

1-Pass Relative-Error L_p -Sampling with Applications

Morteza Monemizadeh^{*†}

David P. Woodruff[‡]

Abstract

For any $p \in [0, 2]$, we give a 1-pass $\text{poly}(\varepsilon^{-1} \log n)$ -space algorithm which, given a data stream of length m with insertions and deletions of an n -dimensional vector a , with updates in the range $\{-M, -M+1, \dots, M-1, M\}$, outputs a sample of $[n] = \{1, 2, \dots, n\}$ for which for all i the probability that i is returned is $(1 \pm \varepsilon) \frac{|a_i|^p}{F_p(a)} \pm n^{-C}$, where a_i denotes the (possibly negative) value of coordinate i , $F_p(a) = \sum_{i=1}^n |a_i|^p = \|a\|_p^p$ denotes the p -th frequency moment (i.e., the p -th power of the L_p norm), and $C > 0$ is an arbitrarily large constant. Here we assume that n, m , and M are polynomially related.

Our generic sampling framework improves and unifies algorithms for **several communication and streaming problems**, including **cascaded norms**, **heavy hitters**, and **moment estimation**. It also gives the first relative-error forward sampling algorithm in a data stream with deletions, answering an open question of Cormode *et al.*

1 Introduction

The streaming model has emerged as an important paradigm for understanding the efficiency of algorithms. In this model the algorithm is typically given only a few passes (usually one) over a massive dataset, and must maintain a small randomized sketch of what it has seen so far. This allows it to later approximately answer global questions about the dataset. This model dates back to the work of Flajolet and Martin [16], as well as Munro and Paterson [31], and has become increasingly popular in the theory community due in a large part to the work of Alon, Matias, and Szegedy [1]. For a survey, see the book by Muthukrishnan [32], or course notes by Indyk [23].

Most streaming problems can be modeled by the evolution of an underlying n -dimensional vector a whose coordinates are initialized to 0. In the *turnstile model* of streaming we see a stream S of $m = \text{poly}(n, M)$ updates of the form (i, x) , where $i \in [n]$ and $x \in \{-M, -M+1, \dots, M-1, M\}$, indicating that the i -th coordinate a_i of a should be

incremented by x . Given the algorithm's sketch of a , the goal is then to output an approximation to $f(a)$, where f is some function. Often f is taken to be the p -norm $\|a\|_p = (\sum_{i=1}^n |a_i|^p)^{1/p}$ of a , for some $p \geq 0$ ($\|a\|_0$ is the number of non-zero coordinates of a), or the k -th frequency moment $F_k(a) = \|a\|_k^k$. Algorithms and lower bounds on the sketch size for these particular f have been extensively studied; see [32] and the references therein. Such statistics provide a measure of similarity, which is useful for massive datasets. Moments are also quite useful for estimating quantities in other contexts, such as cascaded norms [26], earthmover distance [2], entropy [20], geometry [15], independence of distributions [8, 24], and linear algebra [10, 33], to name a few.

In this paper we propose a new twist to the above setting. Instead of trying to approximate the value $f(a)$, we think of there being a non-negative function $g : \mathbb{Z} \rightarrow \mathbb{R}^{\geq 0}$ defining a distribution

$$\pi(a) = \left(\frac{g(a_1)}{G}, \frac{g(a_2)}{G}, \dots, \frac{g(a_n)}{G} \right),$$

where $G = \sum_{i=1}^n g(a_i)$, and the goal is for the algorithm to output a *sample* from this distribution in low space. That is, the probability, over the algorithm's randomness, that coordinate i is output is $\frac{g(a_i)}{G}$. We have not seen such a notion proposed before in its generality, though algorithms for sampling a random non-zero item ($g(x) = 0$ if $x = 0$, otherwise $g(x) = 1$) [12, 13, 27], and a random item based on its frequency in the insertion-only model (so all updates are positive and $g(x) = x$) [1] have been used and studied.

We may also relax this problem to allow an α -*relative-error sampler*, which is an algorithm that for all i , outputs i with probability in the interval

$$\left[(1 - \alpha) \frac{g(a_i)}{G}, (1 + \alpha) \frac{g(a_i)}{G} \right],$$

or an α -*additive-error sampler*, which is an algorithm that for all i , outputs i with probability in the interval

$$\left[\frac{g(a_i)}{G} - \alpha, \frac{g(a_i)}{G} + \alpha \right].$$

^{*}Department of Computer Science, University of Dortmund, Germany morteza.monemizadeh@tu-dortmund.de

[†] Research supported in part by DFG grant So 514/1-2.

[‡]IBM Almaden Research Center, San Jose, CA dpwoodru@us.ibm.com

Notice that an α -relative-error sampler is automatically an α -additive-error sampler. For the case of an α -relative-error sampler, we also allow a small additive error, provided it is at most n^{-C} for an arbitrarily large constant C . This error will not be detected if the sampling subroutine is invoked a polynomial number of times.

We can also choose the algorithm to report the probability $\frac{g(a_i)}{G}$, a probability in the interval

$$\left[(1 - \alpha) \frac{g(a_i)}{G}, (1 + \alpha) \frac{g(a_i)}{G} \right],$$

or a probability in the interval

$$\left[\frac{g(a_i)}{G} - \alpha, \frac{g(a_i)}{G} + \alpha \right],$$

depending on which case we are in. If the sampler has this latter property, we say it is an *augmented sampler*.

In this paper we consider the case $g(a_i) = |a_i|^p$ for $p \in \mathbb{R}^{\geq 0}$, in which case we call the corresponding sampler an L_p -sampler.

The following is our main theorem.

THEOREM 1.1. *For any $p \in [0, 2]$, there is a 1-pass α -relative-error augmented L_p -sampler that runs in $\text{poly}(\alpha^{-1} \log n)$ bits of space. Ignoring an initial data structure initialization stage (which doesn't require looking at the stream), the update time of the L_p -sampler is $\text{poly}(\alpha^{-1} \log n)$. There is also an n^{-C} probability of failure of the algorithm, in which case it can output anything. Here $C > 0$ is an arbitrarily large constant.*

We extend this result in a few ways if we are allowed more than one pass over the stream. Namely, using Theorem 1.1, we can show that for any $p \in [0, 2]$, there is a 2-pass exact (i.e., 0-relative-error) augmented L_p -sampler that uses $\text{polylog } n$ bits of space. The algorithm may also output the special symbol FAIL, which it does with probability less than n^{-C} for any constant $C > 0$.

The idea is to use a form of rejection sampling. That is, we begin by running the 1-pass α -relative-error augmented sampler of Theorem 1.1, with α set to a fixed constant, obtaining a coordinate i as well as a value $\tilde{p}_i = (1 \pm \alpha) \frac{|a_i|^p}{F_p(a)}$.

In parallel we obtain a number V so that $2F_p(a) \leq V \leq 4F_p(a)$ using the algorithm of [28]. In the second pass we compute $|a_i|$ exactly. With probability $|a_i|^p / (V \tilde{p}_i)$ we output i , otherwise we output FAIL. Notice that this probability is less than 1 since

$\tilde{p}_i \geq |a_i|^p / V$ for $\alpha < 1/2$. The probability that coordinate i is sampled is thus $\tilde{p}_i \cdot \frac{|a_i|^p}{V \tilde{p}_i} = \frac{|a_i|^p}{V}$. Hence, the probability that some item is sampled is $F_p/V \geq 1/4$. If no item is sampled, we output FAIL. By repeating this process in parallel $O(\log n)$ times, we ensure the probability of outputting FAIL is less than n^{-C} for an arbitrarily large constant C . We thus have,

THEOREM 1.2. *For any $p \in [0, 2]$, there is a 2-pass exact augmented L_p -sampler that uses $\text{polylog } n$ bits of space. The probability of outputting FAIL is less than n^{-C} for any constant $C > 0$.*

We also give an $O(\log n)$ -pass exact augmented L_p -sampler in Section 4.1 using $O(\log^5 n)$ bits of space. While the number of passes has increased, we have attempted to minimize the number of log factors in the space. This has applications to reducing $\text{poly}(\varepsilon^{-1})$ factors in the communication complexity of two-party L_k -estimation for $k > 2$, see Section 2.4.

THEOREM 1.3. *For any $p \in [0, 2]$, there is an $O(\log n)$ -pass exact augmented L_p -sampler that uses $O(\log^5 n)$ bits of space. The probability of outputting FAIL is less than n^{-C} for any constant $C > 0$.*

2 Applications

Theorem 1.1 leads to many improvements and a unification of well-studied streaming problems:

2.1 Weighted Sampling with Deletions: Cormode, Muthukrishnan, and Rozenbaum [14] state that “A fundamental question that arises is to design algorithms to maintain a uniform sample of the forward distribution under both insertions and deletions or show that it is impossible.” Here, by forward distribution, the authors mean to return a sample i with probability $|a_i| / \|a\|_1$, even if coordinate i undergoes deletions in the stream. Setting $p = 1$ in Theorem 1.1 therefore resolves the main open question of [14] (up to a small relative error). As sampling in the presence of deletions is a useful primitive, we expect this to have many more applications. For instance, Frahling, Indyk, and Sohler [17] study geometric problems such as maintaining approximate range spaces and costs of Euclidean spanning trees. They need a routine which, given a pointset P undergoing multiple insertions and deletions, maintains a random element from P . Applying Theorem 1.1 with $p = 0$ also solves this problem. Another application is that if the support of the underlying vector a is at most k , by running the algorithm of Theorem 1.1 with $p = 0$ at most $O(k \log k)$ times, with constant probability all k non-zero items will be found.

