Frequency Estimation Under Multiparty Differential Privacy: One-shot and Streaming

Ziyue Huang
Hong Kong University of Science and Technology
zhuangbq@cse.ust.hk

Yuan Qiu
Hong Kong University of Science and Technology
yqiuac@cse.ust.hk

Ke Yi
Hong Kong University of Science and Technology
yike@cse.ust.hk

Graham Cormode
University of Warwick
g.cormode@warwick.ac.uk

ABSTRACT

We study the fundamental problem of frequency estimation under both privacy and communication constraints, where the data is distributed among k parties. We consider two application scenarios: (1) one-shot, where the data is static and the aggregator conducts a one-time computation; and (2) streaming, where each party receives a stream of items over time and the aggregator continuously monitors the frequencies. We adopt the model of multiparty differential privacy (MDP), which is more general than local differential privacy (LDP) and (centralized) differential privacy. Our protocols achieve optimality (up to logarithmic factors) permissible by the more stringent of the two constraints. In particular, when specialized to the ε-LDP model, our protocol achieves an error of $\sqrt{k}/(e^{\Theta(\varepsilon)}-1)$ using $O(k\max\{\varepsilon, \log\frac{1}{\varepsilon}\})$ bits of communication and $O(k\log u)$ bits of public randomness, where u is the size of the domain.

PVLDB Reference Format:
Ziyue Huang, Yuan Qiu, Ke Yi, and Graham Cormode. Frequency Estimation Under Multiparty Differential Privacy: One-shot and Streaming. PVLDB, 15(10): 2058 - 2070, 2022.
doi:10.14778/3547305.3547312

1 INTRODUCTION

Providing formal (differential) privacy guarantees for sensitive data in different forms has been a major focus for the data management community in recent years, e.g., [6, 10, 40, 49, 52, 54, 55]. Consider a distributed database with k parties, where each party has some data. We study two settings for privacy-preserving data analysis where an aggregator wants to obtain some aggregated statistics over all data held by the parties: (1) one-shot, where the data is static and the parties conduct a one-time computation; and (2) streaming, where each party receives a stream of items over time and the aggregator wants to monitor the aggregated statistics continuously. These settings are motivated by real-world applications. For example, where each hospital holds some medical records which some third-party public health organization wishes to analyze, and it is the responsibility of the hospital to protect the privacy of their patients.

Proceedings of the VLDB Endowment, Vol. 15, No. 10 ISSN 2150-8097.
doi:10.14778/3547305.3547312

As another example in the distributed streaming model, from the browsing activities of many users, a search engine would like to continually monitor the most popular URLs over the past week without privacy breach [13]. In addition to the privacy constraint, the communication cost often becomes a major bottleneck which limits the scalability of distributed machine learning tasks. For instance, in federated computing, the data is distributed among many mobile devices connected by bandwidth-limited wireless links and the communication is expensive and unreliable [38]. This poses a dual-challenge to the design of the protocol. Recently, there has been a lot of interest in designing protocols under both constraints [1–3, 17, 34, 50]. The most important result obtained in this paper is that, for the frequency estimation problem, it is possible to achieve optimality permissible by the more stringent of the two constraints, or equivalently, the less stringent constraint can be satisfied *for free*.

1.1 Problem Formulation

Multiparty Differential Privacy. In the settings above where multiple parties each possess some data, the standard notion of privacy is *multiparty differential privacy (MDP)* [13, 41, 44, 47]. Denote the parties as $P_1, \ldots, P_k$, and the aggregator as P_0. Suppose each party P_i, $i \geq 1$, has a multiset $\mathcal{D}_i$ of $n_i \geq 1$ items, drawn from a universe $\mathcal{U}$ of size u. Let $N = \sum_{i=1}^{k} n_i$. We use $[n]$ to denote $\{1, \ldots, n\}$.

Definition 1 (Multiparty Differential Privacy [47]). *Let P be a protocol involving parties $(P_0, P_1, \ldots, P_k)$, where P_i has input dataset $\mathcal{D}_i \in \mathcal{U}^{n_i}, i \in [k]$, while P_0 has no input. Consider any party P_i, $i = 0, 1, \ldots, k$, and let A be an adversary controlling $P_{-i} = \{P_0, \ldots, P_k\} - \{P_i\}$. We use $\text{View}_{P_{-i}}(P_{-i} \leftrightarrow (P_0, \ldots, P_k)(\mathcal{D}))$ to denote the random variable that includes everything that A sees when participating in the protocol on input dataset $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_k)$. We say that P is ε-differentially private if for every $i \in [k]$ and every two neighboring datasets $\mathcal{D}, \mathcal{D}' \in (\mathcal{U}^{n_1}, \ldots, \mathcal{U}^{n_k})$ that differ on one item in P_i's input, the following holds for every set T:*

$$\Pr[\text{View}_{P_{-i}}(P_{-i} \leftrightarrow (P_0, \ldots, P_k)(\mathcal{D})) \in T]$$
$$\leq e^{\varepsilon} \cdot \Pr[\text{View}_{P_{-i}}(P_{-i} \leftrightarrow (P_0, \ldots, P_k)(\mathcal{D}')) \in T].$$

The MDP definition above allows arbitrary interactions among the parties. All the protocols designed in this paper, however, only use one-way communications from the parties to a designated aggregator, who can be any particular party. On the other hand, the lower bounds we match under communication or privacy constraints hold even for protocols using arbitrary interactions.

Connections between MDP and DP/LDP. The MDP model is more general than other popular privacy models such as *local differential privacy (LDP)* and *(central) differential privacy (DP)*: Setting $n_i = 1$ for all $i \in [k]$ yields the former while setting $k = 1$ yields the latter. It also generalizes the two-party (i.e., $k = 2$) model [41]. Compared with LDP, MDP more generally applies to the situation where each party holds more than one item, and ensures item-level privacy (the sanitized messages are insensitive to the change of a single item on any party). One can run an LDP protocol in the MDP model, but it would result in $\Omega(\sqrt{N})$ error (for constant ε) and $\Omega(N)$ communication. In the MDP model, we are mostly interested in the case $N \gg k$. As we will see, our protocol can achieve $O(\sqrt{k})$ error with $O(N)$ communication, or $O(\sqrt{N})$ error with $O(\sqrt{Nk})$ communication, both of which are much better than the comparable LDP protocols, and the improvement is more significant for smaller k (with respect to N). Furthermore, the streaming version of MDP naturally corresponds to the event-level privacy model in the streaming setting studied in [13, 14, 28]. Section 9 of [47] gives a more extensive review of MDP.

Frequency Estimation. Denote the frequency of item $j \in [u]$ on party i by $x_{i,j}$, which we call the *local count*. In the *frequency estimation* problem, the aggregator wishes to obtain a synopsis, which can be used to extract an estimate of the *global count* $y_j = \sum_{i=1}^{k} x_{i,j}$ for any $j \in [u]$. As with prior work [7, 22, 50], we aim at an additive error guarantee that holds for a single query with probability $1 - \beta$. A vectorized view of the problem is to consider the local counts $\{x_{i,j}\}_j$ at party i as a vector $\mathbf{x}_i \in \mathbb{N}^u$, and we want to obtain a $\tilde{\mathbf{y}}$ that minimizes $\|\tilde{\mathbf{y}} - \mathbf{y}\|_\infty$, where $\mathbf{y} = \sum_i \mathbf{x}_i$. Setting $\beta = O(1/u)$ plus a union bound converts any error guarantee of the former into one of the latter. Table 1 summarizes the notations used in the paper.

Table 1: Notations used in the paper.

Notation	Meaning
ε	Differential privacy budget
k	Number of parties
u	Domain size
s	Average message size
$x_{i,j}$	Frequency of item j held by party i
n_i	Number of items held by party i
y_j	Total frequency of item j
N	Total number of items
η, ζ	Noises drawn from geometric distribution
g, h	Hash functions used in count-sketch
R	Number of rows in count-sketch
$S_i^{\text{hi}}/S_i^{\text{lo}}$	Local heavy/light hitters at party i
w	Size of the sliding window
m	Number of epochs in the stream
b	Number of time steps in each epoch
β	Failure probability

1.2 Our Results and Prior Work

We describe our results below and compare them with prior work. We often use the $\tilde{O}$ notation to suppress polylogarithmic factors. Please also see Figure 1 for an overview of the results.

One-shot Protocols. Our main one-shot result is an MDP protocol that achieves an error of $\tilde{O}(N/(\sqrt{k}s)) + \sqrt{k}/(e^{\tilde{\Theta}(\varepsilon)} - 1)$ with $\tilde{O}(ks)$ communication, for any given $\varepsilon > 0$ and $s \geq 1$, which controls the communication-utility trade-off. Observe that the first error term is communication-dependent while the second term is privacy-bound. Setting $s = N/k$ reduces the first term to $\tilde{O}(\sqrt{k})$, and the total error is dominated by the second term for $\varepsilon = O(1)$. Note that, regardless of the communication cost, the error has to be $\Omega(\sqrt{k}/\varepsilon)$ for $\varepsilon = O(1)$ [9], even for the special case where each party has just one item (i.e., the LDP model). For applications where saving communication is important, such as collecting data from sensor networks and mobile phones, we may use a smaller s. In this case, the communication-bound term dominates, which also matches the lower bound in the non-private, blackboard communication model under the $\tilde{O}(ks)$ communication constraint [37]. Thus, our one-shot MDP protocol achieves the optimal error subject to the both the communication and the privacy constraint for the case $\varepsilon = O(1)$.

In the LDP model, which is a degenerate case of MDP, we are able to refine the analysis of our MDP protocol. The refined analysis removes the communication-bound term, as well as the logarithmic factor in the exponent $\tilde{\Theta}(\varepsilon)$ in the privacy-bound term. More precisely, we show that our LDP protocol achieves an error of $\tilde{O}\left(\sqrt{k}/(e^{\varepsilon/4} - 1)\right)$ with $O(k \max\{\varepsilon, \log \frac{1}{\varepsilon}\})$ bits of communication while using $O(k \log u)$ bits of public randomness[1]. This matches the recent result of [32][2] in terms of error, while further improving on the communication cost, which is $O(k \max\{\log u, \varepsilon, \log \frac{1}{\varepsilon}\})$ bits. We also prove a lower bound of $\Omega(\sqrt{k}/e^{\varepsilon/2})$ for $\varepsilon = O(\log k)$, regardless of communication cost, improving the previous lower bound $\tilde{\Omega}(k^{1/3})$ [33]. Now, combined with the lower bound $\Omega(\sqrt{k}/\varepsilon)$ for the $\varepsilon = O(1)$ case [9], we have essentially closed the gap for the problem for all meaningful values of ε (note that $\varepsilon \geq \Omega(\log k)$ offers almost no meaningful privacy protection), up to a constant-factor difference in ε.

