 (2023).

[30] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.

[31] Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. 2017. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1265–1277.

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

[33] Adria Gascon, Peter Kairouz, Ziteng Sun, and Ananda Theertha Suresh. 2023. Federated heavy hitter recovery under linear sketching. In *International Conference on Machine Learning*. PMLR, 10997–11012.

[34] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

[35] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. 2021. User-level differentially private learning via correlated sampling. *Advances in Neural Information Processing Systems* 34 (2021), 20172–20184.

[36] Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. 2021. Differentially private quantiles. In *International Conference on Machine Learning*. PMLR, 3713–3722.

[37] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford* 1, 12 (2009), 2009.

[38] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2009. Boosting the accuracy of differentially-private histograms through consistency. *arXiv preprint arXiv:0904.0942* (2009).

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

[40] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal mean estimation under differential privacy. *Advances in Neural Information Processing Systems* 34 (2021), 25993–26004.

[41] Jiankai Jin, Eleanor McMurtry, Benjamin IP Rubinstein, and Olga Ohrimenko. 2022. Are we there yet? timing and floating-point attacks on differential privacy systems. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 473–488.

[42] Peter Kairouz, Ziyu Liu, and Thomas Steinke. 2021. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*. PMLR, 5201–5212.

[43] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[44] B Kashin. 1977. Section of some finite-dimensional sets and classes of smooth functions (in russian) izv. *Acad. Nauk. SSSR* 41 (1977), 334–351.

[45] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[46] Tejas Kulkarni. 2019. Answering range queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data*. 1832–1834.

[47] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. 2021. Learning with user-level privacy. *Advances in Neural Information Processing Systems* 34 (2021), 12466–12479.

[48] Seng Pei Liew, Tsubasa Takahashi, Shun Takagi, Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. 2022. Network shuffling: Privacy amplification via random walks. In *Proceedings of the 2022 International Conference on Management of Data*.

773–787.

[49] Yuhan Liu, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Michael Riley. 2020. Learning discrete distributions: user vs item-level privacy. *Advances in Neural Information Processing Systems* 33 (2020), 20965–20976.

[50] Yuhan Liu, Ananda Theertha Suresh, Wennan Zhu, Peter Kairouz, and Marco Gruteser. 2022. Histogram Estimation under User-level Privacy with Heterogeneous Data. *arXiv preprint arXiv:2206.03008* (2022).

[51] Yurii Lyubarskii and Roman Vershynin. 2010. Uncertainty principles and vector quantization. *IEEE Transactions on Information Theory* 56, 7 (2010), 3491–3501.

[52] Mohamad Mansouri, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. 2023. Sok: Secure aggregation based on cryptographic schemes for federated learning. In *PETS 2023, 23rd Privacy Enhancing Technologies Symposium*, Vol. 2023. 140–157.

[53] Luca Melis, George Danezis, and Emiliano De Cristofaro. 2015. Efficient private statistics with succinct sketches. *arXiv preprint arXiv:1508.06110* (2015).

[54] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 691–706.

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

[56] Shyam Narayanan, Vahab Mirrokni, and Hossein Esfandiari. 2022. Tight and robust private mean estimation with few users. In *International Conference on Machine Learning*. PMLR, 16383–16412.

[57] Krishna Pillutla, Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. 2022. Differentially private federated quantiles with the distributed discrete gaussian mechanism. In *International Workshop on Federated Learning: Recent Advances and New Challenges*.

[58] Daniel Ramage and Stefano Mazzocchi. 2020. Federated Analytics: Collaborative Data Science without Data Collection. Blog. https://blog.research.google/2020/05/federated-analytics-collaborative-data.html

[59] Mher Safaryan, Egor Shulgin, and Peter Richtárik. 2022. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA* 11, 2 (2022), 557–580.

[60] Adam Smith. 2011. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 813–822.

[61] Christos Tzamos, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Ilias Zadik. 2020. Optimal private median estimation under minimal distributional assumptions. *Advances in Neural Information Processing Systems* 33 (2020), 3301–3311.

[62] Jingfeng Wu, Wennan Zhu, Peter Kairouz, and Vladimir Braverman. 2023. Private Federated Frequency Estimation: Adapting to the Hardness of the Instance. *arXiv preprint arXiv:2306.09396* (2023).

[63] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2010. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering* 23, 8 (2010), 1200–1214.

[64] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 1–23.

[65] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).

Notation	Definition
n	total number of users, $n \in \mathbb{N}$
d	domain size of histogram, $d \in \mathbb{N}$
$x_i[j]$	number of item j held by user i, $\mathbf{x}_i \in \mathbb{R}^d$
$y_i[k]$	k-th coordinate of user i's representation, $\mathbf{y}_i \in \mathbb{R}^D$
$Z_i[k]$	k-th coordinate of user i's discrete value to be sent to SecSum protocols, $\mathbf{Z} \in \mathbb{Z}_M^D$
D	dimension of Kashin's representation, $D \in \mathbb{N}$
C	clipping threshold, $C \in \mathbb{R}^+$
U	tight frame, $U \in \mathbb{R}^{d \times D}$
m, m'	cPBM parameter, $m, m' \in \mathbb{N}$
θ, θ'	cPBM parameter, $\theta, \theta' \in [0, 1/4]$
B	range of histogram, $B \in \mathbb{R}$
Δ	Bin size, $\Delta \in \mathbb{R}$
b	number of levels in the binary tree, $b \in \mathbb{N}$
r	r-th round of multi-round communication, $r \in \mathbb{N}$
R	number of communication rounds, $R \in \mathbb{N}$
N	number of users per round, $N \in \mathbb{N}$
η	learning rate, $\eta \in \mathbb{R}$

Table 4: Summary of notations

APPENDIX

Table 4 shows the notations used throughout the paper.

A ADDITIONAL RELATED WORK

Industrial application of federated frequency estimation. Google's Pixel phones leverage federated frequency estimation to show users what songs are playing in the room around them [58].

