# Private Data Stream Analysis for Universal Symmetric Norm Estimation (Full Version)

Anonymous submission

**Abstract**

We study how to release summary statistics on a data stream subject to the constraint of differential privacy. In particular, we focus on releasing the family of *symmetric norms*, which are invariant under sign-flips and coordinate-wise permutations on an input data stream and include $L_p$ norms, $k$-support norms, top-$k$ norms, and the box norm as special cases. Although it may be possible to design and analyze a separate mechanism for each symmetric norm, we propose a general parametrizable framework that differentially privately releases a number of sufficient statistics from which the approximation of all symmetric norms can be simultaneously computed. Our framework partitions the coordinates of the underlying frequency vector into different levels based on their magnitude and releases approximate frequencies for the "heavy" coordinates in important levels and releases approximate level sizes for the "light" coordinates in important levels. Surprisingly, our mechanism allows for the release of an *arbitrary* number of symmetric norm approximations without any overhead or additional loss in privacy. Moreover, our mechanism permits $(1+\alpha)$-approximation to each of the symmetric norms and can be implemented using sublinear space in the streaming model for many regimes of the accuracy and privacy parameters.

## 1 Introduction

The family of $L_p$ norms represent important statistics on an underlying dataset, where the $L_p$ norm of an $n$-dimensional frequency vector $x$ is defined as the number of nonzero coordinates of $x$ for $p=0$ and $L_p(x) = (x_1^p + \ldots + x_n^p)^{1/p}$ for $p>0$. Thus, $L_0$ norm counts the number of distinct elements in the dataset and, e.g., is used to detect denial of service or port scan attacks in network monitoring [ABRS03, EVF03], to understand the magnitude of quantities such as search engine queries or internet graph connectivity in data mining [PSF+01], to manage workload in database design [FST88], and to select a minimum-cost query plan in query optimization [SAC+79]. The $L_1$ norm computes the total number of elements in the dataset and, e.g., is used for data mining [CMR05] and hypothesis testing [IM08], while the $L_2$ norm, e.g., is used for training random forests in machine learning [Bre01], computing the Gini index in statistics [Lor05, Gin12], and network anomaly detection in traffic monitoring [KSZC03, TZ04]. Consequently, $L_p$ estimation has been extensively studied in the data stream model [AMS99, IW05, Ind06, Li08, KNPW11, And17, BVWY18, GW18, WZ20, WZ21]. The simplest streaming model is perhaps the insertion-only model, in which a sequence of $m$ updates increments coordinates of an $n$-dimensional frequency vector $x$ and the goal is to compute or approximate some statistic of $x$ in space that is sublinear in both $m$ and $n$. For a more formal introduction to the streaming model, see Section 2.2.

In many cases, the underlying dataset contains sensitive information that should not be leaked. Hence, an active line of work has focused on estimating $L_p$ norms for various values of $p$, while preserving differential privacy [MMNW11, BBDS12, SST20, BGK+21, WPS21].

**Definition 1.1** (Differential privacy)**.** *[DMNS06] Given $\varepsilon > 0$ and $\delta \in (0,1)$, a randomized algorithm $\mathcal{A} : \mathfrak{U}^* \to \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private if, for every neighboring streams $\mathfrak{S}$ and $\mathfrak{S}'$ and for all $E \subseteq \mathcal{Y}$,*

$$\mathbf{Pr}\left[\mathcal{A}(\mathfrak{S}) \in E\right] \le e^{\varepsilon} \cdot \mathbf{Pr}\left[\mathcal{A}(\mathfrak{S}') \in E\right] + \delta.$$

For example, [BBDS12] showed that the Johnson-Lindenstrauss transformation preserves differential privacy (DP), thereby showing one of the main techniques in the streaming model for $L_2$ estimation already guarantees DP. Similarly, [SST20] showed that the Flajolet-Martin sketch, which is one of the main approaches for $L_0$ estimation in the streaming model, also preserves DP.

Notably, algorithmic designs for $L_p$ estimation in the streaming model differ greatly and require individual analysis to ensure DP, which can be quite difficult due to the complexity of the various techniques. This is especially pronounced in the work of [WPS21], who studied the $p$-stable sketch that estimates the $L_p$ norm for $p \in (0, 2]$ [Ind06][1]. [WPS21] showed that for $p \in (0, 1]$, the $p$-stable sketch preserves DP, but was unable to show DP for $p \in (1, 2]$, even though the general algorithmic approach remains the same. Thus the natural question is whether differential privacy can be guaranteed for an approach that simultaneously estimates the $L_p$ norm in the streaming model, for all $p$. More generally, the family of $L_p$ norms are all symmetric norms, which are invariant under sign-flips and coordinate-wise permutations on an input data stream. Symmetric norms thus also include other important families of norms such as the $k$-support norms and the top-$k$ norms.

In this paper, we show that not only does there exist a differentially private algorithm for the estimation of symmetric norms in the streaming model, but also that there exists an algorithm that privately releases a set of statistics, from which estimates of all (properly parametrized) symmetric norms can be simultaneously computed. To illustrate the difference, suppose we wanted to release approximations of the $L_p$ norm of the stream for $k$ different values of $p$. To guarantee $(\varepsilon, \delta)$-DP for the set of $k$ statistics, we would need, by advanced composition, to demand $\left(\mathcal{O}\left(\frac{\varepsilon}{\sqrt{k}}\right), \mathcal{O}\left(\frac{\delta}{k}\right)\right)$-DP from $k$ instances of a single differentially private $L_p$-estimation algorithm, corresponding to the $k$ different values of $p$. Due to accuracy-privacy tradeoffs, the quality of the estimation will degrade severely as $k$ increases. By comparison, our algorithm releases a single set $C$ of private statistics. By post-processing, we can then estimate the $L_p$ norms for $k$ different values of $p$ while only requiring $(\varepsilon, \delta)$-DP from $C$. Hence, our algorithm can simultaneously handle any large number of estimations of symmetric norms without compromising the quality of approximation.

**Theorem 1.2.** *There exists a $(\varepsilon, \delta)$-differentially private algorithm that outputs a set $C$, from which the $(1 + \alpha)$-approximation to any norm with* maximum modulus of concentration *at most $M$ can be computed, with probability at least $1 - \delta$. The algorithm uses $M^2 \cdot \mathrm{poly}\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log n, \log \frac{1}{\delta}\right)$ bits of space.*

The maximum modulus of concentration of a norm measures the worst-case ratio of the maximum value of a norm on the $L_2$-unit sphere to the median value of a norm on the $L_2$-unit sphere, where the median can be taken over any restriction of the coordinates, and can intuitively quantify the

[1] $L_p$ for $p \in (0, 1)$ does not satisfy the triangle inequality and therefore is not a norm, but is still well-defined/well-motivated and can be computed

complexity of computing a norm. For example, the $L_1$ norm is generally "easy" to compute and has maximum modulus of concentration $\mathcal{O}(\log n)$. See Definition 2.16 for a more formal definition.

We emphasize that prior to our work, there is no algorithm that can handle private symmetric norm estimation for arbitrary symmetric norms, much less simultaneously for all parametrized symmetric norms. Although there is specific analysis for various norm estimation algorithms, e.g., see the discussion on related work, these algorithms require a specific predetermined norm for their input. Thus a separate private algorithm must be run for each estimation, which increases the overall space. Moreover, for a large number of queries, the privacy parameter will need to be much smaller due to the composition of privacy, and thus to ensure privacy, the utility of each algorithm is provably poor. Our algorithm sidesteps both the space and accuracy problems and is the first and only work to do so.

**Applications.** We briefly describe a number of specific symmetric norms that are handled by Theorem 1.2 and commonly used across various applications in machine learning. We first note the following parameterization of the previously discussed $L_p$ norms.

**Lemma 1.3.** *[MS09, KV07] For $L_p$ norms, we have that $\mathrm{mmc}(L) = \mathcal{O}(\log n)$ for $p \in [1, 2]$ and $\mathrm{mmc}(L) = \mathcal{O}\left(n^{1/2-1/p}\right)$ for $p > 2$.*

Thus our algorithm immediately introduces a differentially private mechanism for the approximation of $L_p$ norms that unlike previous work, e.g., [BBDS12, She19, CDKY20, SST20, BGK+21, WPS21], does not need to provide separate analysis for specific values of $p$. Moreover for constant-factor approximation, the space complexity is tight with the *optimal* $L_p$-approximation algorithms that do not consider privacy, up to polylogarithmic factors [KNW10, LW13, Gan15, WZ21].

**Definition 1.4** ($Q$-norm and $Q'$-norm)**.** *We call a norm $L$ a $Q$-norm if there exists a symmetric norm $L'$ such that $L(x) = L'(x^2)^{1/2}$ for all $x \in \mathbb{R}^n$. Here, we use $x^2$ to denote the coordinate-wise square power of $x$. We also call a norm $L'$ a $Q'$-norm if its dual norm is a $Q$-norm.*

The family of $Q'$-norms includes the $L_p$ norms for $1 \le p \le 2$, the $k$-support norm, and the box norm [Bha13] and thus $Q'$-norms have been proposed to regularize sparse recovery problems in machine learning. For instance, [AFS12] showed that $Q'$ norms have tighter relaxations than elastic nets and can thus be more effective for sparse prediction. Similarly, [MPS14] used $Q'$ norms to optimize sparse prediction algorithms for multitask clustering.

**Lemma 1.5.** *[BBC+17] $\mathrm{mmc}(L) = \mathcal{O}(\log n)$ for every $Q'$-norm $L$.*

Theorem 1.2 and Lemma 1.5 thus present a differentially private algorithm for $Q'$-norm approximation that uses polylogarithmic space.

**Definition 1.6** (Top-$k$ norm)**.** *The top-$k$ norm for a vector $x \in \mathbb{R}^n$ is the sum of the largest $k$ coordinates of $|x|$, where we use $|x|$ to denote the coordinate-wise absolute value of $x$.*

The top-$k$ norm is frequently used to understand the more general Ky Fan $k$-norm [WDST14], which is used to regularize optimization problems in numerical linear algebra. Whereas the Ky Fan $k$ norm is defined as the sum of the $k$ largest singular values of a matrix, the top-$k$ norm is equivalent to the Ky Fan $k$ norm when the input vector $x$ represents the vector of the singular values of the matrix.

**Lemma 1.7.** *[BBC+17]* $\text{mmc}(L) = \tilde{\mathcal{O}}\left(\sqrt{\frac{n}{k}}\right)$ *for the top-$k$ norm $L$.*

In particular, the top-$k$ norm for a vector of singular values when $k = n$ is equivalent to the Schatten-1 norm of a matrix, which is a common metric for matrix fitting problems such as low-rank approximation [LW20].

## 1.1 Algorithmic Intuition and Overview

Our starting point is the $L_p$ estimation algorithm of [IW05], which was parametrized by [BBC+17] to handle symmetric norms. For a $(1+\alpha)$-approximation, the algorithm partitions the $n$ coordinates of the frequency vector $x$ into powers of $\xi$-based on their magnitudes, where $\xi > 1$ is a fixed function of $\alpha$. Each partition forms a level set, so that the $i$-th level set consists of the coordinates of $x$ with frequency $[\xi^i, \xi^{i+1})$, but [IW05, BBC+17] showed that it suffices to accurately count the size of each *important* level set and zero out to the other level sets, where a level set is considered important if its size is large enough to contribute an $\frac{\alpha^2}{\log m}$ fraction of the symmetric norm.

**Private symmetric norm estimation in the centralized setting.** To preserve $(\varepsilon, \delta)$-differential privacy, one initial approach would be to treat the statistics as a histogram and add Laplacian noise with scale $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ to the frequency of each element. However, the level sets consisting of elements with frequencies between $[\xi^i, \xi^{i+1})$ for small $i$, say $i = 0$, could be largely perturbed by such Laplacian noise. Fortunately, if $i$ is small, the corresponding level set must contain a large number of elements if it is important, so it seems possible to privately release the size $\Gamma_i$ of the level set. Indeed, we can show that the $L_1$ sensitivity of the vector corresponding to level set sizes is small and so we can add Laplacian noise with scale $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ to each level set size. Hence if the level set has size $\Gamma_i$ roughly $\Omega\left(\frac{1}{\alpha\varepsilon}\right)$, then the Laplacian noise will affect $\Gamma_i$ by a $(1+\alpha)$-factor.

