Federated Heavy Hitter Recovery under Linear Sketching

Adria Gascon [1] **Peter Kairouz** [1] **Ziteng Sun** [1] **Ananda Theertha Suresh** [1]

Abstract

Motivated by real-life deployments of multi-round federated analytics with secure aggregation, we investigate the fundamental communication-accuracy tradeoffs of the heavy hitter discovery and approximate (open-domain) histogram problems under a linear sketching constraint. We propose efficient algorithms based on local subsampling and invertible bloom look-up tables (IBLTs). We also show that our algorithms are information-theoretically optimal for a broad class of interactive schemes. The results show that the linear sketching constraint does increase the communication cost for both tasks by introducing an extra linear dependence on the number of users in a round. Moreover, our results also establish a separation between the communication cost for heavy hitter discovery and approximate histogram in the multi-round setting. The dependence on the number of rounds R is at most logarithmic for heavy hitter discovery whereas that of approximate histogram is $\Theta(\sqrt{R})$. We also empirically demonstrate our findings.

1. Motivation

Collecting and aggregating user data drives improvements in the app and web ecosystems. For instance, learning popular out-of-dictionary words can improve the auto-complete feature in a smart keyboard, and discovering malicious URLs can enhance the security of a browser. However, sharing user data directly with a service provider introduces several privacy risks.

It is thus desirable to only make aggregated data available to the service provider, rather than directly sharing (unanonymized) user data with them. This is typically achieved via multi-party cryptographic primitives, such as a *secure vector summation* protocol (Melis et al., 2016; Bonawitz et al., 2017; Corrigan-Gibbs et al., 2020; Bell et al., 2020). For instance, for closed domain histogram applications, the users can first "one-hot" encode their data into a vector of length d (the size of the domain) and then participate in a secure vector summation protocol to make the aggregate histogram (but never the individual user contributions) available to the service provider.

Federated heavy hitters recovery. The abovementioned solution requires $\Omega(d)$ communication. However, in many real life applications the domain size is very large or even unknown a priori. For example, the set of new URLs can be represented via 8-bit character strings of length 100, and can thus take $d = 256^{100}$ values, which is clearly impossible to support in practice. In such settings, linear[1] sketching is often used to reduce the communication load. For example, Melis et al. use secure count-min sketch aggregation for privacy preserving training of recommender systems, and Corrigan-Gibbs & Boneh rely on count-min sketches for gathering browser statistics, i.e. aggregate histogram queries. Similarly, Hu et al. rely on secure aggregation of variants of Flajolet-Martin sketches for distributed cardinality estimation. Boneh et al. uses sketching to reduce the cost for distributed subset-histogram queries. In the work closest to ours, Chen et al. show that count-sketches can be used to recover the *heavy hitter* items (i.e. frequently appearing items) while reducing the communication overhead. All these protocols operate in the single-round setting.

Sketching in multi-round aggregation schemes. Even though count-sketches are great step towards solving the heavy hitters problem, this approach has only been analyzed in the single round data aggregation setting. However, most commonly deployed systems for federated analytics employ *multi-round* schemes for data aggregation (Bonawitz et al., 2019). This is primarily because (a) not all users are available around the same time, (b) the population may be very large (in the billions of devices) and therefore the server has to aggregate data over batches for bandwidth/compute reasons, and (c) running the cryptographic secure vector summation protocol has compute and communication costs

[1] Google Research. Authorship is in alphabetical order. Correspondence to: Ziteng Sun <zitengsun@google.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] Linearity is necessary because non-linear compression/sketching schemes would not work under the secure vector summation primitive which only makes the sum of client-held vectors available to the server.

that are super linear in the number of users we are aggregating over (Bell et al., 2020; Bonawitz et al., 2017). Further, count sketch based approaches have a decoding runtime that is linear in d, which is infeasible in the open domain setting, and improving it to $\log d$ involves blowing up the communication cost by the same factor.

Our contributions. Our paper thus takes a principled approach towards uncovering the fundamental accuracy-communication tradeoffs of the heavy hitters recovery problem under the linearity constraints imposed by secure vector summation protocols. We show that linearity constraints increase the per-user communication complexity. For a fixed total number of users, as the number of rounds increases, the required communication decreases due to less stringent linearity constraint. Moreover, surprisingly, we show that count-sketches are strictly sub-optimal for this application, and we develop a novel provably optimal approach that combines client-side (local) subsampling with inverse Bloom lookup tables (IBLTs). Roughly speaking, we show (via lower bounds) that in the R-round case, any approach that solves an approximate histogram problem (with additive error) will incur a $\sqrt{R}$ factor penalty in the communication cost, while our optimal approach incurs $\log(R)$. Hence, even non-trivial modifications of count-sketches are strictly sub-optimal.

We also empirically evaluate our proposed algorithms and compare it with count-sketch baselines. Significant advantage of our algorithm is observed, especially when R is large. In the setting of Figure 1, to achieve an F1 score of 0.8, we see a 10x improvement in communication compared to the baseline using Count-sketch.

Organization. We formally define the problem in Section 2 and then discuss our results in Section 3. Algorithms for heavy hitter recovery and approximate histogram are presented in Section 4 and Section 5, respectively. We discuss a practical modification of our algorithm in Section 6 and present the experimental results in Section 7.

2. Problem setup and preliminaries

We consider heavy hitter discovery in the distributed setting with multi-round communication between the users and a central server. Suppose users come in R rounds. In round $r \in [R]$, there are n users, denoted by the set B_r. We assume the sets are pairwise disjoint, *i.e.,* $\forall r \neq r', B_r \cap B_r' = \emptyset$. Each user $i \in B_r$ contributes m_i samples with a contribution bound $m_i \leq m$ from a finite domain $\mathcal{X}$ of size d. Let h_i denote the user's local histogram where $\forall x \in \mathcal{X}$, $h_i(x)$ is the number of times element x appeared in user i's local samples. By assumption, we have $\|h_i\|_1 = m_i \leq m$. Let $h^{(r)}$ be the aggregated histogram in round r, *i.e.,*

$$\forall x \in \mathcal{X}: \qquad h^{(r)}(x) = \sum_{i \in B_r} h_i(x).$$

The aggregated histogram across all R rounds is denoted by $h^{[R]}$ where

$$\forall x \in \mathcal{X}: \qquad h^{[R]}(x) = \sum_{r \in [R]} h^{(r)}(x).$$

The total number of users is denoted by $N{:=}nR$. We will focus on cases where $d \gg Nm$, *i.e.,* the case where the support is large and the data is sparse.

The goal of the server is to learn useful information about the aggregated histogram $h^{[R]}$. More precisely, we consider the two tasks described below.

τ-heavy hitter (ApproxHH). For a given threshold τ, the goal of τ-heavy hitter recovery on the entire data stream is to return a set H such that with probability $1 - \beta$,

1. If $h^{[R]}(x) \geq \tau$, $x \in H$.
2. If $h^{[R]}(x) \leq \tau/10$, $x \notin H$.

τ-approximate histogram (ApproxHist). The goal is to return an approximate histogram $\widehat{h}^{[R]}$ such that with probability $1 - \beta$,

$$\forall x \in \mathcal{X}, \qquad \left|\widehat{h}^{[R]}(x) - h^{[R]}(x)\right| \leq \tau.$$

It can be seen that $\tau/3$-approximate histogram is a harder problem than τ-heavy hitter (HH) since an $\tau/3$-approximate histogram would imply a set of approximate heavy hitters by returning H to be the list of elements with approximate frequency more than $\tau - \tau/2$. Previous work often solves ApproxHH by reducing it to ApproxHist (*e.g.,*, in Chen et al. (2022)). However, as we show in this paper, our work establishes a seperation between the two tasks in terms of the communication complexity in the multi-round setting.

Efficient decoding. Since $d \gg Nm$, we require efficient encoding (run by users) and decoding (run by the server). More precisely, the encoding/decoding time should be polynomial in $N, m, R, \log d, \log(1/\beta)$ and other parameters.

Per-user communication complexity. We focus on distributed settings where each user has limited uplink communication capacity. In particular, each user must compress their local histogram h_i to a message of bit length ℓ, denoted by Y_i. And the server must solve the above tasks based on the received messages. The *communication complexity* of each task is the smallest bit length such that there exists a communication protocol to solve the task.

