Further Study on Frequency Estimation under Local Differential Privacy

Huiyu Fang
Southeast University
Nick_seu@hotmail.com

Liquan Chen✉
Southeast University
lqchen@seu.edu.cn

Suhui Liu
Southeast University
suhuiliu@seu.edu.cn

Abstract

Local Differential Privacy (LDP) protects user privacy while collecting user data without the need for a trusted data collector. Nowadays, LDP protocols have been adopted and deployed by several major technology companies. A basic building block of LDP protocols is the frequency protocol, which estimates the frequency of each value in a specified domain. Although several frequency protocols have been proposed, all these protocols make compromises among the performances of accuracy, computation cost, and communication cost. In this paper, we introduce a precise and convenient equation to evaluate the accuracy of frequency protocols. We use it to analyze the advantages and disadvantages of existing protocols quantitatively. Based on the analysis, we address the shortcomings of these protocols and propose a new protocol, Random Wheel Spinner (RWS), which achieves optimal accuracy with low computation and communication costs simultaneously. Extensive experiments on both synthetic and real-world datasets demonstrate the advantages of our proposed protocols.

1 Introduction

In recent years, a large amount of data has been collected and analyzed to aid in decision-making and service improvement. However, in many cases, these data are obtained from end-user devices and should be considered highly sensitive, describing the behaviors, preferences, or characteristics of individuals. To preserve the privacy of users while collecting data, Local Differential Privacy (LDP) has been proposed and grown to be one of the de facto standards for preserving privacy. In the LDP setting, users perturb their data locally and only report the perturbed data to the data collector. Since the raw data never leaves end-user devices, LDP enables collecting and analyzing data from users while preserving the privacy of users with "plausible deniability". Nowadays, LDP protocols have been adopted and deployed by several major technology companies, including Google [8], Apple [17], Microsoft [7], etc. Examples of usage include collecting web settings to help identify malicious hijacking; collecting commonly used emojis and phrases to enhance typing suggestions; or collecting telemetry data to improve the user experiences.

Frequency estimation is a fundamental task in LDP protocols and aims to learn the frequency of each value across all users. Numerous frequency estimation protocols [1,3,8,12,17, 20,27] have been proposed where data collectors can estimate the frequency of values in a specified domain. These protocols are designed to minimize the estimated error, computation cost, or communication cost to provide unbiased estimates of individual frequencies. However, none of them achieve perfect performance in all respects and situations, and their drawbacks are not negligible in specific cases.

To propose a protocol with the least compromise, we first introduce a new universal and convenient equation to evaluate the analytical mean squared error (MSE) of frequency protocols, which is based on the pure framework [20] to assess the accuracy more precisely across all frequencies and reveal the connection between the variance and the domain size. This equation also helps us to choose the optimal parameters when rounding is required. We then quantitatively analyze the advantages and disadvantages of four state-of-the-art frequency protocols in terms of accuracy, computation cost, and communication cost. Table 2 summarizes the analysis.

According to the analysis, we find that Optimized Unary Encoding (OUE) [20] and Optimized Local Hashing (OLH) [20] do not achieve optimal accuracy performance among UE and LH protocols, especially for the small domains. Although OLH reduces the communication cost compared with OUE, its computation cost is quite high. Subset Selection (SS) [15, 18, 27] has the optimal accuracy but its communication cost is linear with domain size d, which is unacceptable for large domains.

Based on the proposed analytical MSE, we optimize the accuracy of OUE and OLH for the small domain. With domain size $d = 2$ and privacy budget $\varepsilon = 4$, the optimized MSE is reduced to about $1/3$ of the original. We replace the hash function with a reproducible randomization process in the

LH protocols, thus reducing the number of function calls and the computation cost in aggregation from $O(nd\log d)$ to $O(nd)$. With $d = 4096$, the optimized aggregation time is only about $1/91$ of OLH. We fuse the key ideas of LH and SS to propose the Random Wheel Spinner (RWS) protocol which achieves optimal accuracy with low computation and communication costs whether the domain size d is large or small. The communication cost of RWS is $O(\log n)$, which is much lower than $O(d)$ of SS when the domain size is large.

To summarize, this paper makes the following contributions:

- We introduce a universal and convenient equation to evaluate the analytical MSE of frequency protocols, which can assess the accuracy more precisely across all frequencies and reveal the connection between the variance and the domain size. This equation also helps us to choose the optimal parameters when rounding is required.
- We quantitatively analyze the advantages and disadvantages of four state-of-the-art frequency protocols in terms of accuracy, computation cost, and communication cost.
- We partially address the shortcomings of existing protocols and propose a novel Random Wheel Spinner (RWS) protocol which achieves optimal accuracy with low computation and communication costs.
- We conduct experiments on both synthetic and real-world datasets. Experimental results demonstrate the advantages of our proposed protocols.

Roadmap. In Section 2, we provide the preliminaries on LDP frequency protocols and introduce a new equation to evaluate the analytical MSE of frequency protocols. We then apply this equation to study existing frequency protocols in Section 3. Existing protocols are optimized, and a novel RWS protocol is proposed in Section 4. Experimental results are shown in Section 5. We review related work in Section 6 and conclude in Section 7.

2 Preliminaries

2.1 Local Differential Privacy Protocols

In LDP protocols, the data collector wants to gather information from users willing to help the data collector. However, there is no guarantee that the data collector is trusted. To protect their privacy, users send locally perturbed data rather than the raw data to the data collector. Then, the data collector gathers all user reports and obtains overall statistical information through the aggregation algorithm. To satisfy the LDP privacy requirement, the perturbation algorithm applied locally must comply with the following definition.

Definition 1 (ε-Local Differential Privacy)**.** *An algorithm $\mathcal{A}$ satisfies ε-local differential privacy (ε-LDP), where $\varepsilon \geq 0$, if and only if for any inputs v, v', we have*

$$\forall y \in \text{Range}(\mathcal{A}) : \Pr[\mathcal{A}(v) = y] \leq e^{\varepsilon} \Pr[\mathcal{A}(v') = y]$$

where $\text{Range}(\mathcal{A})$ *denotes the set of all possible outputs of* $\mathcal{A}$*.*

By this definition, we can find that any two inputs in LDP have close probabilities of being mapped to the same output, which preserves the privacy of every user even if the data collector is not trusted.

2.2 Problem Definition

Assume there are n users, and each user possesses a value v from a specific domain $[d]$, which is denoted as $\{1, 2, \ldots, d\}$. In frequency estimation, the data collector is to learn the frequencies of each value across all users, and f_i is the frequency of the value i. Frequency estimation is a fundamental task in LDP protocols and a key building block for other advanced tasks, e.g., heavy hitter identification [2, 11, 23], range queries [5, 13, 21], frequent itemset mining [14, 22], etc. Improving frequency estimation performance will enhance the effectiveness of other protocols. More background and introduction on frequency estimation under local differential privacy are presented in the recent survey [26].

2.3 Pure Framework

A very useful framework proposed by Wang [20] is the notion of pure frequency protocols. Most existing frequency protocols can be considered pure and thus can be conveniently analyzed via this framework [6]. The definition of pure frequency protocols is as follows.

Definition 2 (Pure Frequency Protocols)**.** *A protocol is pure if and only if there exist two probability values $p^* > q^*$ in its perturbation algorithm $\mathcal{A}$ such that for all inputs v*

$$\Pr[\mathcal{A}(v) \in \{y | v \in \text{Support}(y)\}] = p^*,$$
$$\forall_{v' \neq v} \Pr[\mathcal{A}(v') \in \{y | v \in \text{Support}(y)\}] = q^*$$

where the set $\{y | v \in \text{Support}(y)\}$ denotes all outputs y that "support" the input v.

Since in a pure frequency protocol, p^* is the probability that the input v is mapped to an output that supports v; and q^* is the probability that every $v' \neq v$ is mapped to an output that supports v, we have the expected number of outputs from n users that support a specific input value i is $E(\sum_{k=1}^{n} \mathbf{1}_{\text{Support}(y_k)}(i)) = nf_i p^* + n(1 - f_i)q^*$. Consequently, the data collector can estimate the frequency f_i of input value i in aggregation using the following equation:

$$\widetilde{f_i} = \frac{\sum_{k=1}^{n} \mathbf{1}_{\text{Support}(y_k)}(i) - nq^*}{n(p^* - q^*)} \tag{1}$$

In addition, the estimation $\widetilde{f_i}$ can be considered the scaled summation of nf_i(resp. $n(1-f_i)$) independent random variables drawn from the Bernoulli distribution with parameter p^*(resp. q^*). Thus, the variance of estimation $\widetilde{f_i}$ is

$$\begin{aligned}\mathrm{Var}[\widetilde{f_i}] &= \frac{nf_ip^*(1-p^*)+n(1-f_i)q^*(1-q^*)}{n^2(p^*-q^*)^2} \\ &= \frac{q^*(1-q^*)}{n(p^*-q^*)^2}+\frac{f_i(1-p^*-q^*)}{n(p^*-q^*)}\end{aligned} \tag{2}$$

If frequency f_i is considered as a variable, the variance $\mathrm{Var}[\widetilde{f_i}]$ increases as f_i increases, and the second term can be viewed as a slope. But with a sufficiently large domain size and no dominant frequency f_i, the second item can be ignored. Thus, we can use the first term to represent the approximate variance as

$$\mathrm{Var}^*[\widetilde{f_i}] = \frac{q^*(1-q^*)}{n(p^*-q^*)^2} \tag{3}$$

Since the f_i is uncertain, existing studies usually use Equation (3) to analyze and optimize the frequency protocols. However, omitting the second term inevitably degrades the precision of the analysis, especially for the small and medium-sized domains. To address this issue, we introduce a new equation of analytical MSE in pure frequency protocols.