Unfortunately, there can be level sets that are both important and small in size. For example, if there is a single element with frequency $m$, then the size of the corresponding level set is just one. Then adding Laplacian noise with scale $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ will severely affect the size of the level set and thus the estimation of the symmetric norm. On the other hand, for $m > \frac{1}{\alpha\varepsilon}$, the frequency of the coordinate is quite large so again it seems like we can just add Laplacian noise with scale $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ and output the noisy frequency of the coordinate.

The main takeaway from these challenges is that we should handle different level sets separately. For the level sets of small coordinates, the important level sets must have large size and thus we can release noisy sizes. For the important level sets of large coordinates, we can release noisy frequencies of the coordinates.

**New approach: classifying and separately handling high, medium, and low frequency levels.** The main takeaway from these challenges is that we should handle different level sets separately. We partition the levels into three groups after defining thresholds $T_1$ and $T_2$, with $T_1 > T_2$. We define the "high frequency levels" as the levels whose coordinates exceed $T_1$ in frequency. The intuition is that because the high frequency levels have such large magnitude, their frequencies can be well-approximated by running an $L_2$-heavy hitters algorithm on the stream $S$.

We define the "medium frequency levels" as the levels whose coordinates are between $T_1$ and $T_2$ in frequency. These coordinates are not large enough to be detected by running an $L_2$-heavy hitters algorithm on the stream $S$. However, the sizes of these level sets must be large if the level set is important. Thus there exists a substream $S_j$ for which a large number of these coordinates are subsampled and their frequencies can be well-approximated by running an $L_2$-heavy hitters algorithm on the substream $S_j$.

Finally, we define the "low frequency levels" as the levels whose coordinates are less than $T_2$ in frequency. These coordinates are small enough that we cannot add Laplacian noise to their frequencies without affecting the level sets they are mapped to. Instead, we show that $L_1$ sensitivity for the level set estimations is particularly small for the low frequency levels. Thus, for these frequency levels, we report the size of the frequency levels rather than the identities of the heavy-hitters. We remark that if our goal was to just approximate the symmetric norms without preserving differential privacy, then it would suffice to just consider the high and medium frequency levels, since the low frequency levels are particularly problematic when Laplacian noise is added to the frequency vector. We also remark that we only use the thresholds $T_1$ and $T_2$ for the purposes of describing our algorithm – in the actual implementation of the algorithm, the thresholds $T_1$ and $T_2$ will be implicitly defined by each of the substreams. We summarize our new approach in Figure 1.

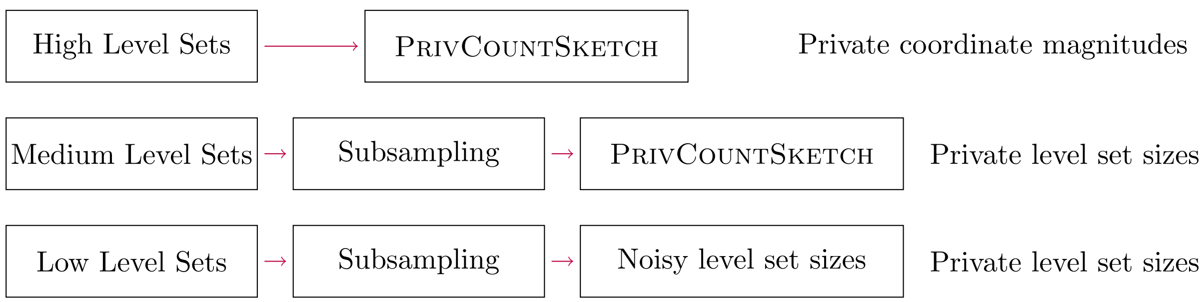


Fig. 1: Illustration of separate handling of the high, medium, and low level sets.

**Private symmetric norm estimation in the streaming model.** Although the previously discussed intuition builds towards a working algorithm, the main caveat is that so far, we have mainly discussed the centralized model, where space is not restricted and so each coordinate and thus each level set size can be counted exactly. In the streaming model, we cannot explicitly track the frequency vector, or even the frequencies of a constant fraction of coordinates. Instead, to estimate the sizes of each level set, [IW05, BBC+17] take the stream $S$ and form $s = \mathcal{O}(\log n)$ substreams $S_1, \ldots, S_s$, where the $j$-th substream is created by sampling the universe of size $n$ at a rate of $\frac{1}{2^{j-1}}$. Then $S_j$ will only consist of the stream updates to the particular coordinates of $x$ that are sampled. Thus in expectation, the frequency vector induced by $S_j$ will have sparsity $\frac{\|x\|_0}{2^{j-1}}$. Similarly, if a level set $i$ has size $\Gamma_i$, then $\frac{\Gamma_i}{2^{j-1}}$ of its members will be sampled in $S_j$ in expectation. It can then be shown through a variance argument that if level set $i$ is important, then there exists an explicit substream $j$ from which $\Gamma_i$ can be well-approximated using the $L_2$-heavy hitter algorithm CountSketch and as a result, the symmetric norm of $x$ can be well-approximated. The main point of the subsampling approach is that if there exists a level set with large size consisting of small coordinates, then the coordinates will not be detected by the CountSketch on $S$, but because $S_j$ has significantly smaller $L_2$ norm, then the coordinates will be detected by CountSketch on $S_j$.

However, adapting the subsampling and heavy-hitter approach introduces additional challenges for privacy. For instance, we can analyze the $L_2$-heavy hitter algorithm CountSketch and show that although the $L_1$ sensitivity of the estimated frequency for a single coordinate is small, the $L_1$ sensitivity of the estimated frequency for all the coordinates is large. Instead, we use the view that CountSketch is a composition function that first only estimates frequencies for the top poly $\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log n\right)$ and then outputs only those estimates that are above a certain threshold. Similarly, the Laplacian noise added to privately use CountSketch can alter the sizes of a significant number of level sets for small coordinates. Thus for the small coordinates (corresponding to the substreams $S_j$ with large $j$), we invoke CountSketch with much higher accuracy, so that with high probability, it will return *exactly* the frequencies for the small coordinates. For example, note that if the frequency $f_k$ of a coordinate $k \in [n]$ is at most $\frac{1}{2\alpha^2\varepsilon}$, then any $(1+\alpha^2\varepsilon)$-approximation to $f_k$ can be rounded to exactly recover $f_k$. This decreases the $L_1$ sensitivity of the vector of estimated level set sizes, therefore allowing us to add Laplacian noise without greatly affecting the quality of approximation.

### 1.2 Related Work

Non-private $L_p$ norm estimation is one of the fundamental problems in the streaming model, beginning with [AMS99]'s seminal work that tracks the inner product of the frequency vector with a random sign vector for $L_2$ estimation (as well as a telescoping argument for integer $p > 0$). [Ind06, Li08] later showed that this approach could be generalized for $p \in (0, 2]$ by tracking the inner product of the frequency vector with a vector with randomly generated $p$-stable variables, which only exist for $p \in (0, 2]$. For $p > 2$, [And17] gave an $L_p$ estimation algorithm using the max-stability property of exponential random variables. More generally, [IW05] introduced the framework of subsampling and using heavy-hitters for $L_p$ estimation, which [BBC$^+$17] parametrized to all symmetric norms. It should be emphasized that these techniques all handle the more general turnstile model, in which $\pm 1$ updates are allowed to each coordinate, rather than single positive increments. Hence our techniques also extend to the turnstile model with a minor change on the conditions.

Symmetric norms have also recently received attention in other big data models as well. [ANN$^+$17] studied approximate near neighbors for general symmetric norms while [LMV$^+$16] studied symmetric norm estimation for network monitoring. [SWY$^+$19] considered Orlicz norm regression and other loss functions where the penalty is a symmetric norm. [BWZ21] gave an algorithm to approximate the symmetric norm in the sliding window model, where updates in the data stream implicitly expire after a fixed amount of time.

Specific cases of private $L_p$ estimation in the streaming model have also been previously well-studied. [CDKY20, SST20] studied private $L_0$ estimation using the Flajolet-Martin sketch, while [WPS21] studied private $L_p$ estimation for $p \in (0, 1]$ using the $p$-stable sketch and [BBDS12, She19, CDKY20, BGK$^+$21] studied private $L_2$ estimation using the Johnson-Lindenstrauss projection. For fractional $p > 1$, private distribution estimation algorithms [ÁCC12, XZX$^+$13, BS15, WLL$^+$20] can be used to approximate the $L_p$ norm, but since the algorithms provide information over a much larger distribution, e.g., much larger histograms of frequencies, the privacy-accuracy trade-off is suboptimal and the space complexity is exponentially worse.

The related problem of privately releasing heavy-hitters in big data models has also been well-studied. [CLSX12] studied the problem of continually releasing $L_1$-heavy hitters in a stream while [DNP$^+$10] studied $L_1$-heavy hitters and other problems in the pan-private streaming model. The heavy-hitter problem has also received significant attention in the local model, e.g., [BS15, DKY17, AS19, BNS19, BNST20], where individual users should locally randomize their data before sending differentially private information to an untrusted server that aggregates the statistics across all users.

## 2 Preliminaries

In this section, we introduce definitions and simple or well-known results from differential privacy, sketching algorithms, and symmetric norms. For notation, we use $[n]$ for an integer $n > 0$ to denote the set $\{1, \ldots, n\}$. We also use the notation $\mathrm{poly}(n)$ to represent a constant degree polynomial in $n$ and we say an event occurs *with high probability* if the event holds with probability $1 - \frac{1}{\mathrm{poly}(n)}$. Similarly, we use $\mathrm{polylog}(n)$ to denote $\mathrm{poly}(\log n)$. Finally, for a parameter $c \geq 1$, we say that $X$ provides a $C$-approximation to a quantity $Y$ if $\frac{X}{C} \leq Y \leq C \cdot X$.

### 2.1 Differential privacy

We first recall standard definitions and results from differential privacy. We define the concept of neighboring streams used in Definition 1.1.

**Definition 2.1** (Neighboring streams). *Data streams $\mathfrak{S}$ and $\mathfrak{S}'$ are neighboring if there exists a single update $i \in [m]$ such that $u_i \neq u'_i$, where $u_1, \ldots, u_m$ are the updates of $\mathfrak{S}$ and $u'_1, \ldots, u'_m$ are the updates of $\mathfrak{S}'$.*

A standard approach to guarantee differential privacy is to add Laplacian noise, which is drawn from the following distribution.

**Definition 2.2** (Laplace distribution). *A random variable $x$ is drawn from the* Laplace distribution *with mean $\mu$ and scale $s > 0$, $x \sim \mathsf{Lap}(\mu, s)$, if the probability density function of $x$ is $\frac{1}{2s} \exp\left(-\frac{|x-\mu|}{s}\right)$. If $\mu = 0$, we say $x \sim \mathsf{Lap}(s)$.*

The amount of Laplacian noise that must be added to an output depends on the $L_1$ sensitivity of the function.

**Definition 2.3** ($L_1$ sensitivity). *We define the $L_1$ sensitivity of a function $f : \mathbb{N}^{|\mathfrak{U}|} \to \mathbb{R}^k$ by*

$$\Delta_f = \max_{x,y \in \mathbb{N}^{|\mathfrak{U}|}, \|x-y\|_1 = 1} \|f(x) - f(y)\|_1.$$

Intuitively, the $L_1$ sensitivity of a function is the largest amount that $f$ can change when a single update in the stream that defines $f$ changes.