Task/Setting	1-round LinSketch	R-round LinSketch	Without LinSketch ($R = N$)
τ-ApproxHH	$\tilde{\Theta}\left(\frac{mN}{\tau}\right)$	$\tilde{\Theta}\left(\frac{mN}{\tau R}\right)$	$\tilde{\Theta}\left(\frac{m}{\tau}\right)$
τ-ApproxHist	$\tilde{\Theta}\left(\frac{mN}{\tau}\right)$	$\tilde{\Theta}\Big(\min\{\frac{mN}{\tau\sqrt{R}}, \frac{mN}{R}\}\Big)$	$\tilde{O}\Big(\min\{\frac{m\sqrt{N}}{\tau}, m\}\Big)$

Table 1. Per-user communication complexity. All described bounds can be acheived by a *non-interactive* protocol with server runtime poly$(m, n, R, \log(d), \log(1/\beta))$. Recall $N = nR$ denotes the total number of users. All bounds cannot be improved up to logarithmic factors even under *interactive* protocols. $\tilde{\Theta}$ omits factor that are logarithmic in τ, R, m and N.

Distributed estimation with linear sketching (LinSketch). A even more stringent communication model is the linear summation model. In each round r, each user $i \in B_r$ can only send a message Y_i from a finite ring G_r based on their local histogram and shared randomness U. For all $i \in B_r$, let

$$Y_i = f_i(h_i, U).$$

Under the linear aggregation model, the server only observes

$$Y^{(r)} = \sum_{i \in B_r} Y_i,$$

where the addition is the additive operation in the ring G_r and by definition, $Y^{(r)} \in G_r$. The reason why we restrict ourselves to a finite ring is for compatibility with cryptographic protocols for secure vector summation (Bonawitz et al., 2017; Bell et al., 2020), which operate in over a finite space. These protocols ensure that any additional information observed by the server beyond $Y^{(r)}$ can in fact be simulated given $Y^{(r)}$, under standard cryptographic assumptions. As mentioned above, we abstract away the specifics of the underlying protocol and assume that the server observes exactly $Y^{(r)}$. For vector summation, it is convenient to think of G_r as $\mathbb{Z}_{q_r}^{\ell}$, i.e. length-ℓ vectors with integer entries mod q_r (we might chose q_r to be prime when we require division, e.g. in the IBLT construction).

If the protocol is *interactive*, for $i \in B_r$, Y_i is allowed to depend on $Y^{(1)}, \ldots, Y^{(r-1)}$. In this case, each f_i is a function of $Y^{(1)}, \ldots, Y^{(r-1)}$. If the protocol is *non-interactive*, f_i's must be fixed independently from previous messages.

The server then must recover heavy hitters (and their frequencies) based on the transcript of the protocol, denoted by

$$\Pi = (Y^{(1)}, \ldots, Y^{(R)}, U).$$

2.1. Connection to other constrained settings.

Below we discuss the connection between our stated setting to other popular constrained settings including the streaming setting, and the general communication constrained setting.

Connection to the streaming setting. When $R = 1$, the setting is similar to the similar to the streaming setting (*e.g.,* in Cormode & Muthukrishnan (2005)) since they both require that all information about the dataset must be "compressed" into a small state. One important difference is that in the distributed setting, the local data is processed independently at each user and only linear operation on the state is permitted due to the linear aggregation operation. For the R-round case, our setting is different since the server observes R states, each with bit length at most ℓ. These states couldn't be viewed as a mega-state with bit length $R \cdot \ell$ since there is a further restriction that each sub-state (corresponding to the aggregated message observed in a round) can only contain information about data in the corresponding round instead of the entire data stream. Another naive way to reduce the problem to the streaming setting is to sum over all R states and obtain a single state with ℓ bits. However, our result implies that this reduction is strictly sub-optimal (see Table 1). To the best of our knowledge, similar settings have not been studied in either the federated analytics literature or the streaming literature.

Connection to distributed estimation without linear sketching. The general communication constrained setting where linear sketching is not enforced could be viewed as a special case of the proposed framework with $n = 1$ and $R = N$ since in this case, the linear aggregation is performed only over one user's message and hence trivial.

We list comparisons to these settings for the considered tasks in Table 1.

3. Results and technique

We consider both approximate heavy hitter recovery and approximate histogram estimation in the linear aggregation model. We establish tight (up to logarithmic factors) communication complexity for both tasks in the single-round and multi-round settings. The results are summarized in Table 1. Our results have the following interesting implications on the communication complexity of these problems.

Linear aggregation increases the communication cost. As shown in Table 1, under LinSketch, for both tasks, the per-user communication would incur a linear dependence on $n = N/R$, the number of users in each round. On the other hand, without linear aggregation constraint, there won't be a linear dependence on n since each user can simply send their m local samples losslessly using $O(m \log d)$ bits. The result establishes the fundamental cost of linear aggregation communication protocols for heavy hitter recovery.

ApproxHist is harder than ApproxHH. As mentioned before, a natural way to obtain heavy hitters is to obtain an approximate histogram and do proper thresholding to select the heavy elements. Although in the single-round case, there is at most a logarithmic gap between the communication complexity for the two problems. In the R-round case, our result shows that this is strictly sub-optimal. More precisely, the communication cost for τ-ApproxHH increases by a factor of $\sqrt{R}$ while that of ApproxHist depends at most logarithmically in R. This implies a gap between the per-user communication cost for τ-ApproxHist and τ-ApproxHH in the multi-round case.

The impact of R. With a fixed total number of users N, our result shows that the per-user communication complexity decreases as R increases. This is due to the fact that as R increases, the linearity constraints are imposed over a smaller group of users with size N/R, and hence less stringent. However, this also comes at the cost that the privacy implication from aggregation becomes weaker.

3.1. Our technique - IBLT with local subsampling

As discussed above, when solving the approximate heavy hitter problem in the multi-round setting, algorithms that rely on obtaining an approximate histogram and thresholding won't give the optimal communication complexity. In the paper, we propose to use invertible bloom lookup tables (IBLTs) (Goodrich & Mitzenmacher, 2011) and local subsampling. At a high-level, IBLT is a bloom filter-type linear data structure that supports efficient listing of the inserted elements and their exact counts. The size of the table scales linearly with the number of unique keys inserted. To reduce the communication cost, we perform local threshold sampling (Duffield et al., 2005a) on users' local datasets. This guarantees that the "light" elements will be discarded with high probability and hence won't take up the capacity of the IBLT data structure. Compared to frequency-oracle based approach, the noise introduced in our subsampling-based approach for each item is proportional to its accumulative count, which gives better estimates for elements with frequencies near the threshold. For elements with counts way above the threshold, the frequency estimate will have a larger error but this won't affect heavy hitter recovery since only whether the count is above τ is crucial to our problem. See detail of the algorithm in Section 4.

3.2. Related work

Linear dimensionality reduction techniques for frequency estimation and heavy hitter recovery has been widely studied to reduce storage or communication cost, such as Count-sketch, Count-min sketch (Charikar et al., 2002; Cormode & Muthukrishnan, 2005; Donoho, 2006; Minton & Price, 2014), and efficient decoding techniques have also been proposed (Cormode & Muthukrishnan, 2006; Gilbert et al., 2010).

The closest to our work is the independent work of (Chen et al., 2022), which studies approximate histogram estimation under linear sketching constraint. The work also establishes gap between communication complexities with/without Secure Aggregation. However, their result is in a more restricted setting of $m = 1$ and $R = 1$. Moreover, our algorithm also has the advantage of being computationally efficient (runtime only depends logartihmically in d), which is important for applications with large support but sparse data. Their work also considers algorithms that guarantee distributed differential privacy guarantees, which we leave as interesting future directions.

4. Approximate heavy hitter under linear aggregation

In this section, we study the approximate heavy hitter problem and show that the problem can be solved with per-user communication complexity $\tilde{O}\left(\frac{mn}{\tau} \log d\right)$, stated in Theorem 4.1.

A natural comparison to make is the heavy hitter recovery algorithm obtained from getting a frequency oracle up to accuracy $\Theta(\tau)$. Since there are R rounds, the naive approach would require an accuracy of $\Theta(\tau/R)$ in each round and classic methods such as Count-min and Count-sketch would require a per-user communication complexity of $\tilde{\Theta}(mnR/\tau)$. In the R-round case, our result improves upon this by a factor of R. In fact, as we show in Theorem 5.2, any frequency oracle-based approach would require per-user communication complexity of at least $\Omega(mn\sqrt{R}/\tau)$. Our result improves upon these and show that the dependence on R is at most logarithmic.