Frequency estimation under the LDP model has been extensively studied in the literature. The earliest protocol, RAPPOR [31, 50], achieves error $O(\sqrt{k}/(e^{\varepsilon/4} - 1))$ with a communication cost $O(ku)$. Such a large communication cost, which is proportional to the universe size, cannot be used for situations such as estimating the frequencies of popular English words (in this case, $u = 26^\ell$ where ℓ is the maximum length of keywords). Subsequent works have aimed at reducing the communication cost to be logarithmic or even independent of u. The *Hadamard Randomized Response (HRR)* algorithm [4, 43] reduces it to $O(k)$ bits while using $O(k \log u)$ bits of public randomness, but the error becomes $\tilde{O}\left(\sqrt{k}/\min\{\varepsilon, 1\}\right)$, which is optimal only for the high privacy regime $\varepsilon < 1$ [9]. For the general privacy regime $\varepsilon = \Omega(1)$ [2, 39], Chen et al. [17] present an

[1] These random bits can be communicated if public randomness is not available.

[2] In fact, we obtained our result concurrently and independently of [32] via a very different approach. Furthermore, [32] does not work in the MDP model, which is our main focus.

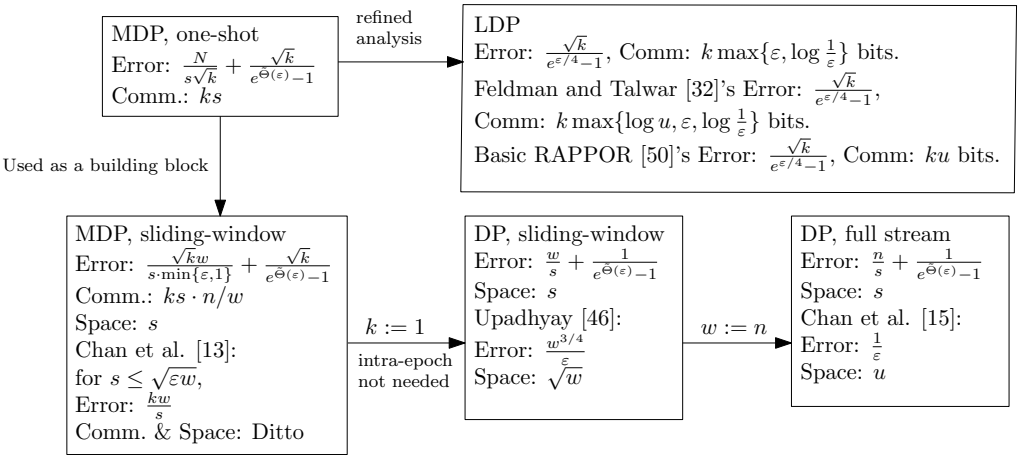


Figure 1: Overview of our results, where s can be any positive integer. All bounds suppress polylogarithmic factors.

improved algorithm that achieves error $O\left(\sqrt{k/\varepsilon}\right)$ using $O(k\varepsilon)$ bits of communication. Finally, the optimal error has been achieved by our protocol and the one in [32] as mentioned above, using $O(k\varepsilon)$ bits and $O(k(\varepsilon + \log u))$ bits of communication, respectively.

Streaming Protocols. The MDP model has a natural streaming version. As with prior work [13, 23, 24], we adopt a *synchronous* timing model, where time is divided into discrete steps, and one item arrives at each party in every time step. Messages sent within a time step all arrive before the next time step. In practice, the parties' clocks might be out-of-sync and messages can be delayed. In this case, we can include timestamps in the messages to simulate the execution, which is particularly easy for our protocol as it only uses one-way messages from the parties to the aggregator. The assumption that one item must arrive at each time step is also without loss of generality. If nothing arrives at a party in a time step, this can be treated as a dummy item.

Perhaps the most useful streaming model is the *sliding-window* model. Here, the goal is for the aggregator to maintain a synopsis such that the global count of any item j, counting all items that have arrived in the last w time steps, can be estimated. However, all messages sent during the entire streaming period, not just those sent in the sliding window, must collectively be ε-DP as in the one-shot MDP model. We present an ε-MDP protocol for this problem that, for any integer $s \ge 1$, achieves error $\tilde{O}(w\sqrt{k}/(s\cdot\min\{\varepsilon,1\})) + \sqrt{k}/(e^{\tilde{\Theta}(\varepsilon)}-1)$ with $\tilde{O}(ks\cdot n/w)$ communication, while using $\tilde{O}(s)$ space on each party, where n is the total number of time steps. Note that the one-shot problem is a special case of this problem, by just setting $n := w, N := kw$, and asking for the synopsis only at the end of the stream. Compared with our one-shot result, the only difference is that the communication-bound error term has an extra $1/\min\{\varepsilon,1\}$ factor. Thus, the communication-error trade-off of our sliding window protocol is still optimal for $\varepsilon = O(1)$. For $\varepsilon = \Theta(1)$, the previous result for this problem [13] gave an error of $\tilde{O}(kw/s)$ under the same communication budget, which is a $\sqrt{k}$-factor from optimal. More importantly, the largest s that can be supported by the protocol of [13] is $O(\sqrt{w})$, which means that the minimum error achievable (regardless of communication) is $\tilde{O}(k\sqrt{w})$. This is a $\sqrt{kw}$-factor from $\tilde{O}(\sqrt{k})$, the smallest error permissible by the privacy constraint, which can be achieved by our protocol by setting $s := w$.

For (private or non-private) streaming algorithms, an important measure of complexity is space. To see that our space-error trade-off is also optimal, simply consider the degenerate case where $k = 1$. This particular case has actually been recently studied by [46], where an algorithm with error $\tilde{O}(w^{3/4}/\varepsilon)$ and space $\tilde{O}(\sqrt{w})$ is presented. When degenerated to the $k = 1$ case, our protocol achieves $\tilde{O}(\frac{w}{s}) + 1/(e^{\tilde{\Theta}(\varepsilon)}-1)$ error with $\tilde{O}(s)$ space. To compare with [46], just set $s = \sqrt{w}$, which yields $\tilde{O}(\sqrt{w}) + 1/(e^{\tilde{\Theta}(\varepsilon)}-1)$ error. In fact, the protocol in [13] yields the same space-error trade-off as ours (but with a limited range $s \le \sqrt{\varepsilon w}$) for the $k = 1$ case, which was overlooked in [46]. In an analogy to the communication-error trade-off, the space-error trade-off is also determined by space or privacy, whichever is more stringent. The space-bound term $\tilde{O}(\frac{w}{s})$ is optimal (assuming $u \ge w$) by well-known lower bounds in the (non-private) streaming literature [11], while the optimality of the privacy-bound term follows from the centralized DP lower bound $\tilde{\Omega}(\frac{1}{\varepsilon})$ for $\varepsilon = O(\log u)$, even for the one-shot problem [47]. Thus, our protocol achieves the optimal space-error trade-off for the full range $1 \le s \le w$ for $\varepsilon = O(\log u)$. On the other hand, [13] achieved optimality only for a partial range $1 \le s \le \sqrt{\varepsilon w}$, while the results of [46] are not optimal. Finally, our protocol spends $\tilde{O}(1)$ time to process each item, so it is time-optimal as well.

Further Applications. Similar to [7, 18, 20], by using our frequency estimation protocol with a dyadic decomposition of the universe, we can also solve many related problems such as heavy hitters, quantiles, and orthogonal range counting (in constant dimensions), at the cost of some extra polylogarithmic factors.

1.3 Other Related Work

We briefly mention results in other models of privacy that are relevant to our study. Most closely related are the notions of *continual observation*, and *pan privacy*, which consider privacy against an adversary who may observe a snapshot of the algorithm's internal state (pan privacy), or when the algorithm continually publishes

updates based on new readings (continual observation). The latter of these most closely matches our (distributed) streaming results, and we leverage similar techniques, such as expressing partial aggregations within a tree structure [15, 28]. More specifically, by setting $w := n$, the sliding-window model degenerates into the problem of monitoring the full stream, i.e., continual observation. Existing solutions for this problem achieves $\tilde{O}(1/\varepsilon)$ error with $\tilde{O}(u)$ space [15]. Our protocol achieves the same error with space $\tilde{O}(n)$, which can be much smaller than $\tilde{O}(u)$. In addition, we provide a full-range space-error trade-off.

We have mentioned the model of Local Differential Privacy (LDP), which corresponds to MDP with $n_i = 1$ for all i in the one-shot setting. The most impactful work in this model is concerned with frequency estimation, and finding heavy hitters, based on building "frequency oracles" from each site's message to estimate item frequencies, as in our setting [4, 8, 27, 31]. Further work has studied a wide variety of data analysis and ML tasks, such as multidimensional statistics, language models and classifiers; we refer the reader to surveys on this topic [19, 51, 53].

Most recently, additional models have been proposed which aim to achieve improved privacy-accuracy tradeoffs by making stronger assumptions. The shuffle model assumes that the identity of the sender can be fully disassociated from the messages sent, either by a trusted "shuffler" entity, or through a cryptographic mix network [5, 30].

2 PRELIMINARIES

2.1 Differential Privacy

The standard (centralized) differential privacy model is a special case of MDP, but we state its definition again for clarity. Let $\mathcal{D} \sim \mathcal{D}'$ denote two neighboring datasets, which differ by one item.

Definition 2 (Differential Privacy [29]). *For $\varepsilon > 0$, an algorithm $\mathcal{M}$ is ε-differentially private (DP) if for any neighboring datasets $\mathcal{D} \sim \mathcal{D}'$ and any $S \subseteq \mathsf{Range}(\mathcal{M})$,*

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \le e^{\varepsilon} \cdot \Pr[\mathcal{M}(\mathcal{D}') \in S].$$

Note that the MDP model degenerates to this definition by setting $k := 1$: P_1 runs $\mathcal{M}$ on $\mathcal{D}_1$ and sends $\mathcal{M}(\mathcal{D}_1)$ to P_0.

For a numeric query q, one common DP mechanism is to add noise drawn from a symmetric geometric distribution (also referred to as "discrete Laplace") [12, 35] calibrated to $\mathrm{GS}_q := \max_{\mathcal{D}\sim\mathcal{D}'} |q(\mathcal{D}) - q(\mathcal{D}')|$, which is known as the (global) *sensitivity* of q.

Definition 3 (Symmetric Geometric Distribution [12, 35]). *Let $\alpha > 1$. We denote by $\mathrm{Geom}(\alpha)$ the symmetric geometric distribution that takes integer values such that the probability mass function at l is $\frac{\alpha-1}{\alpha+1} \cdot \alpha^{-|l|}$.*

The following properties of the symmetric geometric distribution are useful. Let $X \sim \mathrm{Geom}(\alpha)$:

(1) $\mathsf{E}[X] = 0$; $\mathsf{Var}[X] = 2\alpha/(\alpha-1)^2 = O(1/\log^2 \alpha)$.
(2) For every $d > 0$, $\Pr[|X| > d] \le \alpha^{-d}$.
(3) The mechanism $\mathcal{M}(\mathcal{D}) := q(\mathcal{D}) + X$ is ε-DP by setting $\alpha = \exp(\varepsilon/\mathrm{GS}_q)$. Note that in this case, $\mathsf{Var}[X] = O((\mathrm{GS}_q/\varepsilon)^2)$.