**Definition 2.4** (Laplace mechanism). *Given a function $f : \mathbb{N}^{|\mathfrak{U}|} \to \mathbb{R}^k$, we define the Laplace mechanism by*

$$\mathcal{M}_L(x, f, \varepsilon) = f(x) + (X_1, \ldots, X_k),$$

*where $X_i \sim \mathsf{Lap}(\Delta f / \varepsilon)$.*

The Laplace mechanism is one of the most fundamental ways to ensure differential privacy:

**Theorem 2.5.** *[DR14] The Laplace mechanism preserves $(\varepsilon, 0)$-differential privacy.*

Privately releasing multiple statistics that are individually differentially private can also be done, but comes at a slight cost.

**Theorem 2.6** (Composition and post-processing of differential privacy). *[DR14] Let $\mathcal{A}_i : \mathfrak{U}_i \to X_i$ be an $(\varepsilon_i, \delta_i)$-differential private algorithm for $i \in [k]$. Then $\mathcal{A}_{[k]}(x) = (\mathcal{A}_1(x), \ldots, \mathcal{A}_k(x))$ is $\left(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i\right)$-differentially private. Furthermore, if $g_i : X_i \to X_i'$ is an arbitrary random mapping, then $g_i(\mathcal{M}_i(x))$ is $(\varepsilon_i, \delta_i)$-differntially private.*

Although there exists more sophisticated approaches for composition, such as advanced composition, we do not need them for our purposes.

## 2.2 Streaming and Sketching Algorithms

In the streaming model, a frequency vector $x \in \mathbb{R}^n$ is induced by a sequence of updates. In the insertion-only streaming model, $x$ is defined through a stream of $m$ updates $u_1, \ldots, u_m$, where $u_t \in [n]$ for each $t \in [m]$ so that $x_i = |\{t \in [m] \,|\, u_t = i\}|$ for all $i \in [n]$. In other words, $x_i$ is the number of times that $i \in [n]$ appears in the stream. We remark that our techniques generalize to some degree to turnstile streams, where each update is an ordered pair $u_t = (\Delta_t, c_t)$, so that the $t$-th update changes the $c_t$-th coordinate by $\Delta_t$, i.e., $c_t \in [n]$ is a coordinate and $\Delta_t \in [-M, M]$ for some parameter $M > 0$. In this turnstile model, the vector $x$ is defined so that $x_i = \sum_{t: c_t = i} \Delta_t$ for all $i \in [n]$. Although our techniques can apply to the general turnstile model with a minor change on the conditions and assumptions, we shall work with the insertion-only streaming model throughout the remainder of the paper.

Given a frequency vector $x \in \mathbb{R}^n$ on a data stream, the AMS algorithm for $L_2$-estimation first generates a sign vector $\sigma \in \{-1, +1\}^n$ and sets $S_1 = (\langle \sigma, x \rangle)^2$. We remark that to maintain $\sigma$ in small space, it suffices for the coordinates of the sign vector $\sigma$ to be 4-wise independent and therefore it suffices to randomly generate and store a 4-wise independent hash function. The AMS algorithm then repeats this process $b = \frac{6}{\alpha^2}$ independent times to obtain dot products $S_1, \ldots, S_b$, sets $Z^2$ to be the arithmetic mean of $S_1, \ldots, S_b$, and reports $Z$.

We define the $L_2$ norm of a vector $x \in \mathbb{R}^n$ by $L_2(x) = \sqrt{x_1^2 + \ldots + x_n^2}$.

**Definition 2.7** ($\nu$-approximate $\eta$ $L_2$-heavy hitters problem). *Given an accuracy parameter $\nu \in (0, 1)$, a threshold parameter $\eta$, and a frequency vector $x \in \mathbb{R}^n$, compute a set $H \subseteq [n]$ and a set of approximations $\widehat{x_k}$ for all $k \in H$ such that:*

*(1) If $x_k \geq \eta L_2(x)$ for any $k \in [n]$, then $k \in H$, so that $H$ contains all $\eta$ $L_2$-heavy hitters of $x$.*

*(2) There exists a universal constant $C > 0$ so that if $x_k \leq \frac{C\eta}{2} L_2(x)$ for any $k \in [n]$, then $k \notin H$, so that $H$ does not contain any index that is not an $\frac{C\eta}{2}$ $L_2$-heavy hitter of $x$.*

*(3) If $k \in H$ for any $k \in [n]$, then compute $(1 \pm \nu)$-approximation to the frequency $x_k$, i.e., a value $\widehat{x_k}$ such that $(1 - \nu)x_k \leq \widehat{x_k} \leq (1 + \nu)x_k$.*

The well-known CountSketch algorithm provides an estimated frequency to each item and then releases the approximate frequencies of each item that surpasses a threshold proportional to the output of AMS:

**Theorem 2.8** (CountSketch for $\nu$-approximate $\eta$ $L_2$-heavy hitters). *[CCF04] There exists a one-pass streaming algorithm* CountSketch *that takes an accuracy parameter $\nu \in (0, 1)$ and a threshold parameter $\eta^2$ and outputs a list $H$ that contains all indices $k \in [n]$ of an underlying frequency vector $x$ with $x_k \geq \eta\, L_2(x)$ and no index $k \in [n]$ with $x_k \leq \eta(1 - \nu)\, L_2(x)$. For each $k \in H$,* CountSketch *also reports a estimated frequency* $\widehat{x_k}$ *such that* $(1-\nu)x_k \le \widehat{x_k} \le (1+\nu)x_k$. *The algorithm uses* $\mathcal{O}\left(\frac{1}{\eta^2\nu^2}\log^2 n\right)$ *bits of space and succeeds with probability* $1 - \frac{1}{\mathrm{poly}(m)}$.

**Algorithm 1** Heavy-hitter algorithm CountSketch

**Input:** Stream $\mathfrak{S}$, accuracy parameter $\nu \in (0,1)$, and threshold parameter $\eta \in (0,1)$

**Output:** $L_2$ Heavy-hitter algorithm

$r \leftarrow \mathcal{O}(\log n)$, $b \leftarrow \mathcal{O}\left(\frac{1}{\eta^2\nu^2}\right)$

Pick hash functions $h^{(1)}, \ldots, h^{(r)} : [n] \to [b]$ and $s^{(1)}, \ldots, s^{(r)} : [n] \to \{-1,+1\}$

$S_{i,j} \leftarrow 0$ for $(i,j) \in [r] \times [b]$

**for** each update $u_i \in [n]$, $i \in [m]$ **do**

  **for** each $j \in [r]$ **do**

    $b_{i,j} \leftarrow h^{(j)}(u_i)$ and $s_{i,j} \leftarrow s^{(j)}(u_i)$

    $S_{j,b_{i,j}} \leftarrow S_{j,b_{i,j}} + s_{i,j}$

**for** each $i \in [n]$ **do**

  $b_{i,j} \leftarrow h^{(j)}(u_i)$ for each $j \in [r]$

  **return** $\mathrm{median}_{j\in[r]} |S_{j,b_{i,j}}|$ as the estimated frequency for $f_i$

**Lemma 2.9.** *Let* $s \in \{-1,+1\}^n$ *and* $f, f' \in \mathbb{R}^n$ *with* $\max(\|f-f'\|_0, \|f-f'\|_1) \le 2$. *Then* $|\langle s,f\rangle - \langle s,f'\rangle| \le 2$.

*Proof.* Note that we have $\langle s,f\rangle - \langle s,f'\rangle = \langle s, f-f'\rangle$. Since $\max(\|f-f'\|_0, \|f-f'\|_1) \le 2$, then $|\langle s, f-f'\rangle| \le 2\|s\|_\infty = 2$. □

**Lemma 2.10** (Sensitivity of mean)**.** *Let* $x, y \in \mathbb{R}^n$ *so that* $\|x-y\|_\infty \le C$ *for some constant* $C \ge 0$. *Then*

$$\left|\frac{x_1+\ldots+x_n}{n} - \frac{y_1+\ldots+y_n}{n}\right| \le C.$$

*Proof.* By triangle inequality,

$$|(x_1+\ldots+x_n) - (y_1+\ldots+y_n)| \le \sum_{i=1}^n |x_i - y_i| \le Cn,$$

since $|x_i - y_i| \le C$ for all $i \in [n]$. Therefore,

$$\left|\frac{x_1+\ldots+x_n}{n} - \frac{y_1+\ldots+y_n}{n}\right| \le C.$$

□

**Lemma 2.11** (Sensitivity of median)**.** *Let* $x, y \in \mathbb{R}^n$ *so that* $\|x-y\|_\infty \le C$ *for some constant* $C \ge 0$. *Then*

$$|\mathrm{median}(x_1,\ldots,x_n) - \mathrm{median}(y_1,\ldots,y_n)| \le C.$$

*Proof.* Without loss of generality, suppose $x_1 \le \ldots \le x_n$ and first suppose that $n$ is even. Then there exist at least $\frac{n}{2}$ indices $i$ such that $x_i \le \mathrm{median}(x_1, \ldots, x_n)$ and thus at least $\frac{n}{2}$ indices $i$ with $y_i \le \mathrm{median}(x_1, \ldots, x_n) + C$. Hence, $\mathrm{median}(y_1, \ldots, y_n) \le \mathrm{median}(x_1, \ldots, x_n) + C$. Similarly, there exist at least $\frac{n}{2}$ indices $i$ such that $y_i \ge \mathrm{median}(x_1, \ldots, x_n) - C$ and thus, $\mathrm{median}(y_1, \ldots, y_n) \ge \mathrm{median}(x_1, \ldots, x_n) - C$. Therefore,

$$|\mathrm{median}(x_1, \ldots, x_n) - \mathrm{median}(y_1, \ldots, y_n)| \le C,$$

for even $n$.

For odd $n$, there exist at least $\frac{n+1}{2}$ indices $i$ with $x_i \le \mathrm{median}(x_1, \ldots, x_n)$ and at least $\frac{n+1}{2}$ indices $i$ with $x_i \ge \mathrm{median}(x_1, \ldots, x_n)$, and so there are at least $\frac{n+1}{2}$ indices $i$ with $y_i \le \mathrm{median}(x_1, \ldots, x_n) + C$ and we have $\mathrm{median}(y_1, \ldots, y_n) \le \mathrm{median}(x_1, \ldots, x_n) + C$. Similarly, there exist at least $\frac{n+1}{2}$ indices $i$ such that $y_i \ge \mathrm{median}(x_1, \ldots, x_n) - C$, so that

$$|\mathrm{median}(x_1, \ldots, x_n) - \mathrm{median}(y_1, \ldots, y_n)| \le C,$$

for odd $n$ as well. $\square$

**Lemma 2.12** (Sensitivity of CountSketch)**.** *Let $f, f' \in \mathbb{R}^n$ with $\max(\|f - f'\|_0, \|f - f'\|_1) \le 2$. Then we have $|\widehat{f_k} - \widehat{f'_k}| \le 2$ for each estimate of the frequency $f_k, f'_k$ of the $k$-th coordinate, $k \in [n]$ by* CountSketch *with fixed internal randomness.*

*Proof.* Let $b = \mathcal{O}\left(\frac{1}{\alpha^2}\right)$ with a sufficiently large constant. Then CountSketch first generates $r = \mathcal{O}(\log n)$ hash functions $h^{(1)}, \ldots, h^{(r)} : [n] \to [b]$. Fix $i \in [r]$ and $j \in [b]$ and consider the subset $S_{i,j} \subseteq [n]$ defined by $S_{i,j} := \{x \in [n] : h^{(i)}(x) = j\}$. Define $g$ and $g'$ as the restriction of the vectors $f$ and $f'$ to the coordinates of $S_{i,j}$ and $s^{(i,j)}$ as the sign vector $s^{(i)}$ restricted to the coordinates of $S_{i,j}$.