Analytical MSE. We are inspired by the consistency (i.e., the sum of all frequencies is 1) in [24] and the analysis of LDP noise in [9]. As the sum of all frequencies is 1 and each variance $\mathrm{Var}[\widetilde{f_i}]$ can be viewed as independent, we can eliminate the uncertain value f_i in analytical MSE as

$$\begin{aligned}\mathrm{MSE} &= \frac{1}{d}\Sigma_{i=1}^{d}\mathrm{Var}[\widetilde{f_i}] \\ &= \frac{q^*(1-q^*)}{n(p^*-q^*)^2}+\frac{1-p^*-q^*}{nd(p^*-q^*)} \\ &= \frac{q^*(1-q^*)}{n(p^*-q^*)^2}+\frac{1}{nd}+\frac{1-2p^*}{nd(p^*-q^*)}\end{aligned} \tag{4}$$

The empirical MSE is widely used to evaluate the accuracy of frequency protocols but requires extensive experiments. Our proposed equation of analytical MSE gives a universal and convenient way to assess the accuracy of pure frequency protocols more precisely. This equation reveals the connection between variance and domain size d in frequency protocols. Thus, we can apply it to study the best-suited domain size for existing frequency protocols. We can also use this equation to choose the optimal parameters when rounding is required.

3 Frequency Protocols

LDP frequency protocols usually consist of three algorithms: encoding, perturbation, and aggregation. Encoding converts each user's value into a specific format. Perturbation randomizes the encoded value locally, and users then send the perturbed value to the data collector. Aggregation gathers all reports from users to estimate each value's frequency. Existing frequency protocols vary in these algorithms and thus offer different accuracy guarantees, communication costs, and computation costs, resulting in the different best-suited domain sizes. Our study aims to optimize existing protocols in these metrics and find the optimal trade-off.

The following are four commonly used state-of-the-art pure frequency protocols.

3.1 Generalized Randomized Response (GRR)

The randomized response (RR) technique [25] in LDP can be traced back to 1965. Holohan [10] proves that choosing $p=\frac{e^\varepsilon}{e^\varepsilon+1}$ gives the RR technique minimum expected error. Note that the RR technique is only defined for the domain size d = 2. Kairouz [12] extends it to a larger domain called Generalized Randomized Response.

Encoding. In GRR, $\mathrm{Encode}(v)=v$ and $v\in[d]$.

Perturbation. $\mathrm{Perturb}(v)$ outputs $y\in[d]$ as follows

$$\Pr[\mathrm{Perturb}(v)=y]=\begin{cases}p=\frac{e^\varepsilon}{e^\varepsilon+d-1}, & \text{if } v=y\\ q=\frac{1}{e^\varepsilon+d-1}, & \text{if } v\neq y\end{cases}$$

Aggregation. In GRR, each output value i supports the input i. The data collector gathers all the outputs to get the support number on each value. As we have $p^*=p, q^*=q$, the data collector then estimates the frequency using Equation (1), and the analytical MSE is

$$\mathrm{MSE}_{\mathrm{GRR}}=\frac{e^\varepsilon+d-2}{n(e^\varepsilon-1)^2}+\frac{d-2}{nd(e^\varepsilon-1)} \tag{5}$$

Cost. As the output of each user is $y\in[d]$, the communication cost is $\mathrm{O}(\log d)$ and the computation cost is $\mathrm{O}(1)$ on the user side. The data collector's computation cost is $\mathrm{O}(n+d)$ or $\mathrm{O}(n)$ since typically $n\gg d$.

3.2 Optimized Unary Encoding (OUE)

The unary encoding technique converts the user's input into a length-d one-hot encoding vector and then perturbs each bit independently. RAPPOR's implementation [8] uses Symmetric Unary Encoding (SUE), which chooses $p=\frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$ and $q=\frac{1}{e^{\varepsilon/2}+1}$. OUE [20] optimizes the parameters p and q to minimize the approximate variance in Equation (3).

Encoding. $\mathrm{Encode}(v)=[0,\ldots,0,1,0,\ldots,0]$, a length-$d$ one-hot encoding vector B where only the v-th bit is 1.

Perturbation. OUE perturbs vector B bit-by-bit independently into B' as follows

$$\Pr[\mathrm{Perturb}(B_j)=1]=\begin{cases}p=\frac{1}{2}, & \text{if } B_j=1\\ q=\frac{1}{e^\varepsilon+1}, & \text{if } B_j=0\end{cases}$$

Aggregation. In OUE, the input i is supported if the i-th bit in the output vector B is 1. The data collector accumulates

the output vector to obtain all the support numbers. With $p^* = p, q^* = q$, the data collector estimates the frequency using Equation (1). However, with different encoding and perturbation, the analytical MSE is

$$\mathrm{MSE}_{\mathrm{OUE}} = \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} + \frac{1}{nd} > \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} \tag{6}$$

Cost. To output the length-d vector, the communication cost and the computation cost are both $\mathrm{O}(d)$ for each user. The computation cost of the data collector to estimate every value's frequency is $\mathrm{O}(nd)$.

3.3 Optimized Local Hashing (OLH)

OLH [20] uses a hash function to map the input values into a smaller domain $[g]$ and then perturbs the hash value to achieve lower communication costs than OUE with the same accuracy. To minimize the approximate variance in Equation (3), $g = \lfloor e^{\varepsilon}+1 \rfloor$ or $\lceil e^{\varepsilon}+1 \rceil$.

Encoding. $\mathrm{Encode}(v) = \langle H, x \rangle$, where H is generated from hash family $\mathbb{H}$ by uniformly choosing seed s from $[n]$ and $H(v) = x$.

Perturbation. OLH perturbs $\langle H, x \rangle$ into $\langle H, y \rangle$, just like GRR, as follows

$$\forall i \in [g], \Pr[y=i] = \begin{cases} p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, & \text{if } x = i \\ q = \frac{1}{e^{\varepsilon}+g-1}, & \text{if } x \neq i \end{cases}$$

Aggregation. For each user's output s and y, the data collector regenerates the hash function H using the seed s and iterates over domain $[d]$ to find all the values that $H(v) = y$, i.e., the values that the output supports. With $p^* = p, q^* = \frac{1}{g}p + \frac{g-1}{g}q = \frac{1}{g}$, the data collector uses Equation (1) to estimate the frequency. If we ignore the error introduced by rounding g, the accuracy performances of OLH and OUE are the same. The analytical MSE is

$$\mathrm{MSE}_{\mathrm{OLH}} = \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} + \frac{1}{nd} > \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} \tag{7}$$

Cost. The communication cost is $\mathrm{O}(\log n)$ due to the total number of users $n \gg g$. Since the computation cost of the hash function is linear with the message length, the computation cost of one call on the user side is $\mathrm{O}(\log d)$. The data collector needs to call the hash function nd times in aggregation. Thus, its computation cost is $\mathrm{O}(nd \log d)$. Furthermore, there is an additional fixed cost for the initialization and finalization of each hash function call. This indicates that OLH can be very slow with a large domain size.

3.4 Subset Selection (SS)

SS [15, 18, 27] outputs a randomly selected subset of a fixed size k with a specific probability containing the user's private value. To minimize the estimated error, the optimal $k = \lfloor \frac{d}{e^{\varepsilon}+1} \rfloor$ or $\lceil \frac{d}{e^{\varepsilon}+1} \rceil$. In addition, the SS protocol is equivalent to the GRR protocol when $k = 1$. Hence, SS can be viewed as a further extension and optimization of GRR.

Encoding and Perturbation. Let $\mathcal{Z}_1$ be the set of all k-subsets containing the private value v in the domain $[d]$ and $\mathcal{Z}_2$ be the set of all k-subsets not containing the value v in the domain $[d]$, we have

$$\Pr[\mathrm{Perturb}(v) = y] = \begin{cases} p = \frac{ke^{\varepsilon}}{ke^{\varepsilon}+d-k}, & \text{if } y \in \mathcal{Z}_1 \\ q = \frac{d-k}{ke^{\varepsilon}+d-k}, & \text{if } y \in \mathcal{Z}_2 \end{cases}$$

Aggregation. As each value i in the output subset supports the corresponding input value i in SS, we have $p^* = p, q^* = p\frac{k-1}{d-1} + (1-p)\frac{k}{d-1}$. Then, the data collector uses Equation (1) to estimate the frequency. If we ignore the error introduced by rounding k, we have $p^* = 1/2, q^* = \frac{1}{e^{\varepsilon}+1}\frac{d-(e^{\varepsilon}+1)/2}{d-1}$. The q^* of SS is smaller than that of OUE or OLH. However, as d increases, it gets closer and closer to $\frac{1}{e^{\varepsilon}+1}$, which is identical to OUE or OLH. The analytical MSE of SS is

$$\begin{aligned} \mathrm{MSE}_{\mathrm{SS}} &= \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} - \frac{(d-1)e^{2\varepsilon} + (6d-2)e^{\varepsilon} + d - 1}{nd^2(e^{\varepsilon}-1)^2} \\ &< \frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} \end{aligned} \tag{8}$$

Cost. Since the subset size k is linear with the domain size d and selecting k values from domain $[d]$ requires generating at least k random numbers, the communication cost and the computation cost are both $\mathrm{O}(d)$ for each user. The computation cost of the data collector is $\mathrm{O}(nd)$.

3.5 Summary of Frequency Protocols

The listed frequency protocols achieve different accuracy, computation cost, and communication cost combinations. These protocols are summarized together with the following optimized protocols in Table 2. Note that the equations of MSE for OLH and SS do not consider the rounding error of g or k in Table 2.

Among all these protocols, GRR is the simplest and fastest protocol, but its MSE is only optimal for a small domain size d and grows linearly with d, which is unacceptable for a large d. In contrast, OUE and OLH only achieve optimal MSE for a large d. The communication cost of OUE is linear with d. Compared with OUE, OLH achieves a lower communication cost but a relatively higher computation cost on the server side. Whether the domain size d is large or small, SS obtains the optimal MSE. However, its communication cost is also linear with d, which is unacceptable for a large d.

Furthermore, previous research [20] suggests that the intersection point between the MSE results of GRR and OUE (or OLH) is $d = 3e^{\varepsilon} + 2$ by using the approximate variance in Equation (3). However, with the proposed analytical MSE, we can give a more precise intersection $d = e^{\varepsilon} + 3/2 + \sqrt{2e^{2\varepsilon} + 3e^{\varepsilon} + 5/4} \approx 2.41e^{\varepsilon} + 2.56$.

4 Further Optimized Frequency Protocols

As we summarized, existing frequency protocols have their advantages and disadvantages in accuracy, computation cost, and communication cost. This leaves room for further optimization of these protocols. More specifically, the parameters of OUE and OLH are inferred by the approximate Equation (3), which results in OUE and OLH not achieving optimal accuracy performance across the UE and LH protocols, especially for the small domains. OLH is designed for large domains, but its computation cost may need to be lowered for the data collector. SS achieves optimal accuracy performance, but its communication cost is relatively high. In addition, since OUE and OLH have the word "optimized" in their names, our proposed protocols are called further optimized or re-optimized protocols.