The LDP model is another special case of MDP by setting $n_i = 1$ for all $i \in [k]$. The *Hadamard Randomized Response (HRR)* algorithm [4, 43] can be used to solve the frequency estimation problem under LDP. Assume, without loss of generality, that u is a power of 2, and recall that the Hadamard matrix can be defined recursively as

$$H_u = \begin{bmatrix} H_{u/2} & H_{u/2} \\ H_{u/2} & -H_{u/2} \end{bmatrix},$$

where $H_1 = [1]$. Each party i samples an index r_i u.a.r. from $[u]$, encodes her item v_i into a single bit $H[r_i, v_i]$, and then sends it to the aggregator via randomized response. Specifically, each user sends a one-bit message $\mathcal{M}(v_i)$ to the aggregator (if there is no shared randomness, the random index r_i should also be sent using $\log u$ bits), where

$$\mathcal{M}(v_i) = \begin{cases} H[r_i, v_i], & \text{w.p. } \frac{e^{\varepsilon}}{e^{\varepsilon}+1}; \\ -H[r_i, v_i], & \text{otherwise.} \end{cases}$$

The frequency estimator (at the aggregator side) for any item v is $\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1} \sum_{i=1}^{k} \mathcal{M}(v_i) \cdot H[r_i, v]$. The error guarantee of HRR is $O(\sqrt{k \log(1/\beta)}/\min\{\varepsilon, 1\})$ which holds for a single query with probability $1 - \beta$.

Note that the MDP/LDP model allows arbitrary interactions among the parties, and the lower bounds [9] hold under this setting. However, most of existing protocols (including ours) use one-way messages, except for broadcasting some public parameters to all parties before the protocol starts. In this case, it is sufficient for each party to run a ε-DP mechanism $\mathcal{M}$ on her dataset $\mathcal{D}_i$ and send $\mathcal{M}(\mathcal{D}_i)$ to the aggregator. The resulting protocol then trivially satisfies MDP/LDP against active adversaries. It is worth pointing out that one can relax the MDP model by only allowing the adversary to control a smaller number of parties. In this case, one may achieve errors lower than the LDP lower bound of $\Omega(\sqrt{k}/\varepsilon)$ with interactive protocols [45].

2.2 Count Sketch

The *count sketch* [16] of a vector $\mathbf{x}$ of size u is another vector $C(\mathbf{x})$ of size s,

$$C(\mathbf{x})[j] = \sum_{i \in [u]: h(i) = j} g(i) x_i, \quad j = 1, \ldots, s,$$

where $h : [u] \to [s]$ and $g : [u] \to \{-1, +1\}$ are two hash functions. For our analysis, we assume h is pairwise-independent while g is truly random. In some cases, the latter assumption can also be relaxed to pairwise-independence.

The count sketch can be used to extract *point estimates*. For any $i \in [u]$, an estimator for x_i is $\tilde{x}_i = g(i) \cdot C(\mathbf{x})[h(i)]$. It is known that $\mathsf{E}[\tilde{x}_i] = x_i$ and $\mathsf{Var}[\tilde{x}_i] \le \|\mathbf{x}\|_2^2/s$. So by the Chebyshev inequality, the error $|\tilde{x}_i - x_i|$ is $O(\|\mathbf{x}\|_2/\sqrt{s})$ with constant probability, which is an ℓ_2 error guarantee. Meanwhile, the count-sketch also enjoys an ℓ_1 error guarantee that $|\tilde{x}_i - x_i| = O(\|\mathbf{x}\|_1/s)$ with constant probability (Chapter 3.5 in [26]). Note that these two error bounds are in general incomparable. The success probability can be amplified to $1-\beta$ via a standard median trick: creating $O(\log(1/\beta))$ independent instances and returning the median of the estimators. This way, a count sketch can be viewed as a matrix of $O(\log(1/\beta))$ rows and s columns.

3 ONE-SHOT FREQUENCY ESTIMATION

In this section, we build up our approach. The starting point is a relatively simple protocol based on gathering a carefully configured sketch from each party (Section 3.1). This is sufficient to give an accurate result for the general MDP case ($n_i \geq 1$). However, sending a large sketch can be costly when parties have few items, so we show how to reduce the sketch size for the LDP ($n_i = 1$) case in Section 3.2, and to achieve a better communication cost in the case of variable input sizes with an improved protocol in Section 3.3.

3.1 Our Basic Protocol

In addition to the privacy parameter ε and the failure probability β, our protocol uses a parameter $s \geq 1$, which determines the average message size of each party.

Algorithm on Each Party. On each party i, from the local counts $\mathbf{x}_i$ we build a count sketch C_i of R rows and $s_i := \lceil ks \cdot n_i/N \rceil$ columns, where R is the nearest odd number to $\log \frac{3k}{\beta}$. Next, the party perturbs each counter $C_i[r, c]$, $r \in [R], c \in [s_i]$ in the sketch by adding noise $\eta^{(i)}_{r,c}$ drawn from $\mathrm{Geom}(e^{\varepsilon/(2R)})$, to preserve privacy. Then the party sends this noisy count sketch $\tilde{C}_i$ to the aggregator. The communication cost (the total size of all count sketches) is

$$\sum_{i=1}^{k} s_i R = O\left(\sum_{i=1}^{k} ks \cdot \frac{n_i}{N} \log \frac{k}{\beta}\right) = O\left(ks \log \frac{k}{\beta}\right).$$

If there is no public randomness, each party i also needs to send the hash functions used in the count sketch $h^{(i)}_r, g^{(i)}_r$, $r \in [R]$ to the aggregator, which takes $O(kR) = O(k \log \frac{k}{\beta})$ communication.

Privacy Guarantee. It is clear that a count sketch of R rows has a sensitivity of $2R$, so adding noise drawn from $\mathrm{Geom}(e^{\varepsilon/(2R)})$ is sufficient to preserve ε-DP for each party.

Algorithm on Aggregator. After the aggregator has collected the noisy count sketch $\tilde{C}_i$ from each party, for any $j \in [u]$, we use $\tilde{y}_j := \sum_i \mathrm{median}_{r \in [R]}\{g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)]\}$ as the estimator for $y_j := \sum_i x_{i,j}$.

Accuracy. We use a lemma from [48],

Lemma 1 ([48]). *If $\{X_i\}_{i \in [n]}$ are independent random variables, each of which has a symmetric PDF around zero, and n is an odd number, then* $\mathsf{E}\left[\mathrm{median}_{i \in [n]}\{X_i\}\right] = 0$.

To see that $\mathsf{E}[\tilde{y}_j] = y_j$, first we show that the frequency estimator from each row of $\tilde{C}_i$ is unbiased. For any $r \in [R]$,

$$\begin{aligned}\mathsf{E}\left[g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)] - x_{i,j}\right] &= \mathsf{E}\left[g^{(i)}_r(j) \cdot C_i[r, h^{(i)}_r(j)] - x_{i,j}\right] \\ &\quad + \mathsf{E}\left[g^{(i)}_r(j) \cdot \eta^{(i)}_{r, h^{(i)}_r(j)}\right] \\ &= 0.\end{aligned}$$

Moreover, since $g^{(i)}_r$ is a truly random hash function which maps j to ± 1 with equal probability, each random variable

$$g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)] - x_{i,j}$$

has a symmetric PDF around zero. Then, by Lemma 1, we have

$$\begin{aligned}\mathsf{E}\left[\tilde{y}_j - y_j\right] &= \sum_i \mathsf{E}\left[\mathrm{median}_{r \in [R]}\{g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)] - x_{i,j}\}\right] \\ &= 0.\end{aligned}$$

Next, we analyze the error $|\tilde{y}_j - y_j|$. Due to the ℓ_1 error guarantee of the count sketch and the Chebyshev inequality for the Geometric noise, for any $i \in [k]$ and $r \in [R]$, we have

$$\begin{aligned}&\left|g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)] - x_{i,j}\right| \\ &\leq \left|g^{(i)}_r(j) \cdot C_i[r, h^{(i)}_r(j)] - x_{i,j}\right| + \left|g^{(i)}_r(j) \cdot \eta^{(i)}_{r, h^{(i)}_r(j)}\right| \\ &= O\left(\frac{n_i}{s_i} + \frac{1}{e^{\varepsilon/(4R)} - 1}\right) \quad \text{with a constant probability, say, 0.95.}\end{aligned}$$

Note that the frequency estimator from $\tilde{C}_i$ is the median of the estimators from $R = \log \frac{3k}{\beta}$ independent rows. We say that each estimate is *good* if it satisfies the above error bound, which happens with probability 0.95. Let Y be the number of estimates that are not good. The median estimator fails to be good with probability

$$\Pr\left[Y \geq \tfrac{1}{2} \log \frac{3k}{\beta}\right] \leq \exp\left(-81/80 \log \frac{3k}{\beta}\right) \leq \beta/(3k)$$

by the Chernoff bound. Thus, the success probability of the above error guarantee is amplified to $1 - \beta/(3k)$. Applying a union bound, this error guarantee holds for every party $i \in [k]$ with probability $1 - \beta/3$ — let E_1 denote this event. Conditioned upon the event E_1, and since across k parties the random variables $\mathrm{median}_{r \in [R]}\{g^{(i)}_r(j) \cdot \tilde{C}_i[r, h^{(i)}_r(j)] - x_{i,j}\}$ are independent and bounded by $O(N/(ks) + 1/(e^{\varepsilon/(4R)} - 1))$, applying a Hoeffding bound we conclude that

$$|\tilde{y}_j - y_j| = O\left(N\sqrt{\log \frac{1}{\beta}}/(\sqrt{ks}) + \sqrt{k \log \frac{1}{\beta}}/(e^{\varepsilon/(4R)} - 1)\right)$$

with probability $1 - \beta/3$. Finally by the law of total probability, this error guarantee holds unconditionally with probability at least $1 - \beta$.

Theorem 1. *For $s \geq 1$ and $\varepsilon > 0$, our ε-MDP one-shot frequency estimation protocol returns an unbiased estimator for the frequency of any item that with probability at least $1 - \beta$ has error*

$$O\left(\left(\frac{N}{\sqrt{ks}} + \frac{\sqrt{k}}{e^{\varepsilon/(4 \log \frac{3k}{\beta})} - 1}\right) \cdot \sqrt{\log \frac{1}{\beta}}\right).$$

Its expected communication cost is $O\left(ks \log \frac{k}{\beta}\right)$.

Comparison to a simple baseline. The way we combine the sketches is quite different from the standard way of "merging" sketches in the MDP/LDP model. Consider a baseline method, which follows the convention of using linear sketches in the non-private setting: Each party transmits a noisy count sketch of *a fixed size* to the aggregator, then the aggregator merges all noisy count sketches and takes the median estimator from all rows. The common practice (e.g., [7]) is to use the mergeability property of linear sketches, i.e., the aggregator merges the noisy sketches (so all parties must use the same hash functions and the same sketch size) and makes the estimate from the merged sketch. Instead, we make a separate estimate from each noisy sketch and add up the estimates. Thus the parties do not use the same hash functions; actually, as shown in our analysis above, it is critical for the parties to use independent hash functions, as they allow for a higher degree of concentration. Technically, our analysis is mainly based on the unbiasedness of

the median from an odd number of symmetric random variables. Moreover, we use different sketch sizes according to the cardinality of each party, allowing us to utilize the ℓ_1 bound of count sketch in the analysis. We also empirically compare with this basic sketching approach as a baseline in Section 5.