Theorem 4.1. *There exists a non-interactive linear sketching protocol with communication cost $\tilde{O}\left(\frac{mn}{\tau}\right)$ bits per user, which solves the τ-**approximate heavy hitter** problem. Moreover, the running time of the algorithm is $\tilde{O}\left(\frac{mn}{\tau}\right)$.*

The next theorem shows that the above communication complexity is minmax optimal up to logarithmic factors.

Theorem 4.2. *For any τ and interactive linear sketching protocol $\mathcal{A}$ with per-user communication cost $o\left(\frac{mn}{\tau}\right)$, there exists a dataset $h_i, i \in B_r, r \in [R]$, such that $\mathcal{A}$ cannot solve τ-heavy hitter (HH) with success probability at least*

4/5.

Next we will present the protocol that achieves Theorem 4.1 in Section 4 and discuss the proof of the lower bound Theorem 4.2 in Appendix D.1.

At a high level, the protocol relies on two main components: (i) a probabilistic data structure called Invertible Bloom Look up Table (IBLT) introduced by Goodrich & Mitzenmacher, and (ii) local subsampling. We start by introducing IBLTs, starting from the more standard (counting) Bloom filters.

IBLT: Bloom filters with efficient listing. Note that each user's local histogram h_i can be viewed as a sequence of key-value pairs $(x, h_i(x))$. The Bloom filter data structure is a standard linear data structure to representing a set of key-value pairs with keys coming from a large domain. IBLT is a version of Bloom filter that supports an efficient listing operation – while preserving the other nice properties of Bloom counting filters, namely linearity (and thus mergeable by summation), and succintness (linear size in number of indices it holds).[2] These properties are summarized in the following Lemma.

Lemma 4.3 ((Goodrich & Mitzenmacher, 2011))**.** *Consider a collection of local histograms $(h_i)_{i\in[n]}$ over $[d]$ such that $\|\sum_{i\in[n]} h_i\|_0 \leq L_0$.*

For any $\gamma > 0$, there exist local linear sketches $\{f_i\}_{i\in[n]}$ of length $\ell = \tilde{O}(\gamma L_0)$ and an $O(\ell)$ time decoding procedure $\mathrm{Dec}(\cdot)$ *such that*

$$\mathrm{Dec}(\sum_{i\in[n]} f_i(h_i)) = \sum_{i\in[n]} h_i$$

succeeds except with probability at most $O\left(L_0^{2-\gamma}\right)$.

For the purpose of this paper we can focus on the two main operations supported by an IBLT instance $\mathcal{B}$ (see (Goodrich & Mitzenmacher, 2011) for details on deletions and lookups):

- $\texttt{Insert}(k, v)$, which inserts the pair (k, v) into $\mathcal{B}$.
- $\texttt{ListEntries}()$, which enumerates the set of key-value pairs in $\mathcal{B}$.

Note that $f_i(h_i)$ in Lemma 4.3 corresponds to the IBLT $\mathcal{B}_i$ resulting from inserting the set $\{(x, h_i(x)) \mid h_i(x) > 0\}$ into an empty IBLT. Also, $\texttt{ListEntries}()$ corresponds to Dec in Lemma 4.3.

Finally, $\sum_i^n f_i(h_i)$ corresponds to the encoding of the IBLT resulting from inserting the set $\{(x, \sum_i^n h_i(x)) \mid \exists i \in [n] : h_i(x) > 0\}$ into an empty IBLT. In other words, each client $i \in [n]$ computes *local* IBLT $\mathcal{B}_i := f_i(h_i)$, and the (secure) aggregation of the $\mathcal{B}_i$'s results in the *global* IBLT $\mathcal{B} := \sum_i^n f_i(h_i)$. Further details on IBLT are stated in Appendix C.

Reducing capacity via threshold sampling. The second tool in our main protocol is threshold sampling. Note that the guarantee in Lemma 4.3 relies on the number of unique elements in $\sum_{i\in[n]} h_i$, which can be at most mn in the worst-case, leading to an $O(mn)$ not matching our lower bound in Lemma 4.2. For heavy hitter recovery, we reduce the communication cost by local subsampling. More precisely, we use the threshold sampling algorithm from (Duffield et al., 2005b), detailed in Algorithm 1 to achieve the (optimal) dependency $O(mn/\tau)$.

Remark 4.4. Threshold sampling can be replaced by any unbiased local subsampling method that offers sparsity, *e.g.,* binomial sampling where $h'(x) \sim \text{Binomial}(h(x), p)$ for some $p \in (0, 1)$, and similar theoretical guarantee will hold. In this work, we choose threshold sampling due to the property that it minimizes the total variance of h' under an expected sparsity constraint (see Duffield et al. (2005b) for details).

Algorithm 1 Threshold sampling.

```
Input: $h$ : local histogram. $t \in \mathbb{R}_+$ : threshold.
for $x \in \mathrm{supp}(h)$ do
    if $h(x) \geq t$, then
        $h'(x) = h(x)$.
    else
        $$h'(x) = \begin{cases} t & \text{with prob } \frac{h(x)}{t}, \\ 0 & \text{otherwise.} \end{cases}$$
    end if
end for
Return: $h'$.
```

The protocol that achieves the desired communication complexity in Theorem 4.1 is detailed in Algorithm 2.

The algorithm can be viewed as $b{:=}\lceil 20 \log(\frac{40mnR}{\tau\beta})\rceil$ independent runs of a basic protocol, each of which returns a list H_i of potential heavy hitters. And the repetition is to boost the error probability.

In each basic protocol, users first apply Algorithm 1 to subsample to the data, which reduces the number of unique elements while maintaining the heavy hitters upon aggregation. Then the user encodes their samples using IBLTs, whose aggregation is then sent to the server to decode. Since the number of unique elements is reduced through subsampling, the decoding of the aggregated IBLT will be successful with high probabiltiy, hence recovering the aggregation of

[2] In our algorithm, IBLT could be replaced by any data structure with these properties.

Algorithm 2 Subsampled IBLT with LinSketch.

```
Input: $\{h_i\}_{i \in B_r, r \in [R]}$ : local histograms; $d$ : alphabet size; $R$ : number of rounds; $m$ : per-user contribution bound; $n$ : number of users per round; $\tau$ : threshold for heavy hitter recovery; $\beta$ : failure probability.
Let $t = \max\{\tau/2, 1\}$, $b = \lceil 10 \log(\frac{4mnR}{\tau\beta}) \rceil$ and $L_0 = 20\frac{mn}{\tau}\log R, \gamma = \log R$.
for $r \in [R]$ do
    for $j \in [b]$ do
        Each user $i \in B_r$ applies Algorithm 1 with threshold $t$ in to their local histogram with fresh randomness to get $h'_{i,j}$.
        Each user sends message $Y_{i,j} = f_{i,j}(h'_{i,j})$ where $f_{i,j}$'s are mappings from Lemma 4.3 with parameter $L_0, \gamma$ and fresh randomness.
        Server observes $\sum_{i \in B_r} Y_{i,j}$ and computes
            $$\hat{h}_{r,j} = \mathrm{Dec}(\sum_{i \in B_r} Y_{i,j}).$$
        If the decoding is not successful, we set $\hat{h}_{r,j}$ be the all-zero vector.
    end for
end for
for $j \in [b]$ do
    Server computes $\hat{h}_j^{[R]} = \sum_{r \in [R]} \hat{h}_{r,j}$, and obtain list
        $$H_j = \{x \in [d] \mid \hat{h}_j^{[R]} > 0\}.$$
end for
Return:
    $$H = \{x \mid \sum_{j \in [b]} \mathbb{1}\{x \in H_j\} \geq \frac{b}{2}\}.$$
```

subsampled local histograms. The detailed proof of Theorem 4.1 is presented in Appendix A.

5. Approximate histogram under linear aggregation

In this section, we study the task of obtaining an approximate histogram in the multi-round linear aggregation model. The first observation we make is that using Algorithm 2 with threshold τ, we are able to return a list H of heavy hitters such that with high probability, the list contains all x's with frequency more than τ and no tail elements. The approximate histogram algorithm builds on this and further asks each user to send a linear sketching of the their unsampled local data alongside the IBLT data structures in Algorithm 2. The server would then use the aggregation of these linear sketches as a frequency oracle to estimate the frequency of elements in H.