4.1 Re-optimized Accuracy

4.1.1 Re-optimized Unary Encoding (RUE)

Here, we re-optimize the parameters of OUE through the proposed analytical MSE to enhance its accuracy performance. To guarantee ε-LDP, the parameters in the UE protocols need to satisfy the following conditions [20].

Theorem 1 (Privacy of UE)**.** *The UE protocols satisfy* ε*-LDP for*

$$p = \frac{e^{\varepsilon} q}{1 - q + e^{\varepsilon} q} \tag{9}$$

Proof. In the UE protocols, for any inputs v, v', and output vector B, we have

$$\begin{aligned} \frac{\Pr[B|v]}{\Pr[B|v']} &= \frac{\prod_{i=1}^{d} \Pr[B_i|v]}{\prod_{i=1}^{d} \Pr[B_i|v']} \\ &\leq \frac{\Pr[B_v = 1|v]\Pr[B_{v'} = 0|v]}{\Pr[B_v = 1|v']\Pr[B_{v'} = 0|v']} \\ &= \frac{p}{q} \cdot \frac{1-q}{1-p} \\ &= e^{\varepsilon} \end{aligned}$$

This yields $p = \frac{e^{\varepsilon} q}{1-q+e^{\varepsilon} q}$. □

Therefore, with $p^* = p, q^* = q$, plugging the $p = \frac{e^{\varepsilon} q}{1-q+e^{\varepsilon} q}$ into Equation (4), we have

$$\mathrm{MSE}_{\mathrm{UE}} = \frac{((e^{\varepsilon}-1)q+1)^2}{n(e^{\varepsilon}-1)^2 q(1-q)} + \frac{1}{nd} + \frac{1-q-e^{\varepsilon} q}{ndq(1-q)(e^{\varepsilon}-1)} \tag{10}$$

To minimize the analytical $\mathrm{MSE}_{\mathrm{UE}}$, we take its part derivative with respect to q and solve for the re-optimized p and q when the result is 0.

$$\frac{\partial \mathrm{MSE}_{\mathrm{UE}}}{\partial q} = 0 \Rightarrow q = \frac{1}{e^{\varepsilon}\sqrt{\frac{d-1+1/e^{\varepsilon}}{d-1+e^{\varepsilon}}} + 1},$$
$$p = \frac{1}{\sqrt{\frac{d-1+1/e^{\varepsilon}}{d-1+e^{\varepsilon}}} + 1}$$

Compared with OUE, there is a coefficient $h = \sqrt{\frac{d-1+1/e^{\varepsilon}}{d-1+e^{\varepsilon}}}$ integrating both domain size d and privacy budget ε in the re-optimized parameters. It is worth noting that when $d = 2$, we have $h = 1/e^{\varepsilon/2}$, $p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$ and $q = \frac{1}{e^{\varepsilon/2}+1}$ which take the same values as SUE; when d tends to infinity, we have $h = 1$, $p = \frac{1}{2}$ and $q = \frac{1}{e^{\varepsilon}+1}$ which take the same values as OUE.

With the coefficient h, plugging the re-optimized p and q to Equation (10), we have the analytical MSE of RUE as

$$\mathrm{MSE}_{\mathrm{RUE}} = \frac{2e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} + \frac{2e^{\varepsilon}}{nh(e^{\varepsilon}-1)^2} - \frac{2}{ndh(e^{\varepsilon}-1)} \tag{11}$$

4.1.2 Re-optimized Local Hashing (RLH)

The LH protocols use a hash function to reduce the size of the domain in encoding, while the perturbation algorithm is the same as GRR. Since GRR satisfies ε-LDP, the LH protocols satisfy ε-LDP as long as the perturbation algorithm is not altered.

As in RUE, we bring g into the analytical MSE. With $p^* = p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, q^* = \frac{1}{g}p + \frac{g-1}{g}q = \frac{1}{g}$, substituting g into p^* and q^* in Equation (4), we have

$$\mathrm{MSE}_{\mathrm{LH}} = \frac{(e^{\varepsilon}+g-1)^2}{n(e^{\varepsilon}-1)^2(g-1)} + \frac{1}{nd} + \frac{g(g-1-e^{\varepsilon})}{nd(g-1)(e^{\varepsilon}-1)} \tag{12}$$

As in the derivation in RUE, we solve for the re-optimized g to minimize the $\mathrm{MSE}_{\mathrm{LH}}$.

$$\frac{\partial \mathrm{MSE}_{\mathrm{LH}}}{\partial g} = 0 \Rightarrow g = e^{\varepsilon}\sqrt{\frac{d-1+1/e^{\varepsilon}}{d-1+e^{\varepsilon}}} + 1$$

There is also a coefficient $h = \sqrt{\frac{d-1+1/e^{\varepsilon}}{d-1+e^{\varepsilon}}}$ in the re-optimized g. Obviously, as domain size d varies from 2 to infinity, the re-optimized g subsequently varies from $e^{\varepsilon/2}+1$ to $e^{\varepsilon}+1$. Therefore, RLH can be considered equivalent to OLH when the domain is sufficiently large. In practice, g needs to be rounded to an integer, and we choose the integer that minimizes the analytical MSE of Equation (4) or (12) from the two nearest integers.

Neglecting the error introduced by rounding g and plugging the re-optimized g into Equation (12), the analytical MSE of RLH is

$$\mathrm{MSE}_{\mathrm{RLH}} = \frac{2e^{\varepsilon}}{n(e^{\varepsilon}-1)^2} + \frac{2e^{\varepsilon}}{nh(e^{\varepsilon}-1)^2} - \frac{2}{ndh(e^{\varepsilon}-1)} \tag{13}$$

Table 1: The value of coefficient h for different d and ε

	$d=50$	$d=100$	$d=500$	$d=1000$
$\varepsilon=0.5$	0.9897	0.9948	0.9990	0.9995
$\varepsilon=1$	0.9770	0.9884	0.9977	0.9988
$\varepsilon=2$	0.9335	0.9653	0.9928	0.9964
$\varepsilon=3$	0.8426	0.9120	0.9805	0.9901
$\varepsilon=4$	0.6879	0.8029	0.9494	0.9738
$\varepsilon=5$	0.4982	0.6326	0.8779	0.9331

As there is a coefficient h in the parameters of both RUE and RLH, the coefficient h can be regarded as a similarity factor reflecting the degree of similarity between the re-optimized and original parameters. The larger the d or the smaller the privacy budget ε, the closer the coefficient h is to 1, leading to a higher similarity. Table 1 shows the coefficient h for different domain size d and privacy budget ε.

4.1.3 Optimal Parameterizations at Different Frequency Levels

Reviewing Equation (2), (3) and (4), we can find that the analytical MSE takes the same value as $\mathrm{Var}[\widetilde{f_i}]$ when frequency $f_i = 1/d$. Therefore, minimizing analytical MSE is equivalent to minimizing $\mathrm{Var}[\widetilde{f_i}]$ at frequency $f_i = 1/d$, and minimizing approximate $\mathrm{Var}^*[\widetilde{f_i}]$ is equivalent to minimizing $\mathrm{Var}[\widetilde{f_i}]$ at frequency $f_i = 0$. Thus, OUE and OLH can be considered as obtaining parameters by minimizing $\mathrm{Var}[\widetilde{f_i}]$ at frequency $f_i = 0$. In contrast, RUE and RLH solve for the parameters by minimizing $\mathrm{Var}[\widetilde{f_i}]$ at frequency $f_i = 1/d$. Consequently, our re-optimized parameters can better balance the variance between frequent and infrequent values to achieve optimal performance in terms of accuracy. If the parameters are solved at frequencies higher than $1/d$, the variances of the infrequent values are further sacrificed to improve the accuracy of the frequent values. This strategy may be useful for the heavy hitter identification but worsens the total variance.

4.2 Re-optimized Computation Cost

Another limitation of OLH is that it can be very slow with total nd calls of the hash function in aggregation. Reconsidering the utility of hash functions in this protocol, we can find it is designed to divide all the possible inputs into g groups. Thus, a specific output y of LH protocols is a subset (i.e., the y-th group), and all values in this subset are the support inputs. By transmitting the generating seed instead of the subset, OLH reduces the communication cost but needs to regenerate the hash function and the grouping results on the server side. The computation cost of one hash function call is linear with the message length, and it can only identify one value's grouping result in a single call. To determine the complete grouping results, the data collector must make d hash function calls, bringing a computation cost of $\mathrm{O}(d\log d)$ in aggregation for just one user's report. Furthermore, the additional cost incurred by initializing and finalizing each hash function call cannot be ignored.

However, a complete randomized grouping result for domain $[d]$ can be obtained with only $\mathrm{O}(d)$ computation cost if we use the random number generator. Consider generating d random numbers, each taking values ranging from 1 to g. The computation cost of this random process is $\mathrm{O}(d)$, and we only need to call the random function once using NumPy, which reduces the total cost of initialization and finalization. As in OLH, we use the uniformly chosen seed for the random grouping to reduce communication costs and ensure reproducibility. Therefore, we replace the hash function with a randomized grouping process in encoding as:

Encoding. $\mathrm{Encode}(v) = \langle B, x\rangle$, where B is a randomized grouping vector generated by uniformly choosing seed s from $[n]$ and $B(v) = x$. B is a length-d vector, each taking values ranging from 1 to g.

Although we use a random process to replace the hash function, the essence remains the same, so the improved method is called Re-optimized Local Hashing (RLH). With the randomized grouping vector B, the server-side computation cost can be reduced from $\mathrm{O}(nd\log d)$ to $\mathrm{O}(nd)$. The communication cost remains the same because the user reports seed s and $y \in [g]$ to the data collector as in OLH. Since the perturbation algorithm has not changed, RLH still satisfies ε-LDP.

We can also explain the similarity between the UE and LH protocols through the equivalent grouping process. More specifically, both UE and LH protocols' outputs can be considered a subset of grouping results, and their restrictions on parameters are essentially the same, i.e., the outputs should satisfy ε-LDP. Thus, with the same optimization objective, their randomization processes are equivalent.