Comparison to PMG. Chan et al. [13] proposed a one-shot MDP algorithm PMG, which privatizes the MG algorithm [42] by injecting appropriate noise. For a sketch size s, PMG incurs a bias of $O(\frac{w}{s})$ on each party, so the resulting protocol has an error proportional to k, in contrast to the factor-$\sqrt{k}$ in our error bound. Moreover, MG has sensitivity of s, so the error of the streaming protocol in [13] is at least $\tilde{\Omega}(k\sqrt{w})$ (for any s and a constant ε); while count sketch has sensitivity $\tilde{O}(1)$, and our protocol can achieve an error $\tilde{O}(\sqrt{k})$.

Comparison to FreqOracle. The frequency oracle used in [7] combines the idea of count-sketch with the HRR protocol. It uses a Hadamard matrix of size s', where $s' = \tilde{O}(\sqrt{k})$, and two hash functions $h : [u] \to [s'], g : [u] \to \{-1, +1\}$. Each party sends one bit $g(v_i) \cdot H[r_i, h(v_i)]$ via randomized response to the aggregator, where r_i is uniformly drawn from $[s']$. The error of FreqOracle is the same as HRR asymptotically when restricted to the case $\varepsilon < 1$.[3]; their use of the count-sketch is to reduce the running time of identifying the heavy hitters from $\tilde{O}(k^{1.5})$ to $\tilde{O}(k)$. Our protocol differs from FreqOracle in the following aspects: (1) we do not need a Hadamard matrix to reduce the communication; (2) the width of the count-sketch is $\tilde{O}(\sqrt{k})$ in [7], while it is proportional to n_i on each party in our protocol; (3) we combine the count sketches on the aggregator side in a different manner from [7]. On the other hand, the use of geometric noise instead of randomized response is not crucial: Both can be used to achieve error $1/(e^{\Theta(\varepsilon)} - 1)$).

3.2 A Refined Analysis under LDP

Under LDP, which is a special case of MDP, by a more refined analysis based on the ℓ_2 bound of the count sketch, we show that it is sufficient for each party to construct a count sketch of only one row and $s = \lceil (e^{\varepsilon/2} - 1)^2/e^{\varepsilon/2} \rceil$ columns, and only the non-zero entries of the noisy sketch need to be sent to the aggregator. In addition, it suffices for h and g to both be pairwise-independent hash functions.

Theorem 2. *For $\varepsilon > 0$, our ε-LDP frequency estimation protocol returns an unbiased frequency estimator for any item with a variance of $O(ke^{\varepsilon/2}/(e^{\varepsilon/2} - 1)^2)$, or an error of*

$$O\left(\max\left\{\sqrt{k \log \frac{1}{\beta}}/(e^{\varepsilon/4} - 1), \log \frac{1}{\beta}\right\}\right)$$

with probability $1 - \beta$. It uses $O(k \cdot \max\{\varepsilon, \log \frac{1}{\varepsilon}\})$ bits of communication in expectation and $O(k \log u)$ bits of public randomness.

The proof of the theorem is presented in the full version of the paper [36]. Recently, Feldman and Talwar [32] present a general compression scheme to reduce the communication cost of any LDP protocol under standard cryptographic assumptions. When applied to the frequency estimation problem, that ε-LDP protocol achieves the same error as in Theorem 2. However, it is interesting to note that our approach is completely different to the compression scheme. Furthermore, the communication cost using this compression approach is $O(k \max\{\log u, \varepsilon, \log \frac{1}{\varepsilon}\})$, which is higher than ours for the most common privacy regime $1/u^{-\Omega(1)} < \varepsilon < \log u$.

Comparison to OLH. The OLH algorithm [50] improves over HRR for the case $\varepsilon > 1$, although asymptotically it has the same error bound as HRR, i.e., the error is $\Omega(\sqrt{k})$ for $\varepsilon > 1$ [4]. The first step of OLH is similar to ours, where each party hashes the item to $[s]$ for $s = e^{\varepsilon} + 1$, but without a sign hash function. However, the crucial difference is the perturbation step. OLH perturbs the resulting hash value over $[s]$ using randomized response, i.e., reporting the true value with probability $e^{\varepsilon}/(e^{\varepsilon} + s - 1) = 1/2$, otherwise a value u.a.r. over $[s]$. In our algorithm, we treat the hash value as the location in the count sketch. We perturb each counter with geometric noise, and employ the expected sparsity of the noisy sketch to bound the communication cost.

A Lower Bound. To complement our upper bound, we prove (proof in the full version of the paper [36]) a lower bound for $\varepsilon = O(\log k)$ by a reduction to the *1-bit sum* problem under LDP, where each party holds a bit, and the aggregator wishes to estimate the number of 1's.

Theorem 3. *Any LDP protocol for the 1-bit sum problem must have an error of $\Omega(\sqrt{k}/e^{\varepsilon/2})$ for $k = \Omega(e^{\varepsilon})$.*

Note that the lower bound on the 1-bit sum problem also holds for the frequency estimation problem ($u \geq 2$), since any frequency estimation protocol can be used to solve the 1-bit sum problem. For $\varepsilon = O(\log k)$, Ghazi et al. [33] prove a lower bound of $\tilde{\Omega}(k^{1/3})$ (Equation (11) in Theorem 3.3) that is independent of ε. In contrast, our lower bound characterizes the dependency on ε and can be much better than the one in [33], for example, when $\varepsilon = \frac{1}{4} \log k$ the new bound is $\Omega(k^{3/8}) \gg \tilde{\Omega}(k^{1/3})$. Combined with the lower bound of $\Omega(\sqrt{k}/\varepsilon)$ for $\varepsilon = O(1)$ in [9], this implies our LDP protocol is optimal for all $\varepsilon \leq O(\log k)$, up to a constant-factor difference in ε.

3.3 Further Improvement by Frequency Separation

In the protocol described in Section 3.1, the sketch size s_i is proportional to n_i on each party, even if it has only one item with local count n_i. This results in a large number of informationless noisy counters to be sent to the aggregator. In this subsection, we describe a method to reduce the communication cost, which works particularly well on skewed data while providing the same worst case guarantee as Theorem 1. The idea is to divide the local counts into "heavy" and "light" groups. We use an importance sampling based method for the heavy items, while dealing with the rest using count sketch as before. We also demonstrate its effectiveness in the experiments.

More precisely, we separate the local counts $\{x_{i,j}\}_j$ into *local heavy hitters* and *local light hitters*. To preserve privacy, we do so probabilistically, as follows: (1) draw a noise vector $\xi_i \in \mathbb{R}^u$,

[3] More precisely, the error is $O(\sqrt{k} \cdot (e^{\varepsilon} + 1)/(e^{\varepsilon} - 1)) = O(\sqrt{k}/\min\{\varepsilon, 1\})$ The reason is that in FreqOracle or HRR each party uniformly selects an entry of the Hadamard matrix (for reducing communication cost), so even if $\varepsilon \to \infty$ the error is still $O(\sqrt{k})$. Chen et al. [17] improve the error to $O(\sqrt{k}/\sqrt{\varepsilon})$ for $\varepsilon > 1$, by using multiple samples from the Hadamard matrix to reduce the variance.

[4] Note that Wang et al. [50] focus on Var*, which is only one part of the true variance (see Equation (3) in their paper). While Var* decreases exponentially fast in ε, the full variance does not.

where each coordinate is i.i.d. from Geom($e^{\varepsilon/4}$), and perturb $\mathbf{x}_i$ as $\tilde{\mathbf{x}}_i := \mathbf{x}_i + \xi_i$; (2) extract entries $j \in [u]$ such that $\tilde{x}_{i,j} > T$ as the local heavy hitters, for some threshold $T = \Theta(\frac{1}{\varepsilon}\log(ku))$, while the others are the local light hitters. We denote the identities of the local heavy and light hitters at party i as S_i^{hi} and S_i^{lo}, respectively. All error analyses below hold for any fixed separation of the local heavy/light hitters, i.e., conditioned upon S_i^{hi} and S_i^{lo} for all $i \in [k]$; by the law of total probability the error guarantee will hold unconditionally.

In order to avoid running time proportional to u, the above procedure for separating $[u]$ into S^{hi} and S^{lo} can be equivalently done as follows [25]:

(1) Add i.i.d. noise drawn from Geom($e^{\varepsilon/4}$) to the non-zero entries of $\mathbf{x}_i$, and extract the entries with noisy count above the given threshold T as S^{hi}.
(2) For the zero entries of $\mathbf{x}_i$, first draw $m_i \sim \text{Binomial}(u - n_i, p_T)$, where $p_T = \exp((-T+1)\varepsilon/4)/(\exp(\varepsilon/4)+1)$ is the probability that a zero entry has a noisy count above T. Then, uniformly at random select m_i locations from the zero entries of $\mathbf{x}_i$ and add them to S^{hi}.

It is safe for each party to release S_i^{hi} and S_i^{lo}. This is because the local counts $\{x_{i,j}\}_j$ have a sensitivity of 2, thus adding noise drawn from Geom($e^{\varepsilon/4}$) is sufficient to preserve $\varepsilon/2$-DP. Then by the post-processing property of DP, the separation results are $\varepsilon/2$-DP. In the following, we present DP mechanisms for the local heavy and light hitters respectively. The input to these mechanisms are the identities of the local heavy/light hitters and their (original) local counts. Since the locations of the local heavy and light hitters are disjoint, by the parallel composition theorem, it suffices for these two mechanisms to be $\varepsilon/2$-DP.

3.3.1 Local Heavy Hitters.

Algorithm on Each Party. For each $j \in S_i^{\text{hi}}$ at party i, we perturb $x_{i,j}$, with fresh noise $\zeta_{i,j}$ drawn from Geom($e^{\varepsilon'/2}$), as $\hat{x}_{i,j} := x_{i,j} + \zeta_{i,j}$, where $\varepsilon' = \varepsilon/\log\frac{1}{\beta}$. Still, sending all such items would consume a lot of communication so instead we send these perturbed local counts by importance sampling. More precisely, party i sends each pair $(j, \hat{x}_{i,j}/p)$, $j \in S_i^{\text{hi}}$ to the aggregator with probability $p(|\hat{x}_{i,j}|)$, where $p(x) = \min\{ks \cdot x/N, 1\}$. This procedure (perturb and importance sampling) is repeated by $\frac{1}{2}\log\frac{1}{\beta}$ times.

Privacy Guarantee. Because the frequency vector has a sensitivity of 2, adding noise drawn from Geom($e^{\varepsilon'/2}$) provides the guarantee of ε'-DP. By the basic composition theorem across all repetitions, the whole procedure preserves $\varepsilon/2$-DP.