2.2 Cascaded Norms: In [26], Jayram and the second author give 1-pass algorithms for estimating cascaded moments of an $n \times d$ matrix A . Namely, they show that for $k \geq 1$ and $p \geq 2$, the 1-pass space complexity of outputting a $(1 \pm \varepsilon)$ -approximation to $F_k(F_p)(A)$ is

$$n^{1-2/(kp)} d^{1-2/p} \text{poly}(\varepsilon^{-1} \log(nd)).$$

They leave open the question of estimating $F_k(F_p)(A)$ for $k \geq 1$ and $p < 2$, though they prove an $\Omega(n^{1-1/k})$ space lower bound for this problem. The only other work in this regime is due to Cormode and Muthukrishnan [13] who prove an $n^{1/2} \text{poly}(\varepsilon^{-1} \log(nd))$ upper bound for estimating $F_2(F_0)(A)$ assuming that all stream updates are positive.

Theorem 1.1 implies a near-optimal $n^{1-1/k} \text{poly}(\varepsilon^{-1} \log(nd))$ -space 1-pass upper bound for any $k \geq 1$ and $p \in [0, 2]$, which, together with the results of [26], closes the problem for $k \geq 1$ and any $p \geq 0$, up to small factors. Note that for $p < 1$, L_p is not a norm, but the expression for L_p is still well-defined. As our algorithm works in the general turnstile model, this also improves the algorithm of [13] for estimating $F_2(F_0)$.

$F_k(F_p)$ -Estimation(A, ε):

1. Initialize $T = n^{1-1/k} \text{poly}(\varepsilon^{-1} \log(nd))$.
2. For $\ell \in [\log_2 n]$, let $A(\ell)$ be a random subset of 2^ℓ rows of A .
3. Run L_p -Sampler algorithm T times in parallel on each $A(\ell)$.
4. Feed the row IDs of the samples from the $A(\ell)$ into the 1-pass F_k -algorithm of [26].

The 1-pass F_k -estimation algorithm given in [26] is similar to the $n^{1-1/k} \text{poly}(\varepsilon^{-1} \log(nd))$ -space algorithm for estimating F_k of [1], but given a vector (b_1, \dots, b_n) , it requires only sampling $n^{1-1/k} \text{poly}(\varepsilon^{-1} \log(nd))$ coordinate IDs i , rather than approximations to their frequencies (the b_i), where i is sampled with probability $\frac{|b_i|}{\sum_{i=1}^n |b_i|}$. The algorithm also requires such samples from random subsets of 2^ℓ coordinates of the vector, for each $\ell \in [\log_2 n]$. In our case, we run the sampler on the matrices $A(\ell)$, obtaining an entry in a given row i with probability

$$(1 \pm \varepsilon) \frac{\sum_{j=1}^d |A(\ell)_{i,j}|^p}{\sum_{i=1}^n \sum_{j=1}^d |A(\ell)_{i,j}|^p}.$$

Thus, b_i in the F_k subroutine is equal to

$$(1 \pm \varepsilon) \sum_{j=1}^d |A(\ell)_{i,j}|^p.$$

THEOREM 2.1. *For any $p \in [0, 2], k \geq 1$, there is a 1-pass streaming algorithm which, with probability $\geq 1 - 1/\text{poly}(nd)$, outputs a $(1 \pm \varepsilon)$ -approximation to $F_k(F_p)(A)$ using space $n^{1-1/k} \text{poly}(\varepsilon^{-1} \log(nd))$.*

2.3 Heavy Hitters and Block Heavy Hitters:

The classical heavy hitters problem is to report all coordinates i for which $|a_i| \geq \phi \|a\|_p$, where ϕ is an input parameter. For $p = 1$ this is solved by the CountMin data structure of Cormode and Muthukrishnan [13], and for $p = 2$ by the CountSketch data structure by Charikar, Chen, and Farach-Colton [9]. Notice that Theorem 1.1 immediately implies an algorithm for every $p \in [0, 2]$. Indeed, run the augmented sampler of Theorem 1.1 $O(\phi^{-1} \log \phi^{-1})$ times in parallel. Then with constant probability, the list of samples contains all heavy hitters, and using the probabilities returned, one can ensure that if $|a_i|^p \leq (\phi - \varepsilon) \|a\|_p^p$, then i is not reported. Notice that our algorithm works in the turnstile model.

Another immediate application is that of computing block heavy hitters, which is the problem of reporting all rows a^i of an $n \times d$ matrix A for which $\|a^i\|_1$ is at least a ϕ fraction of the L_1 -norm $\|A\|_1$ of A (i.e., $\|A\|_1 = \sum_{j=1}^n \|a^j\|_1$). These rows are called the block heavy hitters, and are a crucial building block in a streaming algorithm of Andoni, Indyk, and Kraughtgamer [4] that constructs a small-size sketch for the Ulam metric under the edit distance. In [3], Andoni, DoBa, and Indyk devise a 1-pass algorithm for this problem using $\text{poly}(\phi^{-1} \log n)$ bits of space.

Notice that if a^i is a block heavy hitter, then if we sample a random entry of A proportional to its absolute value, the probability it comes from row i is at least ϕ . Moreover, based on the number of times an item from row j is sampled, one can detect if

$$\|a^j\|_1 \leq (\phi - \varepsilon) \|A\|_1.$$

Hence, Theorem 1.1 immediately implies the main result of [3]. It should be noted that the proof of Theorem 1.1 does not rely on Nisan's pseudorandom generator or go through p -stable distributions, which could potentially make our block heavy hitters algorithm more practical than that of [3]. Moreover, Theorem 1.1 immediately gives the analogous result for every L_p with $p \in [0, 2]$.

2.4 Moment Estimation: In [11], Coppersmith and Kumar reduce the problem of estimating F_k of a vector to the problem of sampling according to F_2 . In particular Proposition 4.1 of that paper states “If there is a black-box such that

1. it uses $\tilde{O}(1)$ space,
2. it makes $\tilde{O}(1)$ passes over the input,
3. each invocation outputs a random variable Y such that $\Pr[Y = i] \propto a_i^2$,

then there is a data stream algorithm for approximating F_3 that uses $\tilde{O}(n^{1/3})$ space and makes $\tilde{O}(1)$ passes over the input.” As the sampler of Theorem 1.1 satisfies these conditions, we immediately obtain an alternative F_3 -estimation algorithm in optimal space (up to small factors). This generalizes to arbitrary F_k , $k > 2$, and can be implemented in a single pass, giving an alternative algorithm to that of Indyk and the second author [25]. Our algorithm is the first that does not use Nisan’s pseudorandom generator as a subroutine, potentially making it more practical. Moreover, if we consider the two-party communication complexity of L_k -estimation, $k > 2$, we can use an $O(\log n)$ -pass version of our sampler in Theorem 1.3 to improve the dependence on ε of known algorithms [7, 25]. Both of these algorithms are described in Section 4.2.

3 1-Pass L_p -Sampler

3.1 Proof Overview. Here we give an overview for the case $p \in (0, 2]$. The case $p = 0$ is simpler and is handled in Section 3.2.2. As done in the work of Indyk and the second author [25], we conceptually divide the coordinates into classes

$$S_t = \{i \mid |a_i| \in [\eta^{t-1}, \eta^t)\},$$

where $\eta = 1 + \Theta(\varepsilon)$. As in [25], we say that a class S_t *contributes* if

$$|S_t| \eta^{pt} \geq \gamma F_p(a),$$

where $\gamma = \text{poly}(\varepsilon \log^{-1} n)$ is a sufficiently small parameter.

Let $h : [n] \rightarrow [n]$ be a hash function with some amount of limited independence. We form $r = O(\log n)$ substreams $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_r$, where \mathcal{S}_0 denotes the input stream and \mathcal{S}_j denotes the stream updates which pertain only to coordinates i for which $h(i) \leq n2^{-j}$. We say that such an i is a *survivor* (with respect to a given \mathcal{S}_j). As observed in [25], if S_t contributes, then there will be a substream \mathcal{S}_j for

which the survivors i in S_t are heavy hitters with respect to the p -norm, that is, $|a_i|^p \geq \gamma' \|a(j)\|_p^p$, where $a(j)$ denotes the vector a restricted to the survivors in \mathcal{S}_j , and $\gamma' = \text{poly}(\varepsilon \log^{-1} n)$. The heavy hitters in each substream can be found with $\text{poly}(\varepsilon^{-1} \log n)$ space using known heavy hitter algorithms [9, 19], which work for any $p \in (0, 2]$, and even have a fast $\text{poly}(\varepsilon^{-1} \log n)$ update and reporting time [19] in the turnstile model. Here we critically rely on the fact that $p \in (0, 2]$, as otherwise, the space to find such heavy hitters is known to be polynomial in n [5]. We call the algorithm of [19] **HeavyHitters**. By zooming in on the appropriate substream and counting the number of survivors that are heavy hitters, we can estimate $|S_t|$ to within $(1 \pm \Theta(\varepsilon))$ for all S_t that contribute. Notice that here we need the **HeavyHitters** algorithm to also provide $(1 \pm \Theta(\varepsilon))$ -approximations to the values $|a_i|$ for each heavy hitter i , which can be achieved by increasing the space by a $\text{poly}(\varepsilon^{-1} \log n)$ factor. We also need the values $|a_i|$ to not be close to the values η^t , as otherwise we might misclassify i as belonging to S_{t+1} when in fact it belongs to S_t . While in previous work [25, 26] a rerandomization technique was necessary to achieve this, in our work we show that this is not an issue.

For S_t that do not contribute, we only obtain an estimate E for which $0 \leq E \leq (1 + \Theta(\varepsilon))|S_t|$. This sub-sampling approach is the basis of the algorithm of Indyk and the second author [25] for estimating frequency moments.