For each $i \in [r]$ and $k \in [n]$, let $\widehat{f_k^{(i)}}$ be the estimate of $f_k$ by $h^{(i)}$ and $\widehat{\overline{f}_k^{(i)}}$ be the estimate of $f'_k$ by $h^{(i)}$. Since $\max(\|g - g'\|_0, \|g - g'\|_1) \le 2$ and $s^{(i,j)}$ is a sign vector, then by Lemma 2.9,

$$|\langle s^{(i,j)}, g\rangle - \langle s^{(i,j)}, g'\rangle| \le 2$$

and thus

$$|\widehat{f_k^{(i)}} - \widehat{\overline{f}_k^{(i)}}| \le 2$$

for all $i \in [r]$. By Lemma 2.11,

$$\left|\mathrm{median}(\widehat{f_k^{(1)}}, \ldots, \widehat{f_k^{(r)}}) - \mathrm{median}(\widehat{\overline{f}_k^{(1)}}, \ldots, \widehat{\overline{f}_k^{(r)}})\right| \le 2.$$

Because CountSketch outputs the median of the estimated frequencies for each coordinates $k \in [n]$, then it follows that $|\widehat{f_k} - \widehat{f'_k}| \le 2$ for the estimates $\widehat{f_k}, \widehat{f'_k}$ output by CountSketch with fixed internal randomness on the vectors $f$ and $f'$, for all $k \in [n]$. $\square$

We can thus use a private variant PrivCountSketch of CountSketch by adding noise to each coordinate and then using a standard private threshold routine to ensure differential privacy. Specifically, PrivCountSketch first uses the CountSketch data structure to obtain an estimated frequency for each coordinate. It then adds Laplacian noise with scale parameter $\mathcal{O}\left(\frac{1}{\eta^2\nu^2}\right)$

to each estimated frequency, since by Lemma 2.12, the global sensitivity of CountSketch is at most 2. It then acquires a threshold $T$ from the $L_2$ norm estimation algorithm $\mathsf{AMS}$ and releases all coordinates (and estimated frequencies) whose estimated frequencies are at least $\frac{\nu\eta T}{2}+X$, where $X$ is Laplacian noise with scale parameter $\mathcal{O}\left(\frac{1}{\eta^2\nu^2}\right)$. Then PrivCountSketch gives the following guarantees:

**Lemma 2.13.** *There exists a one-pass streaming algorithm* PrivCountSketch *that takes an accuracy parameter $\nu\in(0,1)$ and a threshold parameter $\eta^2$ and outputs a list $H$ that contains all indices $k\in[n]$ of an underlying frequency vector $x$ with $x_k\ge\eta\,L_2(x)$ and no index $k\in[n]$ with $x_k\le\eta(1-\nu)\,L_2(x)$. For each $k\in H$,* PrivCountSketch *also reports a estimated frequency $\widehat{x_k}$ such that $(1-\nu)x_k-\mathcal{O}\left(\frac{\log m}{\eta\nu}\right)\le\widehat{x_k}\le(1+\nu)x_k+\mathcal{O}\left(\frac{\log m}{\eta\nu}\right)$. The algorithm uses $\mathcal{O}\left(\frac{1}{\eta^2\nu^2}\log^2 n\right)$ bits of space and succeeds with probability $1-\frac{1}{\mathrm{poly}(m)}$.*

### 2.3 Symmetric Norms

**Definition 2.14** (Symmetric norm)**.** *A function $L:\mathbb{R}^n\to\mathbb{R}$ is a* symmetric norm *if $L$ is a norm and for all $x\in\mathbb{R}^n$ and any vector $y\in\mathbb{R}^n$ that is a permutation of the coordinates of $x$, we have $L(x)=L(y)$. Moreover, we have $L(x)=L(|x|)$, where $|x|$ is the coordinate-wise absolute value of $x$.*

**Definition 2.15** (Modulus of concentration)**.** *Let $x\in\mathbb{R}^n$ be a random variable drawn from the uniform distribution on the $L_2$-unit sphere $S^{n-1}$ and let $b_L$ denote the maximum value of $L(x)$ over $S^{n-1}$. The median of a symmetric norm $L$ is the unique value $M_L$ such that $\mathbf{Pr}\left[L(x)\ge M_L\right]\ge\frac{1}{2}$ and $\mathbf{Pr}\left[L(x)\le M_L\right]\ge\frac{1}{2}$. Then the ratio $\mathrm{mc}(L):=\frac{b_L}{M_L}$ is the* modulus of concentration *of the norm $L$.*

Although the modulus of concentration quantifies the "average" behavior of the norm $L$ on $\mathbb{R}^n$, norms with challenging behavior can still be embedded in lower-dimensional subspaces. For instance, the $L_1$ norm satisfies $\mathrm{mc}(L)=\mathcal{O}(1)$, but when $x\in\mathbb{R}^n$ has fewer than $\sqrt{n}$ nonzero coordinates, the norm $\max(L_\infty(x),L_1(x)/\sqrt{n})$ on the unit ball becomes identically $L_\infty(x)$ [BBC+17], which requires $\Omega(\sqrt{n})$ space [AMS99] to estimate. Hence, we further quantify the behavior of a norm $L$ by examining its behavior on all lower dimensions.

**Definition 2.16** (Maximum modulus of concentration)**.** *For a norm $L:\mathbb{R}^n\to\mathbb{R}$ and every $k\le n$, define the norm $L^{(k)}:\mathbb{R}^k\to\mathbb{R}$ by $L^{(k)}((x_1,\ldots,x_k)):=L((x_1,\ldots,x_k,0,\ldots,0))$. Then the* maximum modulus of concentration *of the norm $L$ is $\mathrm{mmc}(L):=\max_{k\le n}\mathrm{mc}(L^{(k)})=\max_{k\le n}\frac{b_{L^{(k)}}}{M_{L^{(k)}}}$.*

**Definition 2.17** (Important Levels)**.** *For $x\in\mathbb{R}^n$ and $\xi>1$, we define the level $i$ as the set $B_i=\{k\in[n]\,:\,\xi^{i-1}\le|x_k|\le\xi^i\}$. We define $b_i:=|B_i|$ as the size of level $i$. For $\beta\in(0,1]$, we say level $i$ is $\beta$-*important *if*

$$b_i>\beta\sum_{j>i}b_j,\qquad b_i\xi^{2i}\ge\beta\sum_{j\le i}b_j\xi^{2j}.$$

Informally, level $i$ is $\beta$-important if (1) its size is at least a $\beta$-fraction of the total sizes of the higher levels and (2) its contribution is roughly a $\beta$-fraction of the total contribution of all the lower levels. We would like to show that to approximate a symmetric norm $L(x)$, it suffices to identify the $\beta$-important levels and their sizes for a fixed base $\xi>1$.

**Definition 2.18** (Level Vectors and Buckets)**.** *For $x \in \mathbb{R}^n$ and $\xi > 1$, the* level vector *for $x$ is*

$$V(x) := (\underbrace{\xi^1, \ldots, \xi^1}_{b_1 \textit{ times}}, \underbrace{\xi^2, \ldots, \xi^2}_{b_2 \textit{ times}}, \ldots, \underbrace{\xi^k, \ldots, \xi^k}_{b_k \textit{ times}}, 0, \ldots, 0) \in \mathbb{R}^n,$$

*where each $b_i$ is the size of level $i$. The $i$-th bucket of $V(x)$ is*

$$V_i(x) := (\underbrace{0, \ldots, 0,}_{b_1+\ldots+b_{i-1} \textit{ times}} \underbrace{\xi^i, \ldots, \xi^i}_{b_i \textit{ times}}, \ldots, \underbrace{0, \ldots, 0}_{b_{i+1}+\ldots+b_k \textit{ times}}, 0, \ldots, 0) \in \mathbb{R}^n.$$

*We similarly define the approximate level vectors $\widehat{V(x)}$ and $\widehat{V_i(x)}$ using approximations $\widehat{b_1}, \ldots, \widehat{b_k}$ for $b_1, \ldots, b_k$. We write $V(x) \setminus V_i(x)$ to denote the vector that replaces the $i$-th bucket in $V(x)$ with all zeros and we write $V(x) \setminus V_i(x) \cup \widehat{V_i(x)}$ to denote the vector that replaces the $i$-th bucket in $V(x)$ with $\widehat{b_i}$ instances of $\xi^i$.*

Rather than directly handle the important levels, we define the $\beta$-contributing levels and instead work toward estimating the contribution of the $\beta$-contributing levels.

**Definition 2.19** (Contributing Levels)**.** *Given $x \in \mathbb{R}^n$, a level $i$ defined by base $\xi > 1$ is $\beta$-*contributing *if $L(V_i(x)) \geq \beta L(V(x))$.*

[BBC+17] showed that even if all levels that are not $\beta$-contributing are removed, the contribution of the remaining levels forms a good approximation to $L(x)$.

**Lemma 2.20.** *[BBC+17] Given $x \in \mathbb{R}^n$ and levels defined by a base $\xi > 1$, let $V'(x)$ be the vector obtained by removing all levels that are not $\beta$-contributing from $V(x)$. Then $(1 - \mathcal{O}\left(\log_\xi n\right) \cdot \beta)L(V(x)) \leq L(V'(x)) \leq L(V(x))$.*

Hence for appropriate $\xi > 1$ and $\beta \in (0, 1]$, it suffices to identify the $\beta$-contributing levels, zero out the remaining levels, and determine the contribution of the resulting vector to approximate the symmetric norm $L(x)$.

**Lemma 2.21.** *[BBC+17] Given an accuracy parameter $\alpha \in (0, 1]$, let base $\xi = (1 + \mathcal{O}(\alpha))$, importance parameter $\beta = \mathcal{O}\left(\frac{\alpha^5}{\mathrm{mmc}(\ell)^2 \cdot \log^5 m}\right)$, and $\alpha' = \mathcal{O}\left(\frac{\alpha^2}{\log n}\right)$. Let $\widehat{b_i} \leq b_i$ for all $i$ and $\widehat{b_i} \geq (1 - \alpha')b_i$ for all $\beta$-important levels. Let $\widehat{V}$ be the level vector constructed using the estimates $\widehat{b_1}, \widehat{b_2}, \ldots$ and let $V'$ be the level vector constructed by removing all the buckets that are not $\beta$-contributing in $\widehat{V}$. Then $(1 - \alpha)L(V(x)) \leq L(V'(x)) \leq L(V(x))$.*

To identify the $\beta$-contributing levels, [BBC+17] first notes that the size of the level must be at least a significant fraction of the total size of the higher levels.

**Lemma 2.22.** *[BBC+17] Given $x \in \mathbb{R}^n$, let the level sets be defined by a base $\xi > 1$. If level $i$ is $\beta$-contributing, then there exists some fixed constant $\lambda > 0$ such that*

$$b_i \geq \frac{\lambda \beta^2}{\mathrm{mmc}(\ell)^2 \log^2 n} \cdot \sum_{j > i} b_j.$$

Moreover, [BBC+17] observes that the squared mass of a $\beta$-contributing level must be at least a significant fraction of the total squared mass of the lower levels.

**Lemma 2.23.** *[BBC+17] Given $x \in \mathbb{R}^n$, let the level sets be defined by a base $\xi > 1$. If level $i$ is $\beta$-contributing, then there exists some fixed constant $\lambda > 0$ such that*

$$b_i \xi^{2i} \geq \frac{\lambda \beta^2}{\mathrm{mmc}(\ell)^2 (\log_\xi n) \log^2 n} \cdot \sum_{j \leq i} b_j \xi^{2j}.$$

Observe that together, Lemma 2.22 and Lemma 2.23 imply that a $\beta$-contributing level $i$ must also be an important level as defined in Definition 2.17. Crucially, since Lemma 2.23 states that the squared mass (or the $F_2$ frequency moment) of the $\beta$-contributing levels must be a significant fraction of the total squared mass of the lower levels, then it suggests we might be able to identify the $\beta$-contributing levels through an $L_2$-heavy hitters algorithm after removing the higher levels. Indeed, [BBC+17] show that the problem of identifying the size (and thus the contribution) of the $\beta$-contributing levels can be reduced to the task of finding $\nu$-approximate $\eta$-heavy hitters for specific parameters of $\nu$ and $\eta$.