Communication. The expected communication cost in each repetition is

$$\begin{aligned}
\mathsf{E}\left[\sum_{i=1}^{k}\sum_{j\in S_i^{\text{hi}}} p(\hat{x}_{i,j})\right] &= \mathsf{E}\left[\mathsf{E}\left[\sum_{i=1}^{k}\sum_{j\in S_i^{\text{hi}}} p(\hat{x}_{i,j})\,\middle|\, S^{\text{hi}}\right]\right] \\
&\le \mathsf{E}\left[\mathsf{E}\left[\sum_{i=1}^{k}\sum_{j\in S_i^{\text{hi}}} ks\cdot|\hat{x}_{i,j}|/N\,\middle|\, S^{\text{hi}}\right]\right] \\
&\le \frac{ks}{N}\cdot\mathsf{E}\left[\mathsf{E}\left[\sum_{i=1}^{k}\sum_{j\in S_i^{\text{hi}}} x_{i,j} + |\zeta_{i,j}|\,\middle|\, S^{\text{hi}}\right]\right] \\
&= \frac{ks}{N}\cdot\mathsf{E}\left[\sum_{i=1}^{k}\sum_{j\in S_i^{\text{hi}}} x_{i,j} + \sum_{i=1}^{k} O\left(\frac{1}{\varepsilon}\right)\cdot|S_i^{\text{hi}}|\right] \\
&\le ks + \frac{ks}{N}\cdot O\left(\frac{1}{\varepsilon}\right)\cdot\mathsf{E}\left[\sum_{i=1}^{k}|S_i^{\text{hi}}|\right].
\end{aligned}$$

To bound $\mathsf{E}\left[\sum_{i=1}^{k}|S_i^{\text{hi}}|\right]$, let E denote the event that $|\xi_{i,j}| = O(\frac{1}{\varepsilon}\log(ku))$ simultaneously for every $i \in [k], j \in [u]$. Then we have $\Pr[E] \ge 1 - 1/(ku)$ by the tail property of the Geometric distribution and applying a union bound. Furthermore, conditioned upon E, every local heavy hitter has a frequency of $\Omega(\frac{1}{\varepsilon}\log(ku))$. Then, we have

$$\begin{aligned}
\mathsf{E}\left[\sum_{i=1}^{k}|S_i^{\text{hi}}|\right] &\le \Pr[E]\cdot\mathsf{E}\left[\sum_{i=1}^{k}|S_i^{\text{hi}}|\,\middle|\, E\right] + \Pr[\bar{E}]\cdot ku \\
&\le O\left(\frac{N}{\frac{1}{\varepsilon}\log(ku)}\right).
\end{aligned}$$

Thus the expected communication cost is $O(ks\log\frac{1}{\beta})$ across all repetitions.

Algorithm on Aggregator. In each repetition, let $g_{i,j}$ denote the HT estimator for $\hat{x}_{i,j}$. More precisely, if the aggregator received $\hat{x}_{i,j}$, we use $g_{i,j} = \hat{x}_{i,j}/p(|\hat{x}_{i,j}|)$, otherwise $g_{i,j} = 0$. The aggregator uses $\tilde{y}_j^{\text{hi}} := \sum_i g_{i,j}$ as the estimate for $y_j^{\text{hi}} := \sum_{i:j\in S_i^{\text{hi}}} x_{i,j}$ in each repetition, and takes the median of these estimates across all $\frac{1}{2}\log\frac{1}{\beta}$ repetitions as the final estimator.

Accuracy. It suffices to show that the estimate in each repetition satisfies the desired error guarantee with a constant probability, say, 0.99, then the success probability of the median estimate from all $\frac{1}{2}\log\frac{1}{\beta}$ repetitions can be amplified to $1-\beta$ by the Chernoff bound. First, we show that $\tilde{y}_j^{\text{hi}}$ is an unbiased estimator of y_j^{hi}, let $\hat{x}$ denote $\{\hat{x}_{i,j}\}_{i:j\in S_i^{\text{hi}}}$,

$$\mathsf{E}[\tilde{y}_j^{\text{hi}}] = \mathsf{E}\left[\mathsf{E}\left[\sum_i g_{i,j}\,\middle|\,\hat{x}\right]\right] = \mathsf{E}\left[\sum_{i:j\in S_i^{\text{hi}}}\hat{x}_{i,j}\right] = \sum_{i:j\in S_i^{\text{hi}}} x_{i,j}.$$

Next, we analyze the error $|\tilde{y}_j^{\text{hi}} - y_j^{\text{hi}}|$, which is composed of two parts:

(1) $\sum_i g_{i,j}$ approximates $\sum_{i:j\in S_i^{\text{hi}}}\hat{x}_{i,j}$.
(2) $\sum_{i:j\in S_i^{\text{hi}}}\hat{x}_{i,j}$ approximates $\sum_{i:j\in S_i^{\text{hi}}} x_{i,j}$.

For part (1), let $\beta' = 0.01$, we first show that the error is $O(N/(\sqrt{ks}))$ with probability at least $1-\beta'/2$ for any fixed choice of $\hat{x}$, i.e., conditioned upon the randomness of $\hat{x}$, then by the law of total probability the same error guarantee holds unconditionally. It suffices to consider the worst case that for all $i \in [k]$, $p(|\hat{x}_{i,j}|) < 1$, otherwise $g_{i,j} = \hat{x}_{i,j}$ which is already correct. Let F denote the event that $|\sum_i g_{i,j} - \sum_{i:j\in S_i^{\text{hi}}}\hat{x}_{i,j}| = O(N/(\sqrt{ks}))$. Since $\mathsf{E}[\sum_i g_{i,j} \mid$

$\hat{x}] = \sum_{i:j\in S_i^{\text{hi}}} \hat{x}_{i,j}$ and $|g_{i,j}| \le N/(ks)$, by a Hoeffding bound we have $\Pr[F \mid \hat{x}] \ge 1-\beta'/2$. Then, by the law of total probability,

$$\Pr[F] = \sum_{\hat{x}} \Pr[F \mid \hat{x}] \cdot p(\hat{x}) \ge \sum_{\hat{x}} (1-\tfrac{\beta'}{2}) \cdot p(\hat{x}) = 1-\tfrac{\beta'}{2}.$$

We conclude that the error for part (1) is $O(N/(\sqrt{k}s))$ with probability at least $1-\beta'/2$.

For part (2), since each $\zeta_{i,j}$ is drawn from $\text{Geom}(e^{\varepsilon/(2\log\frac{1}{\beta})})$, by the Chebyshev inequality, it is easy to see that $|\sum_i \zeta_{i,j}| = O(\sqrt{k}/(e^{\varepsilon/(4\log\frac{1}{\beta})}-1))$ with probability at least $1-\beta'/2$.

Finally, applying a union bound, the error $|\tilde{y}_j^{\text{hi}} - y_j^{\text{hi}}|$ is $O(N/(\sqrt{k}s) + \sqrt{k}/(e^{\varepsilon/(4\log\frac{1}{\beta})}-1))$ with probability at least $1-\beta' = 0.99$.

3.3.2 Local Light Hitters. We apply our count sketch based method (with privacy parameter $\varepsilon/4$ and $R = \frac{1}{4}\log\frac{3k}{\beta}$) over the local light hitters. Let n_i^{lo} denote the total (true) frequency of the local light hitters. Recall that the sketch size is proportional to n_i^{lo}, which in this case is sensitive information and cannot be directly released. So we use $s_i := \lceil ks \cdot \tilde{n}_i^{\text{lo}}/N \rceil$ as the sketch size, where $\tilde{n}_i^{\text{lo}} = \min\{n_i^{\text{lo}} + \frac{8}{\varepsilon}\log\frac{2k}{\beta} + \text{Geom}(e^{\varepsilon/4}), n_i\}$ is an upper bound of n_i^{lo} for every i with probability $1-\beta/2$. Conditioned upon $\tilde{n}_i^{\text{lo}} \ge n_i^{\text{lo}}$, the error bound in Theorem 1 holds for local light hitters with probability $1-\beta/2$. Then by a union bound, this error guarantee holds unconditionally. Moreover, since $\tilde{n}_i^{\text{lo}} \le n_i$, the communication bound in Theorem 1 also holds.

Privacy Guarantee. Note that $\|\mathbf{x}_i^{\text{lo}}\|_1$ has a sensitivity of 1, then adding $\text{Geom}(e^{\varepsilon/4})$ noise to it is sufficient to preserve $\varepsilon/4$-DP. And our count sketch based method provides a guarantee of $\varepsilon/4$-DP. Then by the basic composition theorem, the protocol applied on the local light hitters preserves $\varepsilon/2$-DP.

3.4 Applications

A frequency estimation protocol can be used as a basic building block to solve the heavy hitter identification problem, i.e., finding items whose frequency exceeds a threshold ϕN, for some given $0 < \phi < 1$. The most direct approach is to obtain a frequency estimate for every item in the universe $[u]$, but this would be too slow for a large universe, e.g., all keywords up to a certain length. To make it more efficient, a standard technique [7, 18, 20, 26] is to impose a d-adic decomposition over the universe $[u]$, which can also be understood as a complete d-ary tree of $\log_d u$ levels. Specifically, on each level j, the universe $[u]$ is partitioned into u/d^j intervals of length d^j each: $[1, d^j], (d^j, 2d^j], \ldots, (u-d^j, u]$. A frequency estimation protocol is applied on each level such that the frequency of any d-adic interval (i.e., the number of items falling inside this interval) can be estimated. This allows a recursive top-down search procedure to be applied to find the heavy hitters efficiently. Note that to ensure privacy, we need to split the privacy budget ε equally across all levels. We state the following theorem, whose proof follows a rather standard analysis, hence omitted.

Theorem 4. *Given an ε-MDP frequency estimation protocol using communication C such that the frequency of any item can be estimated in time $\mathcal{T}$ within error $\mathcal{E}(\varepsilon,\beta)$ with probability $1-\beta$, then if $\phi > 2\mathcal{E}(\varepsilon/\log_d u, N\beta/\log_d u)/N$, there exists an ε-MDP protocol such that all ϕ-heavy hitters can be found in time $O(\frac{d}{\phi}\log_d u \cdot \mathcal{T})$ and their frequencies can be estimated within error $\mathcal{E}(\varepsilon/\log_d u, N\beta/\log_d u)$ with probability $1-\beta$. The total communication cost is $O(\log_d u \cdot C)$.*

Similar techniques can also be used to solve the orthogonal range counting problem based on a given frequency estimation protocol. In particular, all results in [20] carry over to the MDP model by replacing their LDP frequency estimation protocol with ours.

4 STREAMING PROTOCOLS

In the streaming MDP model, each of the k parties receives a stream of items, one at each time step. Let n be the total number of time steps. For simplicity, we assume that n is known to the protocol in advance; standard techniques can be used to remove this assumption, while incurring some extra logarithmic factors in the error and costs [15]. Let $v_{i,t}$ denote the item received by party i at time step t, and $f(v; t_1, t_2)$ the frequency of a given item v received across all parties between time step t_1 and t_2 (inclusive). In the following, we present a protocol that maintains a synopsis from which an estimate of $f(v; 1, t)$ can be extracted for any v at each time step t; and in the full version of the paper [36] we extend it to the sliding-window model, i.e., we estimate $f(v; t-w+1, t)$ where w is the window length.