As observed by Jayram and the second author [26] (see also [2]) this approach yields a 1-pass $\Theta(\varepsilon)$ -additive-error augmented sampler in the following way. We simply ignore the S_t that do not contribute, and let the \tilde{s}_t be our approximations to the set sizes $|S_t|$ for contributing S_t . We sample a contributing S_t with probability

$$\frac{\tilde{s}_t \eta^{pt}}{\sum_{\text{contributing } S_t} \tilde{s}_t \eta^{pt}}.$$

Given that we chose S_t , we output a heavy hitter in S_t found in the substream \mathcal{S}_j used to estimate S_t . In the work of [26], these heavy hitters were essentially random elements of S_t since h was generated using Nisan’s pseudorandom generator (see, e.g., [22]), so its output distribution was very close to that of a truly random function. By instead appealing to a theorem of Indyk [21], which shows that if h is sufficiently independent then it is also ε -min-wise independent, we could instead take the heavy hitter found in S_t which minimizes h . This can be shown to be a random element of S_t , up to a relative $(1 \pm \varepsilon)$ -

error. It is now straightforward to show that we output i with probability

$$(1 \pm \Theta(\varepsilon)) \cdot \frac{|a_i|^p}{F_p(a)}.$$

The problem with the above approach is that it leads to an additive error rather than a relative error. Indeed, items in non-contributing classes will *never* be sampled. This is problematic if the sampling algorithm is to be used in a subroutine that performs more than $\text{poly}(\varepsilon^{-1} \log n)$ samples, as some items that should be reported by an exact sampler will not be detected.

Our main novelty is the following idea. Suppose we *force* every class to contribute. If we could do this, then we could try to apply the above sampling procedure to obtain a $\Theta(\varepsilon)$ -relative-error augmented sampler. To force each class to contribute, the idea is to inject new coordinates into the stream. That is, let $T = O(\varepsilon^{-1} \log n)$ be the number of classes. For class S_t , we inject $\Theta(\varepsilon)F_p(a)/(T\eta^{pt})$ coordinates i for which $|a_i| \in [\eta^{t-1}, \eta^t]$. It follows that F_p changes by at most a $(1 + \Theta(\varepsilon))$ factor. Moreover, now $|S_t|\eta^{pt} = \Omega(\varepsilon F_p(a)/T)$ for all t , and so provided $\gamma = O(\varepsilon/T) = O(\varepsilon^2/\log n)$, every class now contributes. Actually, for technical reasons, we cannot force a class S_t to contribute if η^{pt} is too close to F_p , but it turns out that these classes can be dealt with separately.

Notice that we do not know $F_p(a)$ in advance, but it suffices to guess a $(1 + \Theta(\varepsilon))$ -approximation to it, and then verify our guess at the end of the stream by running an efficient F_p -approximation algorithm in parallel, taking only $\text{poly}(\varepsilon^{-1} \log n)$ bits of space, e.g., the space optimal algorithm of Kane *et al* [28], or the earlier algorithms of Indyk [22] and Li [29]. The number of guesses we need is only $O(\varepsilon^{-1} \log n)$.

We now run the sampling algorithm above. If we obtain an injected coordinate, then we output **FAIL**, otherwise we output the coordinate and its approximate value returned by **HeavyHitters**. Notice that the injected items contribute at most an $O(\varepsilon)$ mass to the F_p -value, so we output **FAIL** with probability at most $O(\varepsilon)$. By repeating this procedure in parallel a logarithmic number of times, at least one of our samples will not be an injected coordinate with probability at least $1 - n^{-C}$ for an arbitrarily large constant $C > 0$.

The actual proof of Theorem 1.1 is substantially more involved. The reason is that we want additive error probability n^{-C} , while the above “black-box” approach of injecting coordinates and using earlier algorithms of [25] and [26] can be shown to give an additive error of at least $\text{poly}(\varepsilon \log^{-1} n)$. Here we cannot just repeat the algorithm a logarithmic number

of times, since in each repetition a sample is returned with a $\text{poly}(\varepsilon \log^{-1} n)$ bias. We devise a new procedure **Sample-Extraction** to shrink the additive error of certain parts of the algorithm, thereby overcoming this. Some of our new components for achieving this, e.g., our primitive **Set-Estimation** for estimating the level set sizes $|S_t|$ up to a relative error ε , are simpler than alternatives given in [25].

3.2 The Complete Proof. We first describe a solution for $p \in (0, 2]$. In Section 3.2.2 we describe how to sample when $p = 0$, which is simpler. Throughout we use an algorithm that we call **HeavyHitters**, which is given by the following theorem. The algorithm has its roots in the **CountSketch** algorithm of Charikar *et al* [9], but that algorithm did not have fast reporting time for streams in the turnstile model. This property was later achieved by Ganguly *et al* [19], building upon work of Cormode and Muthukrishnan [12].

THEOREM 3.1. ([9, 12, 19]) *Let $0 < \delta < 1$. Let a be a vector of length n initialized to zero. Let \mathcal{S} be a stream of m updates (i, x) to a , where $i \in [n]$ and $x \in \{-M, \dots, +M\}$. There is an algorithm **HeavyHitters** $(\mathcal{S}, B, \delta, \epsilon)$ that, with probability at least $1 - \delta$, returns all coordinates i for which $|a_i|^2 \geq \frac{F_2}{B}$, together with an approximation \tilde{a}_i such that $|a_i| \leq |\tilde{a}_i| \leq (1 + \epsilon)|a_i|$ and $\text{sign}(\tilde{a}_i) = \text{sign}(a_i)$. Here B is an input parameter. The space complexity of **HeavyHitters** is $B \cdot \text{poly}(\varepsilon^{-1} \log n) \log 1/\delta$. The update time is $\text{poly}(\varepsilon^{-1} \log n)$, and the reporting time is $\text{poly}(B\varepsilon^{-1} \log n)$.*

COROLLARY 3.1. *For any $p \in (0, 2]$, with probability at least $1 - \delta$, the output of **HeavyHitters** also contains all items i for which $|a_i|^p \geq \frac{F_p}{B^{p/2}}$.*

Proof. For $p \in (0, 2]$, if $|a_i|^p \geq \frac{F_p}{B^{p/2}}$, then $|a_i|^2 \geq \frac{F_p^{2/p}}{B}$, and, by monotonicity of norms (see, e.g., [18]), $F_p^{2/p} \geq F_2$. Hence, $|a_i|^2 \geq \frac{F_2}{B}$, as needed in order to be found by **HeavyHitters**. For $p < 1$, $F_p^{1/p}$ is not a norm, but it is still a well-defined expression. ♣

We start with the following definition of level sets, similar to that given in [25, 26]. Fix an $\eta = 1 + \Theta(\varepsilon)$. Let $C' > 0$ be a sufficiently large constant. We shall make the simplifying assumption that all values $|a_i|$, if non-zero, are integers of absolute value at least $\tau = C'\varepsilon^{-1}$. This is w.l.o.g., since for each update (i, x) in the input stream \mathcal{S} , we can replace it with the update $(i, \tau x)$. This will also change F_p by a factor of τ^p , but will not affect the distribution we are trying to sample from.

DEFINITION 3.1. Let $B \geq 1$ be a parameter. For $\log_\eta \tau + 1 = \log_\eta(C'\varepsilon^{-1}) + 1 \leq t \leq C_\eta \log n$, for some constant C_η that depends on η , define

$$S_t = \{i \in [n] : |a_i| \in [\eta^{t-1}, \eta^t]\}.$$

Call a level t $(1/B)$ -contributing if

$$|S_t| \cdot \eta^{pt} \geq \frac{F_p}{B}.$$

For a $(1/B)$ -contributing level t , coordinates in S_t will also be called $(1/B)$ -contributing coordinates.

In the description of our algorithms, we assume that we have a value F'_p with $F_p \leq F'_p \leq (1 + \Theta(\varepsilon))F_p$. This is w.l.o.g., since the algorithm can guess each value in a set of $O(\varepsilon^{-1} \log n)$ possible values, and use this as the estimate F'_p . In parallel, the algorithm runs the $O(\varepsilon^{-2} \log n)$ space algorithm of [28] to obtain a $(1 + \Theta(\varepsilon))$ -approximation to F_p , and after processing the stream knows which of its guesses is correct. This only changes the space by a $\text{poly}(\varepsilon^{-1} \log n)$ factor.

Put $T = C_\eta \log n$. We will perform the following transformation to the stream \mathcal{S} , creating a new input stream \mathcal{S}' . Say a $t \in [T]$ is *growing* if

$$\eta^{pt} \leq \frac{\varepsilon^4 F'_p}{5T^3 \log^2 n}.$$

StreamTransformation(\mathcal{S}, p)

- $\mathcal{S}' \leftarrow \mathcal{S}$
- For each $t \in [T]$ for which t is growing:
 1. Allocate $\left\lceil \frac{\varepsilon F'_p}{5T\eta^{pt}} \right\rceil$ new coordinates.
 2. For each new coordinate j , prepend the pair $(j, \lfloor \eta^{t-1/2} \rfloor)$ to the stream \mathcal{S}' .
- Output the stream \mathcal{S}' .

Let a' denote the underlying vector of \mathcal{S}' , which is of length $\text{poly}(n)$. For each new coordinate added by StreamTransformation, we refer to it as an *injected coordinate* (that is, a coordinate that appears in a' but not in a is an injected coordinate). We first show that the injected coordinates with value $\lfloor \eta^{t-1/2} \rfloor$ are indeed in the level set S_t , and are away from the upper boundary of that level set (the value η^t).