Secure aggregation protocols. We note that there are cryptographic primitives that work with floating-point representation, applicable to common DP mechanisms like Gaussian mechanism [3]. However, they do not scale well for cross-device applications of our interest, involving at least thousands of users. Moreover, these primitives may be vulnerable to floating-point representation attack, increasing attack surfaces [41]. As far as we know, SecAgg is the leading cryptographic solution that is succesfully applied to real-world products, [5] and we henceforth focus on studying it in this paper. We also note that there are also alternative secure technologies applicable to large-scale cross-device applications, such as the use of trusted execution environment and shuffling [8, 11, 21, 32, 48], but they require the use of specialized hardware and face other risks (e.g., side-channel attacks).

User-level privacy. Previous work in the literature studies user-level privacy with assumptions on data distribution [24, 35, 47, 49, 56], except [4, 40, 50], whereas ours does not require such assumptions. [4, 40, 50] however put attention mostly on the central-DP settings, unlike ours, which take DDP constraints and federated settings into account.

[5] https://research.google/blog/distributed-differential-privacy-for-federated-learning/

Applications of Kashin's representation. We note that Kashin's representation has applications in the local DP model, and in the context of communication efficiency [18, 31, 59].

B ALGORITHM FOR COMPUTING THE KASHIN'S REPRESENTATION

The tight frame U is generated as follows: one generates a random orthogonal matrix of size D and selects the first d columns (in `scipy`, it is `scipy.stats.ortho_group.rvs(dim=D).T[:, 0:d]`). To compute the Kashin's representation, we use the algorithm outlined in the original paper [51]:

Input:

- A tight frame $(u_i)_{i=1}^D$ in $\mathbb{R}^d$ which satisfies the uncertainty principle with parameters $\eta, \nu \in (0, 1)$.
- A vector $x \in \mathbb{R}^d$ and a number of iterations r.

Output: Kashin's decomposition of x with level $K = (1-\eta)^{-1}\nu^{-1/2}$ and with accuracy $\eta^r \|x\|_2$. The algorithm looks for coefficients $a_1, \ldots, a_D$ such that

$$\left\| x - \sum_{i=1}^{D} a_i u_i \right\|_2 \le \eta^r \|x\|_2,$$
$$\max_i |a_i| \le \frac{K}{\sqrt{D}} \|x\|_2.$$

Initialize coefficients and truncation level:

$$a_i \leftarrow 0, \quad i = 1, \ldots, D;$$
$$M \leftarrow \frac{\|x\|_2}{\sqrt{\nu D}}.$$

Iterate the following r times:

- Compute the Kashin's representation of x and truncate at level M:

$$b_i \leftarrow \langle x, u_i \rangle, \quad \hat{b}_i \leftarrow t_M(b_i), \quad i = 1, \ldots, N.$$

- Reconstruct and compute the error:

$$Tx \leftarrow \sum_{i=1}^{D} \hat{b}_i u_i; \quad x \leftarrow x - Tx.$$

- Update Kashin's coefficients and the truncation level:

$$a_i \leftarrow a_i + \sqrt{D}\hat{b}_i, \quad i = 1, \ldots, D;$$
$$M \leftarrow \eta M.$$

C ADDITIONAL STATEMENTS AND PROOFS

C.1 RDP-to-DP conversion

Lemma 3 (RDP-to-DP conversion [7, 14]). A mechanism $\mathcal{M}$ satisfying $(\lambda, \epsilon(\lambda))$-RDP also satisfies (ϵ, δ)-DP, where $1 < \delta < 0$ is arbitrary and ϵ is given by

$$\epsilon = \min_{\lambda} \left(\epsilon(\lambda) + \frac{\log(1/\lambda\delta) + (\lambda - 1)\log(1 - 1/\lambda)}{\lambda - 1} \right). \quad (8)$$

C.2 Proof of Lemma 2

Let us reiterate Lemma 2 below.

Lemma 4 (cPBM RDP bounds)**.** The SecSum of the cPBM outputs of n users is differentially private, and its RDP of order λ bounded by

$$\epsilon(\lambda) \le \frac{md}{\lambda - 1} \log \max\left(\Pi_1, \Pi_2\right),$$

where

$$\Pi_1 = \sum_{m=0}^{n'+1} \mathrm{Bin}(m, n'+1, \frac{1}{2}+\theta) e^{\lambda \xi_m},$$

$$\xi_m = \log\left(\frac{1+2\theta}{1-2\theta}\frac{n'+1-m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{m}{n'+1}\right),$$

$$\Pi_2 = \sum_{m=0}^{n'+1} \mathrm{Bin}(m, n', \frac{1}{2}-\theta)\left(\frac{1}{2}+\theta\right)\left(\frac{m}{n'-m+1}\frac{1+2\theta}{1-2\theta} + \frac{1-2\theta}{1+2\theta}\right) e^{-\lambda \mu_m},$$

$$\mu_m = \log\left(\frac{1+2\theta}{1-2\theta}\frac{m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{n'+1-m}{n'+1}\right),$$

and $n' = \lceil \frac{n-1}{2} \rceil$, $\mathrm{Bin}(m, n, p) = \binom{n}{m} p^m (1-p)^{n-m}$.

Proof. Since each coordinate of $\mathbf{x}$ has bounded norm, the overall RDP can be obtained by composing the one-dimensional (scalar) case of cPBM by d times.

For the scalar case, Lemma 3.6 of [20] tells us that the RDP of PBM is equivalent to calculating the maximum of the following two divergences:

(1) $D_\lambda\left(P_{\mathrm{Ber}(\frac{1}{2}-\theta)+\mathrm{Binom}(n', \frac{1}{2}+\theta)} || P_{\mathrm{Binom}(n'+1, \frac{1}{2}+\theta)}\right)$

(2) $D_\lambda\left(P_{\mathrm{Binom}(n'+1, \frac{1}{2}-\theta)} || P_{\mathrm{Ber}(\frac{1}{2}+\theta)+\mathrm{Binom}(n', \frac{1}{2}-\theta)}\right)$

We give the explicit expression of these below. Note that for both the divergences, they are a summation of the event-level divergence over m from 0 to $n'+1$, $\sum_{m=0}^{n'+1} \frac{P(m)^\lambda}{Q(m)^{\lambda-1}}$. Fixing m, we expand the Bernoulli term to evaluate the divergences explicitly. We also write $\mathrm{Bin}(m, n, p) = \binom{n}{m} p^m (1-p)^{n-m}$.