The above protocol leads to near optimal performance in the single-round case. However, the R-round case is trickier since the error will build up along all R rounds and the naive application of the sketching algorithm will lead to an error that depends linearly in R. This can be solved by carefully designing the correlation among hash functions in all R rounds and we show that the dependence on R can be reduced to $\sqrt{R}$. We further show that the $\sqrt{R}$ dependence is in fact optimal by proving a matching lower bound, stated in Theorem 5.2.

At a high-level, to improve the dependence on R, we use Count-sketches where the location hashes are fixed across rounds while the sign hashes are generated with fresh randomness. The details of the algorithm are described in Algorithm 3. The proof follows from the guarantee in Theorem 4.1 and standard analysis for the Count-sketch algorithm. We defer the complete proof to Appendix B.

Theorem 5.1. *In the R-round setting, there exists a linear aggregation protocol with communication cost* $\tilde{O}\Big(\min\{\frac{mn\sqrt{R}}{\tau}, mn\}\Big)$ *per user, which solves the* τ-***approximate histogram*** *problem. Moreover, the running time of the algorithm is* $\tilde{O}\Big(\min\{\frac{mn\sqrt{R}}{\tau}, mn\}\Big)$.

Lower bound for ApproxHist We prove the following lower bound on ApproxHist, which shows that the bound in Theorem 5.1 is tight up to logarithmic factors, establishing the seperation between the sample complexity for

Theorem 5.2. *For any* τ *and R-round ApproxHist protocol with per-user communication cost* $o\Big(\min\{\frac{mn\sqrt{R}}{\tau}, mn\}\Big)$, *there exists a dataset* $\{h_i\}_{i \in B_r, r \in [R]}$, *such that the protocol cannot solve* τ-***approximate histogram*** *with error probability at most 1/5.*

6. Practical adaptive tuning for instance-specific bounds

In practical scenarios, the per-user communication cost ℓ is often determined by system constraints (*e.g.,* delay tolerance, bandwidth constraint) and the goal is to recovery heavy hitters with the small enough τ under a fixed communication cost $\ell_{\max}$. While we have shown in Theorem 4.2, in the worst case, we can only reliably recover heavy hitters with frequency at least $\Omega(\frac{mn}{\ell_{\max}})$. However, since the successful decoding of IBLTs only requires the number of *unique* elements in a round to be small, when users' data is more favorable, it is possible to obtain better instance-specific bounds when the data is more concentrated on "heavy" elements.

Algorithm 3 R-round ApproxHist with LinSketch

```
1: Input: $\{h_i\}_{i \in B_r, r \in [R]}$ : local histograms; $d$ : alphabet size; $R$ : number of rounds; $m$ : per-user contribution bound; $n$ : number of users per round; $\tau$ : error for approximate histogram; $\beta$ : failure probability.
2: if $\tau \leq \sqrt{R}$ then
3:     Users implement Algorithm 2 with $\tau = 1$ and return the histogram obtained in Line 11.
4: end if
5: Let $w = \lceil \frac{10mn\sqrt{R}}{\tau} \rceil$ and $b = \lceil \log\left(\frac{4mnR}{\tau\beta}\right) \rceil$.
6: Get the same set of location hash functions $\{g_j : [d] \to [w]\}_{j \in [w]}$ for all rounds. And the independent sets of sign hashes $\{s_{j,r} : [d] \to \{\pm 1\}\}_{j \in [w], r \in [R]}$ across rounds.
7: for $r \in [R]$ do
8:     (In Parallel) Each user $i \in B_r$ implements the protocol in Algorithm 2 and sends messages $Y_i$.
9:     (In Parallel) User $i \in B_r$ encode $j \in [b]$ and $k \in [w]$,
           $$T_i(j,k) = \sum_x \mathbb{1}\{g_j(x) = k\}\, s_{j,r}(x) \cdot h_i(x).$$
10: end for
11: Server obtains a list $H$ of heavy hitters from the the messages $Y_i$'s.
12: Server obtains $\forall r \in [R], T^{(r)} = \sum_{i \in B_r} T_i$ and constructs $\hat{h}$, where $\forall x \in H$
       $$\hat{h}(x) = \text{Median}\left( \{ \sum_{r \in [R]} T^{(r)}(j, g_j(x)) \cdot s_{j,r}(x) \}_{j \in [b]} \right),$$
    and $\forall x \notin H, \hat{h}(x) = 0$.
13: Return $\hat{h}$.
```

When interactivity across rounds is allowed, we give an adaptive tuning algorithm for the subsampling parameter, which can be implemented when interactivity is allowed. The details of the algorithm are described in Algorithm 4. At a high level, our algorithm is based on an estimate for $\|\sum_{i \in B_r} h_i'\|_0$ where h_i's are the subsampled histograms. When the decoding is successful, we can compute $\|\sum_{i \in B_r} h_i'\|_0$ exactly from the recovered histogram. When the decoding is not successful, we rely on an analysis based on the "core size" of a random hypergraph (Molloy, 2005) introduced by the hashing process to get an estimate of $\|\sum_{i \in B_r} h_i'\|_0$. We discuss this in details in Appendix C. Under the assumption that for a fixed subsampling parameter t, $\|\sum_{i \in B_r} h_i'\|_0$ will be relatively stable across rounds, we can then increase/decrease t based on past estimates of the data process.

We will empirically demonstrate the effectiveness of our tuning procedure. We leave proving rigorous guarantees on the adaptive tuning algorithm as an interesting future direction.

Algorithm 4 Adaptive subsampled IBLT

```
Input: Communication budget $C$, number of users $n$, user contribution bound $m$.
    Update: A tuning function that updates the subsampling parameter based on past observations.
Set $t_0 = \Theta\left(\frac{nB}{C}\right)$.
for $r = 0, 1, 2, \ldots, R$ do
    Each user $i \in B_r$ applies Algorithm 1 with threshold $t$ in to their local histogram with fresh randomness to get $h_i'$.
    Each user sends message $Y_i = f_i(h_i')$ where $f_i$'s are mappings from Lemma 4.3 with parameter $L_0, \gamma$ and fresh randomness.
    Server observes $\sum_{i \in B_r} Y_i$ and computes
        $$\hat{h}_r = \text{Dec}(\sum_{i \in B_r} Y_i)$$
    If the decoding is not successful, we let $\hat{h}_{r,j}$ be the all-zero vector.
    if The decoding is successful, then
        Set $\hat{s}_r = \|\hat{h}_r\|_0$.
    else
        Get an estimate $\hat{s}_r$ for $\|\sum_{i \in B_r} h_i'\|_0$ based on $\sum_{i \in B_r} Y_i$ using (3) and (4) (Appendix C).
    end if
    Set
        $$t_{r+1} = \textbf{Update}(t_r, C, \hat{s}_r).$$
end for
```

7. Experiments

In this section, we empirically evaluate our proposed algorithms (Algorithms 2 and 4) for the task of heavy hitter recovery and compare it with baseline methods including (1) Count-sketch based method; (2) IBLT-based method without subsampling (Algorithm 2 with $\tau = 1$). We measure communication cost in units of words (denoted as C) and each word unit is an int16 object (can be communicated with 2 bytes) in python and C++ for implementation purposes. We will mainly focus on string data with characters from ROOT consisting of lower-case letters, digits and special symbols in $\{' \, @ \, \# \, - \, ; \, * \, : \, . \, / \, \_\}$. Our code is available at https://github.com/google-research/federated. Below are the details of our implementation.

Count-median sketch. We use H hash functions, each with domain size W and the total communication cost is

$C = H \cdot W$ words[3]. In the R-round setting, for each round r, we loop over all $x \in \mathcal{X}$ and compute an estimate $\hat{h}_r(x)$ and the recovered heavy hitters are those with $\sum_r \hat{h}_r(x) \geq \tau$. Note that in the open-domain setting, $d = |\mathcal{X}|$ can be large and this decoding procedure can be computationally infeasible. There are more computationally feasible variants including tree-based decoding but these come at the cost of higher communication cost or lower utility. We stick to the described version in this work and show that our proposed algorithms outperform this computationally expensive version. The advantage will be be at least as large when comparing to the more computationally feasible versions.

IBLT-based method. In our experiment, each IBLT data structure is of size $8L_0$, where L_0 is the targeted capacity for IBLT. We state more details about how the size is computed in Appendix C.