LEMMA 3.1. For all $t \geq \log_\eta(C'\varepsilon^{-1}) + 1$,

$$\eta^{t-1} \leq \lfloor \eta^{t-1/2} \rfloor \leq \frac{\eta^t}{1 + \Theta(\varepsilon)}.$$

Proof.

$$\begin{aligned} \frac{\lfloor \eta^{t-1/2} \rfloor}{\eta^{t-1}} &\geq \frac{\eta^{t-1/2} - 1}{\eta^{t-1}} = \eta^{1/2} - \frac{1}{\eta^{t-1}} \\ &\geq \eta^{1/2} - \frac{\varepsilon}{C'} = 1 + \Theta(\varepsilon) - \frac{\varepsilon}{C'}, \end{aligned}$$

and the latter is at least 1 for sufficiently large C' . For the other direction, $\lfloor \eta^{t-1/2} \rfloor \leq \frac{\eta^t}{\eta^{1/2}} \leq \frac{\eta^t}{1 + \Theta(\varepsilon)}$. ♣

LEMMA 3.2. For growing t , the number of injected coordinates added is at least $\varepsilon^{-3} T^2 \log^2 n$.

Proof. For growing t , the number of coordinates added is $\left\lceil \frac{\varepsilon F'_p}{5T\eta^{pt}} \right\rceil \geq \frac{\varepsilon F'_p}{5T\varepsilon^4 F'_p / (5T^3 \log^2 n)} = \varepsilon^{-3} T^2 \log^2 n$. ♣

LEMMA 3.3. $F_p(a) \leq F_p(a') \leq (1 + \varepsilon/2)F_p(a)$, for ε less than a sufficiently small constant.

Proof. The left inequality is obvious. For the right inequality, fix a growing t . Then the added contribution of this level set to F_p is at most $\lceil \varepsilon F'_p / (5T\eta^{pt}) \rceil \cdot \eta^{p(t-1/2)}$. By Lemma 3.2, $\lceil \varepsilon F'_p / (5T\eta^{pt}) \rceil \geq \varepsilon^{-3} T^2 \log^2 n$, which in particular implies $\varepsilon F'_p / (5T\eta^{pt}) \geq 1$. Hence, $\lceil \varepsilon F'_p / (5T\eta^{pt}) \rceil \cdot \eta^{p(t-1/2)} \leq 2\varepsilon \eta^{pt-p/2} F'_p / (5T\eta^{pt}) \leq 2\varepsilon F'_p / (5T)$. Hence, the added contribution is at most $2\varepsilon F'_p / (5T) \leq \varepsilon F_p(a) / (2T)$, for ε small enough so that F'_p is close enough to $F_p(a)$. The lemma follows by summing over all t . ♣

LEMMA 3.4. With respect to a' , each growing t is $\varepsilon / (40T)$ -contributing.

Proof. Each growing t contributes at least

$$\lfloor \eta^{t-1/2} \rfloor^p \cdot \frac{\varepsilon F'_p}{5T\eta^{pt}} \geq \frac{\eta^{pt-p/2}}{2^p} \cdot \frac{\varepsilon F'_p}{5T\eta^{pt}}$$

to $F_p(a')$, which is at least

$$\frac{\eta^{-p/2} \varepsilon F'_p}{5T2^p} \geq \frac{\eta^{-p/2} \varepsilon F_p(a)}{5T2^p} \geq \frac{\eta^{-p/2} \varepsilon F_p(a')}{5T2^p(1 + \frac{\varepsilon}{2})} \geq \frac{\varepsilon F_p(a')}{40T},$$

where the second to last inequality follows from Lemma 3.3, and the last for small enough ε . ♣

We start with the following assumption: for all i , if $i \in S_t$, then $|a'_i| \leq \eta^t / (1 + \beta\varepsilon)$ for a small constant $\beta > 0$. Intuitively, this is to ensure that the $|a'_i|$ are away from their upper boundaries, so that when HeavyHitters returns $(1 + \Theta(\varepsilon))$ -approximations to the $|a'_i|$, which are guaranteed to be at least $|a'_i|$, the i

can be classified correctly into the corresponding S_t . We note that for an injected coordinate j with value $\lfloor \eta^{t-1/2} \rfloor$ for some t , this assumption already holds by Lemma 3.1. In Section 3.2.1, we show that this assumption is unnecessary. We just use it now for ease of presentation.

We can consider operating on stream \mathcal{S}' as opposed to \mathcal{S} , since **StreamTransformation** can be implemented in $\text{poly}(\varepsilon^{-1} \log n)$ space in a preprocessing phase, i.e., without looking at the pairs in \mathcal{S} . The following is our main sampling algorithm.

L_p -Sampler(\mathcal{S}', p)

1. $L \leftarrow \text{HeavyHitters}(\mathcal{S}', A, n^{-C}, \varepsilon/C)$, where $A = (5\eta^p T^3 \varepsilon^{-4} \log^2 n)^{2/p}$. Let the $(\tilde{a}')_i$ be estimates of the a'_i for all $i \in L$.
2. $B = (6400 \cdot 1600 \varepsilon^{-3} \eta^p T^3 \log^3 |a'|)^{2/p}$.
3. Independently, for $z = 1, 2, \dots, C \log n$,
 - (a) Put $\{L_j^z \mid \text{all } j\} = \text{ListHH}(\mathcal{S}', B)$.
 - (b) Put $\{\tilde{s}_t^z \mid \text{growing } t\} = \text{Set-Estimation}(\{L_j^z \mid \text{all } j\})$.
4. For growing t , put $\tilde{s}_t = \text{median}_z \tilde{s}_t^z$.
5. $\{(x_t, (\tilde{a}')_{x_t}) \mid \text{growing } t\} = \text{Sample-Extraction}(\mathcal{S}', B)$.
6. $G = \sum_{i \in L, \text{ not in growing } t} |(\tilde{a}')_i|^p + \sum_{\text{growing } t} \tilde{s}_t \cdot |(\tilde{a}')_{x_t}|^p$.
7. Choose a sample u from the distribution:

$$\Pr[u = i] = \frac{|(\tilde{a}')_i|^p}{G} \text{ if } i \in L, \text{ not in growing } t;$$

$$\Pr[u = x_t] = \frac{\tilde{s}_t \cdot |(\tilde{a}')_{x_t}|^p}{G}.$$
8. Repeat steps (1-7) independently $C \log n$ times. If in all repetitions the sample u returned is an injected coordinate, then report **FAIL**; else return $(u, (\tilde{a}')_i)$ if $i \in L$, or the value $(u, (\tilde{a}')_{x_t})$. Do this from the first repetition for which the sample u obtained is not an injected coordinate.

One of the subroutines that algorithm L_p -Sampler invokes is **ListHH**, which we describe next. This is an algorithm which takes in the stream \mathcal{S}' and a parameter B , sub-samples the stream at a logarithmic number of different rates, using independent hash functions $h_j : [|a'|] \rightarrow [|a'|]$, and then invokes **HeavyHitters** on the substream with B as an input parameter. Note that here $|a'|$ refers to the dimension of a' , which

is known (up to a constant factor) to the streaming algorithm (assuming n or an upper bound on n is known). Let C be a sufficiently large constant.

ListHH(\mathcal{S}', B)

1. For $j \in [\log |a'|]$, independently sample functions $h_j : [|a'|] \rightarrow [|a'|]$ from a set of pairwise independent hash functions.
2. Let \mathcal{S}_j be the restriction of \mathcal{S}' to those pairs (i, x) for which $h_j(i) \leq |a'|2^{-j}$.
3. Return, for each $j \in [\log |a'|]$,

$$L_j = \text{HeavyHitters}(\mathcal{S}_j, B, 1/(C \log |a'|), \varepsilon/C).$$

The **ListHH** algorithm is invoked $C \log n$ times in step 3 of L_p -Sampler, once for each value of z . Moreover, the following **Set-Estimation** algorithm is also invoked for each value of z .

Set-Estimation($\{L_j \mid \text{all } j\}$)

1. For growing t ,
 - (a) Choose the largest j for which L_j contains at least $1600 \varepsilon^{-2} T^2 \log^2 |a'|$ elements of S_t . If there is such a j ,
 - $S'_t = S_t \cap L_j$ and $j(t) = j$.
 - $\tilde{s}_t = 2^{j(t)} \cdot |S'_t|$.
2. Return \tilde{s}_t for each t for which a j was found, and return 0 for the other growing t .

Let us fix a value of z in step 3, and analyze the output of **ListHH** and **Set-Estimation**. For this value of z , let V_j denote the set of coordinates i in a' for which $h_j(i) \leq |a'|2^{-j}$ in **ListHH**.

The Random Events. Define the events:

- \mathcal{E} : for all growing t and all $j \in [\log |a'|]$, $|S_t \cap V_j| \leq 40|S_t|2^{-j}T \log |a'|$.
- \mathcal{F} : for all growing t and all $j \in [\log |a'|]$, if $\mathbf{E}[|S_t \cap V_j|] \geq 40\varepsilon^{-2}T \log |a'|$, then $|S_t \cap V_j| \in [(1 - \varepsilon)|S_t|2^{-j}, (1 + \varepsilon)|S_t|2^{-j}]$.
- \mathcal{G} : for all $j \in [\log |a'|]$, $F_p(a'(j)) \leq 40 \log |a'| \cdot 2^{-j} F_p(a')$, where $a'(j)$ is the restriction of a' to the coordinates in V_j .
- \mathcal{H} : all invocations of **HeavyHitters** by **ListHH** succeed.