4.1 Full-stream Protocol

Let $s \ge 1$ and $\Delta := \lceil n/s \rceil \cdot \sqrt{k}$. We divide the stream into $m := \sqrt{k} \cdot n/\Delta = \min\{n, s\}$ epochs of $b := \Delta/\sqrt{k}$ time steps each. We say that an epoch is *complete* if items in all time steps in this epoch have been received, otherwise we say that it is *active*. Express the current time as $t = q \cdot b + r$ where $q, r \in \mathbb{Z}$ and $0 \le r < b$. To estimate $f(v; 1, t)$, we estimate $f(v; 1, qb)$ (i.e., over all complete epochs) and $f(v; qb+1, t)$ (i.e., over the current active epoch) separately using different methods. The *intra-epoch* protocol, which estimates $f(v; qb+1, t)$, operates on a per time step basis, so its error (variance) grows linearly as time goes on and would be too large beyond one epoch. The *inter-epoch* protocol works on the epoch level; it responds slower to the stream but its error only grows logarithmically.

4.1.1 Intra-epoch protocol. First of all, note that we only need to run the intra-epoch protocol when $b > 1$, or $s < n$.

Algorithm on Each Party. Each party samples each time step t with probability $p = b^{-1} = \sqrt{k}/\Delta$ independently. If t is sampled, the party encodes the item $v_{i,t}$ as $\mathcal{M}(v_{i,t})$ using HRR (with privacy parameter $\varepsilon/2$) and sends it to the aggregator.

Algorithm on Aggregator. The aggregator collects the messages received during the current active epoch and calculates

$$\tilde{f}(v; qb+1, t) = \tfrac{1}{p} \cdot \textstyle\sum_i \sum_{t'\in S_i} \tilde{f}_{\mathcal{M}(v_{i,t'})}(v)$$

as the estimation for $f(v; qb+1, t)$, where S_i denotes the time steps sampled at each party i during this epoch, and $\tilde{f}_{\mathcal{M}(v_{i,t'})}(v)$ denotes the frequency estimator for item v used in the HRR protocol.

Communication. Since in each epoch of size b each party samples an item with probability $p = b^{-1}$, the expected communication cost in each epoch is $O(k \cdot b \cdot b^{-1}) = O(k)$, and across all epochs the total is $O(km) = O(ks)$.

Accuracy. We first show that the estimator is unbiased. Let the random variable $Z_{i,t'} \in \{0,1\}$ indicate whether the time step t' gets sampled, and let $X_{i,t'} := \tilde{f}_{\mathcal{M}(v_{i,t'})}(v)$ denote the frequency estimator used in the HRR protocol. Then the estimator can be written as

$$\tilde{f}(v; qb+1, t) = \frac{1}{p} \cdot \sum_i \sum_{qb<t'\le t} Z_{i,t'} X_{i,t'}.$$

Note that we have $\mathsf{E}[X_{i,t'}] = x_{i,t'}$, where $x_{i,t'} \in \{0,1\}$ indicates whether $v_{i,t'}$ is identical to v. By the independence of $Z_{i,t'}$ and $X_{i,t'}$, we conclude that $\tilde{f}(v; qb+1, t)$ is an unbiased estimator for $f(v; qb+1, t)$.

Since our algorithm essentially runs HRR over the sampled time steps, the error consists of two parts: the error due to sampling and the error due to HRR:

$$\begin{aligned}
&|\tilde{f}(v; qb+1, t) - f(v; qb+1, t)| \\
&= \left| \frac{1}{p} \cdot \sum_i \sum_{qb<t'\le t} Z_{i,t'} X_{i,t'} - \sum_i \sum_{qb<t'\le t} x_{i,t'} \right| \\
&\le \left| \frac{1}{p} \cdot \sum_i \sum_{qb<t'\le t} Z_{i,t'} \cdot (X_{i,t'} - x_{i,t'}) \right| \\
&\quad + \left| \frac{1}{p} \cdot \sum_i \sum_{qb<t'\le t} Z_{i,t'} x_{i,t'} - \sum_i \sum_{qb<t'\le t} x_{i,t'} \right|.
\end{aligned}$$

We first bound the first error term. Denote the number of items sampled during this epoch as $Z^{(qb:(q+1)b)} := \sum_i \sum_{qb<t'\le(q+1)b} Z_{i,t'}$. Observe that $\mathsf{E}[Z^{(qb:(q+1)b)}] = k \cdot b \cdot p = k$ and $\mathsf{Var}[Z^{(qb:(q+1)b)}] \le k$. Hence, by the Bernstein inequality, we have $Z^{(qb:(q+1)b)} = O(\max\{k, \log\frac{1}{\beta}\})$ with probability at least $1 - \beta/3$. Conditioned upon this event and by the accuracy guarantee of HRR, with probability at least $1-\beta/3$, the first error term is bounded by

$$O\left(\frac{1}{p}\sqrt{\max\{k, \log\frac{1}{\beta}\}\log\frac{1}{\beta}}/\min\{\varepsilon,1\}\right) = O(\Delta \log\frac{1}{\beta}/\min\{\varepsilon,1\}).$$

Then the same error guarantee holds unconditionally with probability at least $1 - 2\beta/3$.

For the second error term, we first bound the variance

$$\begin{aligned}
\mathsf{Var}\left[\frac{1}{p}\cdot\sum_i\sum_{qb<t'\le t} Z_{i,t'}x_{i,t'}\right] &= \sum_i \sum_{qb<t'\le t} \mathsf{E}\left[\left(\frac{1}{p}\cdot Z_{i,t'}x_{i,t'} - x_{i,t'}\right)^2\right] \\
&= \sum_i \sum_{qb<t'\le t} \frac{1-p}{p} x_{i,t'}^2 \\
&\le k \cdot \frac{\Delta}{\sqrt{k}} \cdot \frac{1-p}{p} \\
&\le \Delta^2.
\end{aligned}$$

Note that each $|\frac{1}{p}\cdot Z_{i,t'}x_{i,t'} - x_{i,t'}|$ is bounded by $O\left(\frac{1}{p}\right) = O\left(\frac{\Delta}{\sqrt{k}}\right)$. By the Bernstein inequality, we conclude that the error for the second part is $O(\Delta \log\frac{1}{\beta})$ with probability at least $1-\beta/3$.

Finally, applying a union bound, with probability at least $1-\beta$, the error of the estimator $\tilde{f}(v; qb+1, t)$ is

$$O\left(\Delta \log\frac{1}{\beta}/\min\{\varepsilon,1\}\right) = O\left(n\sqrt{k}\log\frac{1}{\beta}/(s\cdot\min\{\varepsilon,1\})\right).$$

Privacy Guarantee. Observe that the sampling procedure is data-independent and the information of each item is released only once. So the intra-epoch protocol guarantees $\varepsilon/2$-DP, as provided by the HRR protocol.

Space/time. Observe that each party does not need store any historical information before time step t, so the intra-epoch protocol needs $O(1)$ space on each party. Sampling and running HRR take $O(1)$ time per time step.

4.1.2 Inter-epoch protocol. To obtain estimation for $f(v; 1, qb)$, we make use of a dyadic structure, which naturally corresponds to a tree representation, imposed over all epochs. Specifically, we build $\log m$ levels, and for each level l the epoch's time steps are divided into $n/(2^l \cdot b)$ consecutive blocks of size $2^l \cdot b$ each. More precisely, for $0 \le l < \log m, 1 \le j \le n/(2^l \cdot b)$, let $B_{l,j} = \{t \mid (j-1)\cdot 2^l \cdot b < t \le j \cdot 2^l \cdot b\}$ denote the j-th block at level l. Note that each block on level 0 corresponds to an epoch. Essentially, the inter-epoch protocol runs our one-shot algorithm for each block, but using different parameters.

Algorithm on Each Party. Each party i maintains a count sketch of $R = \frac{1}{2}\log\frac{2k\log m}{\beta}$ rows and $s\sqrt{\log m}\cdot|B|/n$ columns for the items within each block B, where l denotes the level of this block and $|B| = 2^l \cdot b$ denotes the number of time steps in B. After B completes, we add i.i.d. noise draw from $\mathrm{Geom}(e^{\varepsilon/(4R\log m)})$ to each counter in the count sketch, then send this noisy count sketch to the aggregator.

Algorithm on Aggregator. As in our one-shot protocol, from the noisy count sketches (across all parties) corresponding to each block, the aggregator can obtain a frequency estimator for any item within this block. Furthermore, we know that the interval $[1, qb]$ can be decomposed into at most $\log m$ disjoint dyadic blocks, at most one from each level. Thus, to obtain an estimation for $f(v; 1, qb)$, we just add up the frequency estimates for v from these blocks.

The analysis for accuracy, communication, privacy and space of the inter-epoch algorithm is presented in the full version [36].

Combining the intra-epoch and inter-epoch algorithm, we obtain the following result.

Theorem 5. *For $\varepsilon > 0$ and $s \ge 1$, our ε-MDP streaming frequency estimation protocol is able to return, at each time step, an unbiased estimator for the frequency of any item. With probability at least $1-\beta$, the error of the estimator is*

$$O\left(\frac{n\sqrt{k}\log\frac{1}{\beta}}{s\cdot\min\{\varepsilon,1\}} + \frac{\sqrt{k\log s\log\frac{1}{\beta}}}{e^{\varepsilon/(4\log s\log\frac{k\log s}{\beta})}-1}\right)$$

for $s < n$, or

$$O\left(\frac{n\sqrt{k\log\frac{1}{\beta}}}{s} + \frac{\sqrt{k\log n\log\frac{1}{\beta}}}{e^{\varepsilon/(4\log n\log\frac{k\log n}{\beta})}-1}\right)$$

for $s \geq n$ (in this case the intra-epoch algorithm is not needed). Its expected communication cost is $O(ks \cdot \log^{1.5} s \log \frac{k \log s}{\beta})$, and it takes $O(s \cdot \sqrt{\log s} \log \frac{k \log s}{\beta})$ space and $O(\log s \log \frac{k \log s}{\beta})$ time to process each item on each party.

The streaming MDP model degenerates into the continual observation model with $k := 1$. In this case, each epoch has $b = \lceil n/s \rceil$ items, so we may just discard them, and only run the inter-epoch algorithm. We then obtain a streaming algorithm with the following space-error trade-off:

Corollary 1. *For $s \geq 1$, our ε-DP frequency estimation algorithm runs on a single stream using space $O(s \cdot \sqrt{\log s} \log \frac{\log s}{\beta})$. At any time, an estimate for the frequency of any item can be extracted that, with probability $1 - \beta$, has error*

$$O\left(\frac{n\sqrt{\log \frac{1}{\beta}}}{s} + \frac{\sqrt{\log s \log \frac{1}{\beta}}}{e^{\varepsilon/(4 \log s \log \frac{\log s}{\beta})} - 1}\right).$$

We remark that the results in Section 3.4 also hold in the streaming model.