**Lemma 2.24.** *[BBC+17] Let $s = \mathcal{O}(\log n)$. If a level $i$ is $\beta$-important, then either $\xi^{2i} \geq \frac{\alpha^2 \beta \varepsilon^2}{\log^2 m} F_2(x)$ or there exists $j \in [s]$ such that $b_i \geq \frac{2^j \log^2 m}{\alpha^2 \varepsilon^2}$ and $\xi^{2i} \in \left[\frac{\alpha^2 \beta \varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^j}, \frac{\alpha^2 \beta \varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^{j-1}}\right]$.*

Lemma 2.24 implies that if level $i$ is $\beta$-important, then either (1) it will be identified by using PrivCountSketch with threshold $\frac{\alpha^2 \beta}{\log^2 m}$ on the stream or (2) its contribution can be well-approximated by using PrivCountSketch with threshold $\frac{\alpha^2 \beta \varepsilon^2}{\log^2 m}$ on a substream formed by sampling coordinates of the universe with probability $\frac{1}{2^j}$. We thus split our algorithm and analysis to handle these cases. In particular, we call a frequency level $i$ "high" if $\xi^{2i} \geq \frac{\alpha^2 \beta \varepsilon^2}{\log^2 m} F_2(x)$. We call a frequency level $i$ "medium" if $\xi^{2i} \geq \frac{\alpha^2 \beta' \varepsilon^2}{2^j} F_2(x) > T$ and $b_i \geq \mathcal{O}\left(\frac{2^j \log^2 m}{\alpha^2 \varepsilon^2}\right)$ for a certain $\beta' > 0$ and a threshold $T$. We call a frequency level $i$ "low" if $\xi^{2i} \geq \frac{\alpha^2 \beta' \varepsilon^2}{2^j} F_2(x)$ and $b_i \geq \mathcal{O}\left(\frac{2^j \log^2 m}{\alpha^2 \varepsilon^2}\right)$, but $T \geq \frac{\alpha^2 \beta' \varepsilon^2}{2^j} F_2(x)$.

# 3 Private Symmetric Norm Estimation Algorithm

In this section, we give our algorithm that releases a set of private statistics from which an arbitrary number of symmetric norms can be well-approximated. In particular, recall that Lemma 2.21 suggests that it suffices to approximate the sizes of the important levels and identity the non-important levels, so that their contributions can be set to zero. We partition the levels into three groups after defining thresholds $T_1$ and $T_2$, with $T_1 > T_2$. Recall that we define the "high frequency levels" as the levels whose coordinates exceed $T_1$ in frequency, the "medium frequency levels" as the levels whose coordinates are between $T_1$ and $T_2$ in frequency, and the "low frequency levels" as the levels whose coordinates are less than $T_2$ in frequency.

The intuition is that because the high frequency levels have such large magnitude, their frequencies can be well-approximated by running an $L_2$-heavy hitters algorithm on the stream $S$. On the other hand, the medium frequency level coordinates are not large enough to be detected by running an $L_2$-heavy hitters algorithm on the stream $S$, but the sizes of these level sets must be large if the level set is important and therefore, there exists a substream $S_j$ for which a large number of these coordinates are subsampled and their frequencies can be well-approximated by running an $L_2$-heavy hitters algorithm on the substream $S_j$. Finally, the low frequency level coordinates are small enough that we cannot add Laplacian noise to their frequencies without affecting the level sets they are mapped to. We instead show that $L_1$ sensitivity for the level set estimations is particularly small for the low frequency levels and thus, we report the size of the level sets of the low frequency levels rather than the identities of the heavy-hitters.

We emphasize that we only use the thresholds $T_1$ and $T_2$ for the purposes of describing our algorithm – in the actual implementation of the algorithm, the thresholds $T_1$ and $T_2$ will be implicitly defined by each of the substreams. For example, the items with threshold larger than $T_1$ will automatically be revealed through the stream $S$, while the items with thresholds between $T_1$ and $T_2$ will be revealed through the substreams $S_j$ with $2^j > \frac{\log n}{\beta'\alpha\varepsilon}$ for explicit parameters $\alpha$, $\beta'$, and $\varepsilon$. More specifically, note that Algorithm 2 sets $\beta' = \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$ or more specifically $\beta' = \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m}$. Then $\beta' \cdot F_2(x)$ corresponds to the threshold $T_1$, which is utilized in the proofs of Section 3.1. Similarly, Algorithm 3 leverages the quantity $\frac{\log n}{\beta'\alpha\varepsilon}$ to define the threshold $T_2$, which is then utilized in the proofs of Section 3.2.

### 3.1 Recovery of High Frequency Levels

In this section, we describe our algorithm for recovering the high frequency levels, whose coordinates have sufficiently large magnitude and thus their frequencies can be well-approximated by running an $L_2$-heavy hitters algorithm on the stream $S$. Moreover, with high probability, adding Laplacian noise will not affect the level sets because the frequencies are so large. Thus it simply suffices to return the noisy estimated frequencies of each of the elements in the high frequency levels. This algorithm is the simplest of our cases and we give the algorithm in full in Algorithm 2.

**Algorithm 2** Algorithm to privately estimate the high levels

**Input:** Privacy parameter $\varepsilon > 0$, accuracy parameter $\alpha \in (0, 1)$

**Output:** Private estimation of the frequencies of the coordinates of the high frequency levels

$\beta \leftarrow \mathcal{O}\left(\frac{\alpha^5}{\text{mmc}(L)^2 \log^5 m}\right)$, $\beta' \leftarrow \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$

Run PrivCountSketch on the stream $S$ with threshold $\alpha^2\beta'$ and failure probability $\frac{1}{\text{poly}(m)}$

**for** each heavy-hitter $k \in [n]$ reported by PrivCountSketch **do**

  Let $\widetilde{f_k}$ be the frequency estimated by PrivCountSketch

  $\widehat{x_k} \leftarrow \widetilde{x_k} + \mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$

  **return** $\widehat{x_k}$

We first show that coordinates in high frequency levels are identified and their frequencies are accurately estimated.

**Lemma 3.1.** *Suppose $x_k^2 \geq \frac{\alpha^2\beta\varepsilon^2}{\log^2 m} F_2(x)$ and $m = \frac{\Omega(\log^5 m)}{\alpha^5\beta^2\varepsilon^5}$. Then with high probability, Algorithm 2 outputs $\widehat{x_k}$ such that*

$$(1 - \alpha^2)x_k \leq \widehat{x_k} \leq x_k.$$

*Proof.* Consider Algorithm 2. Since $x_k^2 \geq \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m} F_2(x)$ and we call PrivCountSketch with threshold $\alpha^2\beta'$ with $\beta' := \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$, then with high probability, the output $\widetilde{x_k}$ satisfies

$$(1 - \mathcal{O}\left(\alpha^2\right))x_k \leq \widetilde{x_k} \leq x_k.$$

We then add Laplacian noise $\mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$ to $\widetilde{x_k}$ to form $\widehat{x_k}$. Since $x_k^2 \ge \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m}\,F_2(x) = \beta'\,F_2(x)$ and $F_2(x) \ge m$, then with high probability, the Laplacian noise is at most an $\alpha^2$ fraction of $\widehat{x_k}$ for $\frac{\mathcal{O}(\log m)}{\beta'\varepsilon} \le \alpha^2 m$ or equivalently, $m \ge \frac{\Omega(\log m)}{\alpha(\beta')^2\varepsilon} \ge \frac{\Omega\left(\log^5 m\right)}{\alpha^5\beta^2\varepsilon^5}$. Hence with high probability,

$$(1-\alpha^2)x_k \le \widehat{x_k} \le x_k.$$

□

Similarly, we show that if a coordinate does not have high frequency, it will not be output by Algorithm 2.

**Lemma 3.2.** *Suppose $x_k^2 < \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m}\,F_2(x)$ and $m = \frac{\Omega\left(\log^5 m\right)}{\alpha^5\beta^2\varepsilon^5}$. Then with high probability, Algorithm 2 outputs $\widehat{x_k}$ such that*

$$\widehat{x_k} < \frac{3\alpha^2\beta\varepsilon^2}{4\log^2 m}\,F_2(x).$$

*Proof.* Since $x_k^2 < \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m}\,F_2(x)$ and we call PrivCountSketch with threshold $\alpha^2\beta'$ with $\beta' := \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$, then the output $\widetilde{x_k}$ satisfies

$$|(\widetilde{x_k})^2 - (x_k)^2| \le 2\alpha^2\beta'\,F_2(x).$$

We then add Laplacian noise $\mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$ to $\widetilde{x_k}$ to form $\widehat{x_k}$. Since $F_2(x) \ge m$, then with high probability, the Laplacian noise is at most an $\alpha^2\beta'$ fraction of $F_2(x)$ for $\frac{\mathcal{O}(\log m)}{\beta'\varepsilon} \le \alpha^2 m$ or equivalently, $m \ge \frac{\Omega(\log m)}{\alpha(\beta')^2\varepsilon} \ge \frac{\Omega\left(\log^5 m\right)}{\alpha^5\beta^2\varepsilon^5}$. Hence with high probability,

$$|(\widetilde{x_k})^2 - (x_k)^2| \le \frac{\alpha^2\beta\varepsilon^2}{4\log^2 m}\,F_2(x).$$

Since $x_k^2 < \frac{\alpha^2\beta\varepsilon^2}{2\log^2 m}\,F_2(x)$, then it follows that

$$\widehat{x_k} < \frac{3\alpha^2\beta\varepsilon^2}{4\log^2 m}\,F_2(x).$$

□

We now show that Algorithm 2 preserves differential privacy.

**Lemma 3.3.** *Algorithm 2 is $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$-differentially private for $\delta = \frac{1}{\mathrm{poly}(m)}$.*

*Proof.* By Lemma 2.12, the sensitivity of PrivCountSketch is at most 2 and the failure probability is $\frac{1}{\mathrm{poly}(m)}$. Thus by adding Laplacian noise $\mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$ to $\widetilde{x_k}$, each estimated frequency is $\left(\frac{\beta'\varepsilon}{4}, \frac{\delta}{4\beta}\right)$-differentially private for $\delta = \frac{1}{\mathrm{poly}(m)}$. Since PrivCountSketch with threshold $\beta'$ can release at most $\frac{1}{\beta}$ estimated frequencies and post-processing does not cause loss in privacy, then by Theorem 2.6, Algorithm 2 is $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$. □

Finally, we analyze the space complexity of Algorithm 2.

**Lemma 3.4.** *Algorithm 2 uses space* $\mathrm{mmc}(L)^2 \cdot \mathrm{poly}\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log m\right)$.

*Proof.* The space complexity follows from running a single instance of PrivCountSketch with threshold $\alpha^2\beta'$ and failure probability $\frac{1}{\mathrm{poly}(m)}$, where $\beta' = \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$ and $\beta = \mathcal{O}\left(\frac{\alpha^5}{\mathrm{mmc}(L)^2 \log^5 m}\right)$. □

## 3.2 Recovery of Medium Frequency Levels

In this section, we describe our algorithm for recovering the medium frequency levels, whose coordinates do not have sufficiently large magnitude to be detected by running an $L_2$-heavy hitters algorithm on the stream $S$, but have sufficiently large size, so that there exists some $j \in [s]$ across the $s$ subsampling levels such that the coordinates can be detected by running an $L_2$-heavy hitters algorithm on the stream $S_j$. On the other hand, their magnitudes are sufficiently large so that with high probability, adding Laplacian noise will not affect the level sets. We give the algorithm in full in Algorithm 3.