LEMMA 3.5.

$$\Pr[\mathcal{E} \wedge \mathcal{F} \wedge \mathcal{G} \wedge \mathcal{H}] \geq 9/10.$$

Proof. First for the event \mathcal{E} , observe that $\mathbf{E}[|S_t \cap V_j|] = |S_t|2^{-j}$, and so by a Markov and a union bound over all t and all j , we obtain

$$\Pr[\mathcal{E}] \geq 39/40.$$

To bound $\Pr[\mathcal{F}]$, fix a growing t and a j satisfying

$$\mathbf{E}[|S_t \cap V_j|] \geq 40\varepsilon^{-2}T \log |a'|.$$

Let $I_x^{t,j}$ be an indicator for the event that an $x \in S_t$ is also in V_j . Put $I^{t,j} = \sum_{x \in S_t} I_x^{t,j}$. Then

$$\mathbf{E}[I^{t,j}] = |S_t|2^{-j} \geq 40\varepsilon^{-2}T \log |a'|.$$

Since h_j is drawn from a pairwise-independent family, $\mathbf{Var}[I^{t,j}] \leq \mathbf{E}[I^{t,j}]$. By Chebyshev's inequality,

$$\begin{aligned} \Pr[|I^{t,j} - \mathbf{E}[I^{t,j}]| \geq \varepsilon \mathbf{E}[I^{t,j}]] &\leq \frac{1}{\varepsilon^2 \mathbf{E}[I^{t,j}]} \\ &\leq \frac{1}{\varepsilon^2 \cdot 40\varepsilon^{-2}T \log |a'|} \leq \frac{1}{40T \log |a'|}. \end{aligned}$$

By a union bound over growing t and the j satisfying $\mathbf{E}[|S_t \cap V_j|] \geq 40\varepsilon^{-2}T \log |a'|$,

$$\Pr[\mathcal{F}] \geq 39/40.$$

For the event \mathcal{G} , for a fixed j , we have $\mathbf{E}[F_p(a'(j))] = 2^{-j}F_p(a')$. By a Markov bound,

$$\Pr[F_p(a') \geq 40 \log |a'| \cdot 2^{-j}F_p(a')] \leq \frac{1}{40 \log |a'|}.$$

By a union bound,

$$\Pr[\mathcal{G}] \geq 39/40.$$

Finally for the event \mathcal{H} , as **HeavyHitters** is invoked $\log |a'|$ times with error probability $1/(C \log |a'|)$, for a sufficiently large constant C , we get that

$$\Pr[\mathcal{H}] \geq 39/40.$$

By a union bound, $\Pr[\mathcal{E} \wedge \mathcal{F} \wedge \mathcal{G} \wedge \mathcal{H}] \geq 9/10$. ♣

LEMMA 3.6. Fix a $z \in [C \log n]$. Then with probability at least $9/10$, for all growing t , there is a $j = j(t)$ assigned to t by **Set-Estimation**, and also

$$\tilde{s}_t^z \in [(1 - \varepsilon)|S_t|, (1 + \varepsilon)|S_t|].$$

Proof. We condition on $\mathcal{E}, \mathcal{F}, \mathcal{G}$, and \mathcal{H} jointly occurring. Fix a growing t . We first show there is a $j = j(t)$ assigned to t by **Set-Estimation**. By Lemma 3.2,

$$|S_t| \geq \varepsilon^{-3}T^2 \log^2 n.$$

It follows that, for small enough ε , there is a unique $j^* \geq 0$ for which

$$(3.1) \quad 3200\varepsilon^{-2}T^2 \log^2 |a'| \leq 2^{-j^*}|S_t| < 6400\varepsilon^{-2}T^2 \log^2 |a'|.$$

(recall that $\log |a'| = O(\log n)$). Since S_t is growing, by Lemma 3.4, $|S_t| \cdot \eta^{pt} \geq \frac{\varepsilon F_p(a')}{40T}$. Hence,

$$2^{-j^*}|S_t|\eta^{pt} \geq \frac{\varepsilon 2^{-j^*}F_p(a')}{40T}.$$

Since event \mathcal{G} occurs,

$$F_p(a'(j^*)) \leq 40 \log |a'| \cdot 2^{-j^*}F_p(a'),$$

and so

$$(3.2) \quad 2^{-j^*}|S_t|\eta^{pt} \geq \frac{\varepsilon F_p(a'(j^*))}{1600T \log |a'|}.$$

Since $2^{-j^*}|S_t| < 6400\varepsilon^{-2}T^2 \log^2 |a'|$ by (3.1), we have

$$\eta^{p(t-1)} \geq \frac{\varepsilon^3 F_p(a'(j^*))}{6400 \cdot 1600\eta^p T^3 \log^3 |a'|}.$$

Now we use the fact that event \mathcal{H} occurs, and so **HeavyHitters**($\mathcal{S}_{j^*}, B, 1/(C \log |a'|), \varepsilon/C$) succeeds in reporting a list $L_{j^*}^z$ containing all i for which $|a'_i|^p \geq \frac{F_p(a'(j^*))}{B^{p/2}}$, where

$$B = (6400 \cdot 1600\varepsilon^{-3}\eta^p T^3 \log^3 |a'|)^{2/p}.$$

In particular, $L_{j^*}^z$ contains $V_{j^*} \cap S_t$, and these coordinates i will be correctly classified into S_t given our assumption that the $|a'_i|$ are away from their upper boundaries. Finally, notice that

$$\mathbf{E}[|S_t \cap V_{j^*}|] = 2^{-j^*}|S_t|,$$

which is at least $3200\varepsilon^{-2}T^2 \log^2 |a'|$ by (3.1). Hence, since event \mathcal{F} occurs,

$$|V_{j^*} \cap S_t| \geq 3200(1 - \varepsilon)\varepsilon^{-2}T^2 \log^2 |a'| \geq 1600\varepsilon^{-2}T^2 \log^2 |a'|.$$

It follows that t is assigned a value $j(t)$ in **Set-Estimation**.

Now we show that $\tilde{s}_t^z \in [(1 - \varepsilon)|S_t|, (1 + \varepsilon)|S_t|]$. We must show that

$$2^{j(t)}|S'_t| \in [(1 - \varepsilon)|S_t|, (1 + \varepsilon)|S_t|],$$

for the $j(t)$ assigned to t by **Set-Estimation** (which need not equal j^*), and where $S'_t = S_t \cap L_{j(t)}^z$. Now $j(t)$ is such that $|L_{j(t)}^z \cap S_t| \geq 1600\epsilon^{-2}T^2 \log^2 |a'|$. Since event \mathcal{E} occurs,

$$|S_t \cap V_{j(t)}| \leq 40|S_t|2^{-j(t)}T \log |a'|.$$

Using that $|S_t \cap V_{j(t)}| \geq |S_t \cap L_{j(t)}^z|$,

$$|S_t|2^{-j(t)} = \mathbf{E}[|S_t \cap V_{j(t)}|] \geq 40\epsilon^{-2}T \log |a'|.$$

But then since event \mathcal{F} occurs, it follows that

$$|S_t \cap V_{j(t)}| \in [(1 - \epsilon)|S_t|2^{-j(t)}, (1 + \epsilon)|S_t|2^{-j(t)}].$$

The same analysis for j^* in (3.2) shows that for $j(t)$,

$$2^{-j(t)}|S_t|\eta^{pt} \geq \frac{\epsilon F_p(a'(j(t)))}{1600T \log |a'|}.$$

Since we have shown that $L_{j^*}^z$ contains at least $1600\epsilon^{-2}T^2 \log^2 |a'|$ elements of S_t , we must necessarily have $j(t) \geq j^*$ since the **Set-Estimation** algorithm chooses the largest value of j for which L_j^z contains at least $1600\epsilon^{-2}T^2 \log^2 |a'|$ elements of S_t . But then

$$2^{-j(t)}|S_t| \leq 2^{-j^*}|S_t| < 6400\epsilon^{-2}T^2 \log^2 |a'|,$$

and so

$$\eta^{p(t-1)} \geq \frac{\epsilon^3 F_p(a'(j(t)))}{6400 \cdot 1600\eta^p T^3 \log^3 |a'|}.$$

Since we are conditioning on event \mathcal{H} occurring, **HeavyHitters**($\mathcal{S}_{j(t)}, B, 1/(C \log |a'|), \epsilon/C$) succeeds in reporting a list $L_{j(t)}^z$ containing $V_{j(t)} \cap S_t$, and these coordinates i will be correctly classified into S_t given that the $|a'_i|$ are away from their upper boundaries. It follows that $|S'_t| = |V_{j(t)} \cap S_t|$ lies in

$$[(1 - \epsilon)|S_t|2^{-j(t)}, (1 + \epsilon)|S_t|2^{-j(t)}].$$

Hence, $\tilde{s}_t^z \in [(1 - \epsilon)|S_t|, (1 + \epsilon)|S_t|]$, as desired. ♣

COROLLARY 3.2. *With probability $\geq 1 - n^{-\Theta(C)}$, for an absolute constant hidden in the $\Theta(\cdot)$,*

$$\tilde{s}_t = \text{median}_z \tilde{s}_t^z \in [(1 - \epsilon)|S_t|, (1 + \epsilon)|S_t|].$$

We now turn to steps 5 and beyond, which are based on the following **Sample-Extraction** algorithm. Recall that the underlying vector is a' (the output of **StreamTransformation**) with length $|a'| = \text{poly}(n)$. As above, let $C > 0$ be a sufficiently large constant.