Comparison with Chan et al. There are two major differences between our protocol with Chan et al. [13]: (1) the choice of frequency summary technique (the MG algorithm v.s. count sketch); and (2) the introduction of an intra-epoch protocol. (1) is important as we explained in Section 3.1. (2) is important to achieve optimal error: the protocol in [13] doesn't need such an intra-epoch component, because that one-shot algorithm already has an error proportional to k, so it can simply ignore all items inside an epoch. We aim at the optimal error proportional to $\sqrt{k}$, which requires a more careful handling of the intra-epoch items.

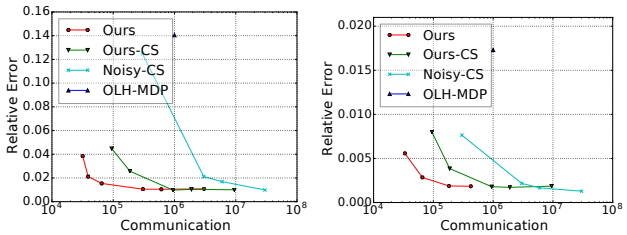
5 EXPERIMENTS

In this section, we perform experiments to evaluate our methods on frequency estimation and finding heavy hitters.

5.1 Frequency Estimation

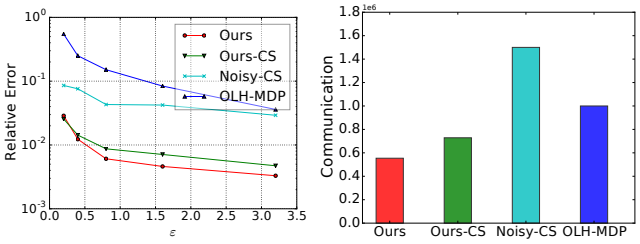
We compare our methods with the simple baseline method (denoted as `Noisy-CS`) mentioned in Section 3.1, where the aggregator simply merges all noisy count sketches and extract point estimates. We also compare with running OLH [50] in MDP, where we apply the OLH randomizer to each item. This can be equivalently viewed as an N-party LDP protocol. We use `Ours-CS` to denote denote our basic protocol described in Section 3.1 and `Ours` the method based on frequency separation described in Section 3.3.

For frequency estimation, we utilize synthetic datasets generated from Zipf distribution with skewness 1.5 and 2.0, and measure the error for the frequent items which jointly take over 85% of the total cardinality. We set the number of parties k to 1000, the total cardinality N to 1 million, and the privacy parameter ε to 1.0. In Figure 2, we report the communication-error trade-off. Note that the communication cost of `OLH-MDP` is fixed to N. We can observe that `OLH-MDP` is impractical in the MDP setting, as it has an error $\Omega(\sqrt{N})$ which can be much larger than $\sqrt{k}$. It can be observed that `Ours` offers over 3× improvement in communication while achieving the same error with other methods. We can also observe



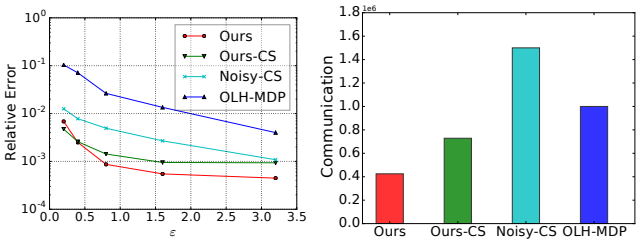
(a) Skewness 1.5 **(b) Skewness 2.0**

Figure 2: Relative Error vs. communication on Zipf datasets.



(a) Relative Error **(b) Communication**

Figure 3: Accuracy vs. ε on Zipf datasets with skewness 1.5.



(a) Relative Error **(b) Communication**

Figure 4: Accuracy vs. ε on Zipf datasets with skewness 2.0.

that, beyond a certain threshold, increasing communication does not lead to a smaller error, since the privacy constraint dominates the error in this case. In Figure 3 and 4, we perform the experiments by varying ε and fixing the communication budget. As expected, the error decreases when ε increases. `Ours` achieves the lowest error, while, for instance, using 4× less communication than `Noisy-CS`.

5.2 Identifying Heavy Hitters

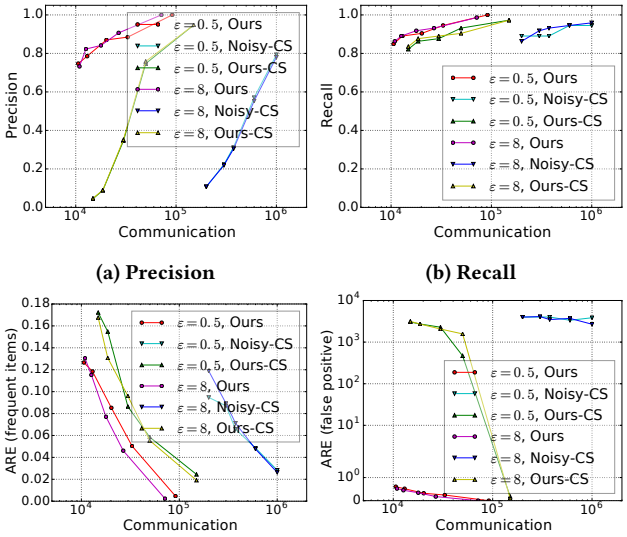
We utilize two real-world datasets, the Kosarak dataset[5] and the 2006 AOL search queries[6]. The Kosarak dataset consists of 990,002 clicks over 41,270 unique web pages. We assume that these web pages are known in advance, so the universe size u is 41,270, and we obtain the frequency estimates of all web pages to find the heavy hitters. The AOL dataset consists of both the search queries by the users and the URL of the web pages they clicked. We extracted

[5] http://fimi.ua.ac.be/data/.
[6] http://www.cim.mcgill.ca/~dudek/206/Logs/AOL-user-ct-collection/.

1,890,569 click records over 184,304 unique URLs. We consider these URLs as unknown, simulating the case of a larger universe. More precisely, as in [21], we strip the URLs of the leading domain prefixes such as "www." and "https://", and truncate all URLs to a length of 6 characters. The alphabet size of a single character in the AOL dataset is 49, so the universe size u is 49^6. We use the dyadic decomposition method as described in Section 3.4 to find heavy hitters on the AOL dataset, and the branching factor d is set to 49.

We evaluate the results using the following standard metrics (see e.g., [18]): (1) Recall, i.e., the number of true heavy hitters reported over the number of all true heavy hitters; (2) precision, i.e., the number of true heavy hitters reported over the number of answers reported; and (3) the average relative error of the reported frequencies, measured separately for the true heavy hitters and the false positive answers. For some experiments, we report the F-score, which is the harmonic mean of precision and recall. All experimental results are averaged over 5 repetitions. We use MurmurHash3 as the hash functions in the Count Sketch.

5.2.1 One-Shot Results. In the one-shot experiments, we use the following default values of of parameters: the frequency threshold ϕ is 0.001, the (expected) message size s is $5/(\phi\sqrt{k})$ (note that this leads to an error guarantee of $\phi N/5$), the privacy parameter ε is 2, the number of parties k is 100, and the number of rows of the Count Sketch is 3. For the Kosarak dataset, we uniformly partition the data across all parties. We partition the AOL dataset non-uniformly, where the largest party may have 10 times of the data than that of the smallest party.

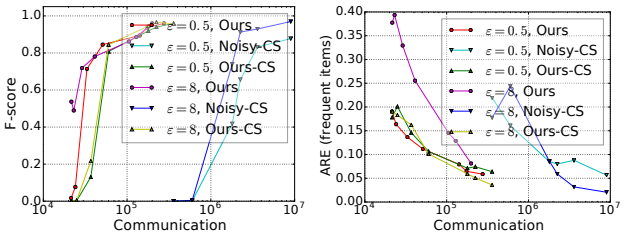


(a) Precision (b) Recall

(c) ARE (frequent items) (d) ARE (false positive)

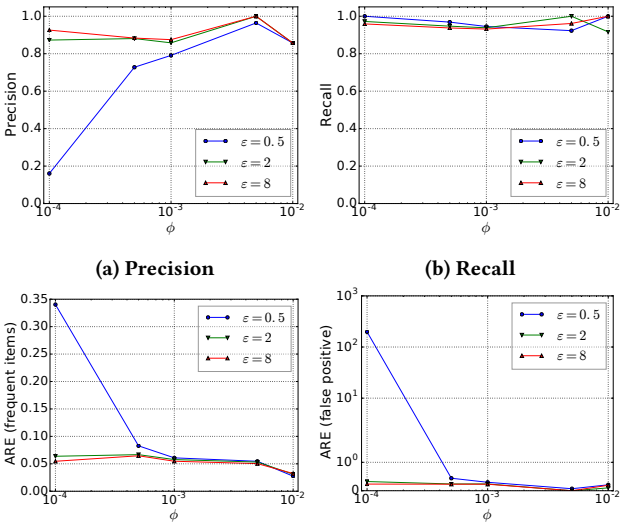
Figure 5: Accuracy vs. Communication on Kosarak.

In Figure 5 and 6, we perform the experiments by varying s and report the communication-error trade-off. We observe that `Ours` outperforms `Ours-CS` and `Noisy-CS`, that is, `Ours` uses less communication while achieving same error guarantee in practice. This improvement in communication is more pronounced for skewed dataset, as suggested by our theoretical analysis.



(a) F-score (b) ARE (frequent items)

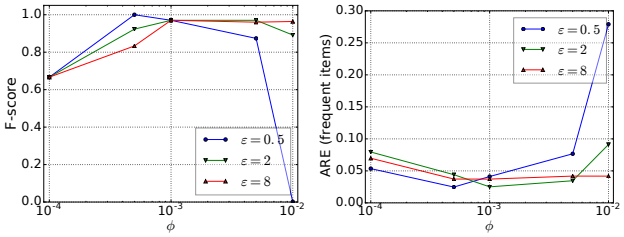
Figure 6: Accuracy vs. Communication on AOL.



(a) Precision (b) Recall

(c) ARE (frequent items) (d) ARE (false positive)

Figure 7: Accuracy vs. ϕ on Kosarak.

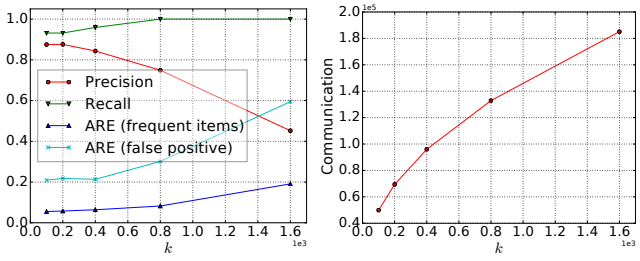


(a) F-score (b) ARE (frequent items)

Figure 8: Accuracy vs. ϕ on AOL.

Figure 7 and 8 show the accuracy results for varying ϕ from 10^{-4} to 10^{-2} and $\varepsilon = 0.5, 2, 8$, while other parameters remain fixed. It is shown that our method usually has high accuracy in practice for various frequency thresholds. It is also expected that the error becomes smaller when the privacy parameter ε is larger.