**Algorithm 3** Algorithm to privately estimate the medium levels

**Input:** Privacy parameter $\varepsilon > 0$, accuracy parameter $\alpha \in (0,1)$
**Output:** Private estimations of the sizes of the medium frequency levels

$\beta \leftarrow \mathcal{O}\left(\frac{\alpha^5}{\mathrm{mmc}(L)^2 \log^5 m}\right)$, $\beta' \leftarrow \mathcal{O}\left(\frac{\alpha^3\beta\varepsilon^2}{\log^2 m}\right)$, $\xi \leftarrow (1 + \mathcal{O}(\varepsilon))$
$\gamma \leftarrow (1/2, 1)$ uniformly at random, $\ell \leftarrow \lceil \log_\xi(2m) \rceil$, $s \leftarrow \mathcal{O}(\log n)$
**for** $j \in [s]$ with $2^j > \frac{\log n}{\beta'\alpha\varepsilon}$ **do**
  Form stream $S_j$ by sampling elements of $[n]$ with probability $\frac{1}{2^j}$
  Run PrivCountSketch$_j$ on stream $S_j$ with threshold $\alpha^2\beta'\varepsilon^2$ and failure probability $\frac{1}{\mathrm{poly}(m)}$
  **for** each heavy-hitter $k \in [n]$ reported by PrivCountSketch$_j$ **do**
    Let $\widehat{x_k}$ be the frequency estimated by PrivCountSketch$_j$
    **if** $\widehat{x_k} > \frac{\log n}{\beta'\alpha\varepsilon}$ **then**
      $\widetilde{x_k} \leftarrow \widehat{x_k} + \mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$
  **for** $i \in [\ell]$ with $\frac{m^2}{2^{j+1}} > \gamma\xi^{2i} \geq 2^j > \mathcal{O}\left(\frac{\log n}{\beta'\alpha^2\varepsilon}\right)$ **do**
    Let $\widetilde{b_i}$ be the number of indices $k \in [n]$ such that $\gamma\xi^{2i} \leq \widetilde{x_k} < \gamma\xi^{2i+2}$
    $\widehat{b_i} \leftarrow \frac{2^j}{(1+\mathcal{O}(\alpha))}\widetilde{b_i}$
    **return** $\widehat{b_i}$

We first upper bound the second frequency moment (and hence the $L_2$ norm) of each substream. This is necessary because we want to detect the coordinates of the medium frequency levels as $L_2$-heavy hitters for each substream, but if the substream has overwhelmingly large $L_2$ norm, then we will not be able to find coordinates of the medium frequency levels. However, it may not be true that $F_2(S_j)$ is significantly smaller than $F_2(S)$ with high probability. For example, if there were a single large element, then the probability it is sampled at level $s$ is $\frac{1}{2^s}$, which is roughly $\frac{1}{n} > \frac{1}{\mathrm{poly}(m)}$. Instead, we note that PrivCountSketch benefits from the stronger *tail guarantee*, which states that not only does PrivCountSketch with threshold $\eta < 1$ detect the elements $k$ such that $(f_k)^2 \geq \eta F_2(S)$, but it also detects the elements $k$ such that $(f_k)^2 \geq \eta F_2(S_{\mathrm{tail}(1/\eta)})$, where $S_{\text{tail}(1/\eta)}$ is the frequency vector $f$ induced by $S$, with the largest $\frac{1}{\eta}$ entries instead set to zero [BCI+17, BGL+18].

**Lemma 3.5.** *With high probability, we have that $F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) \le \frac{200\log m}{2^j} F_2(x)$ for all $j \in [s]$.*

*Proof.* For each $j \in [s]$, we have that $\mathbb{E}\left[F_2(S_j)\right] = \frac{F_2(x)}{2^j}$. By Chernoff bounds with $\mathcal{O}\left(\log n\right)$-wise limited independence, we have that

$$\mathbf{Pr}\left[F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) > \frac{200\log m}{2^j} F_2(x)\right] \le \frac{1}{\text{poly}(m)}.$$

Since $s \le 2\log m$, then by a union bound over all $j \in [s]$, we have that $F_2(S_j) \le (200\log m)F_2(x)$ for all $j \in [s]$. $\square$

We now show that conditioned on the event that the $L_2$ norm of the subsampled streams are not too large, then we can well-approximate the frequency of any coordinate of the medium frequency levels, provided that they are sampled in the substream.

**Lemma 3.6.** *Suppose $i$ is a $\beta$-important level and $k \in [n]$ is in level $i$, so that $x_k \in [\xi^i, \xi^{i+1})$. If $F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) \le \frac{200\log m}{2^j} F_2(x)$ for all $j \in [s]$, then $k$ is sampled in stream $S_j$ with $2^j > \frac{\log n}{\beta'\alpha\varepsilon}$, then with high probability, Algorithm 3 outputs $\widehat{x_k}$ such that*

$$(1-\alpha^2)x_k \le \widehat{x_k} \le x_k.$$

*Proof.* Consider Algorithm 3. By Lemma 2.24, $x_2^2 \in \left[\frac{\alpha^2\beta\varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^j}, \frac{\alpha^2\beta\varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^{j-1}}\right]$. Conditioned on the event that $F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) \le \frac{200\log m}{2^j} F_2(x)$ for all $j \in [s]$, then $x_k^2 \ge \frac{\alpha^2\beta\varepsilon^2}{200\log m} F_2(S_j)$. We call PrivCountSketch with threshold $\alpha^2\beta'\varepsilon^2 = \mathcal{O}\left(\frac{\alpha^4\beta\varepsilon^3}{\log^2 m}\right)$. Thus with high probability, the output $\widetilde{x_k}$ satisfies

$$(1-\mathcal{O}\left(\alpha^2\right))x_k \le \widetilde{x_k} \le x_k.$$

We then add Laplacian noise $\mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$ to $\widetilde{x_k}$ to form $\widehat{x_k}$. Since $x_k^2 \ge \mathcal{O}\left(\frac{\log n}{\beta'\alpha^2\varepsilon}\right)$, then with high probability, the Laplacian noise is at most an $\alpha^2$ fraction of $\widehat{x_k}$. Hence with high probability,

$$(1-\alpha^2)x_k \le \widehat{x_k} \le x_k.$$

$\square$

Unfortunately, Lemma 3.6 only provides guarantees for the coordinates of the medium frequency levels that are sampled. Thus, we still need to use Lemma 3.6 to show that a good estimator to the sizes of the medium frequency levels can be obtained from the estimates of the coordinates of the medium frequency levels that are sampled. In particular, we show that rescaling the empirical sizes of the medium frequency levels forms a good estimator to the actual sizes of the medium frequency levels.

**Lemma 3.7.** *Consider a $\beta$-important level $i$ with $\xi^{2i} \in \left[\frac{\beta\alpha^2\varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^j}, \frac{\beta\alpha^2\varepsilon^2}{\log^2 m} \cdot \frac{F_2(x)}{2^{j-1}}\right]$ for some integer $j > 0$ and $\xi^i > \frac{\log n}{\beta'\alpha\varepsilon}$. If $F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) \le \frac{200\log m}{2^j} F_2(x)$ for all $j \in [s]$, then $k$ is sampled in stream $S_j$ with $2^j >$, then with high probability, Algorithm 3 outputs $\widehat{b_i}$ such that*

$$(1-\mathcal{O}\left(\alpha\right))b_i \le \widehat{b_i} \le b_i,$$

*where $b_i$ is the size of level $i$.*

*Proof.* Suppose $i$ is a $\beta$-important level. Then by Lemma 2.24 and a shifting of the index $j$, $b_i \ge \mathcal{O}\left(\frac{2^j \log^2 m}{\alpha^2 \varepsilon^2}\right)$. Thus in $S_j$, the expected number of items $E_j$ from level $i$ is at least $\frac{\log^2 m}{\alpha^2 \varepsilon^2}$ and the variance $V_j$ is at most $E_j$. Hence by Chernoff bounds with $\mathcal{O}(\log n)$-wise limited independence, we have that the number of items $N_j$ from level $i$ satisfies

$$(1 - \mathcal{O}(\alpha))b_i \le 2^j \cdot N_j \le (1 + \mathcal{O}(\alpha))b_i,$$

with high probability. [BBC+17] show that due to the uniformly random chosen $\gamma \in (1/2, 1)$, we further have

$$(1 - \mathcal{O}(\alpha))N_j \le (1 + \mathcal{O}(\alpha))\widehat{b_i} \le (1 + \mathcal{O}(\alpha))N_j,$$

with high probability. Since $s \le 2\log m$, then by a union bound over all $j \in [s]$, we have that with high probability, Algorithm 3 outputs $\widehat{b_i}$ such that

$$(1 - \mathcal{O}(\alpha))b_i \le \widehat{b_i} \le b_i.$$

□

We now show that Algorithm 3 preserves differential privacy.

**Lemma 3.8.** *Algorithm 3 is $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$-differentially private for $\delta = \frac{1}{\mathrm{poly}(m)}$.*

*Proof.* By Lemma 2.12, the sensitivity of PrivCountSketch is at most 2 and the failure probability is $\frac{1}{\mathrm{poly}(m)}$. Thus by adding Laplacian noise $\mathsf{Lap}\left(\frac{8}{\beta'\varepsilon}\right)$ to $\widetilde{x_k}$, each estimated frequency is $\left(\frac{\beta'\varepsilon}{4}, \frac{\delta}{4\beta}\right)$-differentially private for $\delta = \frac{1}{\mathrm{poly}(m)}$. Since PrivCountSketch with threshold $\beta'$ can release at most $\frac{1}{\beta}$ estimated frequencies, then by Theorem 2.6, Algorithm 3 is $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$. □

It remains to analyze the space complexity of Algorithm 3.

**Lemma 3.9.** *Algorithm 3 uses space* $\mathrm{mmc}(L)^2 \cdot \mathrm{poly}\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log m\right)$.

*Proof.* The space complexity follows from running $s$ instances of PrivCountSketch with threshold $\alpha^2\beta'$ and failure probability $\frac{1}{\mathrm{poly}(m)}$, where $\beta' = \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$ and $\beta = \mathcal{O}\left(\frac{\alpha^5}{\mathrm{mmc}(L)^2 \log^5 m}\right)$. Since $s = \mathcal{O}(\log n)$ and we assume $n \le m$ so that $\mathcal{O}(\log n) = \mathcal{O}(\log m)$, then the space complexity follows. □

### 3.3 Recovery of Low Frequency Levels

In this section, we describe our algorithm for recovering the low frequency levels, whose coordinates have magnitude small enough that we cannot add Laplacian noise to their frequencies without affecting the corresponding level set sizes. We instead report the sizes of the level sets for the low frequency levels rather than the identities and approximate frequencies of the heavy-hitters. Thus we must add Laplacian noise to the sizes of the level sets; we show that $L_1$ sensitivity for the level set estimations is particularly small for the low frequency levels and thus the Laplacian noise does not greatly affect the estimates of the level set sizes. We note that this approach does not work for the high frequency levels because the high frequency levels may have small level set sizes, so that adding Laplacian noise to the sizes can significantly affect the resulting estimates of the level set sizes. Similarly, it is more challenging to argue the low $L_1$ sensitivity for the level set estimations for the medium frequency levels. Hence, both the algorithm and analysis are especially well-catered to the low frequency levels. We give the algorithm in full in Algorithm 4.

We first show that the estimates of the level set sizes for the low frequency levels are accurate.