We consider fixed subsampling and adaptive subsampling. For fixed subsampling, when the max number of items in each round is upper bounded by $M_{\max}$, we set the subsampling parameter in Algorithm 1, to be $t = \max\{1, \min\{\frac{M_{\max}}{L_0}, \frac{\tau}{2}\}\}$. In practice, $M_{\max}$ can be obtained by system parameters such as the number of users in a round and the maximum contribution by a single user. Setting $t \leq \tau/2$ guarantees that the heavy hitters will be kept with high probability (Lemma A.2). And setting $t = \frac{M_{\max}}{L_0}$ guarantees that with high probability, the decoding of IBLT in each round will succeed and we can obtain more information. We set $b = 1$ in our experiments, the estimated and the heavy hitters are defined as those with estimated frequency at least τ. In the adaptive algorithm (Algorithm 4), for the update rule, we use

$$t_{r+1} = 0.5t_r + 0.5t_r \times \frac{\hat{s}_r}{C}$$

We leave designing better update rules as future work.

Client data simulation. We take the ground-truth distribution of strings in the Stackoverflow dataset in Tensorflow Federated and truncate them to the first 3 characters in set ROOT. This is to make sure that the computation is feasible for Count-median Sketch. And the data universe is of size $d = 97336$. In each round, we take $M_r \sim \mathcal{N}(M, M/10)$ *i.i.d.* samples from the this distribution and encode them using the algorithms mentioned above. In the experiment, we assume all samples come from different users ($m = 1$). For Count-sketch, this won't affect the performance. For IBLT with threshold sampling, this will be equivalent to IBLT with binomial sampling. And by (Duffield et al., 2005a), this will only increase the variance of the noise introduced in the sampling process. The metric we use is the F1 score of real heavy hitters (heavy hitters with true cumulative frequency at least τ) and the estimated heavy hitters.

We take $R \in \{10, 30, 50, 100\}$, $\tau \in \{20, 50, 100, 200\}$, $M \in \{1000, 2000, 5000, 10000\}$ and $C \in \{100, 200, 500, 1000, 2000, 5000, 8000, 10000, 20000, 30000, 40000\}$. For Count-median method, we take the max F1 score over all $H \in \{5, 7, 9, 11\}$ for each communication cost. We run each experiment for 5 times and compute the mean and standard deviation of the obtained F1 score. Our proposed algorithms consistently outperforms the sketching based method. Below we list and analyze a few representative plots.

In Figure 1, we plot the F1 score comparison under different communication costs when $R = 30, \tau = 50, M = 10000$. It can be seen that our proposed algorithms significantly outperforms the Count-sketch method. Among the IBLT-based methods, Subsampled IBLT with adaptive tuning is performing the best. For non-interactive algorithms, subsampled IBLT with fixed subsampling probability is better compared to the unsampled counter part for a wide range of small capacity.

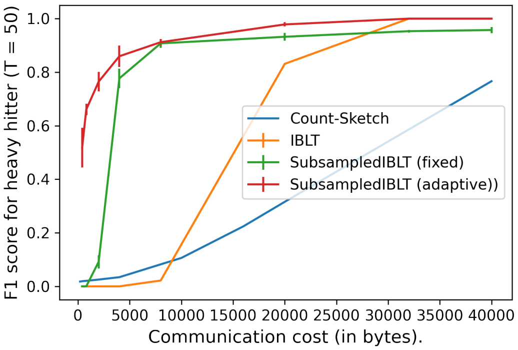

Figure 1. F1 score comparison under different communication cost ($R = 30, \tau = 50, M = 10000$). Each F1 score is an average of 5 runs and the error bar represents 3x the standard deviation of the runs.

In Figure 2, we plot the F1 score comparison under different round numbers when $C = 10000, \tau = 200, M = 10000$. As we can see, the performance of Count-sketch decreases significantly when the number of rounds increase while the performance of IBLT-based methods remains relatively flat, which is consistent with the theoretical results. The slight increase in the F1 score when R increases might be due to the *i.i.d.* generating process of the data in each round. As R increases, we get more information about the underlying distribution and this effect outweighs additional noise introduced by multiple rounds. Better understanding of this effect is an interesting further direction.

[3] In our experiments, mn will be between $\sim$ 1000 and $\sim$ $10,000$, and hence one word is enough to store an entry in the sketch.

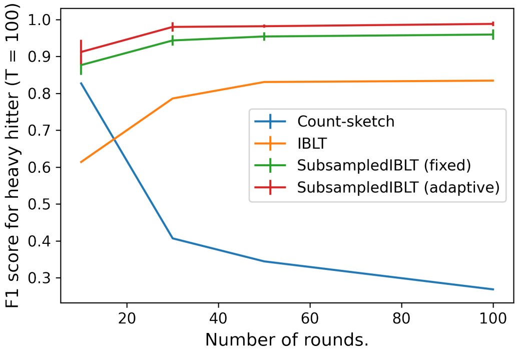

Figure 2. F1 score comparison with different number of rounds ($\tau = 200, M = 10000, C = 10000$). Each F1 score is an average of 5 runs and the error bar represents 3x the standard deviation of the runs.

In Figure 3, we further demonstrate our adaptive tuning method by showing that it is comparable with the best possible subsampling parameter in a candidate set. More specifically, we run subsampled IBLT with $t \in \{1, 1.25, 2, 5, 10, 20, 50, 100\}$ for all communication costs. And the F1 score for SubsampledIBLT (best fixed) is defined as the best F1 score among these candidates. Our result shows that the performance of tha adaptive algorithm is in-par with the best fixed subsampling parameter. It outperforms the best fixed subsampling parameter in certain cases because the set of subsampling parameters we choose from has limited granularity and hence the adaptive algorithm might find better parameters for the underlying instance.

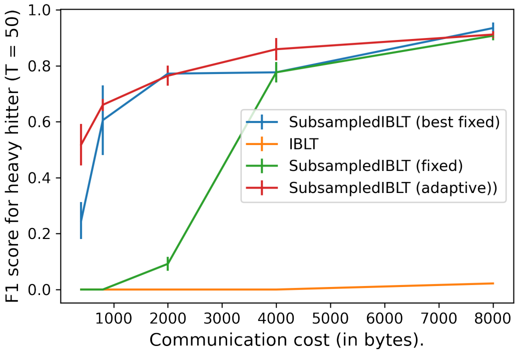

Figure 3. F1 score comparison (adaptive vs best fixed probability) ($\tau = 200, M = 10000, C = 5000$). Each F1 score is an average of 5 runs and the error bar represents 3x the standard deviation of the runs.

8. Conclusion

We provided lower bounds and matching upper bounds for central tasks in multi-round distributed data analysis: heavy hitters recovery and approximate histograms over large domains. Our findings show how porting single-round approaches based on standard sketching does not achieve optimality, and how this can be cleverly achieved via subsampled IBLTs. Several interesting and non-trivial questions remain to be addressed, including (a) developing distributed differential privacy schemes that are provably optimal for this problem, and (b) developing (non-linear) cryptographic (or other secure) primitives that allow us to extract heavy hitters with smaller (sublinear in mn) communication.

9. Acknowledgments

The authors thank Wennan Zhu for early discussions on the work, and Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, Amer Sinha, and Ameya Velingker for proposing IBLT as the linear data structure in federated heavy hitter recovery.

References

Acharya, J., Canonne, C. L., Sun, Z., and Tyagi, H. Unified lower bounds for interactive high-dimensional estimation under information constraints. *arXiv preprint arXiv:2010.06562*, 2020.

Bar-Yossef, Z., Jayram, T., Kumar, R., and Sivakumar, D. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2003.11.006. URL https://www.sciencedirect.com/science/article/pii/S0022000003001855. Special Issue on FOCS 2002.

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. Secure single-server aggregation with (poly)logarithmic overhead. In *CCS*, pp. 1253–1269. ACM, 2020.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3133982. URL https://doi.org/10.1145/3133956.3133982.

Bonawitz, K. A., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. Towards federated learning at scale: System design. In *MLSys*. mlsys.org, 2019.

Boneh, D., Boyle, E., Corrigan-Gibbs, H., Gilboa, N., and Ishai, Y. Lightweight techniques for private heavy hitters. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 762–776, 2021. doi: 10.1109/SP40001.2021.00048.

Braverman, M., Garg, A., Ma, T., Nguyen, H. L., and Woodruff, D. P. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 1011–1020, 2016.