Furthermore, we can find that all UE, LH, and SS can be considered outputting a subset. In these protocols, the critical difference for SS to achieve optimal accuracy is to fix subset size. Hence, there only exist two probabilities for each subset to be output in SS: a high probability of the subset including the user's value and a low probability of the subset excluding the user's value. In contrast, there are various probabilities for different-sized subsets to be output in OUE and OLH, bringing tighter restrictions on the parameters to satisfy ε-LDP.

4.3 Fusion of Local Hashing and Subset Selection with RWS

The SS protocol derives the parameter k by minimizing the L_2 loss, which is equivalent to minimizing the proposed analytical MSE. The SS protocol achieves optimal performance in accuracy by outputting a fixed-size subset but its communication cost is linear with d, and the LH protocols can

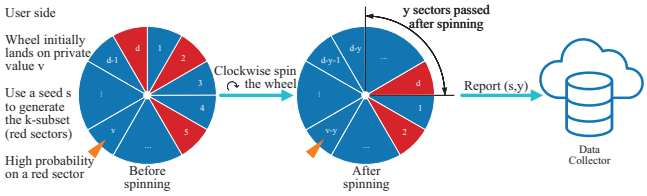


Figure 1: Random Wheel Spinner protocol

efficiently reduce the communication cost by using local hashing technique (i.e., using a seed instead of generating results in reports). To address the communication issue, we propose the Random Wheel Spinner (RWS) protocol, which combines the advantages of LH and SS.

Figure 1 presents the randomized model of the RWS protocol. Assume a spinning wheel is divided into d sectors, each of which is marked with one of the values 1 to d. The wheel initially lands on the user's private value v. The user randomly generates a subset with a fixed size k from domain $[d]$ via random seed $s \in [n]$, i.e., randomly select k values from domain $[d]$ without replacement. The optimal $k = \lfloor \frac{d}{e^\varepsilon+1} \rfloor$ or $\lceil \frac{d}{e^\varepsilon+1} \rceil$ that minimize the analytical MSE of Equation (4) or (15), and $k \geq 1$.

The wheel is biased so that the probability of landing on a specific value in the k-subset is $\frac{e^\varepsilon}{ke^\varepsilon+d-k}$, while the other values have equal probabilities of $\frac{1}{ke^\varepsilon+d-k}$. The user then spins the wheel to get a random value $(v-y)\%d$ where y is the number of sectors passed after spinning. The valid y can only be from 1 to d because the wheel "wraps around" every d sectors (i.e., 360 degrees). After spinning, the user only reports the seed (instead of the k-subset) and the number y to the data collector. The formal encoding, perturbation, and aggregation algorithms are as follows:

Encoding. Encode$(v) = v$ and $v \in [d]$. Selects k values from domain $[d]$ without replacement to generate the uniformly chosen k-subset using randomly chosen seed $s \in [n]$.

Perturbation. Perturb(v) randomly outputs $y \in [d]$ as follows

$$\forall i \in [d],$$
$$\Pr[y=i] = \begin{cases} p = \frac{e^\varepsilon}{ke^\varepsilon+d-k}, & \text{if } (v-i)\%d \in k\text{-subset} \\ q = \frac{1}{ke^\varepsilon+d-k}, & \text{if } (v-i)\%d \notin k\text{-subset} \end{cases}$$

where $(v-i)\%d$ is the value that the wheel lands on after i sectors of rotation. Because $[d]$ starts at 1 in the paper, we specify that $0\%d = d$. This is not required if the practical implementation starts at 0.

Aggregation. For each user's report of s and y, the data collector regenerates the k-subset and processes it with the operation $(k\text{-subset}+y)\%d$. Each value in the processed subset supports the corresponding input value. As $(v-y)\%d \in k$-subset can be equivalently converted to $v \in (k\text{-subset}+y)\%d$, the probability that the processed k-subset contains the private value v is $p^* = kp$. For any other value, it is included with probability $q^* = p^*\frac{k-1}{d-1} + (1-p^*)\frac{k}{d-1}$. Then, the data collector uses Equation (1) to estimate the frequency. If we ignore the error introduced by rounding k and $k = \frac{d}{e^\varepsilon+1}$, we have $p^* = 1/2, q^* = \frac{1}{e^\varepsilon+1}\frac{d-(e^\varepsilon+1)/2}{d-1}$, which are identical to SS protocol. The analytical MSE is

$$\begin{aligned} \text{MSE}_{\text{RWS}} &= \frac{4e^\varepsilon}{n(e^\varepsilon-1)^2} - \frac{(d-1)e^{2\varepsilon}+(6d-2)e^\varepsilon+d-1}{nd^2(e^\varepsilon-1)^2} \\ &< \frac{4e^\varepsilon}{n(e^\varepsilon-1)^2} \end{aligned} \tag{14}$$

Cost. Typically, the total number of users $n \gg d$, thus the communication cost is $O(\log n)$ which is the same as the LH protocols. Since the subset size k is linear with the domain size d, the computation cost of generating a subset is $O(d)$, and the total computation cost in aggregation is $O(nd)$.

Theorem 2 (Privacy of RWS)**.** *The proposed RWS protocol satisfies* ε*-LDP.*

Proof. In the RWS protocol, for any two inputs v, v', and output seed s and angular displacement y, we have

$$\frac{\Pr[y|v,s]}{\Pr[y|v',s]} = \frac{\Pr[\text{Perturb}(v)=y]}{\Pr[\text{Perturb}(v')=y]} \leq \frac{p}{q} = e^\varepsilon$$

□

Theorem 3 (The optimal k of RWS)**.** *The RWS protocol achieves the optimal analytical MSE at* $k = \frac{d}{e^\varepsilon+1}$

Proof. In the RWS protocol, we have $p^* = kp, q^* = kp\frac{k-1}{d-1} + (1-kp)\frac{k}{d-1}$. Plugging these values into Equation (4), we have

$$\begin{aligned} \text{MSE}_{\text{RWS}} = &\frac{(ke^\varepsilon+d-k-e^\varepsilon)(ke^\varepsilon+d-k-1)}{nk(e^\varepsilon-1)^2(d-k)} + \frac{1}{nd} \\ &+ \frac{(d-1)(d-k-ke^\varepsilon)}{nkd(e^\varepsilon-1)(d-k)} \end{aligned} \tag{15}$$

To minimize the analytical MSE, we take its part derivative with respect to k and solve for k when the result is 0.

$$\begin{aligned} &\frac{\partial \text{MSE}_{\text{RWS}}}{\partial k} = 0 \\ &\Rightarrow \frac{(d-1)^2(k(e^\varepsilon+1)-d)(k(e^\varepsilon+1)+d)}{nk^2d(e^\varepsilon-1)^2(d-k)^2} = 0 \\ &\Rightarrow k = \frac{d}{e^\varepsilon+1} \end{aligned}$$

□

Comparison with SS. The proposed RWS achieves the same optimal accuracy as SS, but the communication cost is significantly reduced from $O(d)$ to $O(\log n)$ for large domains.

Table 2: Comparison of frequency protocols

Frequency Protocols	Analytical MSE			Aggregation cost	Communication cost
	Equation	$d=2$	$d\to\infty$		
GRR	$\frac{e^\varepsilon+d-2}{n(e^\varepsilon-1)^2}+\frac{d-2}{nd(e^\varepsilon-1)}$	$\frac{e^\varepsilon}{n(e^\varepsilon-1)^2}$	∞	$O(n)$	$O(\log d)$
OUE	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{1}{nd}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{1}{2n}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd)$	$O(d)$
RUE	$\frac{2e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{2e^\varepsilon}{nh(e^\varepsilon-1)^2}-\frac{2}{ndh(e^\varepsilon-1)}$	$\frac{e^{\varepsilon/2}}{n(e^{\varepsilon/2}-1)^2}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd)$	$O(d)$
OLH	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{1}{nd}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{1}{2n}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd\log d)$	$O(\log n)$
RLH	$\frac{2e^\varepsilon}{n(e^\varepsilon-1)^2}+\frac{2e^\varepsilon}{nh(e^\varepsilon-1)^2}-\frac{2}{ndh(e^\varepsilon-1)}$	$\frac{e^{\varepsilon/2}}{n(e^{\varepsilon/2}-1)^2}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd)$	$O(\log n)$
SS	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}-\frac{(d-1)e^{2\varepsilon}+(6d-2)e^\varepsilon+d-1}{nd^2(e^\varepsilon-1)^2}$	$\frac{e^\varepsilon}{n(e^\varepsilon-1)^2}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd)$	$O(d)$
RWS	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}-\frac{(d-1)e^{2\varepsilon}+(6d-2)e^\varepsilon+d-1}{nd^2(e^\varepsilon-1)^2}$	$\frac{e^\varepsilon}{n(e^\varepsilon-1)^2}$	$\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$	$O(nd)$	$O(\log n)$

$h=\sqrt{\frac{d-1+1/e^\varepsilon}{d-1+e^\varepsilon}}$. For more precise equations of LH, SS, and RWS that consider the rounding error of g or k, please refer to Equation (4), (12) and (15) and bring the rounded parameters to these equations.

We can roughly assume that the worst-case scenario is n = 8 billion since the current world population is about 8 billion. In this worst-case scenario, $O(\log n)\approx 33$. Furthermore, SS directly outputs a subset related to the user's private value. In contrast, the subset generated using a random seed in RWS is independent of the user's private value, thus making it possible to reduce the communication cost.

Comparison with the Wheel Mechanism. The Wheel Mechanism [19] is a continuous version of LH protocols and its output cannot suggest a fixed number of possible inputs. It uses the seed to hash the private value to a floating point $x\in[0,1.0)$ and perturbs x in the range of $[0,1.0)$ where the probability mass in the coverage area $[x,x+c)$ (we use c instead of p in [19] to avoid confusion.) is higher than the rest. Like LH protocols, it reports the seed and perturbed value to the data collector. For different perturbed values, the hash seed produces various numbers of possible coverage areas (i.e., possible inputs in the domain). The true coverage probability $\frac{c\cdot e^\varepsilon}{c\cdot e^\varepsilon+1-c}$ and false coverage probability c in [19] are equivalent to the p^* and q^* respectively in the pure framework. As $p^*=\frac{e^\varepsilon}{e^\varepsilon+g-1}, q^*=\frac{1}{g}$ in LH protocols, we can find these probabilities are identical if we make $c=\frac{1}{g}$. Thus, the Wheel Mechanism is equivalent to the LH protocols and its optimal accuracy is the same as RLH with $c=\frac{1}{h\cdot e^\varepsilon+1}$. In contrast, our proposed RWS uses the seed to generate a k-subset independent of the private value and ensures that every output suggests a fixed number of possible inputs. This makes the restrictions on the parameters p^* and q^* more relaxed in RWS, thus achieving higher accuracy compared with the Wheel Mechanism and LH protocols.