First term. Since

$$P(m) = \left(\frac{1}{2}-\theta\right)\mathrm{Bin}(m-1, n', \frac{1}{2}+\theta) + \left(\frac{1}{2}+\theta\right)\mathrm{Bin}(m, n', \frac{1}{2}+\theta),$$

$$Q(m) = \mathrm{Bin}(m, n'+1, \frac{1}{2}+\theta),$$

and

$$\frac{\mathrm{Bin}(m-1, n', \frac{1}{2}+\theta)}{\mathrm{Bin}(m, n'+1, \frac{1}{2}+\theta)} = \frac{m}{n'+1}\frac{1}{\frac{1}{2}+\theta},$$

$$\frac{\mathrm{Bin}(m, n', \frac{1}{2}+\theta)}{\mathrm{Bin}(m, n'+1, \frac{1}{2}+\theta)} = \frac{n'+1-m}{n'+1}\frac{1}{\frac{1}{2}-\theta},$$

rearranging the terms, we get

$$\left(\frac{P(m)}{Q(m)}\right)^\lambda = \left(\frac{1+2\theta}{1-2\theta}\frac{n'+1-m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{m}{n'+1}\right)^\lambda =: e^{\lambda \xi_m}$$

This produces the expression of Π_1 in Lemma 2 summing up m from 0 to $n'+1$.

Second term. Similarly,

$$\frac{P(m)}{Q(m)} = \frac{\mathrm{Bin}(m, n'+1, \frac{1}{2}-\theta))}{(\frac{1}{2}+\theta)\mathrm{Bin}(m-1, n', \frac{1}{2}-\theta) + (\frac{1}{2}-\theta)\mathrm{Bin}(m, n', \frac{1}{2}-\theta)}$$

$$= \frac{\binom{n'+1}{m}(\frac{1}{2}-\theta)^m(\frac{1}{2}+\theta)^{n'+1-m}}{(\frac{1}{2}+\theta)\binom{n'}{m-1}(\frac{1}{2}-\theta)^{m-1}(\frac{1}{2}+\theta)^{n'+1-m} + (\frac{1}{2}-\theta)\binom{n'}{m}(\frac{1}{2}-\theta)^m(\frac{1}{2}+\theta)^{n'-m}}$$

$$= \frac{1}{\frac{m}{n'+1}\frac{1}{\frac{1}{2}-\theta}(\frac{1}{2}+\theta) + \frac{n'+1-m}{n'+1}\frac{1}{\frac{1}{2}+\theta}(\frac{1}{2}-\theta)}$$

$$= \frac{1}{\frac{1+2\theta}{1-2\theta}\frac{m}{n'+1} + \frac{1-2\theta}{1+2\theta}\frac{n'+1-m}{n'+1}}$$

$$=: e^{-\mu_m}$$

This yields the μ term of Π_2 in Lemma 2 . Noting that

$$Q(m) = \left(\frac{1}{2}+\theta\right)\mathrm{Bin}(m-1, n', \frac{1}{2}-\theta) + \left(\frac{1}{2}-\theta\right)\mathrm{Bin}(m, n', \frac{1}{2}-\theta)$$

$$= \mathrm{Bin}(m, n', \frac{1}{2}-\theta)\left(\frac{1}{2}+\theta\right)\left(\frac{m}{n'-m+1}\frac{\frac{1}{2}+\theta}{\frac{1}{2}-\theta} + \frac{\frac{1}{2}-\theta}{\frac{1}{2}+\theta}\right)$$

$$= \mathrm{Bin}(m, n', \frac{1}{2}-\theta)\left(\frac{1}{2}+\theta\right)\left(\frac{m}{n'-m+1}\frac{1+2\theta}{1-2\theta} + \frac{1-2\theta}{1+2\theta}\right),$$

we obtain Π_2 combining the above two expressions and summing up m from 0 to $n'+1$. ∎

C.3 Proof of Theorem 1

First, we write $\mathbf{x}$'s in its Kashin's representation. From Equation 1 and the definition of Kashin's representation,

$$\|\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}\|_2^2 = \sum_{j=1}^{d}\left(\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k=1}^{D} U_{jk}(y_i[k] - \hat{y}_i[k])\right)\right)^2$$

$$= \sum_{j=1}^{d}\left(\sum_{k=1}^{D} U_{jk}(y_\cdot[k] - \hat{y}_\cdot[k])\right)^2$$

Here, $y_\cdot[k] \coloneqq \frac{1}{n}\sum_{i=1}^{n} y_i[k]$. Using Cauchy–Schwarz inequality,

$$\sum_{j=1}^{d}\left(\sum_{k=1}^{D} U_{jk}(y_\cdot[k] - \hat{y}_\cdot[k])\right)^2 \le \sum_{k'=1}^{D}\sum_{j=1}^{d} U_{jk'}^2 \sum_{k=1}^{D}(y_\cdot[k] - \hat{y}_\cdot[k])^2$$

$$= d\|y_\cdot - \hat{y}_\cdot\|_2^2$$

due to the orthogonal properties of U. This shows that up to a multiplicative constant, the expected l_2-error of $\mathbf{x}$ is equivalent to that of y. We henceforth analyze the expected l_2-error in terms of the Kashin's representation, i.e., $\|y_\cdot - \hat{y}_\cdot\|_2^2$.

By Jensen's inequality,

$$\mathbb{E}\|\mathbf{y}_\cdot - \hat{\mathbf{y}}_\cdot\|_2 \le \sqrt{\mathbb{E}\sum_{k=1}^{D}(y_\cdot[k] - \hat{y}_\cdot[k])^2}$$

By noting that

$$(\mathbb{E}\left[\hat{y}_\cdot[k]\right] - y_\cdot[k])^2 = (\mathbb{E}\left[\hat{y}_\cdot[k]\right])^2 - 2\mathbb{E}\left[\hat{y}_\cdot[k]\right] y_\cdot[k] + y_\cdot[k]^2,$$

and that $\mathrm{Var}(\hat{y}.[k]) = \mathbb{E}(\hat{y}.[k]^2) - (\mathbb{E}[\hat{y}.[k]])^2$, we perform bias-variance decomposition as follows.