Charikar, M., Chen, K., and Farach-Colton, M. Finding frequent items in data streams. In *Automata, Languages and Programming: 29th International Colloquium, ICALP 2002 Málaga, Spain, July 8–13, 2002 Proceedings 29*, pp. 693–703. Springer, 2002.

Chen, W.-N., Özgür, A., Cormode, G., and Bharadwaj, A. The communication cost of security and privacy in federated frequency estimation, 2022. URL https://arxiv.org/abs/2211.10041.

Cormode, G. and Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005. ISSN 0196-6774. doi: https://doi.org/10.1016/j.jalgor.2003.12.001. URL https://www.sciencedirect.com/science/article/pii/S0196677403001913.

Cormode, G. and Muthukrishnan, S. Combinatorial algorithms for compressed sensing. In *2006 40th Annual Conference on Information Sciences and Systems*, pp. 198–201. IEEE, 2006.

Corrigan-Gibbs, H. and Boneh, D. Prio: Private, robust, and scalable computation of aggregate statistics. In *NSDI*, pp. 259–282. USENIX Association, 2017.

Corrigan-Gibbs, H., Boneh, D., Chen, G., Englehardt, S., Helmer, R., Hutten-Czapski, C., Miyaguchi, A., Rescorla, E., and Saint-Andre, P. Privacy-preserving firefox telemetry with prio. https://rwc.iacr.org/2020/slides/Gibbs.pdf, 2020.

Donoho, D. L. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Duffield, N., Lund, C., and Thorup, M. Learn more, sample less: control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51(5): 1756–1775, 2005a. doi: 10.1109/TIT.2005.846400.

Duffield, N., Lund, C., and Thorup, M. Learn more, sample less: control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51(5): 1756–1775, 2005b. doi: 10.1109/TIT.2005.846400.

Gilbert, A. C., Li, Y., Porat, E., and Strauss, M. J. Approximate sparse recovery: optimizing time and measurements. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 475–484, 2010.

Goodrich, M. T. and Mitzenmacher, M. Invertible bloom lookup tables. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 792–799, 2011. doi: 10.1109/Allerton.2011.6120248.

Han, Y., Özgür, A., and Weissman, T. Geometric lower bounds for distributed parameter estimation under communication constraints. *IEEE Transactions on Information Theory*, 67(12):8248–8263, 2021. doi: 10.1109/TIT.2021.3108952.

Hu, C., Li, J., Liu, Z., Guo, X., Wei, Y., Guang, X., Loukides, G., and Dong, C. How to make private distributed cardinality estimation practical, and get differential privacy for free. In *USENIX Security Symposium*, pp. 965–982. USENIX Association, 2021.

Jayram, T. S. Hellinger strikes back: A note on the multiparty information complexity of and. In Dinur, I., Jansen, K., Naor, J., and Rolim, J. (eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 562–573, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-03685-9.

Melis, L., Danezis, G., and Cristofaro, E. D. Efficient private statistics with succinct sketches. In *NDSS*. The Internet Society, 2016.

Minton, G. T. and Price, E. Improved concentration bounds for count-sketch. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 669–686. SIAM, 2014.

Mitzenmacher, M. and Upfal, E. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017. ISBN 110715488X.

Molloy, M. Cores in random hypergraphs and boolean formulas. *Random Structures & Algorithms*, 27(1):124–135, 2005.

A. Proof of Theorem 4.1

Note that the algorithm can be viewed as $b{:=}\lceil 20\log(\frac{40mnR}{\tau\beta})\rceil$ independent runs of a basic protocol, each of which returns a list H_i of potential heavy hitters. We assume $b \geq 260$, else we take $b' = \max\{b, 260\}$ and the result will change by at most a constant factor.

The next lemma states that the probabilities of heavy elements and tail elements falling in the list.

Lemma A.1. *All H_j defined in Algorithm 2 satisfy that, if $h^{[R]}(x) \geq \tau$,*

$$\Pr\left(x \in H_j\right) \geq 4/5.$$

Else if $h^{[R]}(x) \leq \tau/10$,

$$\Pr\left(x \in H_j\right) \leq \frac{2h^{[R]}(x)}{\tau}.$$

Before proving the lemma, we first show how Theorem 4.1 can be implied by Lemma A.1.

By Lemma A.1, for x with $h^{[R]}(x) \geq \tau$, we have

$$\Pr\left(x \in H\right) \geq \Pr\left(\mathrm{Binom}(b, 4/5) \geq b/2\right) \geq 1 - \frac{\beta\tau}{40mnR},$$

where the last inequality follows from standard concentration bounds for Binomial random variables (*e.g.,* Chernoff bound (Mitzenmacher & Upfal, 2017)).

Hence by union bound, we have

$$\Pr\Big(\{x \in [d] \mid h^{[R]}(x) \geq \tau\} \subset H\Big) \geq 1 - \frac{\beta}{40}.$$

For any x, with $h^{[R]}(x) \leq \tau/10$, by Lemma A.1, we have

$$\Pr\left(x \in H\right) \leq \Pr\left(\mathrm{Binom}\left(b, \frac{2h^{[R]}(x)}{\tau}\right) \geq b/2\right) \leq \frac{b+1}{2}\left(\frac{8e}{5} \cdot \frac{2h^{[R]}(x)}{\tau}\right)^{b/2},$$

where the last inequality follows from Binomial tail bound (see Lemma E.1).

Hence by union bound we have

$$\begin{aligned}
&\Pr\Big(\{x \in [d] \mid h^{[R]}(x) \leq \tau/10\} \cap H \neq \emptyset\Big) \\
\leq& \sum_{x: h^{[R]}(x) \leq \tau/10} \frac{b+1}{2}\left(\frac{16eh^{[R]}(x)}{5\tau}\right)^{b/2} \\
\leq& \frac{20mnR}{\tau}\frac{b+1}{2}\left(\frac{8e}{25}\right)^{b/2} \qquad (1) \\
\leq& \frac{20mnR}{\tau}e^{-\frac{b}{20}} \qquad (2) \\
\leq& \frac{\beta}{2},
\end{aligned}$$

where (1) follows from $x^{b/2} + y^{b/2} \leq (x+y)^{b/2}$, and hence we can combine symbols to increase the sum of tail probability and end up with at most $\frac{20mnR}{\tau}$ symbols with frequencies at most $\tau/10$. (2) follows from the inequality $(x + 1/2)(8e/25)^x \leq e^{-x/10}$ for $x \geq 130$.

By union bound, we get the guarantee claimed in Theorem 4.1.

Proof of Lemma A.1: The proof mainly consists of two parts. We will first show that local subsampling will keep each heavy hitter with a high probability and each tail element with a low probability, stated in Lemma A.2. We will then show that after local subsampling, the number of unique elements in each round will decrease so that the decoding in Algorithm 2 will succeed with high probability.

Lemma A.2. *Let $h_j^{'[R]}$ be the aggregation of locally subsampled histogram for run j, i.e.,*

$$h_j^{'[R]} = \sum_{r \in [R]} \sum_{i \in B_r} h_{i,j}.$$

Then if $h^{[R]}(x) \geq \tau$,

$$\Pr\left(h_j^{'[R]}(x) > 0\right) \geq 1 - \frac{1}{e^2}.$$

Else if $h^{[R]}(x) \leq \tau/10$,

$$\Pr\left(x \in H_j\right) \leq \frac{2h^{[R]}(x)}{\tau}.$$

Proof. When $h^{[R]}(x) \geq \tau$,

$$\Pr\left(h_j^{'[R]}(x) > 0\right) = 1 - \Pi_{r \in [R], i \in B_r} \min\{1 - \frac{2h_{i,j}(x)}{\tau}, 0\} \geq 1 - \Pi_{r \in [R], i \in B_r} e^{-\frac{2h_{i,j}(x)}{\tau}} = 1 - e^{-\frac{2h^{[R]}(x)}{\tau}} \geq 1 - \frac{1}{e^2}.$$

When $h^{[R]}(x) \leq \tau/10$

$$\Pr\left(h_j^{'[R]}(x) > 0\right) = 1 - \Pi_{r \in [R], i \in B_r}\left(1 - \frac{2h_{i,j}(x)}{\tau}\right) \leq 1 - \left(1 - \sum_{r \in [R], i \in B_r} \frac{2h_{i,j}(x)}{\tau}\right) = \frac{2h^{[R]}(x)}{\tau}.$$

□

The next lemma shows that with high probability, the number of elements in each round will decrease by least a factor of τ.