4.4 Summary of Further Optimized Frequency Protocols

We summarize the accuracy, computation cost, and communication cost of listed protocols in Table 2. Note that the equations of MSE for OLH, RLH, SS, and RWS do not consider the rounding error of g or k. To get a more precise equation, we should bring the rounded g or k into the analytical MSE, but it may be too complicated and cannot intuitively show the connection between the MSE and the domain size d. The aggregation cost is the computation cost of the data collector. We omit the computation cost on the user side because this cost is too small for the user to perceive. The longest experimental perturbation time for a user among all protocols is less than 0.04 seconds, even if the domain size is 1 million.

We enhance the accuracy of OUE and OLH for small domains and recall them as RUE and RLH, respectively. We also reduce the computation cost of OLH on the server side from $O(nd\log d)$ to $O(nd)$ effectively. Our proposed RWS fuses the key ideas of SS and LH to achieve optimal accuracy with low computation and communication costs. Among all protocols (except GRR), RWS achieves optimal performance in every metric. Although GRR has a minimal computation cost, it performs poorly when the domain size d is large; when d is small, the difference in computation and communication costs among protocols is negligible.

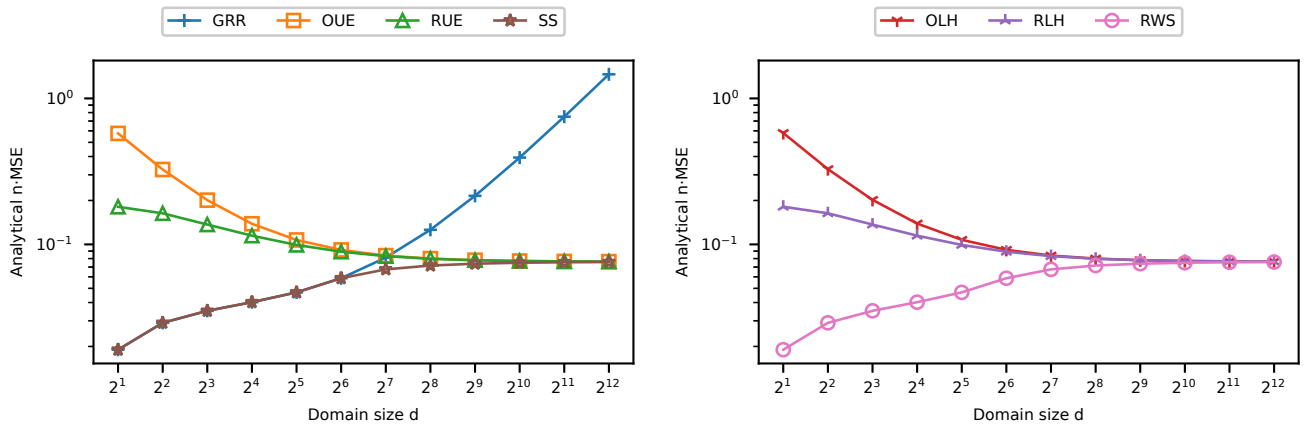
In UE, LH, SS, and RWS, a tight MSE bound $\frac{4e^\varepsilon}{n(e^\varepsilon-1)^2}$ exists when d tends to infinity. Notably, it is a tight lower bound for UE and LH, while for SS and RWS, it is a tight upper bound.

Numerical values of analytical n·MSE for different frequency protocols using Equation (4) are given in Table 3 and Figure 2. Since the values of OUE and OLH are very similar, as are the RUE and RLH, and SS and RWS, we plot values on two separate figures to make the results (especially for the

Table 3: Numerical values of analytical n·MSE for different frequency protocols with $\varepsilon = 4$

	GRR	OUE	RUE	OLH	RLH	SS	RWS
$d = 2$	0.01901	0.5760	0.1811	0.5798	0.1812	0.01901	0.01901
$d = 2^4$	0.04020	0.1385	0.1148	0.1390	0.1148	0.04020	0.04020
$d = 2^7$	0.08123	0.08383	0.08311	0.08389	0.08311	0.06747	0.06747
$d = 2^{10}$	0.3934	0.07700	0.07699	0.07701	0.07699	0.07491	0.07491

The values of OUE and OLH are slightly different because we consider the rounding error, as are RUE and RLH.



(a) Varying d, fixing $\varepsilon = 4$

(b) Varying d, fixing $\varepsilon = 4$

Figure 2: Numerical values of analytical n·MSE for different frequency protocols

following comparison of empirical and analytical MSE) more readable.

Guideline. With different domain sizes, our analysis provides the following guidelines for choosing protocols.

- When domain size d is small, more precisely, when $d < \sqrt{2e^{2\varepsilon} + 0.25} + 1.5$ (i.e., the optimal $k = 1$), GRR, SS and RWS have the optimal MSE.
- When $d > \sqrt{2e^{2\varepsilon} + 0.25} + 1.5$ and the communication cost $O(d)$ is acceptable, we should use SS or RWS.
- When d is so large that the MSE of RLH and RWS tends to $\frac{4e^{\varepsilon}}{n(e^{\varepsilon}-1)^2}$ and the communication cost $O(d)$ is unacceptable, we should use RLH or RWS which have the communication cost of $O(\log n)$.
- If only one protocol can be chosen for various domain sizes, RWS is all you need. It offers the optimal MSE with low computation and communication costs whether d is large or small.

5 Experiments

5.1 Experimental Setup

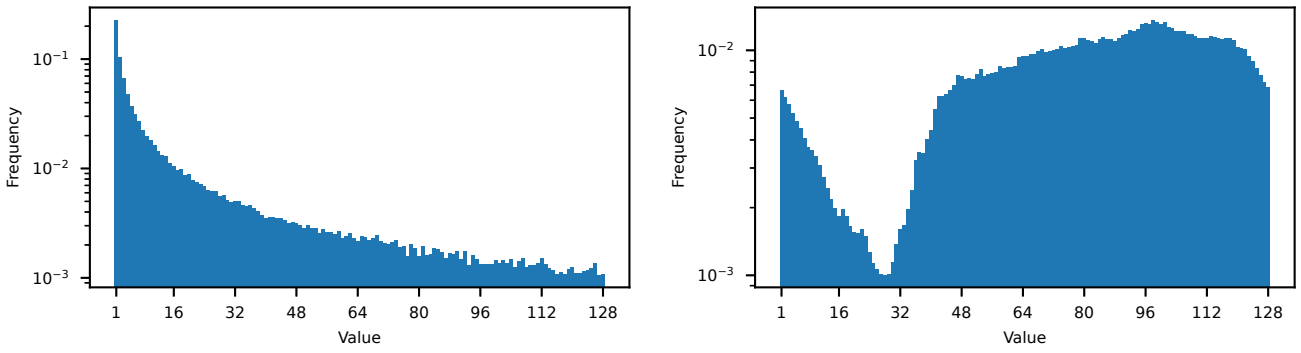
Datasets. We conduct experiments on the following two datasets (one synthetic and one real-world).

Synthetic Zipf's dataset. The synthetic data is generated from Zipf's distribution with parameter $s = 1.1$, similar to experiments in [20]. For every given domain size, we sample 100,000 points (i.e., $n = 100,000$).

Taxi pickup time dataset. Taxi pickup time dataset comes from March 2024 New York Yellow Taxi Trip Records [16]. There are 3,582,628 records in this dataset (i.e., $n = 3,582,628$). We normalize the time records to [0,1] and categorize them at different given domain sizes.

Figure 3 shows the frequency distribution of datasets used for experiments. The frequencies in Zipf's dataset vary dramatically, while the frequencies, especially the top frequencies, in Taxi dataset are relatively even.

Setup. All frequency protocols are implemented in Python 3.10.4 using Numpy 1.26.4 and xxhash 3.4.1. All experiments are conducted on a PC with AMD Ryzen 9 7950X and



(a) Zipf's

(b) Taxi

Figure 3: True frequencies of values in the experimental datasets with $d = 2^7$

64GB memory. Unless stated otherwise, all experiments are performed 100 times with their results averaged to reduce randomness.

Metrics. We verify the correctness of the analytical MSE and evaluate the accuracy and aggregation cost of all frequency protocols on two datasets.

We compare the empirical and analytical MSE to verify the correctness of our analysis. The empirical MSE, $\frac{1}{d}\sum_{v\in[d]}(\widetilde{f}_v - f_v)^2$, is also used to evaluate the accuracy of all protocols. It measures the mean of the squared errors, i.e., the average squared difference between each estimate and ground truth. As data collectors are usually more interested in frequent values, we also compute the MSE results on top k frequent values instead of the full domain. The empirical aggregation time is used to evaluate the computation cost on the server side of all frequency protocols.

5.2 Verification of Analysis

We introduce the analytical MSE to analyze the accuracy of frequency protocols and further optimize the parameters of OUE and OLH. To verify the analytical MSE, we now show that the analytical MSE matches the empirically measured MSE.

Figure 4 shows the empirical and analytical MSE results for all protocols on both synthetic and real-world datasets with $\varepsilon = 4$. The empirical results match very well with the analytical results in both datasets, although there is a slight fluctuation when the domain size is small. This is because MSE is the average of the variances of all the values in the domain, and the smaller the domain size, the smaller the number (i.e., the sample size) of averages and the greater the fluctuations according to the law of large numbers. We run the experiments 100 times with the results averaged to reduce the fluctuations. More experiments could further reduce fluctuations but would be very time-consuming.