**Algorithm 4** Algorithm to privately estimate the low levels

**Input:** Privacy parameter $\varepsilon > 0$, accuracy parameter $\alpha \in (0,1)$

**Output:** Private estimations of the sizes of the low frequency levels

$\beta \leftarrow \mathcal{O}\left(\frac{\alpha^5}{\text{mmc}(L)^2 \log^5 m}\right)$, $\beta' \leftarrow \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon}{\log n}\right)$, $\xi \leftarrow (1+\mathcal{O}(\varepsilon))$

$\gamma \leftarrow (1/2,1)$ uniformly at random, $\ell \leftarrow \lceil \log_{\xi}(2m) \rceil$, $s \leftarrow \mathcal{O}(\log n)$

**for** $j \in [s]$ with $2^j \le \frac{\log n}{\beta'\alpha\varepsilon}$ **do**

  Form stream $S_j$ by sampling elements of $[n]$ with probability $\frac{1}{2^j}$

  Run $\textsc{PrivCountSketch}_j$ on stream $S_j$ with threshold $\beta'' := \mathcal{O}\left(\frac{\beta'\alpha^2\varepsilon^3}{\log^2 n}\right)$

  **for** each heavy-hitter $k \in [n]$ reported by $\textsc{PrivCountSketch}_j$ **do**

    Let $\widehat{x_k}$ be the frequency estimated by $\textsc{PrivCountSketch}_j$

  **for** $i \in [\ell]$ with $\mathcal{O}\left(\frac{\log n}{\beta'\alpha^2\varepsilon}\right) \ge 2^{j+1} > \gamma\xi^{2i} \ge 2^j$ **do**

    Let $\widetilde{b_i}$ be the number of indices $k \in [n]$ such that $\gamma\xi^{2i} \le \widehat{x_k} < \gamma\xi^{2i+2}$

    $\widehat{b_i} \leftarrow \frac{2^j}{(1+\mathcal{O}(\alpha))}\left(\widetilde{b_i} + \mathsf{Lap}\left(\frac{8}{\varepsilon}\right)\right)$

    **return** $\widehat{b_i}$

**Lemma 3.10.** *Consider a $\beta$-important level $i$ with $\xi^{2i} \in \left[\frac{\beta\alpha^2\varepsilon^2}{\log^2 m}\cdot\frac{F_2(x)}{2^j}, \frac{\beta\alpha^2\varepsilon^2}{\log^2 m}\cdot\frac{F_2(x)}{2^{j-1}}\right]$ for some integer $j > 0$ and $\xi^i \le \frac{\log n}{\beta'\alpha\varepsilon}$. If $F_2((S_j)_{1/(\alpha^2\beta'\varepsilon^2)}) \le \frac{200\log m}{2^j} F_2(x)$ for all $j \in [s]$, then $k$ is sampled in stream $S_j$ with $2^j >$, then with high probability, Algorithm 4 outputs $\widehat{b_i}$ such that*

$$(1-\mathcal{O}(\alpha))b_i \le \widehat{b_i} \le b_i,$$

*where $b_i$ is the size of level set $i$.*

*Proof.* Suppose $i$ is a $\beta$-important level. Hence by a shifting of the index $j$ in Lemma 2.24, we have that $b_i \ge \mathcal{O}\left(\frac{2^j \log^2 m}{\alpha^2\varepsilon^2}\right)$. Therefore, the expected number of items $E_j$ from level $i$ sampled in the substream $S_j$ is at least $\frac{\log^2 m}{\alpha^2\varepsilon^2}$ and the variance $V_j$ is at most $E_j$. Thus by Chernoff bounds with $\mathcal{O}(\log n)$-wise limited independence, the number of items $N_j$ from level $i$ satisfies

$$(1-\mathcal{O}(\alpha))b_i \le 2^j \cdot N_j \le (1+\mathcal{O}(\alpha))b_i,$$

with high probability. [BBC+17] show that due to the uniformly random chosen $\gamma \in (1/2,1)$, we further have

$$(1-\mathcal{O}(\alpha))N_j \le (1+\mathcal{O}(\alpha))\widehat{b_i} \le (1+\mathcal{O}(\alpha))N_j,$$

with high probability. Since $s \le 2\log m$ and $\mathsf{Lap}\left(\frac{8}{\varepsilon}\right)$ is at most an $\varepsilon$-fraction of $b_i \ge \mathcal{O}\left(\frac{2^j\log^2 m}{\alpha^2\varepsilon^2}\right)$ with high probability, then by a union bound over all $j \in [s]$, we have that with high probability, Algorithm 3 outputs $\widehat{b_i}$ such that

$$(1-\mathcal{O}(\alpha))b_i \le \widehat{b_i} \le b_i.$$

□

We then show that Algorithm 4 is differentially private.

**Lemma 3.11.** *Algorithm 4 is* $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$*-differentially private for* $\delta = \frac{1}{\mathrm{poly}(m)}$.

*Proof.* Note that since each instance of $\textsc{PrivCountSketch}_j$ uses threshold $\beta'' := \mathcal{O}\left(\frac{\beta'\alpha^2\varepsilon^3}{\log^2 n}\right)$ on a stream $S_j$ with $F_2(S_j) \le \frac{200\log m}{2^j} F_2(x)$, then for any $k \in [n]$ with $x_k \le \mathcal{O}\left(\frac{\log n}{\beta'\alpha^2\varepsilon}\right)$, we have that $\textsc{PrivCountSketch}_j$ outputs $x_k$ exactly. Hence, at most two estimates of the sizes of the level sets $\widehat{b_i}$ can change, and then can change by at most one. Thus the sensitivity is at most 2, so it suffices to add Laplcian noise $\mathsf{Lap}\left(\frac{8}{\varepsilon}\right)$ to each estimate $\widehat{b_i}$ to obtain $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$-differentially private for $\delta = \frac{1}{\mathrm{poly}(m)}$. $\square$

Finally, we argue the space complexity of Algorithm 4.

**Lemma 3.12.** *Algorithm 4 uses space* $\mathrm{mmc}(L)^2 \cdot \mathrm{poly}\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log m\right)$.

*Proof.* Similar to Algorithm 3, the space complexity follows as a result of running $s$ instances of $\textsc{PrivCountSketch}$ with threshold $\alpha^2\beta'$ and failure probability $\frac{1}{\mathrm{poly}(m)}$, where $\beta' = \mathcal{O}\left(\frac{\alpha^2\beta\varepsilon^2}{\log^2 m}\right)$ and $\beta = \mathcal{O}\left(\frac{\alpha^5}{\mathrm{mmc}(L)^2\log^5 m}\right)$. Since $s = \mathcal{O}(\log n)$ and we assume $n \le m$ so that $\mathcal{O}(\log n) = \mathcal{O}(\log m)$, then the space complexity follows. $\square$

## 3.4 Putting Things Together

We would like to combine the subroutines from the previous sections to output a private dataset for symmetric norm estimation. Thus it remains to describe how to privately partition the coordinates into the high, medium, and low frequency levels. To that end, we remark that by Lemma 2.12, the sensitivity of $\textsc{PrivCountSketch}$ in Algorithm 1 is at most 2. Moreover, although $\textsc{PrivCountSketch}$ actually provides an estimated frequency for each coordinate, for our purposes, we only need estimated frequencies for the $L_2$-heavy hitters and there are at most $K := \mathcal{O}\left(\frac{1}{\eta^2}\right)$ possible $L_2$-heavy hitters with whichever threshold $\eta$ that we choose, e.g., $\eta = \alpha^2\beta'$ in Algorithm 2. Thus it suffices to observe that we can privately partition the coordinates into the high, medium, and low frequency levels by first privately outputting the top $K$ estimated frequencies and then partitioning the coordinates according to their noisy estimated frequencies, which can be viewed as post-processing. In particular, [QSZ21] observes that it suffices to add Laplacian noise with scale $\frac{8}{\eta\varepsilon}$ to each of the frequencies and then outputting the top $K$ noisy estimated frequencies to achieve $\frac{\varepsilon}{4}$-differential privacy.

We now finally put together the results from the previous sections to show the following result:

**Theorem 3.13.** *There exists a* $(\varepsilon, \delta)$*-differentially private algorithm that outputs a set* $C$*, for* $\delta = \frac{1}{\mathrm{poly}(m)}$*. From* $C$*, the* $(1+\alpha)$*-approximation to any norm with* maximum modulus of concentration *at most* $M$ *can be computed, with probability at least* $1-\delta$*. The algorithm uses* $M^2 \cdot \mathrm{poly}\left(\frac{1}{\alpha}, \frac{1}{\varepsilon}, \log m\right)$ *bits of space.*

*Proof.* Note that from Lemma 3.1 and Lemma 3.2, the frequencies of the coordinates in the high frequency levels are well-approximated with high probability. Similarly, from Lemma 3.7 and Lemma 3.10, the sizes of the level sets of the medium and low frequency levels are well-approximated with high probability. Moreover, all the level sets are partitioned into the high, medium, or low frequency levels. We would like to say that by Lemma 2.21, these statistics are sufficient to recover a $(1+\alpha)$-approximation to any norm with *maximum modulus of concentration* at most $M$ and so we achieve a $(1+\alpha)$-approximation to any norm with maximum modulus of concentration at most $M$ that with high probability. Indeed, in an idealized process where $\xi^i \le \widehat{f_k} \le \xi^{i+1}$ if and only if $k$ is sampled by the substream $j$ assigned to level $i$ and $\xi^i \le f_k < \xi^{i+1}$, Lemma 2.21 would show that we achieve a $(1+\alpha)$-approximation to any norm with maximum modulus of concentration at most $M$ that with high probability. However, this may not always be the case because the frequency $f_k$ may lie near the boundary of the interval $[\xi^i, \xi^{i+1})$ and the estimate $\widehat{f_k}$ may lie outside of the interval, in which case $\widehat{f_k}$ is used toward the estimation of some other level set. Thus, our algorithm randomizes the boundaries of the level sets by instead defining the level sets as $[\gamma\xi^i, \gamma\xi^{i+1})$ for some $\gamma \in (1/2, 1)$ chosen uniformly at random. Since we call PrivCountSketch with threshold at most $\alpha^2\beta'$, then the probability that item $k \in [n]$ is misclassified over the choice of $\gamma$ is at most $\mathcal{O}\left(\alpha^2\beta'\right)$. Furthermore, if $k$ in level set $i$ is misclassified, it can only be classified into level set $i-1$ or $i+1$, causing at most an incorrect multiplicative factor of two. Then in expectation across all $k \in [n]$, the error due to the misclassification is at most an $\mathcal{O}\left(\alpha^2\beta'\right)$ fraction of the symmetric norm. Hence by Markov's inequality, the error due to the misclassification is at most an additive $\frac{\alpha}{2}$ fraction of the symmetric norm with probability at least 0.99. To obtain high probability of success, it then suffices to take the median across $\mathcal{O}(\log m)$ independent instances, finally showing correctness of our algorithm.

The private partitioning of the coordinates into the high, medium, and low frequency levels is $\frac{\varepsilon}{4}$-differentially private. Each of the three sets of statistics released by the high, medium, and low frequency levels are $\left(\frac{\varepsilon}{4}, \frac{\delta}{4}\right)$-differentially private, by Lemma 3.3, Lemma 3.8, and Lemma 3.11. Then $(\varepsilon, \delta)$-differential privacy follows from the composition of differential privacy, i.e., Theorem 2.6.

Finally, the space complexity follows from Lemma 3.4, Lemma 3.9, and Lemma 3.12. □

We remark that our algorithm is presented as having unlimited access to random bits but is analyzed using $\mathcal{O}(\log m)$-wise independence, so it can be properly derandomized to provide the space guarantees without needing to store a large number of random bits. Alternatively, our algorithm can also be derandomized using Nisan's pseudorandom generator, which induces an extra multiplicative factor of $\mathcal{O}(\log m)$ in the space overhead [Nis92].

Finally, we remark that the failure probability can be raised from $\delta = \frac{1}{\mathrm{poly}(m)}$ to arbitrarily $\delta > 0$ using additional space overhead polylog $\frac{1}{\delta}$, since the space dependency in each subroutine on the failure probability $\delta$ is polylog $\frac{1}{\delta}$. Thus, we obtain Theorem 1.2.