The algorithm, like the **ListHH** algorithm, performs sub-sampling at a logarithmic number of different rates. The first difference is that the functions g_j

used to do the sub-sampling are now $C \log(\epsilon^{-1}|a'|)$ -wise independent, and map $[C\epsilon^{-1}|a'|]$ to $[C\epsilon^{-1}|a'|]$. This will allow us to apply a theorem of Indyk [21] which relates hash function families with limited independence to ϵ -min-wise independent families, which we shall define in the analysis. The second difference is that the surviving coordinates, for a given level of sub-sampling, are further hashed into D buckets using a D -wise independent hash function. This second bucketing operation is repeated independently $E = C \log n$ times. The goal of this bucketization is to guarantee that in a sub-sampling level j for which items from S_t are extracted, all surviving items from S_t are reported with very high probability. This requires showing that the heavy items, i.e., those from $\cup_{t' \geq t} S_{t'}$ in the substreams, as well as the items from non-growing level sets, are perfectly hashed into the D buckets for some repetition $e \in E$, with high probability. Given all surviving items from S_t , which is the set of coordinates i satisfying $g_j(i) \leq C\epsilon^{-1}|a'|2^{-j}$, we can find the $i \in S_t$ which minimizes g_j . Given that g_j is from an ϵ -min-wise independent family, this i is a random element of S_t , up to a small relative error.

Sample-Extraction(\mathcal{S}', B)

1. $D \leftarrow C(\epsilon^{-4}T^4 \log^2 n)^2$, and
 $E \leftarrow C \log n$.
2. For $j \in [\log(C\epsilon^{-1}|a'|)]$, independently sample $g_j : [C\epsilon^{-1}|a'|] \rightarrow [C\epsilon^{-1}|a'|]$ from a $(C \log(\epsilon^{-1}|a'|))$ -wise independent family of hash functions.
For $e \in [E]$, independently sample $f_{j,e} : [C\epsilon^{-1}|a'|] \rightarrow [D]$ from a D -wise independent family.
3. For $j \in [\log(C\epsilon^{-1}|a'|)]$, $d \in [D]$, and $e \in [E]$, let $\mathcal{S}_{j,d,e}$ be the restriction of \mathcal{S}' to those pairs (i, x) for which $g_j(i) \leq (C\epsilon^{-1}|a'|)2^{-j}$ and $f_{j,e}(i) = d$. For $B' = O(T^4\epsilon^{-3} \log^2 n)^{2/p}$, $M_{j,d,e} = \text{HeavyHitters}(\mathcal{S}_{j,d,e}, B', n^{-C}, \epsilon/C)$.
4. For growing t , choose the largest $j = j(t)$ for which $\cup_{d,e} M_{j,d,e}$ contains $\geq 1600\epsilon^{-2}T^2 \log^2 n$ elements of S_t . If there is such a j :
 - $S''_t = S_t \cap (\cup_{d,e} M_{j(t),d,e})$.
 - Let w_t be the element of S''_t that minimizes $g_{j(t)}$.
5. Return $(w_t, (\tilde{a}')_{w(t)})$ for those t for which a $j(t)$ was found.

LEMMA 3.7. With probability $1 - n^{-\Omega(C)}$, for each growing t a pair $(w_t, a'_{w(t)})$ is returned by **Sample-Extraction**. Moreover, conditioned on returning a pair for S_t , for all $\alpha \in S_t$ we have

$$\Pr[w_t = \alpha] = (1 \pm \Theta(\varepsilon)) \cdot \frac{1}{|S_t|} \pm n^{-\Omega(C)}.$$

Proof. We let U_j be the set of coordinates for which $g_j(i) \leq (C\varepsilon^{-1}|a'|)2^{-j}$, and $a''(j)$ be the vector a restricted to coordinates in U_j . Define the random events:

- \mathcal{E}' : for all growing t and $j \in [\log(C\varepsilon^{-1}|a'|)]$,
 $|S_t \cap U_j| \leq 2|S_t|2^{-j} + O(\log n)$.
- \mathcal{F}' : for all growing t and $j \in [\log(C\varepsilon^{-1}|a'|)]$, if
 $\mathbf{E}[|S_t \cap U_j|] \geq C^2\varepsilon^{-2} \log n$,

then

$$|S_t \cap U_j| \in [(1 - \varepsilon)|S_t|2^{-j}, (1 + \varepsilon)|S_t|2^{-j}].$$

- \mathcal{H}' : all invocations of **HeavyHitters** by **Sample-Extraction** succeed.

We need the following proposition.

PROPOSITION 3.1. $\Pr[\mathcal{E}' \wedge \mathcal{F}' \wedge \mathcal{H}'] \geq 1 - n^{-\Theta(C)}$, for an absolute constant in the $\Theta(\cdot)$.

Proof. Since g_j is $C \log(\varepsilon^{-1}|a'|)$ -wise independent for sufficiently large C , one can use the following theorem:

THEOREM 3.2. (Lemma 2.3 of [6]) Let $X_i \in [0, 1]$, $1 \leq i \leq n$, be t -wise independent for $t \geq 4$ an even integer, $X = \sum_{i=1}^n X_i$, and $A > 0$. Then $\Pr[|X - \mathbf{E}[X]| \geq A] \leq 8 \left(\frac{t\mathbf{E}[X] + t^2}{A^2} \right)^{t/2}$.

We can assume $\log(\varepsilon^{-1}|a'|) = \Theta(\log n)$, as otherwise ε is so small that we can just store the entire vector a' in $\text{poly}(\varepsilon^{-1} \log n)$ bits of space. Hence, the independence of g_j is $O(C \log n)$, where the constant in the big-Oh is absolute. Suppose we set

$$A = |S_t|2^{-j} + C^2 \log n.$$

Using the fact that $\mathbf{E}[|S_t \cap U_j|] = |S_t|2^{-j}$ and Theorem 3.2,

$$\begin{aligned} \Pr[|S_t \cap U_j| > 2|S_t|2^{-j} + C^2 \log n] \\ \leq 8 \left(\frac{t\mathbf{E}[|S_t \cap U_j|] + t^2}{A^2} \right)^{t/2} \\ \leq 8 \left(\frac{O(C \log n)|S_t|2^{-j} + O(C^2 \log^2 n)}{|S_t|^2 2^{-2j} + C^4 \log^2 n} \right)^{C \cdot \Omega(\log n)}. \end{aligned}$$

Now, there are two cases: (1) $|S_t|2^{-j} \leq C^2 \log n$, and (2) $|S_t|2^{-j} > C^2 \log n$. In both cases the RHS is at most $8(1/2)^{C \cdot \Omega(\log n)}$, and so it can be bounded by $n^{-\Theta(C)}$, which upper bounds $\Pr[\neg \mathcal{E}']$.

To upper bound $\Pr[\neg \mathcal{F}']$, we set

$$A = \varepsilon \mathbf{E}[|S_t \cap U_j|] = \varepsilon |S_t|2^{-j}.$$

Then by Theorem 3.2,

$$\begin{aligned} \Pr[||S_t \cap U_j| - |S_t|2^{-j}| \geq \varepsilon |S_t|2^{-j}] \\ \leq 8 \left(\frac{O(C \log n)|S_t|2^{-j} + O(C^2 \log^2 n)}{\varepsilon^2 |S_t|^2 2^{-2j}} \right)^{C \cdot \Omega(\log n)}. \end{aligned}$$

Using the premise of event \mathcal{F}' , namely, that $|S_t|2^{-j} \geq C^2\varepsilon^{-2} \log n$, the RHS can be bounded by $n^{-\Theta(C)}$, which upper bounds $\Pr[\neg \mathcal{F}']$.

Since **HeavyHitters** is invoked $\text{poly}(\varepsilon^{-1} \log n)$ times by **Sample-Extraction** with error parameter n^{-C} , by a union bound,

$$\Pr[\mathcal{E}' \wedge \mathcal{F}' \wedge \mathcal{H}'] \geq 1 - n^{-\Theta(C)},$$

for an absolute constant in the $\Theta(\cdot)$. ♣

We condition on these three events in the remainder of the proof of Lemma 3.7.

Fix a t that is growing. Below we will show that a $j(t)$ is assigned to t in step 4 of **Sample-Extraction**. Next, we show that **Sample-Extraction** ensures that with probability $\geq 1 - n^{-\Theta(C)}$, all coordinates in $U_{j(t)} \cap S_t$ are in S_t'' . This is stronger than the guarantee of Lemma 3.6, which could only guarantee this with constant probability (or by minor modifications, $1 - \text{poly}(\varepsilon \log^{-1} n)$ probability). One obstacle is that there may be no concentration in the random variables $F_p(a''(j))$, and if they are too large, then we cannot collect the heavy hitters in the corresponding substream. In the proof of Lemma 3.6 it sufficed to bound the $F_p(a''(j))$ using Markov's inequality. Here, we further partition the streams \mathcal{S}_j into D pieces $\mathcal{S}_{j,1}, \dots, \mathcal{S}_{j,D}$. We do this independently E times, so that for all j ,

$$\forall e \in [E], \mathcal{S}_j = \cup_{d \in [D]} \mathcal{S}_{j,d,e}.$$

Let $a''(j, d, e)$ denote the restriction of the vector $a''(j)$ to coordinates that go to the d -th bucket in the e -th independent repetition.