$$\sqrt{\mathbb{E}\sum_{k=1}^{D}(\mathbf{y}.[k]-\hat{\mathbf{y}}.[k])^2} = \sqrt{\|\mathbb{E}[\hat{\mathbf{y}}.]-\mathbf{y}.\|_2^2 + D\cdot\mathrm{Var}(\hat{y}.)}$$
$$\leq \|\mathbb{E}[\hat{\mathbf{y}}.]-\mathbf{y}.\|_2 + \sqrt{D\cdot\mathrm{Var}(\hat{y}.)}$$

where $\mathrm{Var}(\hat{y}.) \leq \frac{C^2}{4mn\theta^2}$. Note that

$$\begin{aligned}\|\mathbb{E}[\hat{\mathbf{y}}.]-\mathbf{y}.\|_2 &= \frac{1}{n}\Big\|\sum_{i:\|\mathbf{y}_i\|_2>C}\left(\mathbf{y}_i - \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|_2}\right)\Big\|_2\\ &= \frac{1}{n}\sqrt{\sum_{i:\|\mathbf{y}_i\|_2>C}\sum_{k=1}^{D} y_i^2[k](1-\frac{C}{\|\mathbf{y}_i\|_2})^2}\\ &= \frac{1}{n}\sqrt{\sum_{i:\|\mathbf{y}_i\|_2>C}(\|\mathbf{y}_i\|_2 - C)^2}\\ &\leq \frac{1}{n}\sum_{i=1}^{n}\max(0,\|\mathbf{y}_i\|_2 - C),\end{aligned}$$

using the fact that l_2 distance is always smaller than l_1 distance. Summarizing, we have $\mathbb{E}\|\bar{\mathbf{x}}-\hat{\mathbf{x}}\|_2 \leq d\cdot\|\mathbf{y}.-\hat{\mathbf{y}}.\|_2 \leq \frac{1}{n}\sum_{i=1}^{n}\max(0,\|\mathbf{y}_i\|_2 - C) + C\sqrt{\frac{D}{4mn\theta^2}}$, as required.

C.4 Proof of Proposition 1

For each bin, the variance is $1/4mn^2\theta^2$ according to the argument given in Section 3.2. By summing up all bins, B/Δ in total, we obtain the desired variance.

C.5 Proof of Proposition 2

Any quantile can be determined using at most $\log_2(B/\Delta)$ nodes [16, 27]. The conclusion follows from the argument presented in the proof of Proposition 1.

C.6 Proof of Proposition 3

Notice that at each level of the binary tree, the range can only cut through one of the nodes. Moreover, at level l, $|s_L[w] - s_R[w]|$ can only be at most 2^{l-1}. the variance of the quantile $\hat{\gamma} := \frac{\hat{c}[1:v]}{n}$ is then upper-bounded by (slightly abusing notations by writing $\hat{H}_l[w]$ as $\hat{H}_l$)

$$\mathrm{Var}(\frac{\hat{c}[1:v]}{n}) \leq \sum_l \frac{2^{2l-2}}{2^{2l}}\mathrm{Var}(\frac{\hat{H}_l}{n}) = \sum_l \frac{1}{4}\mathrm{Var}(\frac{\hat{H}_l}{n})$$

The variance of $\frac{\hat{H}_l}{n}$ applied with PBM with parameters θ and m is bounded by $\frac{1}{4mn^2\theta^2}$. Writing the discretization error as $\Delta =: B/2^b$, we have $\mathrm{Var}(\hat{\gamma}) \leq \frac{\log_2(B/\Delta)}{16mn^2\theta^2}$.

C.7 Proof of Proposition 4

The underlying DP mechansim for both federated quantile and frequency estimation is cPBM, which satisfies RDP as described in Section 3, with one difference in federated quantile estimation: as we apply cPBM independently to $2^b - 1$ Haar coefficients, the RDP of quantile estimation equals to those given in Lemma 2 with d set to $2^b - 1$ instead. By RDP composition and Lemma 2, we obtain the required statement.

D MULTI-ROUND FEDERATED FREQUENCY ESTIMATION

Suppose there are R rounds of user-server communication. Let N be the number of users per round, and assume that each user only reports once in the whole cycle of FFE. Then, we have that the product $N \cdot R$ equals n, the population. Let $x^{(r)}$ be the user data contributing to FFE, and $\hat{x}^{(r)}$ be the estimated frequency at round $r \in [R]$. Our target is to estimate $\hat{x} = \frac{1}{R}\sum_{r=1}^{R}\hat{x}^{(r)}$ with minimum expected l_2-error.

Since at each round, the number of users is now N instead of n, one may wonder if we should modify the optimal quantile in Equation 5 to $1-\sqrt{\frac{D}{4mN\theta^2}}$-th quantile. In fact, the optimal quantile is still $1-\sqrt{\frac{D}{4mn\theta^2}}$, under mild assumptions, as shown below.

Corollary 1. Let $\hat{x}^{(r)}$ be the estimated frequency at round r. Assume that users sending reports each round are i.i.d. samples, i.e., $\mathbb{E}\left[\frac{1}{R}\sum_{r=1}^{R}\frac{1}{N}\sum_{i=1}^{N}\hat{x}_i^{(r)}\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\hat{x}_i^{(r)}\right] = \frac{1}{n}\sum_{i=1}^{n}\hat{x}_i$. Setting the threshold to $1-\sqrt{\frac{D}{4mn\theta^2}}$ at each round to estimate $\hat{x}_r$ minimizes the upper bound of the expected l_2-error as in Equation 4.

Proof. In the multi-round setting, the upper bound given in Equation 4 stays the same except with the single-round estimator $\hat{\mathbf{x}}$ replaced by the multi-round estimator, $\frac{1}{R}\sum_{r=1}^{R}\hat{\mathbf{x}}^{(r)}$. Let $\mathbf{y}_i^r$ be the Kashin's representation of user i at round r. By repeating the argument of the proof of Theorem 1, we arrive at

$$\begin{aligned}\mathbb{E}\|\bar{\mathbf{x}}-\hat{\mathbf{x}}\|_2 &\leq \frac{1}{R}\sum_{r=1}^{R}\frac{1}{N}\sum_{i=1}^{N}\max(0,\|\mathbf{y}_i^r\|_2 - C) + C\sqrt{\frac{D}{4mn\theta^2}}\\ &\approx \frac{1}{N}\sum_{i=1}^{N}\max(0,\|\mathbf{y}_i^{(r)}\|_2 - C) + C\sqrt{\frac{D}{4mn\theta^2}}\end{aligned}$$

for all r using the i.i.d. assumption. Similar to Equation 5, we optimize the threshold to obtain

$$0 = -\frac{1}{n}\sum_{i:\|y_i^{(r)}\|_2>C} 1 + \sqrt{\frac{D}{4mn\theta^2}}$$

Hence, setting the threshold such that the quantile is $1-\sqrt{\frac{D}{4mn\theta^2}}$ at each round minimizes the upper bound of the expected l_2-error. ■

Online quantile estimation. Next, we consider how to estimate the optimal threshold in an online and private fashion. We take inspirations from [5] by requiring the users to send extra information (privately) about the threshold along with the usual frequencies at each round. Algorithm 7 outlines our protocol, MFFE.