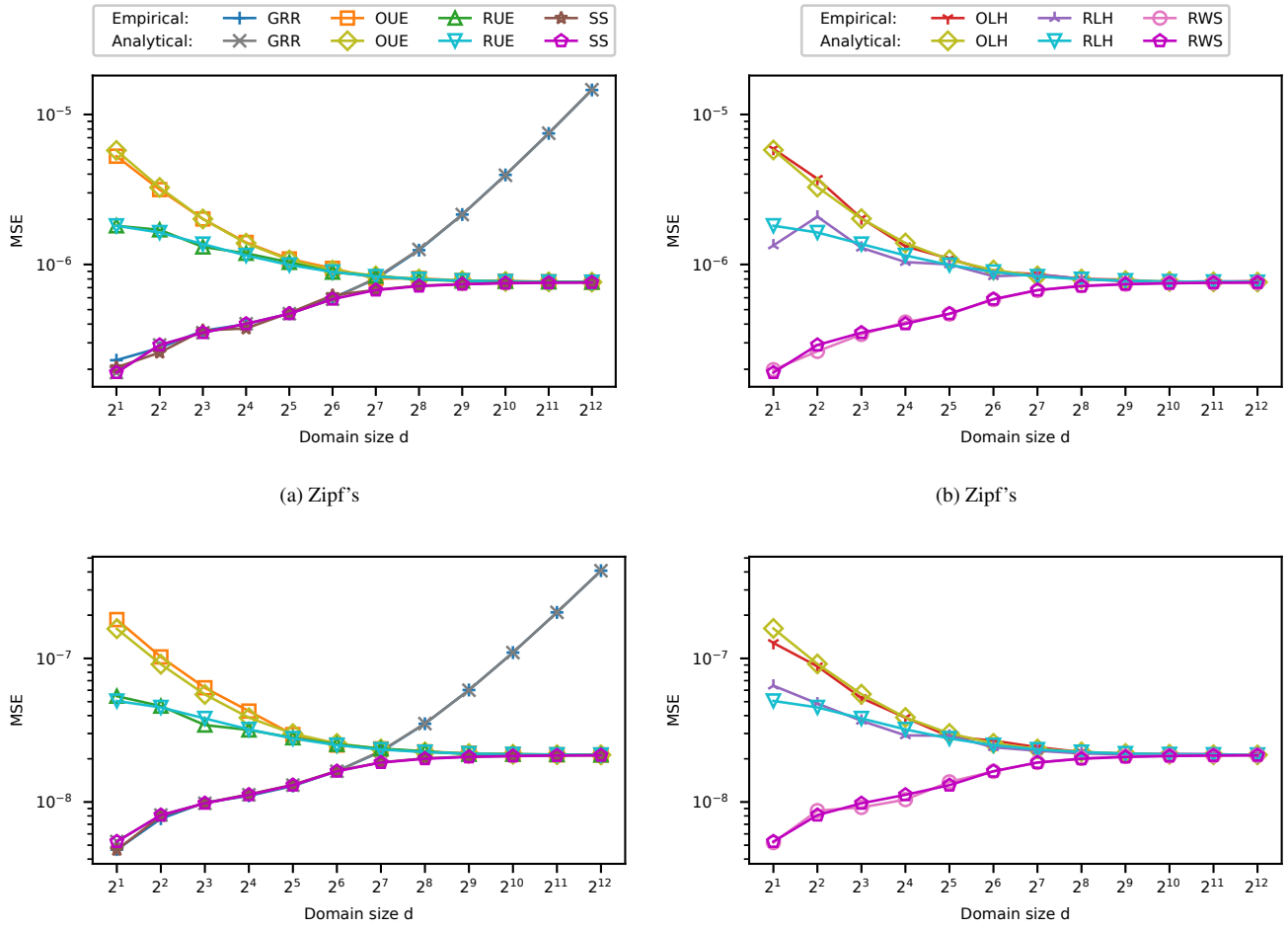
5.3 Accuracy Evaluation

To compare the accuracy of all protocols straightforwardly, we plot each protocol's full-domain MSE results on the same subfigure in Figure 5. The empirical results on Zipf's dataset are very similar to those of Taxi dataset, which is consistent with our analysis. That is, the sum of all frequencies is 1, which makes the full-domain MSE independent of the distribution of the dataset.

In Figure 5a and 5c, we fix $\varepsilon = 4$ and vary the domain size d. We can find that RWS and SS achieve almost identical performance, as well as the performance of OUE and OLH, and RUE and RLH, respectively. RWS and SS achieve the best accuracy in all protocols regardless of domain size. GRR achieves the same MSE with RWS and SS when $d <= 2^6$. This is because the optimal subset size $k = 1$ in RWS and SS when $d < \sqrt{2e^{2\varepsilon} + 0.25} + 1.5 \approx 78.7$ according to the guideline. When $k = 1$, RWS and SS are equivalent to GRR. However, when $d > 2^7$, the performance of GRR starts to be surpassed by other protocols. OUE and OLH perform poorly when the domain size is small. RUE and RLH enhance the accuracy of OUE and OLH for small domains, and when $d >= 2^6$, they achieve very close performances to OUE and OLH. When $d >= 2^{10}$, all protocols except GRR have very close performance due to tending to the analytical bound. We can find an intersection of GRR and other protocols (except RWS and SS) at about $d = 2^7$, which is consistent with our analytical intersection $2.41e^{\varepsilon} + 2.56 \approx 134$.

In Figure 5b and 5d, we fix $d = 2^7$ and vary the privacy budget ε. We can find that the effect of ε is opposite to that of d. More specifically, the effect of increasing ε on the empirical MSE is equivalent to decreasing d. As with the performance with varying d, RWS and SS are optimal for all ε values. GRR performs poorly when ε is small and performs better than UE and LH protocols when $\varepsilon > 4$. In contrast, UE and LH protocols perform much better than GRR when ε is small.

As data collectors are usually more interested in frequent



(a) Zipf's

(b) Zipf's

(c) Taxi

(d) Taxi

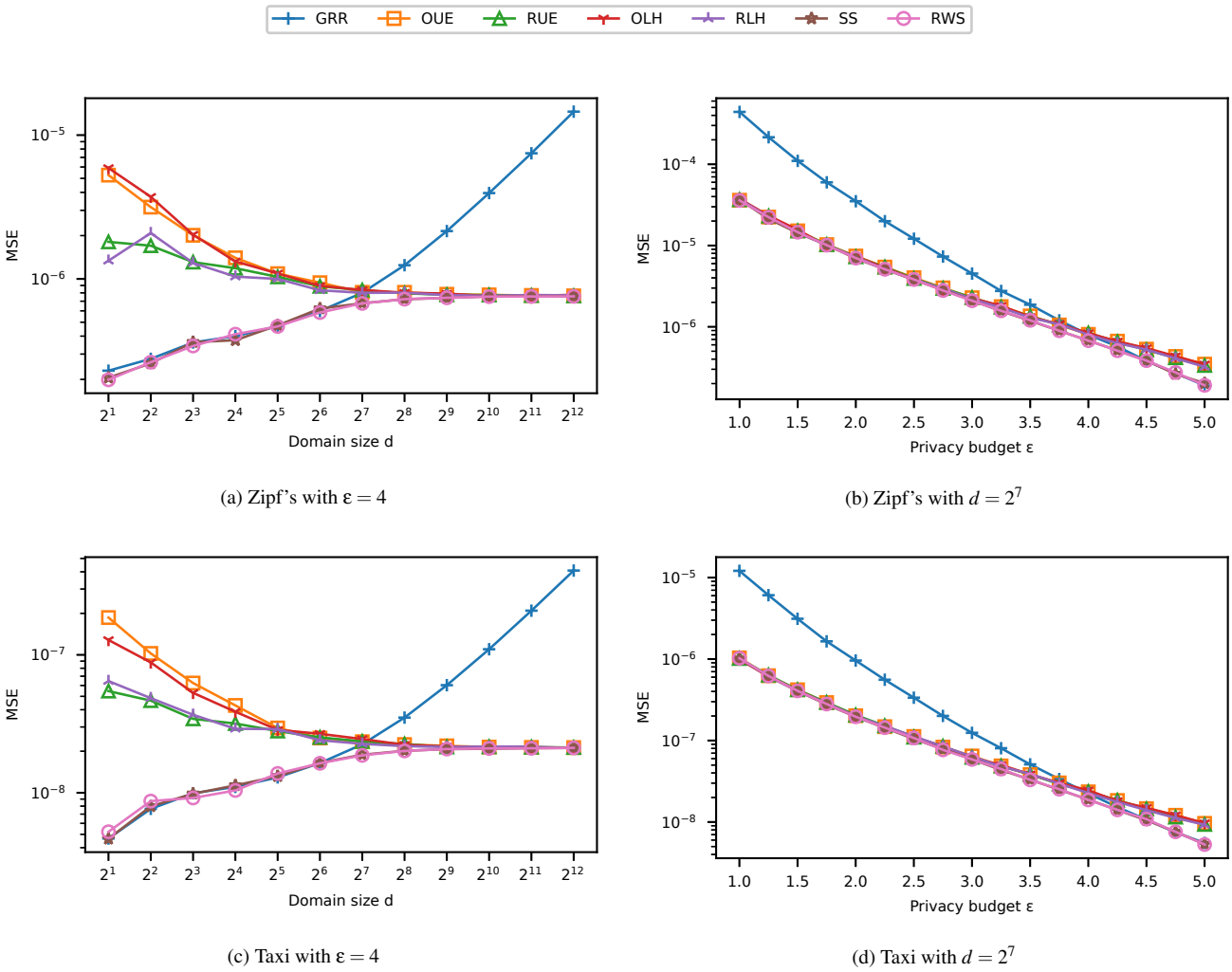
Figure 4: Comparing empirical and analytical MSE results with $\varepsilon = 4$

values, we give the empirical MSE results on top k values from 2 to 30 with $d = 2^7$ and $\varepsilon = 4$ in Figure 6. No frequency protocols are aware of the top k values in processing. Because all protocols except GRR have very similar pure parameters of p^* and q^* for a large d, their performance on top k values would be identical if d is large. Thus, we choose a middle-sized $d = 2^7$ instead of a large d to compare the accuracy of all protocols for frequent values. All protocols except RWS and SS achieve similar full-domain MSE results at $d = 2^7$, which gives us a better perspective to compare the treatment of frequent values among protocols.

We can find that the results on Zipf's dataset are quite different from those on Taxi dataset due to the different distributions of the two datasets. Notably, RWS and SS perform very close to each other and outperform other protocols in both datasets.