PROPOSITION 3.2. With probability $\geq 1 - n^{-\Theta(C)}$, a $j(t)$ is assigned to t in step 4 of **Sample-Extraction** and all coordinates in $U_{j(t)} \cap S_t$ are in S_t'' , i.e.,

$$\Pr[S_t'' = U_{j(t)} \cap S_t] \geq 1 - n^{-\Theta(C)}.$$

Proof. Fix a $j \in [\log(C\varepsilon^{-1}|a'|)]$. We now bound the number of elements in a level set S_t that is not growing. By definition,

$$\eta^{pt} > \frac{\varepsilon^4 F'_p}{5T^3 \log^2 n},$$

and so

$$(3.3) \quad \eta^{p(t-1)} \geq \frac{\varepsilon^4 F'_p}{\eta^p 5T^3 \log^2 n}.$$

But by Lemma 3.3,

$$F'_p \geq F_p(a) \geq \frac{F_p(a')}{1 + \varepsilon/2}.$$

Hence, by (3.3), the number of elements in S_t can be at most

$$\frac{(1 + \varepsilon/2)\eta^p 5T^3 \log^2 n}{\varepsilon^4} = O\left(\frac{T^3 \log^2 n}{\varepsilon^4}\right).$$

Summing over all non-growing sets, the number of such items is

$$O\left(\frac{T^4 \log^2 n}{\varepsilon^4}\right).$$

Next, for a $t' \geq t$ that is growing, since event \mathcal{E}' occurs,

$$(3.4) \quad |S_{t'} \cap U_j| \leq 2|S_{t'}|2^{-j} + O(\log n).$$

Moreover, $S_{t'}$ cannot be much larger than S_t , as otherwise S_t could not be $\varepsilon/(40T)$ -contributing, as per Lemma 3.4. That is,

$$|S_t|\eta^{pt} \geq \varepsilon F_p(a')/(40T),$$

and since $t' \geq t$ all coordinates in $S_{t'}$ have absolute value at least η^{t-1} . Then necessarily,

$$|S_{t'}| \leq 40T\eta^p \varepsilon^{-1} |S_t|.$$

Combining this inequality with that of (3.4),

$$|S_{t'} \cap U_j| = O(T\varepsilon^{-1}|S_t|2^{-j} + \log n),$$

and so

$$|\cup_{t' \geq t} S_{t'} \cap U_j| = O(T^2 \varepsilon^{-1} |S_t| 2^{-j} + T \log n).$$

By Lemma 3.2,

$$|S_t| \geq \varepsilon^{-3} T^2 \log^2 n.$$

Hence, for small enough ε , there is a unique $j^* \geq 0$ for which

$$3200\varepsilon^{-2} T^2 \log^2 n \leq |S_t| 2^{-j^*} < 6400\varepsilon^{-2} T^2 \log^2 n.$$

In the remainder of the proof we restrict our attention to those j for which $j \geq j^*$. For such j ,

$$|\cup_{t' \geq t} S_{t'} \cap U_j| = O(\varepsilon^{-3} T^4 \log^2 n).$$

Now, fix an $e \in [E]$.

Since the $f_{j,e}$ are chosen from a D -wise independent family for $D = C(\varepsilon^{-4} T^4 \log^2 n)^2$, it follows that for a sufficiently large constant $C > 0$, with probability $\geq 1/2$, none of the items in $\cup_{t' \geq t} S_{t'} \cap U_j$ or in non-growing level sets go to the same bucket (i.e., agree on the function $f_{j,e}$).

We now show that conditioned on $\mathcal{E}' \wedge \mathcal{F}' \wedge \mathcal{H}'$ and assuming that the items in $\cup_{t' \geq t} S_{t'} \cap U_j$ are perfectly hashed, for each $i \in U_j \cap S_t$, the coordinate i is returned by **HeavyHitters**($S_{j,d,e}, B, n^{-C}, \varepsilon/C$).

For this, we need a bound on $F_p(a''(j, d, e))$ for each bucket $d \in [D]$ in an iteration $e \in [E]$ containing an element of S_t . Fix a $d \in [D]$. Since event \mathcal{E}' occurs, the number $y_{t'}$ of items in a growing $t' < t$ that collide in the d -th bucket is $O(|S_{t'}|2^{-j} + \log n)$.

Since S_t is $\varepsilon/(40T)$ -contributing,

$$|S_t|\eta^{pt} \geq \frac{\varepsilon F_p(a')}{40T},$$

and since

$$|S_{t'}|\eta^{p(t'-1)} \leq F_p(a'),$$

we obtain the bound

$$|S_{t'}| \leq 40T\varepsilon^{-1}\eta^p |S_t|\eta^{pt-pt'}.$$

Since $j \geq j^*$, we have

$$2^{-j}|S_t| \leq 2^{-j^*}|S_t| = O(\varepsilon^{-2} T^2 \log^2 n),$$

and so we obtain $y_{t'} = O(T^3 \varepsilon^{-3} \eta^{pt-pt'} \log^2 n)$. It follows that the contribution to $F_p(a''(j, d, e))$ is at most

$$\eta^{pt'} y_{t'} = O(T^3 \varepsilon^{-3} \eta^{pt} \log^2 n).$$

Hence, the total contribution from all $t' < t$ to $F_p(a''(j, d, e))$ is at most

$$O(T^4 \varepsilon^{-3} \eta^{pt} \log^2 n).$$

But the contribution of the single item in S_t in this bucket to $F_p(a''(j, d, e))$ is at least η^{pt-p} . Since **HeavyHitters** is invoked with parameter $B' = O(T^4 \varepsilon^{-3} \log^2 n)^{2/p}$, the single item in S_t in this bucket will be returned, using the fact that event \mathcal{H}' occurs, and thus **HeavyHitters** succeeds.

Since $E = C \log n$, it follows that with probability $\geq 1 - n^{-\Theta(C)}$, there is a value of e for which the

items in $\cup_{t' \geq t} S_{t'} \cap U_j$ are perfectly hashed, and hence with this probability

$$\cup_{d,e} M_{j,d,e} = S_t \cap U_j,$$

for any $j \geq j^*$. Now, notice that

$$\mathbf{E}[|S_t \cap U_{j^*}|] = |S_t| 2^{-j^*} \geq 3200 \varepsilon^{-2} T^2 \log^2 n.$$

Hence, since event \mathcal{F}' occurs,

$$|S_t \cap U_{j^*}| \geq (1 - \varepsilon) \mathbf{E}[|S_t \cap U_{j^*}|] \geq 1600 \varepsilon^{-2} T^2 \log^2 n.$$

This implies that in step 4 of **Sample-Extraction** the value j^* satisfies the criterion that $\cup_{d,e} M_{j^*,d,e}$ contains at least $1600 \varepsilon^{-2} T^2 \log^2 n$ elements of S_t . Since **Sample-Extraction** sets $j(t)$ to be the largest such j , the value t will be assigned a value $j(t) \geq j^*$.

The above implies that

$$\Pr[S_t'' = U_{j(t)} \cap S_t] \geq 1 - n^{-\Theta(C)}.$$

This concludes the proof of the proposition. ♣

Now if $S_t'' = U_{j(t)} \cap S_t$, then all coordinates i in S_t for which $g_{j(t)}(i) \leq (C \varepsilon^{-1} |a'|) 2^{-j(t)}$ are in S_t'' , and there are at least $1600 \varepsilon^{-2} T^2 \log^2 n \geq 1$ of them. In particular, we can find the coordinate i for which

$$g_{j(t)}(i) = \min_{i' \in S_t} g_{j(t)}(i').$$

We need the following definition.

DEFINITION 3.2. A family of functions $\mathcal{H} \subseteq [N] \rightarrow [N]$ is called (ε, s) -min-wise independent if for any $X \subseteq N$ with $|X| \leq s$, and any $x \in [N] \setminus X$, we have

$$\Pr_{h \in \mathcal{H}}[h(x) < \min_{X} h(X)] = (1 \pm \varepsilon) \cdot \frac{1}{|X| + 1}.$$

We need the following theorem of Indyk.

THEOREM 3.3. ([21]) There exist constants $c, c' > 1$ such that for any $\varepsilon > 0$ and $s \leq \varepsilon N/c$, any $(c' \log 1/\varepsilon)$ -wise independent family of functions is (ε, s) -min-wise independent.

Here we apply the theorem with $N = C \varepsilon^{-1} |a'|$ and $s = |a'|$. Our family of hash functions is also $(C \log \varepsilon^{-1} |a'|)$ -wise independent. Hence, for $C > 0$ a sufficiently large constant, we have for all $i \in S_t$,

$$\Pr[g_j(i) = \min_{i' \in S_t} g_j(i')] = (1 \pm \Theta(\varepsilon)) \cdot \frac{1}{|S_t|}.$$

This finishes the proof of Lemma 3.7. ♣

THEOREM 3.4. For any $p \in [0, 2]$, the probability that L_p -Sampler(S', p) returns $(i, (1 \pm \Theta(\varepsilon))a_i)$ is

$$(1 \pm \varepsilon) \frac{|a_i|^p}{F_p(a)} \pm n^{-C}$$

for an arbitrarily large constant $C > 0$. The space complexity is $\text{poly}(\varepsilon^{-1} \log n)$ bits. Ignoring the time of **StreamTransformation**, which can be performed without looking at the data stream, the update time is $\text{poly}(\varepsilon^{-1} \log n)$.

Proof. For $p \in (0, 2]$, by Corollary 3.2 and Lemma 3.7, with probability at least $1 - n^{-\Theta(C)}$,

$$\tilde{s}_t \in [(1 - \varepsilon)|S_t|, (1 + \varepsilon)|S_t|]$$

and for each growing t a sample w_t is returned with probability $(1 \pm \Theta(\varepsilon))|S_t|^{-1} \pm n^{-\Theta(C)}$.