# References


[ABRS03] Aditya Akella, Ashwin Bharambe, Mike Reiter, and Srinivasan Seshan. Detecting ddos attacks on isp networks. In *Proceedings of the Twenty-Second ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams*, pages 1–3, 2003. 1

[ÁCC12] Gergely Ács, Claude Castelluccia, and Rui Chen. Differentially private histogram publishing through lossy compression. In *12th IEEE International Conference on Data Mining, ICDM*, pages 1–10, 2012. 6

[AFS12] Andreas Argyriou, Rina Foygel, and Nathan Srebro. Sparse prediction with the $k$-support norm. In *Advances in Neural Information Processing Systems 25: Annual Conference on Neural Information Processing Systems*, pages 1466–1474, 2012. 3

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. 1, 6, 11

[And17] Alexandr Andoni. High frequency moments via max-stability. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 6364–6368, 2017. 1, 6

[ANN+17] Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Approximate near neighbors for general symmetric norms. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theoryof Computing, STOC*, pages 902–913, 2017. 6

[AS19] Jayadev Acharya and Ziteng Sun. Communication complexity in locally private distribution estimation and heavy hitters. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 51–60, 2019. 6

[BBC+17] Jaroslaw Blasiok, Vladimir Braverman, Stephen R. Chestnut, Robert Krauthgamer, and Lin F. Yang. Streaming symmetric norms via measure concentration. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 716–729, 2017. 3, 4, 5, 6, 11, 12, 13, 18, 19, 27

[BBDS12] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 410–419, 2012. 2, 3, 6

[BCI+17] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. Bptree: An $l_2$ heavy hitters algorithm using constant memory. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 361–376, 2017. 17

[BGK+21] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Judy Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via JL projections. *CoRR*, abs/2102.03013, 2021. 2, 3, 6

[BGL+18] Vladimir Braverman, Elena Grigorescu, Harry Lang, David P. Woodruff, and Samson Zhou. Nearly optimal distinct elements and heavy hitters on sliding windows. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 7:1–7:22, 2018. 17

[Bha13] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013. 3

[BNS19] Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *ACM Trans. Algorithms*, 15(4):51:1–51:40, 2019. 6

[BNST20] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42, 2020. 6

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 1

[BS15] Raef Bassily and Adam D. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 127–135, 2015. 6

[BVWY18] Vladimir Braverman, Emanuele Viola, David P. Woodruff, and Lin F. Yang. Revisiting frequency moment estimation in random order streams. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 25:1–25:14, 2018. 1

[BWZ21] Vladimir Braverman, Viska Wei, and Samson Zhou. Symmetric norm estimation and regression on sliding windows. In *Computing and Combinatorics - 27th International Conference, COCOON, Proceedings*, pages 528–539, 2021. 6

[CCF04] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004. 8

[CDKY20] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. Differentially-private multi-party sketching for large-scale statistics. *Proc. Priv. Enhancing Technol.*, 2020(3):153–174, 2020. 3, 6

[CLSX12] T.-H. Hubert Chan, Mingfei Li, Elaine Shi, and Wenchang Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *Privacy Enhancing Technologies - 12th International Symposium, PETS Proceedings*, pages 140–159, 2012. 6

[CMR05] Graham Cormode, S. Muthukrishnan, and Irina Rozenbaum. Summarizing and mining inverse distributions on data streams via dynamic inverse sampling. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 25–36, 2005. 1

[DKY17] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 3571–3580, 2017. 6

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings*, pages 265–284, 2006. 2

[DNP+10] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science - ICS. Proceedings*, pages 66–80, 2010. 6

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. 7, 8

[EVF03] Cristian Estan, George Varghese, and Mike Fisk. Bitmap algorithms for counting active flows on high speed links. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 153–166, 2003. 1

[FST88] Sheldon J. Finkelstein, Mario Schkolnick, and Paolo Tiberio. Physical database design for relational databases. *ACM Trans. Database Syst.*, 13(1):91–128, 1988. 1

[Gan15] Sumit Ganguly. Taylor polynomial estimator for estimating frequency moments. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP Proceedings, Part I*, pages 542–553, 2015. 3

[Gin12] Corrado Gini. Variabilità e mutabilità. *Reprinted in Memorie di metodologica statistica*, 1912. 1

[GW18] Sumit Ganguly and David P. Woodruff. High probability frequency moment sketches. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 58:1–58:15, 2018. 1

[IM08] Piotr Indyk and Andrew McGregor. Declaring independence via the sketching of sketches. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 737–745. SIAM, 2008. 1

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. 1, 2, 6

[IW05] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 202–208, 2005. 1, 4, 5, 6, 27

[KNPW11] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC*, pages 745–754, 2011. 1

[KNW10] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1161–1178, 2010. 3

[KSZC03] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. Sketch-based change detection: Methods, evaluation, and applications. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247, 2003. 1

[KV07] Bo'az Klartag and Roman Vershynin. Small ball probability and dvoretzky's theorem. *Israel Journal of Mathematics*, 157(1):193–207, 2007. 3

[Li08] Ping Li. Estimators and tail bounds for dimension reduction in $l_\alpha$ $(0 < \alpha \leq 2)$ using stable random projections. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 10–19, 2008. 1, 6

[LMV$^+$16] Zaoxing Liu, Antonis Manousis, Gregory Vorsanger, Vyas Sekar, and Vladimir Braverman. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proceedings of the ACM SIGCOMM 2016 Conference*, pages 101–114, 2016. 6

[Lor05] Max O Lorenz. Methods of measuring the concentration of wealth. *Publications of the American statistical association*, 9(70):209–219, 1905. 1

[LW13] Yi Li and David P. Woodruff. A tight lower bound for high frequency moment estimation with small error. In *Approximation, Randomization, and Combinatorial*

*Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM. Proceedings*, pages 623–638, 2013. 3

[LW20] Yi Li and David P. Woodruff. Input-sparsity low rank approximation in schatten norm. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 6001–6009, 2020. 4

[MMNW11] Darakhshan J. Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. Pan-private algorithms via statistics on sketches. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS*, pages 37–48, 2011. 2

[MPS14] Andrew M. McDonald, Massimiliano Pontil, and Dimitris Stamos. Spectral k-support norm regularization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3644–3652, 2014. 3

[MS09] Vitali D Milman and Gideon Schechtman. *Asymptotic theory of finite dimensional normed spaces: Isoperimetric inequalities in riemannian manifolds*, volume 1200. Springer, 2009. 3

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992. 21

[PSF+01] Christopher R Palmer, Georgos Siganos, Michalis Faloutsos, Christos Faloutsos, and Phillip B Gibbons. The connectivity and fault-tolerance of the internet topology, 2001. 1

[QSZ21] Gang Qiao, Weijie J. Su, and Li Zhang. Oneshot differentially private top-k selection. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, pages 8672–8681, 2021. 20

[SAC+79] Patricia G. Selinger, Morton M. Astrahan, Donald D. Chamberlin, Raymond A. Lorie, and Thomas G. Price. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*, pages 23–34, 1979. 1

[She19] Or Sheffet. Differentially private ordinary least squares. *J. Priv. Confidentiality*, 9(1), 2019. 3, 6

[SST20] Adam D. Smith, Shuang Song, and Abhradeep Thakurta. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 2, 3, 6

[SWY+19] Zhao Song, Ruosong Wang, Lin F. Yang, Hongyang Zhang, and Peilin Zhong. Efficient symmetric norm regression via linear sketching. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 828–838, 2019. 6

[TZ04] Mikkel Thorup and Yin Zhang. Tabulation based 4-universal hashing with applications to second moment estimation. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 615–624, 2004. 1

[WDST14] Bin Wu, Chao Ding, Defeng Sun, and Kim-Chuan Toh. On the moreau-yosida regularization of the vector k-norm related functions. *SIAM J. Optim.*, 24(2):766–794, 2014. 3

[WLL+20] Tianhao Wang, Milan Lopuhaä-Zwakenberg, Zitao Li, Boris Skoric, and Ninghui Li. Locally differentially private frequency estimation with consistency. In *27th Annual Network and Distributed System Security Symposium, NDSS*, 2020. 6

[WPS21] Lun Wang, Iosif Pinelis, and Dawn Song. Differentially private fractional frequency moments estimation with polylogarithmic space. *arXiv preprint arXiv:2105.12363*, 2021. 2, 3, 6

[WZ20] David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. *CoRR*, abs/2011.07471, 2020. 1

[WZ21] David P. Woodruff and Samson Zhou. Separations for estimating large frequency moments on data streams. In *48th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 198, pages 112:1–112:21, 2021. 1, 3, 26

[XZX+13] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *VLDB J.*, 22(6):797–822, 2013. 6

# A Additional Intuition

## A.1 Maximum Modulus of Concentration

Let $x \in \mathbb{R}^n$ be a random variable drawn from the uniform distribution on the $L_2$-unit sphere $S^{n-1}$ and let $b_L$ denote the maximum value of $L(x)$ over $S^{n-1}$. The median of a symmetric norm $L$ is the unique value $M_L$ such that $\mathbf{Pr}\left[L(x) \ge M_L\right] \ge \frac{1}{2}$ and $\mathbf{Pr}\left[L(x) \le M_L\right] \ge \frac{1}{2}$. Then recall that the ratio $\mathrm{mc}(L) := \frac{b_L}{M_L}$ is the *modulus of concentration* of the norm $L$.

For a vector $x \in \mathbb{R}^n$, the $L_1$ norm is defined as $L_1(x) = \sum_{i=1}^n |x_i|$. The maximum value of $L_1(x)$ for a vector $x$ from the $L_2$-unit sphere $S^{n-1}$ is $L_1(x) = \sqrt{n}$ for the flat vector $\left(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}\right)$. Thus we have $b_{L_1} = \sqrt{n}$. It turns the median value of $L_1(x)$ for a vector $x$ from the $L_2$-unit sphere $S^{n-1}$ is also $M_{L_1} = \tilde{\Omega}(\sqrt{n})$ and so $\mathrm{mc}(L_1) \le \mathrm{polylog}(n)$.

On the other hand, for the $L_3$ norm defined as $L_3(x) = \left(\sum_{i=1}^n |x_i|^3\right)^3$, the maximum value of $L_3(x)$ for a vector $x$ from the $L_2$-unit sphere $S^{n-1}$ is $L_3(e_i) = 1$ for a unit vector $e_i$., while the median value of $L_3(x)$ for a vector $x$ from the $L_2$-unit sphere $S^{n-1}$ is roughly $M_{L_3} = \mathcal{O}\left(n^{-1/6}\right)$ and so $\mathrm{mc}(L_3) = \Omega(n^{-1/6})$. Thus we should expect the complexity of the estimating the $L_3$ norm to be significantly more challenging than estimating the $L_1$ norm, and indeed this reflects known upper and lower bounds in the streaming model, e.g., see the discussion in [WZ21].

## A.2 Intuition on [IW05] and [BBC$^+$17]

The main intuition of the celebrated Indyk-Woodruff norm estimation algorithm [IW05, BBC$^+$17] is to decompose a norm $\ell(x)$ on input vector $x$ into the contribution by each of its coordinates, which can then be partitioned into level sets, based on how much they contribute to the norm $\ell(x)$. We can then approximate each of the contributions of the level sets by subsampling the universe and estimating the sizes of each universe through the heavy-hitters of each subsample.

Recall the definition of the important levels in Definition 2.17. Intuitively, an important level if its size is "significant" compared to all the higher levels and its contribution is "significant" compared to all the lower levels, so that the important level contributes a "significant" amount to the overall norm, which is formalized by Lemma 2.24 to be the $\beta$-contributing levels, i.e., see Definition 2.19.

The norm estimation algorithms of [IW05, BBC$^+$17] then reconstructs an estimate of the level vector, i.e., see Definition 2.18, by removing all the levels that are not $\beta$-contributing and using a $(1+\varepsilon')$-approximation to the sizes of all $\beta$-important levels, for some fixed value of $\varepsilon' > 0$, which is a function of the accuracy parameter $\varepsilon$ and the maximum modulus of concentration $\text{mmc}(\ell)$ of the norm $\ell$. Lemma 2.24 then shows that this approach suffices to obtain a $(1+\varepsilon)$-approximation to $\ell(x)$.

In particular, we can use $\nu$-approximate $\eta$-heavy hitters algorithms to roughly estimate the size $b_i$ of all $\beta$-important levels, because each $\beta$-important level must have either (1) large size, i.e., a large number of coordinates achieving a certain range of frequencies, or (2) large contribution, i.e., a small number of coordinates with significantly large value or (3) both. If the $\beta$-important level has large contribution but small size, then the significantly large coordinates will immediately be recognized as heavy-hitters. Otherwise, if the $\beta$-important level has large size, then a large number of these coordinates in the level set will be subsampled. Then these sampled coordinates of the level set will become heavy-hitters at some level $i$ in which $\Theta\left(\frac{1}{\varepsilon^2}\right)$ of these coordinates are subsampled, since the expected $\ell$ norm of the sampled coordinates will be also be significantly smaller. The size $b_i$ of these level sets with a large number of coordinates can be then roughly estimated by rescaling the number of sampled coordinates by the inverse of the sampling probability, though additional care is required to formalize this argument, e.g., randomized boundaries for each of the level sets.