Figure 6a gives the top k values' MSE results on Zipf's dataset. In this dataset, the most frequent value occurs about twice as often as the second most frequent value, three times as often as the third most frequent value, etc. Since the variance of a particular value decreases as its frequency decreases, all protocols' MSE results on top k values decrease as k increases. We can see that GRR achieves the worst MSE on top k values, although it has a similar or slightly better full-domain MSE to other protocols except RWS and SS at $d = 2^7$ and $\varepsilon = 4$. The result of OUE is slightly better than that of OLH. Our proposed RUE and RLH achieve better MSE results on the top k values than OUE and OLH, respectively. This is consistent with our theoretical analysis that RUE and RLH enhance the accuracy of frequent values to achieve better full-domain MSE. We can find that RUE and RLH perform even slightly better than RWS and SS when $k <= 6$.

Figure 6b gives the top k values' MSE results on Taxi dataset. In contrast to Zipf's dataset, the frequencies of top



(a) Zipf's with $\varepsilon = 4$

(b) Zipf's with $d = 2^7$

(c) Taxi with $\varepsilon = 4$

(d) Taxi with $d = 2^7$

Figure 5: Empirical MSE results of different frequency protocols

k values are quite even in this dataset. All protocols' results fluctuate as k increases. OLH achieves the worst MSE values on top k values, while GRR performs between OLH and OUE. The proposed RUE and RLH still achieve better MSE results on the top k values than OUE and OLH, respectively. RWS and SS outperform other protocols for all k values.

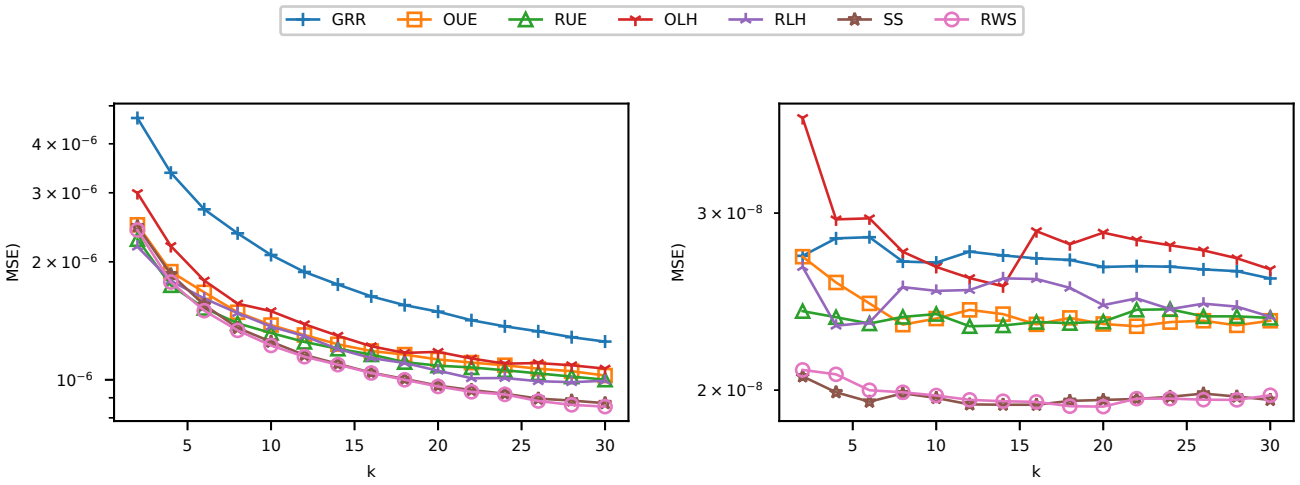
Furthermore, both Figures 6a and 6b indicate that RUE and RLH are superior to OUE and OLH when k is small, that is, for high frequent values. This matches the discussion in Section 4.1.3, where RUE and RLH improve the accuracy of the frequent values by solving for the optimal parameters at frequency $f_i = 1/d$ instead of 0. Solving the parameters at frequency $f_i = 1/d$ can also achieve the optimal MSE. However, if the data collector is only interested in frequent values (like heavy hitter identification), it can solve for the parameters at a frequency higher than $1/d$.

5.4 Aggregation Cost Evaluation

To show the aggregation cost of all frequency protocols, we measure the runtime of each protocol's aggregation process while fixing $\varepsilon = 4$, varying d. The experiment is run only once, as there is basically no randomness in the running time. To measure the cost more precisely, we do not take advantage of multiprocessing or parallelism, which can massively reduce the running time in practice.

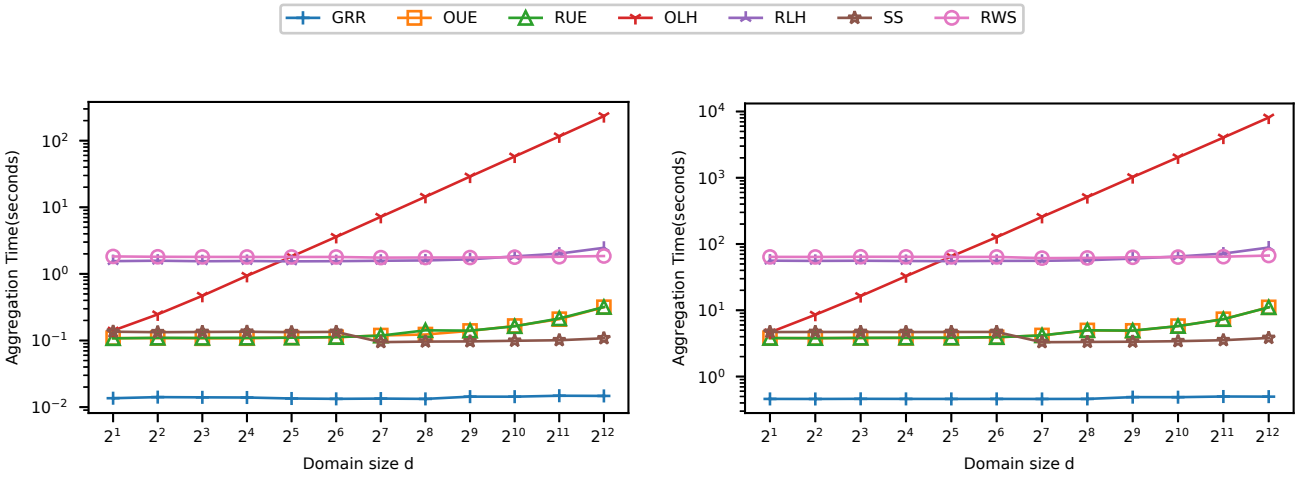
Figure 7 shows the aggregation times of all protocols. The running times on Zipf's dataset are very similar to the results on Taxi dataset, except for the numerical differences. The reason is that the aggregation time is linear with n for all protocols and $n = 100,000$ and $3,582,628$ in Zipf's and Taxi datasets, respectively.

We can find that the running time of OLH grows the fastest with d, while GRR has the shortest running time, which complies with our theoretical analysis. The proposed RLH is de-



(a) Zipf's (b) Taxi

Figure 6: Empirical MSE results on top k values from 2 to 30 with $d = 2^7$ and $\varepsilon = 4$



(a) Zipf's (b) Taxi

Figure 7: Aggregation Times of different frequency protocols with $\varepsilon = 4$

signed to reduce the running time in aggregation. Compared with OLH, the running time of the RLH grows much slower as d grows. When $d = 2^{12}$ in Taxi dataset, the running time of RLH is 88.5s while the running time of OLH is 8101.2s (about 2.25 hours), 90 times more than RLH. This gap will increase further as d increases. When d is small, the running times of RLH and RWS are the most; this is because Numpy and its random number generator are optimized for large amounts of data (i.e., they perform poorly with small sizes of data) and the cost of initialization and finalization in regenerating the subsets dominates when d is small. However, the proposed RLH and RWS have much lower runtime growth rates than OLH, especially RWS, which is even lower than OUE and RUE. As expected, OUE and RUE achieve almost identical running times in aggregation since their implementations are identical except for different parameters. SS achieves a similar running time to OUE and RUE, but its growth rate is much lower. While the running times of SS, RWS, OUE, and RUE are all linearly with d, the running times of SS and RWS are more precisely linear with $k = \frac{d}{e^{\varepsilon}+1}$, which is smaller than d. There is a drop between $d = 2^6$ and 2^7 in the running time of SS. We believe the time drop is due to the various performances in Numpy's array with different subset sizes k and we can find that subset size $k = 1$ and 2 at $d = 2^6$ and 2^7, respectively. In addition, as $k = 1$ when $d <= 2^6$, users can report the subset instead of the seed to eliminate the cost of

regenerating users' subsets in the aggregation of RWS. In this case, the running time of RWS is similar to that of SS.

5.5 Discussion

In summary, we run experiments on synthetic and real-world datasets to verify the correctness of our analysis and evaluate the accuracy and aggregation time of all frequency protocols.

Experimental results demonstrate that the empirical MSE matches our analytical MSE very well. The proposed RUE and RLH achieve better accuracy for small domains than OUE and OLH by enhancing the accuracy of frequent values. RLH is also much faster in aggregation than OLH for large domains. The proposed RWS achieves the same accuracy as SS, which is optimal for all protocols regardless of domain size d and privacy budget ε. The growth rate of aggregation time in RWS is similar to SS and lower than OUE and RUE as d increases. The communication cost of RWS is much lower than SS for large domains. Therefore, RWS achieves optimal accuracy with low computation and communication costs simultaneously.

6 Related Work

Frequency estimation is a fundamental task in LDP protocols. We present a detailed comparison of four state-of-the-art LDP frequency protocols in this paper. GRR [12] performs best for small domains, but its accuracy worsens rapidly for large domains. OUE [20] optimizes the parameters of Basic RAPPOR [8] and achieves the optimal accuracy for large domains, but suffers high communication costs. Compared with OUE, OLH [20] reduces the communication cost for large domains but with complex server-side computation costs. Additionally, OUE and OLH are not as accurate as GRR for small domains. SS [15, 18, 27] achieves the optimal accuracy for both small and large domains but suffers high communication costs as OUE. We partially address the weaknesses of these protocols and propose the RWS protocol that achieves optimal accuracy with low computation and communication costs.