Strategy advantages. This strategy has several attractive properties:

- efficiency: user participates only once as the threshold is estimated along with frequency measurement in the same round and only a few extra bits are required.
- The online estimation problem is convex and the geometric update guarantees convergence [5].

Algorithm 7 Multi-round FFE protocol (MFFE)

Inputs: Data x_i. Target quantile γ^*. Tight frame U. Parameters $R, N, \eta, \theta, \theta', m, m', C$
Output: Estimated frequencies
$C^{(1)} \leftarrow C$
For $r \in [R]$:
  User-side:
    Step 1: Run OcPBM on $x_i^{(r)}$ with $\theta = \theta, m = m, C = C^{(r)}$ and SecSum
    Step 2: Run cPBM with parameters $\theta = \theta', m = m', C = 1$ on $y_i^{(r)} = \mathbb{I}_{\|y_i^{(r)}\|_2 > C^{(r)}}$, and SecSum
  Server-side: Update $\hat{x} \leftarrow \frac{1}{R}\hat{x}^{(r)}, C^{(r+1)} \leftarrow C^{(r)} \exp(-\eta(\hat{\gamma}^{(r)} - \gamma^*))$, and send $C^{(r+1)}$ to users
Return: $\hat{x}$

- The update rule allows quick convergence to the true quantile even if the initial values are quite different.

Overall DP guarantees. Note that each user contributes her data only once during the whole FFE process. Thus, we can think of R-round FFE as splitting the whole datasets to R disjoint chunks and applying ϵ-RDP mechanism to each of them separately. Parallel composition tells us that the overall RDP guaranteed is still ϵ. Therefore, privacy accounting can be done as follows. At any round r, the online quantile estimation method introduced above has RDP guarantees equal to those given in Lemma 2 with d set to 1. This is to be composed with the FFE's RDP at the same round to obtain the overall DP guarantees at round r, which is also the DP guarantees for the *whole* FFE process by parallel composition.[6]

E MORE EXPERIMENTAL RESULTS

Verifying Theorem 1. In Figure 6, we plot the upper bound derived from Theorem 1 and the true l_2-error of our proposed algorithms for all four datasets, varying the threshold, C. It can be seen that our theoretical upper bounds indeed upper-bound the true errors in all cases.

Federated quantile estimation. In Figure 7, we compare algorithms of federated quantile estimation for four datasets. It is clear that our proposed DHT protocol has over better performance than the flat-histogram and HH approaches. One can also see that without further prior knowledge on the range of $\|\mathbf{y}\|_2$, our choice of B is conservative, as the quantile reaches 1 at $\|\mathbf{y}\|_2$ much smaller than the chosen B, i.e., the maximum possible value of $\|\mathbf{y}\|_2$. Even so, we can still obtain pretty good estimates of the quantile as shown in the Figure.

To make a more quantitative comparison, we vary the quantile in the range $[0.8, 0.81, ..., 0.99]$ (which is quantiles of interest in this work, i.e., Equation 5), and estimate the corresponding DP threshold. Then, we measure the average "bin error": we calculate the absolute difference between the estimated one and those obtained from the un-noised histogram, divide by Δ, the bin size, and take the average over all the quantile queries (this corresponds to average error in the unit of bin size) to produce Table 5. We can see that DHT performs better than the flat approach for all datasets with lower average bin errors.

[6] Alternatively, we could consider privacy amplification via random subsampling to enhance the privacy guarantees, instead of performing parallel composition as above, though this may not be preferable in real-world cross-device situations where user is not always available to be sampled randomly [43]. For this reason, we opt for the parallel composition approach in this study.

	i.i.d.	hetero	Foursquare	Sentiment140
DHT+DDP	2.0	0.16	2.36	0.86
Flat	4.66	6.84	11.68	2.7

Table 5: Comparison of federated quantile estimation approaches. Shown are the average bin error of estimated quantiles. Here, the parameters are $d = 100, \theta' = 0.2, m' = 30, b = 6$. See text for other details.

Estimated frequencies. In Figure 11, we plot the true per-user frequencies versus the estimated per-user frequencies with various algorithms for the Foursquare dataset with the two-round FFE protocol. As argued in the main text, the `Normalize` and `Sample` have significantly larger l_2-errors compared to quantile-based methods. There is no huge visible differences among the quantile-based methods.

Additional baselines. We consider sampling at most 4, 16,..., 256 items from each user, as a baseline for bounding user contribution. [7] The result is shown in Table 6, outperformed by our proposals in the main text.