Figure 9 is performed by varying k from 100 to 1, 600, while other parameters like ϕ remain fixed. As we set the average message size s to $5/(\phi N)$ by default, our theory suggests that the error guarantee

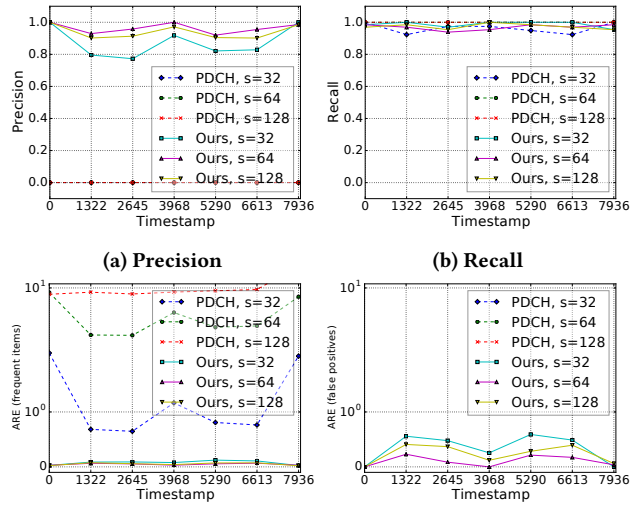


(a) Accuracy vs. k **(b) Communication vs. k**

Figure 9: Accuracy/Communication vs. k on Kosarak.

of our method is $\tilde{O}(\phi N)$, while the communication cost is $\tilde{O}(\sqrt{k}/\phi)$. We plot the curve of the communication versus k in Figure 9b. It shows that the communication cost of our method is indeed grow sublinearly with k, which confirms our theoretical analysis. Moreover, the accuracy shown in Figure 9a slightly degrades as k increases, because the error $(\sqrt{k}/\varepsilon)$ due to privacy constraint becomes larger.

5.2.2 Streaming. For the streaming experiments, we use the following default values of parameters: the number of parties $k = 100$, the privacy parameter $\varepsilon = 4$, the frequency threshold $\phi = 0.005$, the width of count sketch is 5. We set the window size $w = n/10$ on the Kosarak dataset and $w = n$ (i.e., it degenerates to the full stream case) on the AOL dataset, where n is the length of the stream.

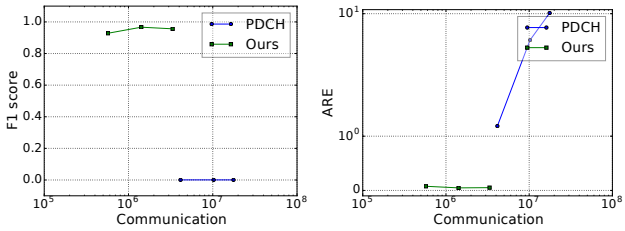


(a) Precision **(b) Recall**

(c) ARE (frequent items) **(d) ARE (false positive)**

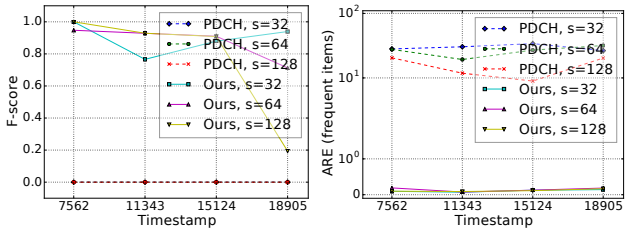
Figure 10: Accuracy over time on Kosarak.

We compare our method with the PDCH protocol [13] using the same s, which leads to the same asymptotic communication bound. We report the accuracy measures every $w/6$ timestamps in Figure 10 and 12, when the time elapsed from 0 to w. We also report the accuracy measures vs. the actual communication cost in Figure 11 and 13. Observe that increasing s does not always result in better



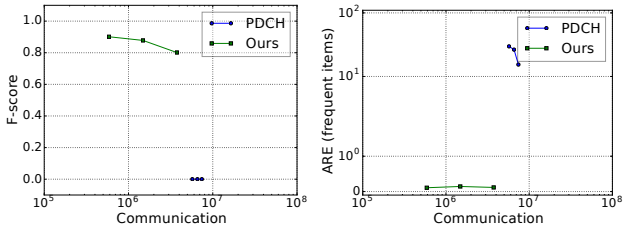
(a) F-score **(b) ARE**

Figure 11: Accuracy vs. communication on Kosarak



(a) F-score **(b) ARE (frequent items)**

Figure 12: Accuracy over time on AOL.



(a) F-score **(b) ARE (frequent items)**

Figure 13: Accuracy vs. communication on AOL

accuracy in practice. In our method, this is because increasing s also leads to more epochs, making the noise magnitude required in the inter-epoch part larger, which grows polylogarithmically with s. In PDCH, the accuracy may drop significantly when increasing s, due to the large sensitivity of the PMG summary used in their protocol. More precisely, for a MG summary of size s, PMG injects noises of magnitude $\tilde{\Theta}(s/\varepsilon)$ onto it to preserve differential privacy. In practice, the noises can even be orders of magnitudes larger than the real frequencies, making PDCH impractical. Specifically, in our setting, the noise in PDCH is roughly 10 times larger than the real frequencies, so a 0-frequency item can be easily reported as a frequent item, making the ARE (false positive) infinite.

ACKNOWLEDGMENTS

The first three authors have been supported by HKRGC under grants 16201318, 16201819, and 16205420. GC was supported in part by EPSRC grant EP/V056883/1 and European Research Council grant ERC-2014-CoG 647557.

REFERENCES

[1] J. Acharya and Z. Sun. Communication complexity in locally private distribution estimation and heavy hitters. In *International Conference on Machine Learning*, pages 51–60. PMLR, 2019.

[2] J. Acharya, Z. Sun, and H. Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1120–1129, 2019.

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

[4] Apple. Apple differential privacy technical overview. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf, 2017. [Last accessed on 5-June-2022].

[5] B. Balle, J. Bell, A. Gascón, and K. Nissim. The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, pages 638–667. Springer, 2019.

[6] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 273–282. ACM, 2007.

[7] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta. Practical locally private heavy hitters. In *Neural Information Processing Systems*, pages 2285–2293, 2017.

[8] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta. Practical locally private heavy hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42, 2020.

[9] R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *ACM STOC*, pages 127–135, 2015.

[10] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 128–138, 2005.

[11] A. Chakrabarti. Data stream algorithms. *Computer Science*, 49:149, 2015.

[12] T. H. Chan, E. Shi, and D. Song. Optimal lower bound for differentially private multi-party aggregation. In *European Symposium on Algorithms*, pages 277–288. Springer, 2012.

[13] T.-H. H. Chan, M. Li, E. Shi, and W. Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 140–159. Springer, 2012.

[14] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 2011.

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

[16] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

[17] W.-N. Chen, P. Kairouz, and A. Özgür. Breaking the communication-privacy-accuracy trilemma. In *Advances in Neural Information Processing Systems*, 2020.

[18] G. Cormode and M. Hadjieleftheriou. Methods for finding frequent items in data streams. *The VLDB Journal*, 19(1):3–20, 2010.

[19] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang. Privacy at scale: Local differential privacy in practice. In *ACM International Conference on Management of Data*, pages 1655–1658, 2018.

[20] G. Cormode, T. Kulkarni, and D. Srivastava. Answering range queries under local differential privacy. *Proceedings of the VLDB Endowment*, 12(10):1126–1138, 2019.

[21] G. Cormode, S. Maddock, and C. Maple. Frequency estimation under local differential privacy [experiments, analysis and benchmarks]. *arXiv preprint arXiv:2103.16640*, 2021.

[22] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[23] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. *ACM Transactions on Algorithms*, 7:21, 03 2011.

[24] G. Cormode, S. Muthukrishnan, K. Yi, and Q. Zhang. Continuous sampling from distributed streams. *Journal of The ACM*, 59:1–25, 04 2012.

[25] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran. Differentially private summaries for sparse data. In *International Conference on Database Theory*, pages 299–311, 2012.

[26] G. Cormode and K. Yi. *Small Summaries for Big Data.* Cambridge University Press, 2020.

[27] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*, pages 3571–3580, 2017.

[28] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *ACM STOC*, pages 715–724, 2010.

[29] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[30] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479, 2019.

[31] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067. ACM, 2014.

[32] V. Feldman and K. Talwar. Lossless compression of efficient private local randomizers. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3208–3219, 18–24 Jul 2021.

[33] B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. On the power of multiple anonymous messages. *arXiv preprint arXiv:1908.11358*, 2019.

[34] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *International Conference on Machine Learning*, pages 3505–3514. PMLR, 2020.

[35] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

[36] Z. Huang, Y. Qiu, K. Yi, and G. Cormode. Frequency estimation under multiparty differential privacy: One-shot and streaming. *arXiv preprint arXiv:2104.01808*, 2021.

[37] Z. Huang and K. Yi. The communication complexity of distributed epsilon-approximations. *SIAM Journal on Computing*, 46(4):1370–1394, 2017.

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

[39] P. Kairouz, S. Oh, and P. Viswanath. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems*, 27:2879–2887, 2014.

[40] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *IEEE International Conference on Data Engineering*, pages 277–286. IEEE Computer Society, 2008.

[41] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan. The limits of two-party differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE, 2010.

[42] J. Misra and D. Gries. Finding repeated elements. *Science of computer programming*, 2(2):143–152, 1982.

[43] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin. Collecting and analyzing data from smart device users with local differential privacy. *CoRR*, abs/1606.05053, 2016.

[44] M. Pathak, S. Rane, and B. Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, pages 1876–1884, 2010.

[45] E. Shi, T.-H. Chan, E. Rieffel, and D. Song. Distributed private data analysis: Lower bounds and practical constructions. *ACM Transactions on Algorithms*, 13:1–38, 12 2017.

[46] J. Upadhyay. Sublinear space private algorithms under the sliding window model. In *International Conference on Machine Learning*, pages 6363–6372. PMLR, 2019.

[47] S. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

[48] L. Wang, G. Luo, K. Yi, and G. Cormode. Quantiles over data streams: an experimental study. In *ACM SIGMOD International Conference on Management of Data*, pages 737–748, 2013.

[49] N. Wang, X. Xiao, Y. Yang, T. D. Hoang, H. Shin, J. Shin, and G. Yu. Privtrie: Effective frequent term discovery under local differential privacy. In *IEEE International Conference on Data Engineering*, pages 821–832, 2018.

[50] T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 729–745, 2017.

[51] T. Wang, X. Zhang, J. Feng, and X. Yang. A comprehensive survey on local differential privacy toward data statistics and analysis in crowdsensing. *CoRR*, abs/2010.05253, 2020.

[52] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Trans. Knowl. Data Eng.*, 23(8):1200–1214, 2011.

[53] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K. Lam. Local differential privacy and its applications: A comprehensive survey. *CoRR*, abs/2008.03686, 2020.

[54] D. Zhang, R. McKenna, I. Kotsogiannis, G. Bissias, M. Hay, A. Machanavajjhala, and G. Miklau. ϵKTELO: A framework for defining differentially private computations. *ACM Trans. Database Syst.*, 45(1):2:1–2:44, 2020.

[55] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4):25:1–25:41, 2017.