Lemma A.3. *With probability at least $1 - 1/32$, we have*

$$\max_{r \in [R]} \{\|h'_r\|_0\} = O\Big(\frac{mn}{\tau} \log R\Big).$$

Proof. Since all rounds are independent, it would be enough to show that $\forall i$, with probability at least $1 - 1/32R$, we have

$$\|h'_r\|_0 = O\Big(\frac{mn}{\tau} \log R\Big).$$

To see this, we have

$$\Pr\left(\|h'_r\|_0 \geq \frac{2mn}{\tau} \log R\right) \leq \Pr\left(\text{Binom}\left(mn, \frac{1}{\tau}\right) \geq \frac{2mn}{\tau} \log R\right) \leq \frac{1}{32R},$$

where the first step follows from that the left hand side is maximized when all mn elements in h_r are distinct, and the second step follows from standard binomial tail bound when $mn > 4\tau$ and $R > 32$. □

Finally, it would be enough to show that when the condition in Lemma A.3 holds, the decoding of the aggregated IBLT will succeed with high probability. This is true since by Lemma 4.3 and union bound, we have

$$\Pr\left(\forall j, \hat{h}_j^{[R]} = h_j^{'[R]}\right) \geq 1 - R \cdot (\frac{mn}{\tau} \log R)^{2-\gamma} \geq 1 - 1/32,$$

where the last inequality holds when $mn > 4\tau$ and $R > 32$. Combining the above and Lemmas A.2 and A.3, we conclude the proof since $1/e^2 + 1/32 + 1/32 \leq 1/5$. □

B. Proof of Theorem 5.1

We start with the case when $\tau \leq \sqrt{R}$. In this case, Algorithm 3 implements Algorithm 2 with $\tau = 1$ and returns the obtained histogram in Line 11. Notice that when $\tau = 1$, the subsampling step is trivial and each user encodes their entire histogram. Hence as long as long the decoding of IBLT succeeds (as promised in the performance analysis of Algorithm 2), we recover the histogram perfectly, *i.e.,* $\hat{h}^{[R]} = h^{[R]}$. And the communication cost will be $\tilde{\Theta}(mn)$.

Next we focus on the case when $\tau \geq \sqrt{R}$. We will condition on the event that the list H obtained in Line 8 of Algorithm 3 is a τ approximate heavy hitter set and hence setting $\hat{x} = 0$ for $x \notin H$ won't introduce error larger than τ.

The rest of the proof follows similarly as the standard proof for Count-sketch. Since $b = \lceil \log \left(\frac{4mnR}{\tau\beta} \right) \rceil$, it would be enough to prove that $\forall x \in \mathcal{X}$, with probability at least 2/3, we have

$$| \sum_{r \in [R]} T_r(j, g_j(x)) \cdot s_{j,r}(x) - h^{[R]}(x)| = O(\tau).$$

Let

$$\begin{aligned}
\hat{h}_j(x) :=& \sum_{r \in [R]} T^{(r)}(j, g_j(x)) \cdot s_{j,r}(x) \\
=& \sum_{r \in [R]} \sum_{x'} \mathbb{1} \left\{ g_j(x') = g_j(x) \right\} s_{j,r}(x') s_{j,r}(x) \cdot h^{(r)}(x') \\
=& \sum_{x'} \mathbb{1} \left\{ g_j(x') = g_j(x) \right\} \sum_{r \in [R]} s_{j,r}(x') s_{j,r}(x) \cdot h^{(r)}(x') \\
=& h^{[R]}(x) + \sum_{x' \neq x} \mathbb{1} \left\{ g_j(x') = g_j(x) \right\} \sum_{r \in [R]} s_{j,r}(x') s_{j,r}(x) \cdot h^{(r)}(x')
\end{aligned}$$

Then we have $\mathbb{E} \left[\hat{h}_j(x) = h^{[R]}(x) \right]$. Next we provide a bound on the variance. Let $H_{10\tau/\sqrt{R}}$ be the set of elements with frequency at least $10\tau/\sqrt{R}$, then we have $|H_{10\tau/\sqrt{R}}| \leq \frac{mn\sqrt{R}}{10\tau}$. Since $w = \lceil \frac{10mn\sqrt{R}}{\tau} \rceil$, we have with probability at least 5/6,

$$\sum_{x' \in H_{10\tau/\sqrt{R}}, x' \neq x} \mathbb{1} \left\{ g_j(x') = g_j(x) \right\} = 0.$$

Conditioned on this event, we have

$$\begin{aligned}
\mathbb{E} \left[\left(\hat{h}_j(x) - h^{[R]}(x) \right)^2 \right] &= \mathbb{E} \left[\left(\sum_{x' \notin H_{10\tau/\sqrt{R}}, x' \neq x} \mathbb{1} \left\{ g_j(x') = g_j(x) \right\} \sum_{r \in [R]} s_{j,r}(x') s_{j,r}(x) \cdot h^{(r)}(x') \right)^2 \right] \\
&\leq \frac{\max_{x' \notin H_{10\tau/\sqrt{R}}} h^{[R]}(x) \sum_{x' \notin H_{10\tau/\sqrt{R}}} h^{[R]}(x)}{w} \\
&\leq \tau^2.
\end{aligned}$$

Hence with probability at least $5/6$, we have

$$\mathbb{E} \left[\left| \hat{h}_j(x) - h^{[R]}(x) \right| \right] \leq \sqrt{6}\tau.$$

We conclude the proof by a union bound over the two events.

C. Additional details on IBLT

Intuition on `ListEntries` for IBLT. The intuition behind the IBLT construction is as follows: Start with an array $\mathcal{B}$ of length ℓ containing 4-tuples of the form $(0, 0, 0, 0)$. To insert pair (x, v) hash the tuple $(x, \tilde{x}, v, 1)$ into k locations $l_1, \ldots, l_k$

in $\mathcal{B}$ based on the key x, where $\tilde{x} := G(x)$ is a hash of x into a sufficiently large domain so that collision probability is sufficiently unlikely. Then add, using component-wise sum, (x, $\tilde{x}$, v, 1) to the contents of $\mathcal{B}$ in all locations $l_1, \ldots, l_k$. The ListEntries/Dec operation corresponds to the result of the following procedure: (1) find an entry (x_{sum}, $\tilde{x}_{sum}$, v_{sum}, j) such that $G(x_{sum}/j) = \tilde{x}_{sum}/j$ holds, (2) add $(x_{sum}/j, v_{sum})$ to the output, and (3) remove the pair $(x_{sum}/j, v_{sum})$ by subtracting (x_{sum}, $\tilde{x}_{sum}$, v_{sum}, j) from the entries $l'_1, \ldots, l'_k$ in the array $\mathcal{B}$ to which an insertion would add the tuple for key x_{sum}/j and get back to step (1). The process of listing entries a.k.a "peeling off" $\mathcal{B}$. might terminate before the IBLT is empty. This is the failure procedure in Lemma 4.3, which corresponds to the natural procedure to find a 2-core in a random graph (Goodrich & Mitzenmacher, 2011).

Sketch size. The above intuition corresponds to the IBLT construction variant from (Goodrich & Mitzenmacher, 2011) that can handle duplicates. It can be implemented with four length ℓ vectors with entries in $[d]$, $\mathtt{Im}(G)$, $[mn]$, $[mn]$, respectively. In terms of concrete parameters (see (Goodrich & Mitzenmacher, 2011) for details), $k = 3, \ell > 1.3L_0$, and $G = \mathbb{Z}_p$ with $p = 2^{31} - 1$ give good performance, and require $1.3L_0(32 + \log_2 d + 2\log_2(mn))$ bits. For the experiment setting considered in Section 7, this is will take at most $8L_0$ words.

Cardinality estimation from saturated IBLT. Lemma 4.3 tells us that a tight bound L_0 on the number of distinct non-zero indices in the intended histogram, can save us space in an IBLT encoding. However, getting that bound wrong results in an undecodable IBLT. While in the single round case all is lost, in the multi-round setting we leverage a property of undecodable IBLTs that helps update our L_0 bound for subsequent rounds after a failed round. This is the main ingredient for our adaptive tuning heuristic presented in Section 6.