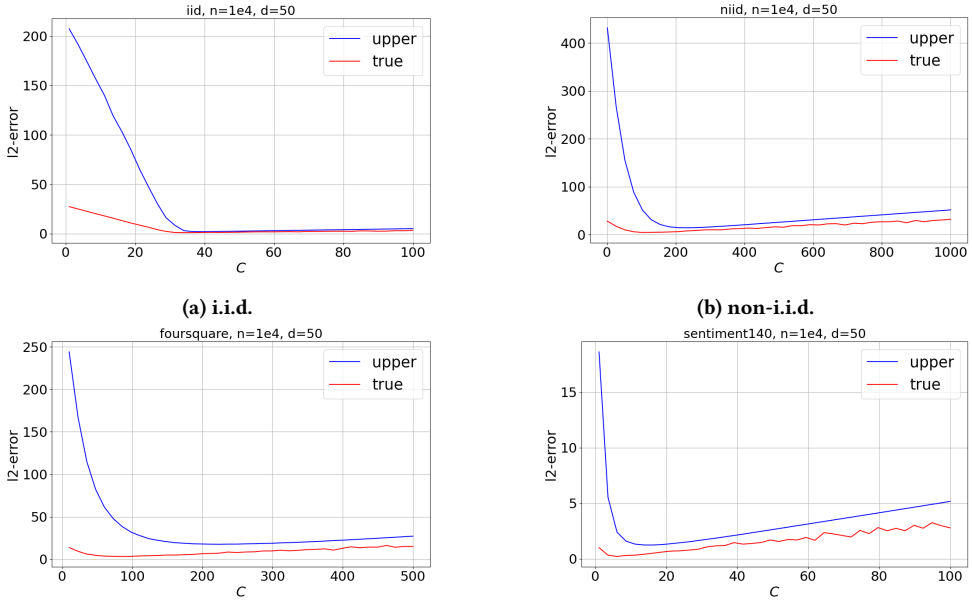
DP mechanisms other than cPBM. Lastly, we briefly study the performance of using alternatives of cPBM as the underlying DP mechanisms. We utilize our cPBM instead of e.g. Discrete Gaussian (DDG), as the underlying DP mechanism. The reason is, these approaches has infinite range of noise, leading to overflows, have other disadvantages such as increased communication costs at low privacy budget, as outlined in [20]. Also, as mentioned in Section 3, they have complicated relations between clipping and bias that cannot be optimized to obtain near-minimal error as done in Theorem 1. To see this, we study one of the state-of-the-art alternative, Skellam Mixture mechanism [9] by adapting it to our study presented in Table 2. For example, for i.i.d. dataset ($d = 50$), we obtain an error of 26.9 (ours is 1.16) using their default clipping thresholds.

E.1 Multi-round Federated Frequency Estimation

Online quantile estimation. We first study how to set the learning rate η and the initial values of C. Notice that the geometric update has an exponential dependence on η multiplied by $\hat{\gamma}^{(r)} - \gamma^*$ (Algorithm 7), of which the absolute value in expectation is less than 1. We set η to be of the same order, i.e., 1. For initial C, the initial values do not affect its convergence to C^* albeit the speed of convergence may differ. Setting C to $O(1)$ achieves reasonably fast convergence (see Figure 8 for plots). In our full experiments, we set $C = \eta = 1$ without more refined tuning.

Experimental details of MFFE. We consider $n = 10^5$, both $d = 50$ and $d = 100$. The parameters set for performing Algorithm 1 is $m = 15$. The parameters of performing online quantile estimation is

[7] After sampling, we use cPBM to encode the per-user histogram's Kashin's representation. User contribution is bounded as the l_1-norm is bounded after sampling, and l_2-norm, relevant quantity when projecting to the representation space, is always less than or equal to l_1-norm.



(a) i.i.d. **(b) non-i.i.d.**

(c) Foursquare. **(d) Sentiment140.**

Figure 6: Upper bound of expected l_2-error (blue) versus the true error plotted against C.

Dataset	d	Sample(4)	Sample(16)	Sample(64)	Sample(256)
i.i.d.	50	28.18 ± 0.0	27.5 ± 0.01	28.29 ± 0.03	28.28 ± 0.1
	100	24.55 ± 0.0	24.16 ± 0.01	24.63 ± 0.03	24.66 ± 0.1
hetero	50	28.43 ± 0.0	28.62 ± 0.02	28.63 ± 0.04	28.71 ± 0.12
	100	24.72 ± 0.0	24.85 ± 0.01	24.83 ± 0.02	24.89 ± 0.07
Foursquare	50	17.73 ± 0.0	17.48 ± 0.01	17.88 ± 0.05	17.98 ± 0.15
	100	18.02 ± 0.0	17.58 ± 0.01	18.13 ± 0.02	18.1 ± 0.21
Sentiment140	50	1.32 ± 0.0	1.35 ± 0.02	1.44 ± 0.03	1.98 ± 0.06
	100	1.38 ± 0.0	1.39 ± 0.01	1.49 ± 0.03	2.02 ± 0.09

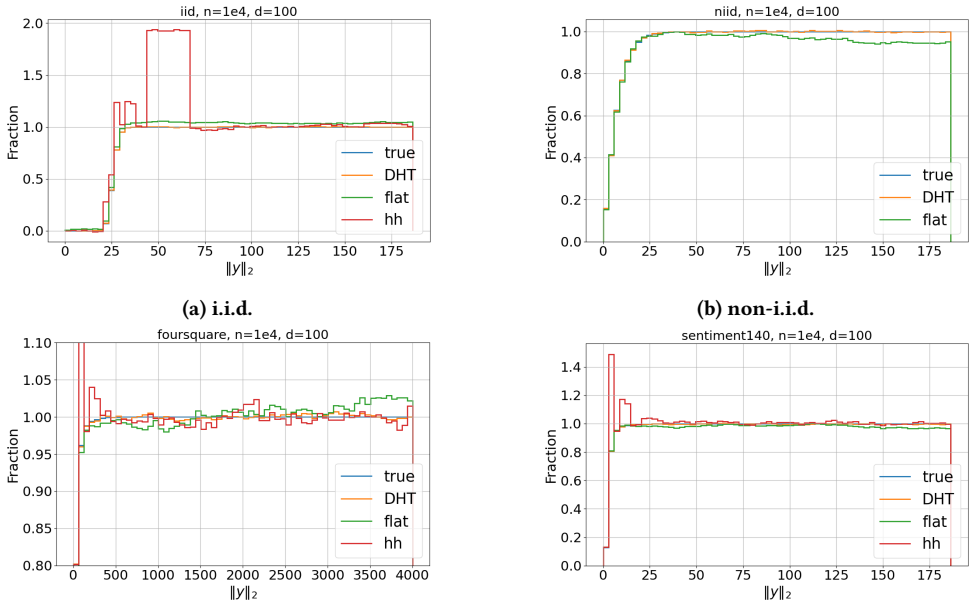
Table 6: l_2-errors of two-round federated frequency estimation. Mean (standard deviation) over 5 runs.

set to be $\theta' = \theta$ and $m' = 15$. Thus, we are paying a privacy of $1/d$-factor of those used for frequency estimation on online quantile estimation. The total privacy spent (under parallel composition assumption) is $\epsilon = 7.5$ (10.7) for $d = 50$ ($d = 100$) and $\delta = 10^{-5}$. Table 3 shows the full results. We see that our method, private $\hat{C}$, has error very close to the optimal C^*, and outperforms other baselines. Comparison with C_{small} also validates Corollary 1. In Figure 10, we show that similar results hold for larger d's.