Several other frequency protocols have been proposed recently but still with tradeoffs. Hadamard Response [1] is similar to SS with a fixing $k = d/2$ and reduces the communication cost with Hadamard transform. Since the optimal $k = d/(e^{\varepsilon}+1)$ in SS, the accuracy of Hadamard Response is unsatisfactory. Fast Local Hashing [6] uses a limited range of hash functions to achieve computational gains but sacrifices some accuracy compared with OLH. The Wheel Mechanism [19] is a continuous version of LH protocols, thus inheriting all the advantages and disadvantages of OLH.

Wang [20] proposes a pure framework that can be used to conveniently analyze frequency protocols that satisfy the pure definition. Since Definition 1 of ε-LDP requires that frequency protocols treat each value in the domain equally, protocols should ensure some kind of interchangeability and symmetry (i.e., pure) among the values in pursuit of optimal accuracy. There are only a few non-pure frequency protocols. For instance, RAPPOR (Basic RAPPOR with Bloom filters) is not pure due to collisions caused by Bloom filters. Without Bloom filters, Basic RAPPOR is pure. RAPPOR integrates Bloom filters to handle non-categorical values but with accuracy loss.

In real-world deployments, Google implements RAPPOR in Chrome for browser telemetry. Basic RAPPOR collects statistics on categorical values (client properties in browser telemetry, numerical and ordinal values with buckets). For non-categorical values, Bloom filters technique is integrated to map the values to a processable domain. Our proposed RWS can replace the Basic RAPPOR in RAPPOR to collect the categorical data straightforwardly. For non-categorical values, RWS can be applied to each hash result after Bloom filters by splitting ε. Due to the lower communication cost of RWS than Basic RAPPOR for large domains, the domain size of hash results can be larger to reduce collisions. PROCHLO [4] implements the ESA (Encode, Shuffle, Analyze) design for privacy-preserving software monitoring. By introducing the trusted Shuffler, PROCHLO provides high utility while protecting user privacy. Encoders in PROCHLO may utilize LDP frequency protocols for categorical values, in which case RWS can be used to improve utility.

7 Conclusion

This paper presents a further study on frequency estimation under local differential privacy. A universal equation is introduced to evaluate the analytical MSE of LDP frequency protocols. It enables us to assess the accuracy of frequency protocols precisely and conveniently. We then quantitatively analyze the advantages and disadvantages of existing protocols in terms of accuracy, computation cost, and communication cost. RUE and RLH are proposed to enhance the accuracy of OUE and OLH for small domain sizes. RLH also reduces the aggregation time massively compared with OLH. We fuse the key ideas of LH and SS and further propose the RWS protocol, which achieves optimal accuracy with low computation and communication costs simultaneously. Experimental results verify the correctness of our analysis and demonstrate the advantages of our proposed protocols in both synthetic and real-world datasets.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China (U22B2026) and the Big Data Computing Center of Southeast University.

Ethics Considerations

We have read the ethics considerations discussions in the conference call for papers, the detailed submissions instructions, and the guidelines for ethics document. We believe this research was done ethically, and that the team's next-step plans are ethical. As we use the publicly available dataset from TLC Trip Record Data which is anonymized for use and we do not collect any personal information, we determined that there are no ethical concerns. For the innovations with both positive and negative potential outcomes, this paper aims to improve the utilities of LDP frequency protocols. With better accuracy in frequency protocols, the top technology companies can learn the data distribution among users with a smaller privacy budget or sample size. As the privacy budget ε determines users' privacy guarantee, users may benefit from a small ε, which means more privacy-preserving. As long as the privacy budget remains the same, improving the accuracy of protocols will not affect users' privacy guarantee. All reviewers agree that there are no risks associated with this paper, and if any, are appropriately mitigated.

Open Science

We mainly use Python 3.10.4 with Numpy to run experiments and Matplotlib to draw the results. Pyarrow is used to read the taxi dataset from TLC Trip Record Data. We run the experiments on a PC with AMD Ryzen 9 7950X and 64GB memory and the code should be compatible with any recent versions. To ensure the code runs properly, it is recommended to use a computer with at least 32 GB of memory. Our artifact is available on GitHub at https://github.com/SEUNICK/LDP_Frequency_Protocols and Zenodo at https://zenodo.org/records/14715748.

The components of the artifact are as follows:

- *fp:* Source code folder of all frequency protocols in this paper.
- *yellow_tripdata_2024-03.parquet:* The taxi dataset from TLC Trip Record Data.
- *main.py:* Main entrance for experiments (using multiprocessing to save time and memory) and save results in the *results* folder.
- *runtime.py:* Get the aggregation runtime (without multiprocessing) of every frequency protocol in this paper.
- *drawxx.py:* Draw the experimental results into figures in this paper and save them in the *draw* folder. More specifically:
  - *drawAnaMSE.py:* Draw Figure 2.
  - *drawDis.py:* Draw Figure 3.
  - *drawAnaEmpMSE.py:* Draw Figure 4.
  - *drawEmpMSE.py:* Draw Figure 5.
  - *drawTopk.py:* Draw Figure 6.
  - *drawRuntime.py:* Draw Figure 7.
- *README.md*

References

[1] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 1120–1129. PMLR, 16–18 Apr 2019. https://proceedings.mlr.press/v89/acharya19a.html.

[2] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2285–2293, Red Hook, NY, USA, 2017. Curran Associates Inc. https://dl.acm.org/doi/10.5555/3294771.3294989.

[3] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 127–135, New York, NY, USA, 2015. https://dl.acm.org/doi/10.1145/2746539.2746632.

[4] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, page 441–459. Association for Computing Machinery, 2017. https://dl.acm.org/doi/10.1145/3132747.3132769.

[5] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. Answering range queries under local differential privacy. *Proc. VLDB Endow.*, 12(10):1126–1138, Jun 2019. https://dl.acm.org/doi/10.14778/3339490.3339496.

[6] Graham Cormode, Samuel Maddock, and Carsten Maple. Frequency estimation under local differential privacy. *Proc. VLDB Endow.*, 14(11):2046–2058, Jul 2021. https://dl.acm.org/doi/10.14778/3476249.3476261.

[7] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. NIPS'17, page

3574–3583, Red Hook, NY, USA, 2017. Curran Associates Inc. https://dl.acm.org/doi/10.5555/3294996.3295115.

[8] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 1054–1067, New York, NY, USA, 2014. https://dl.acm.org/doi/10.1145/2660267.2660348.

[9] Huiyu Fang, Liquan Chen, Yali Liu, and Yuan Gao. Locally differentially private frequency estimation based on convolution framework. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2208–2222, Los Alamitos, CA, USA, May 2023. https://ieeexplore.ieee.org/document/10179389.

[10] Naoise Holohan, Douglas J. Leith, and Oliver Mason. Optimal differentially private mechanisms for randomised response. *IEEE Transactions on Information Forensics and Security*, 12(11):2726–2735, 2017. https://ieeexplore.ieee.org/document/7967624.

[11] Jinyuan Jia and Neil Zhenqiang Gong. Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2008–2016, 2019. https://ieeexplore.ieee.org/document/8737527.

[12] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2436–2444. JMLR.org, 2016. https://dl.acm.org/doi/10.5555/3045390.3045647.

[13] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Škoric. Estimating numerical distributions under local differential privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 621–635, New York, NY, USA, 2020. https://dl.acm.org/doi/10.1145/3318464.3389700.

[14] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 192–203, New York, NY, USA, 2016. https://dl.acm.org/doi/10.1145/2976749.2978409.

[15] Pengzhan Wang Yiwen Nie Hongli Xu Wei Yang Xiang-Yang Li Chunming Qiao Shaowei Wang, Liusheng Huang. Mutual information optimally local private discrete distribution estimation, 2016. https://arxiv.org/abs/1607.08025.

[16] New York City Taxi and Limousine Commission. Tlc trip record data, 2024. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[17] Apple Differential Privacy Team. Learning with privacy at scale, 2017. https://machinelearning.apple.com/research/learning-with-privacy-at-scale.

[18] Shaowei Wang, Liusheng Huang, Yiwen Nie, Xinyuan Zhang, Pengzhan Wang, Hongli Xu, and Wei Yang. Local differential private data aggregation for discrete distribution estimation. *IEEE Transactions on Parallel and Distributed Systems*, 30(9):2046–2059, 2019. https://ieeexplore.ieee.org/document/8640266.

[19] Shaowei Wang, Yuqiu Qian, Jiachun Du, Wei Yang, Liusheng Huang, and Hongli Xu. Set-valued data publication with local privacy: tight error bounds and efficient mechanisms. *Proc. VLDB Endow.*, 13(8):1234–1247, April 2020. https://dl.acm.org/doi/10.14778/3389133.3389140.

[20] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, Vancouver, BC, August 2017. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao.

[21] Tianhao Wang, Bolin Ding, Jingren Zhou, Cheng Hong, Zhicong Huang, Ninghui Li, and Somesh Jha. Answering multi-dimensional analytical queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 159–176, New York, NY, USA, 2019. https://dl.acm.org/doi/10.1145/3299869.3319891.

[22] Tianhao Wang, Ninghui Li, and Somesh Jha. Locally differentially private frequent itemset mining. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 127–143, Los Alamitos, CA, USA, May 2018. https://ieeexplore.ieee.org/document/8418600.

[23] Tianhao Wang, Ninghui Li, and Somesh Jha. Locally differentially private heavy hitter identification. *IEEE Transactions on Dependable and Secure Computing*, 18(2):982–993, 2021. https://ieeexplore.ieee.org/document/8758350.

[24] Tianhao Wang, Milan Lopuhaä-Zwakenberg, Zitao Li, Boris Skoric, and Ninghui Li. Locally differentially private frequency estimation with consistency. In *NDSS'20: Proceedings of the NDSS Symposiu*, San Diego, CA, USA, 23-26 Feb 2020. https://doi.org/10.14722/ndss.2020.24157.

[25] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. https://www.jstor.org/stable/2283137.

[26] Mengmeng Yang, Taolin Guo, Tianqing Zhu, Ivan Tjuawinata, Jun Zhao, and Kwok-Yan Lam. Local differential privacy and its applications: A comprehensive survey. *Computer Standards Interfaces*, 89:103827, 2024. https://www.sciencedirect.com/science/article/pii/S0920548923001083.

[27] Min Ye and Alexander Barg. Optimal schemes for discrete distribution estimation under local differential privacy. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 759–763, 2017. https://ieeexplore.ieee.org/document/8006630.