Let $\mathcal{B}$ be an undecodable IBLT, and let S be the size of the undecoded graph of $\mathcal{B}$. Also let ℓ be the size of $\mathcal{B}$, and let N the (unknown) number of distinct elements inserted in $\mathcal{B}$ (note that N corresponds to the correct bound L_0 that enables decoding). By (Molloy, 2005), we have the following relation: For large enough N, if $S < \ell$, we have

$$\frac{S}{C} = 1 - e^{-x}(1 + x) + o(1), \tag{3}$$

where x is the greatest solution to

$$\frac{6N}{C} = \frac{2x}{(1 - e^{-x})^2}. \tag{4}$$

Hence we can have an estimate for N (and thus a correct choice for L_0 in a subsequent round) based on S and C. We first solve (3) ignoring the $o(1)$ term to get x and then plug x and C into (4) to get an estimate for N. As mentioned above we leverage this fact in Section 6.

D. Proof of lower bounds.

D.1. Proof of Theorem 4.2

We will focus on the case when $R = 1$ since the claimed bound doesn't depend on R and we can assume there is no data in other $R - 1$ rounds. We will consider the case when $10 < \tau < n/4$.

We prove the theorem by a reduction to the set disjointness problem (Bar-Yossef et al., 2004; Jayram, 2009). The set disjointness problem ($\textsc{Dist}_{t,d}$) considers the setting where t users where user i has a set of elements $S_i \subset \{1, 2, \ldots, d\}$. The goal is to distinguish between the following two chase with success probability at least $4/5$.

1. All S_i's disjoint.
2. There exists $x \in [d]$ such that for all $i, j \in [t]$, $S_i \cap S_j = \{x\}$.

And the goal is to minimize the size of the transcript of all communications among all users. More specifically, we will use the following lemma:

Lemma D.1 ((Jayram, 2009)). *Any protocol that solves* $\textsc{Dist}_{t,d}$ *must have a transcript of size at least* $\Theta(d/t)$.

Next we show that $\textsc{Dist}_{t,d}$ with $t = \tau$ and $d = mn/2$ can be reduced to the approximate heavy hitter problem. We divide users into $\tau + 1$ groups. For $i \in [\tau]$, the ith group has $n_i = \lceil |S_i|/m \rceil$ users. And let $\tilde{S}_i$ be set of all elements held by users in group i. We partition S_i to subsets of size at most m and distribute them to users in group i arbitrarily. This can be done since $mn_i \geq |S_i|$. The total number of users in the first τ groups is $\sum_{i\in[\tau]} n_i \leq \frac{d+\tau}{m} + \tau \leq n$. The $\tau+1$ group has $n - \sum_{i\in[\tau]} n_i$ users and each user has zero element.

Suppose there exists a τ-ApproxHH linear sketch algorithm with communication cost per-user $o(\frac{mn}{\tau})$. When S_i's are disjoint, all elements in $[d]$ will have frequency $1 < \tau/10$. The algorithm should output an empty list. When S_i's have an unique intersection, the element will have frequency τ, and hence the algorithm should output a list with size 1. By distinguishing between the two cases, the ApproxHH algorithm can be used to solve $\textsc{Dist}_{t,d}$.

Moreover, under linear sketching constraint, the size of the transcript is the same as the per-user communication. Hence we conclude the proof by noticing that this violates Lemma D.1.

D.2. Proof of Theorem 5.2

Here we prove a stronger version of the lower bound where in each round r, the communication among users is not limited but the users in B_r must compress $h^{(r)}$ to an element $Y^{(r)} \in G_r$ with $|G_r| \leq 2^\ell$, which is observed by the server. And the server will then obtain an approximate histogram $\widehat{h}^{[R]}$ based on $\Pi = (Y^{(1)}, \ldots, Y^{(R)}, U)$. For a given τ, next we show that any protocol with $\ell = o\Big(\min\{\frac{mn\sqrt{R}}{\tau}, mn\}\Big)$ won't solve τ-approximate heavy hitter with error probability at most 1/5. We will focus on the case when $\tau \geq \sqrt{R}$ and $\ell = o\Big(\frac{mn\sqrt{R}}{\tau}\Big)$. When $\tau < \sqrt{R}$, the bound follows by setting $\tau = \sqrt{R}$ and the fact that the problem gets harder as τ decreases. To simply the proof, we assume $R \geq 400$ without loss of generality.

We consider histograms $h^{(r)}, \forall r \in [R]$ supported over the domain 10ℓ and are generated *i.i.d.* from a distribution P. Let Z be uniformly distributed over $\{\pm 1\}^{5\ell}$, and under distribution P_Z, we have $\forall r \in [R], i \in [5\ell]$,

$$h^{(r)}(2i) = \begin{cases} \frac{mn}{5\ell} & \text{with prob } \frac{1}{2} + \frac{10}{\sqrt{R}} Z_i. \\ 0 & \text{with prob } \frac{1}{2} - \frac{10}{\sqrt{R}} Z_i. \end{cases}$$

and

$$h^{(r)}(2i-1) = 1 - h^{(r)}(2i).$$

It can be check that $\|h^{(r)}\|_1 = mn$ with probability 1. We prove the theorem by contradiction. If the protocol solves τ-approximate heavy hitter with error probability at most 1/5, let

$$\hat{Z}_i = \mathbb{1}\left\{\hat{h}^{[R]}(2i) > \frac{mnR}{10\ell}\right\}.$$

We have

$$\begin{aligned}\Pr\left(\hat{Z}_i \neq Z_i\right) &\leq \Pr\left(\left|\hat{h}^{[R]}(2i) - h^{[R]}(2i)\right| \geq \frac{mn\sqrt{R}}{\ell}\right) + \Pr\left(\left|h^{[R]}(2i) - \frac{mnR}{5\ell}\left(\frac{1}{2} + \frac{10}{\sqrt{R}} Z_i\right)\right| \geq \frac{mn\sqrt{R}}{\ell}\right) \\ &\leq \frac{1}{5} + \frac{1}{25} = \frac{6}{25},\end{aligned}$$

where the first probability is bounded by the success probability of the algorithm and the second probability is bounded using Hoeffding bound. Hence we have

$$\sum_{i\in[5\ell]} I(Z_i; \Pi) \geq \sum_{i\in[5\ell]} I(Z_i; \hat{Z}_i) \geq \sum_{i\in[5\ell]} (1 - H(\frac{5}{26})) \geq 2\ell,$$

where $H(p)$ is the Shannon entropy of a Bernoulli random variable with success probability $6/25$.

To upper bound $\sum_{i\in[5\ell]} I(Z_i; \Pi)$, we notice that the vector

$$\left(\frac{5\ell h^{(r)}(2)}{mn}, \frac{5\ell h^{(r)}(4)}{mn}, \ldots, \frac{5\ell h^{(r)}(10\ell)}{mn}\right).$$

follows a product distribution with the marginal of each coordinate being a Bernoulli distribution. Hence by standard arguments on communication-limited estimation of product of Bernoulli random variables (*e.g.,* in (Braverman et al., 2016;

Han et al., 2021; Acharya et al., 2020)). In particular, following almost the same steps as in Acharya et al. (2020, Section 7.1),

$$\sum_{i\in[5\ell]} I(Z_i;\Pi) \le R\cdot\left(\frac{1}{\sqrt{R}}\right)^2 \ell = \ell,$$

which leads to a contradiction. This concludes the proof.

E. Binomial tail bound.

Lemma E.1. *Let* $X \sim \mathrm{Binom}(n,p)$ *be a binomial distribution, when* $n > 10$ *and* $p < 1/5$*, we have*

$$\Pr\left(X \ge n/2\right) \le \frac{n+1}{2}\left(\frac{8ep}{5}\right)^{n/2}.$$

Proof.

$$\begin{aligned}
\Pr\left(X \ge n/2\right) &= \sum_{i=\lfloor (n+1)/2\rfloor}^{n} \Pr\left(X = i\right) \\
&= \sum_{i=\lfloor (n+1)/2\rfloor}^{n} \binom{n}{i}(1-p)^{n-i}p^i \\
&\le \frac{n+1}{2}\binom{n}{\lceil n/2\rceil}((1-p)p)^{n/2} \qquad (5)\\
&\le \frac{n+1}{2}(2e)^{n/2}\left(\frac{4p}{5}\right)^{n/2} \qquad (6)\\
&= \frac{n+1}{2}\left(\frac{8ep}{5}\right)^{n/2},
\end{aligned}$$

where (5) follows from $\Pr\left(X = i\right)$ is monotonically decreasing when $i \ge n/2$ and (6) follows from standard bounds on binomial coefficients. □