TFFE versus MFFE. We discuss in what situation a practitioner should deploy TFFE/MFFE when both options are viable in practice. The answer is one should always perform TFFE. There are several reasons for this. (a) Security-wise, TFFE is preferable as running SecAgg fewer times reduces privacy leakage risks, e.g., eavesdropping. (b) Algorithm-wise, MFFE makes imprecise frequency estimation in early rounds of aggregation when the estimated threshold has not converged yet. (c) In terms of privacy-utility trade-offs, TFFE is also favorable. Recall that the RDP is proportional to the inverse of the number of aggregated users in a round. The TFFE benefits maximally privacy-wise from aggregating the whole population, while the MFFE does not receive similar boost of privacy due to smaller per-round number of users.

To illustrate this, we conduct the following experiment. For MFFE, we fix $N = 1000$ and other parameters set to be equal to the MFFE experiments presented previously, and vary the total population, n (and hence the number of rounds, R). Then, we measure the l_2-error. For TFFE, we vary n and adjust m, m' such that the both MFFE and TFFE are compared under the same privacy budget. In addition, we consider the communication cost of both protocols. Secure multi-party computation of [10] requires $O(\log N)$ ($O(\log n)$) communication overhead per user, in addition to the communication overhead of the MFFE (TFFE) protocol for communicating user vectors. We calculate the TFFE/MFFE ratio of communication overhead for different values of n.

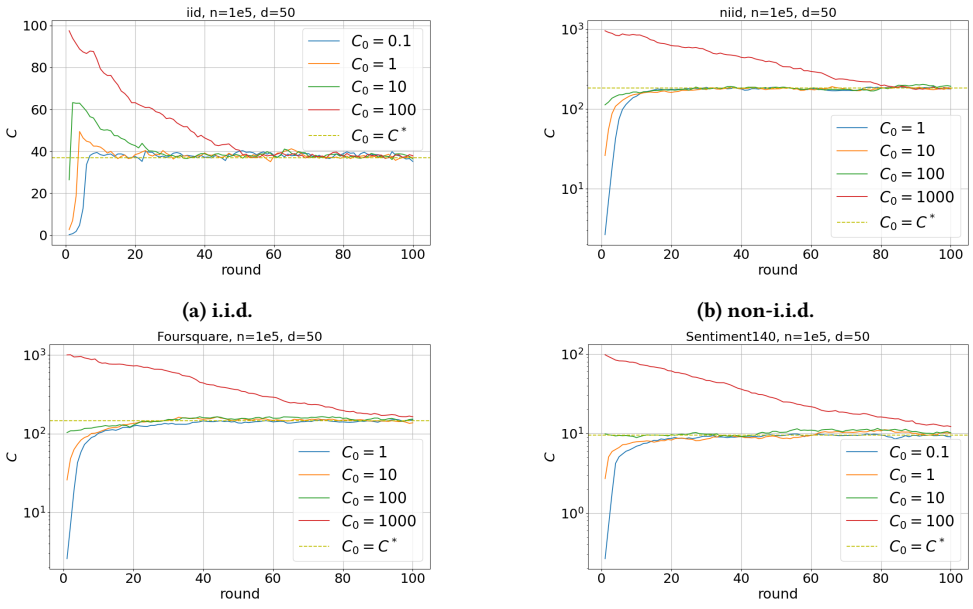
Figure 12a show the results, where TFFE is always better than MFFE. However, the communication overhead of TFFE also increases with n, due to larger number of bits sent, m, m' (to compensate the increase of ϵ at larger values of n). To see how TFFE



(a) i.i.d. **(b) non-i.i.d.**

(c) Foursquare. **(d) Sentiment140.**

Figure 7: Federated quantile estimation.

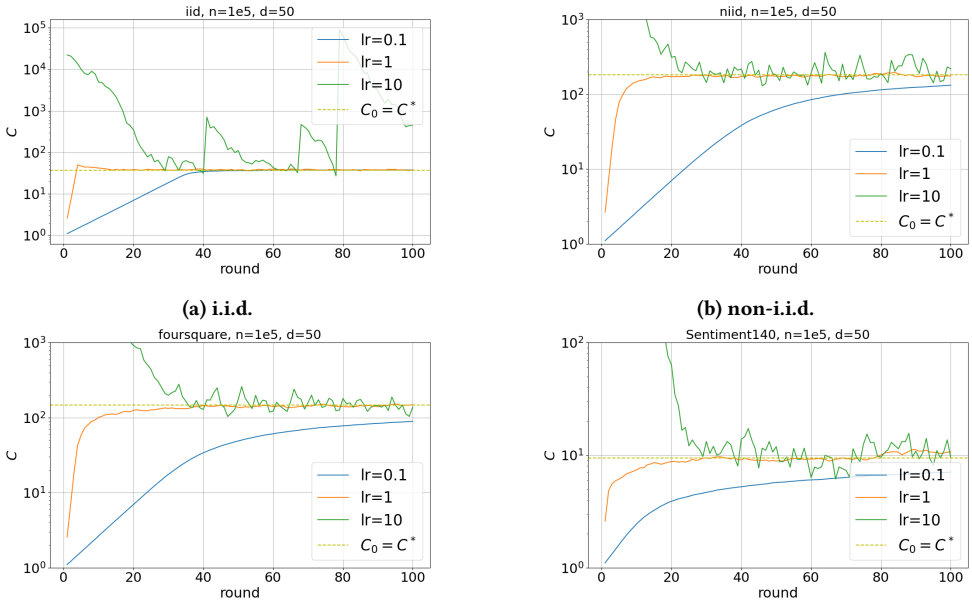


(a) i.i.d. **(b) non-i.i.d.**

(c) Foursquare. **(d) Sentiment140.**

Figure 8: Evolution of C for various values of initial C (learning rate set to 1).

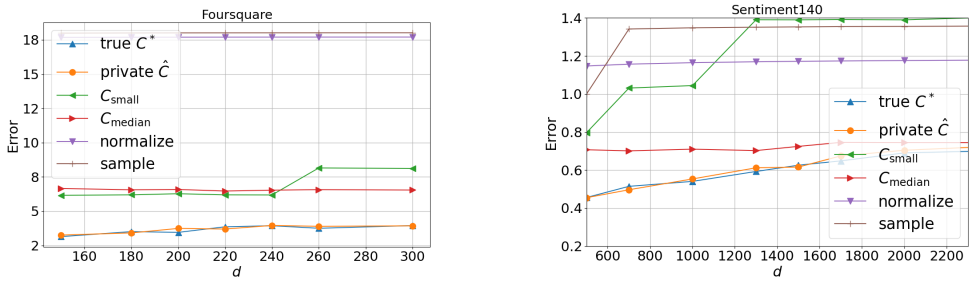
and MFFE peforms under the same communication constraints, we perform another experiment with TFFE, adjusting $m = 15, m' = 3$ such that both TFFE and MFFE send roughly the same amount of bits. From Figure 12b, we can see that TFFE and MFFE perform similarly. Additionally, we note that TFFE requires smaller privacy budget to achieve the same utility as MFFE (MFFE reqiuires $\epsilon = 7.5$), demonstrating better privacy-utility trade-offs. In summary, we recommend the deployment of TFFE whenever the situation allows.



(a) i.i.d. **(b) non-i.i.d.**

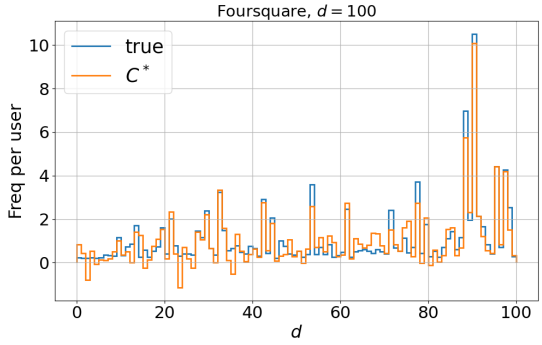
(c) Foursquare. **(d) Sentiment140.**

Figure 9: Evolution of C for various values of learning rate (initial C set to 1).

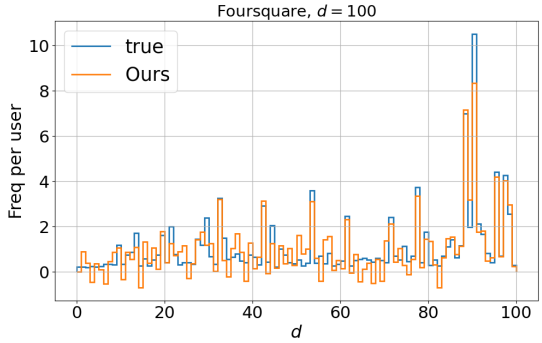


(a) Foursquare. **(b) Sentiment140.**

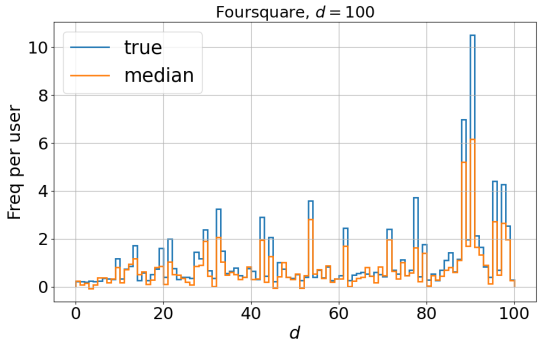
Figure 10: Comparisons of various algorithms varying d for multi-round FFE.



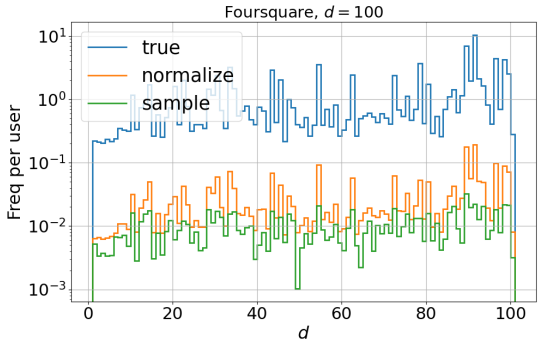
(a) True C^*



(b) Private $\hat{C}$

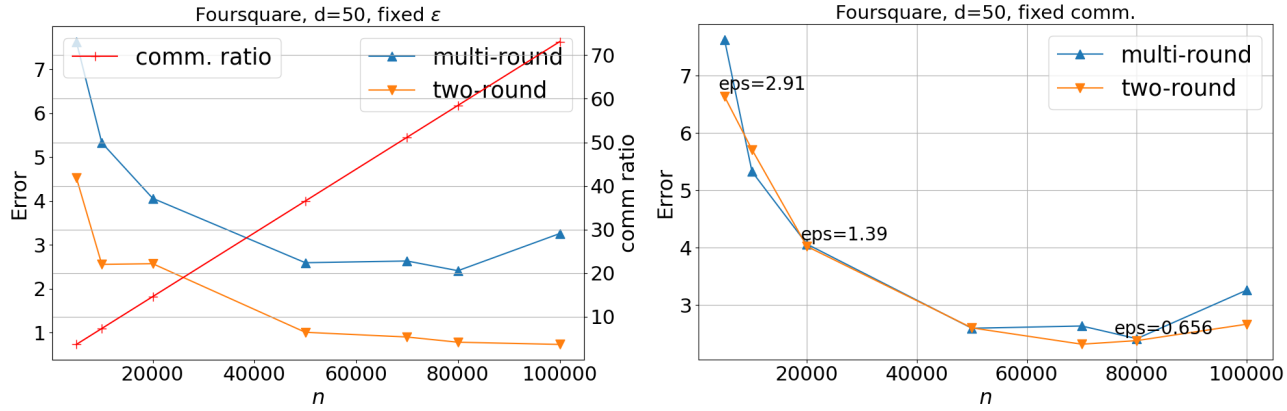


(c) C_{median}



(d)

Figure 11: Average per-user frequencies for Foursquare dataset.



(a) (b)

Figure 12: Comparing two-round/multi-round federated frequency estimation, fixing (a) privacy budget, (b) communication cost. See text for details.