We also condition on the event that **HeavyHitters** succeeds in returning all coordinates not in growing S_t in step 1; this only adds an additional n^{-C} to the error probability. Notice that for such coordinates i , if i occurs in S_t , then we have

$$|a'_i|^p \geq \eta^{(t-1)p} \geq \frac{\varepsilon^4 F_p'}{\eta^p 5 T^3 \log^2 n},$$

where the latter inequality follows by definition. Since $F_p' \geq F_p$, it follows that i is $(\varepsilon^4 / (\eta^p 5 T^3 \log^2 n))$ -contributing, and so the **HeavyHitters** algorithm with paramater $A = (5 \eta^p T^3 \varepsilon^{-4} \log^2 n)^{2/p}$ will report i .

It now follows that in Step 7,

$$\Pr[u = i] = (1 \pm \Theta(\varepsilon)) \frac{|a_i|^p}{F_p} \pm n^{-\Theta(C)}.$$

By Lemma 3.3, the total contribution of injected coordinates to $F_p(a')$ is $O(\varepsilon) F_p(a')$. Hence, in step 8, the probability that in $C \log n$ repetitions all samples are injected coordinates is at most $n^{-\Theta(C)}$. The statement of the theorem now follows by adjusting C and ε by constant factors. It is also easy to see that the space of the overall algorithm is $\text{poly}(\varepsilon^{-1} \log n)$ bits. The update time is dominated by that of the **HeavyHitters** subroutines, and is $\text{poly}(\varepsilon^{-1} \log n)$. We show how to remove the assumption that the $|a'_i|$ are away from their boundaries in Section 3.2.1. Finally, we show how to build an augmented L_0 -sampler in Section 3.2.2. ♣

3.2.1 Removing the assumption The above algorithm holds under the assumption that for all i , if $i \in S_t$, then $|a'_i| \leq \eta^t / (1 + \beta \varepsilon)$, for a small constant $\beta > 0$, allowing **HeavyHitters** to accurately classify the

coordinates that it finds. This assumption is easy to remove given an additional pass, since one can compute the $|a_i|$ exactly and then perform classification. To achieve a 1-pass algorithm, we make the following observation.

We note that the guarantees of our algorithm do not change by more than a factor of $\eta^p = 1 + \Theta(\varepsilon)$ provided the classification of coordinates i into the level sets S_t is *consistent*. That is, if coordinate i is returned by multiple **HeavyHitters** invocations, then each invocation must classify it into the same level set S_t . Notice that consistency is easily enforced since the total number of items returned, across all **HeavyHitters** invocations, is $\text{poly}(\varepsilon^{-1} \log n)$, and hence the algorithm can simply remember a table indicating how previous coordinates returned were classified.

This effectively takes the underlying vector a' , and multiplies some coordinates by a value of at most η (those that are misclassified into their neighboring level set). Sampling from the resulting vector is equivalent to sampling from a' , up to a factor of η^p . We need consistency for estimating the set sizes $|S_t|$, since we do not want to count one coordinate towards multiple S_t . Notice that unlike the algorithm of Indyk and the second author [25], we do not have to worry about some level sets no longer contributing because of misclassification. This is because, as argued earlier, all injected coordinates are not near their boundaries, by definition, so they will be correctly classified, and so all growing sets will still be $\text{poly}(\varepsilon \log^{-1} n)$ -contributing (items from the non-growing sets do not undergo classification).

3.2.2 An Augmented L_0 -Sampler In this section we can work directly on the vector a , without introducing injected coordinates as needed for L_p -sampling, $p > 0$.

THEOREM 3.5. *There exists a 1-pass algorithm L_0 -Sampler which, given a stream \mathcal{S} of an underlying vector a , outputs a random non-zero coordinate i together with the value a_i , such that for all non-zero a_i , the probability that L_0 -Sampler outputs (i, a_i) is*

$$(1 \pm \varepsilon) \cdot \frac{1}{\|a\|_0} \pm n^{-C}$$

for an arbitrarily large constant $C > 0$. The space complexity is $\text{poly}(\varepsilon^{-1} \log n)$ bits, and the update time is $\text{poly}(\varepsilon^{-1} \log n)$.

Proof. Let $C > 0$ be a sufficiently large constant. We first assume that $\|a\|_0 \geq C$. W.l.o.g., $C\varepsilon^{-1}n$ is

a power of 2. We choose $C \log(\varepsilon^{-1}n)$ independent hash functions $h_j : [C\varepsilon^{-1}n] \rightarrow [C\varepsilon^{-1}n]$ from a family of $C \log(\varepsilon^{-1}n)$ -wise independent hash functions. For $j = 0, \dots, \log(C\varepsilon^{-1}n)$, we say that a coordinate i of a survives with respect to h_j if $h_j(i) \leq 2^{-j}(C\varepsilon^{-1}n)$. Let \mathcal{S}_j be the restriction of updates in \mathcal{S} to coordinates that survive with respect to h_j . Let $a(j)$ denote the restriction of the underlying vector a to coordinates that survive with respect to h_j . On each \mathcal{S}_j , we run a $\text{poly}(\log n)$ -space L_0 -estimation algorithm to estimate $a(j)$ up to a factor of $(1 \pm 1/3)$ in the general turnstile model with error probability $n^{-\Theta(C)}$, e.g., the algorithm of [28]. Denote the resulting estimate E_j . Further, for each \mathcal{S}_j choose $C \log n$ independent C^3 -wise independent hash functions $\psi_{j,r} : [C\varepsilon^{-1}n] \rightarrow [C^3]$, for $r \in [C \log n]$. We maintain the counters:

$$c_{j,r,d} = \sum_{\ell \text{ s.t. } h_j(\ell) \leq 2^{-j}C\varepsilon^{-1}n \text{ and } \psi_{j,r}(\ell)=d} a_\ell.$$

We repeat the entire procedure in the previous paragraph $C \log n$ times in parallel. We find some repetition for which there is a j for which $E_j \in [C/16, C]$. If there is no such j , then we output the symbol **FAIL**. Otherwise, we choose the first repetition for which such a j was found.

From the counters $c_{j,r,d}$, one can recover the list L_j of all non-zero coordinates i in $a(j)$, together with their values a_i . This follows since for each fixed value of r , the at most $\frac{3}{2} \cdot E_j = \frac{3C}{2}$ survivors with respect to h_j are perfectly hashed with probability $\geq 2/3$ (for large enough C). Hence, with probability $\geq 1 - n^{-\Theta(C)}$, for all survivors i we have,

$$a_i = \text{median}_r c_{j,r,\psi_{j,r}(i)}.$$

If the list L_j does not contain at least $(3/2)C/16$ coordinates, then we output **FAIL**. Else, we find the coordinate i in L_j which minimizes h_j , and output (i, a_i) .

We can assume that all invocations of the L_0 -estimation algorithm succeed. Now, assuming that $\|a\|_0 \geq C$, for each independent repetition, we claim that there exists a value of j for which

$$E_j \in \left[\frac{C}{16}, C \right]$$

with constant probability. To show this, by the definition of E_j , it suffices to show that there exists a value of j for which with constant probability,

$$\|a(j)\|_0 \in \left[\frac{3}{2} \cdot \frac{C}{16}, \frac{3}{4} \cdot C \right] = \left[\frac{3C}{32}, \frac{3C}{4} \right]$$

To see the latter, consider the unique value of j for which

$$\frac{C}{6} \leq 2^{-j} \|a\|_0 < \frac{C}{3}.$$

For each non-zero coordinate i , let X_i be an indicator variable which is one iff i survives with respect to h_j . Let $X = \sum_i X_i$. Then $\mathbf{E}[X] = 2^{-j} \|a\|_0$, and by pairwise-independence, $\mathbf{Var}[X] \leq \mathbf{E}[X]$. By Chebyshev's inequality,

$$\Pr \left[|X - \mathbf{E}[X]| \geq \frac{\mathbf{E}[X]}{3} \right] \leq \frac{9\mathbf{Var}[X]}{\mathbf{E}^2[X]} \leq \frac{54}{C} < \frac{1}{3},$$

where the last inequality follows for large enough C . This implies

$$\|a(j)\|_0 \in \left[\frac{3C}{32}, \frac{3C}{4} \right].$$

Hence, with probability $1 - n^{-\Omega(C)}$, in some repetition we will find a j for which $E_j \in [C/16, C]$.

Finally, we appeal to Theorem 3.3. We apply the theorem with $N = C\varepsilon^{-1}n$ and $s = n$. Our family of hash functions is also $C \log \varepsilon^{-1}n$ -wise independent. Hence, for $C > 0$ a sufficiently large constant, we have for all i such that $a_i \neq 0$,

$$\Pr \left[h_j(i) = \min_{i' \text{ s.t. } h_j(i') \neq 0} h_j(i') \right] = (1 \pm \varepsilon) \cdot \frac{1}{\|a\|_0}.$$

It remains to handle the case $\|a\|_0 < C$. We will in fact show how to solve the case $\|a\|_0 \leq 2C$. The idea is just to use the perfect hashing data structure described above. Namely, choose $C \log n$ independent C^3 -wise independent hash functions $\psi_r : [n] \rightarrow [C^3]$, for $r \in [C \log n]$. We maintain the counters:

$$c_{r,d} = \sum_{\ell \text{ s.t. } \psi_r(\ell)=d} a_\ell.$$

As described above, with probability $\geq 1 - n^{-\Theta(C)}$, for all i we have,

$$a_i = \text{median}_r c_{r,\psi_r(i)}.$$

Hence, we can recover the vector a in this case, after which L_0 -sampling is trivial.

In parallel we run a $\text{poly}(\log n)$ -space L_0 -estimation algorithm which can distinguish between the cases (1) $\|a\|_0 \leq C$ and (2) $\|a\|_0 \geq 2C$ with probability $\geq 1 - n^{-\Theta(C)}$. In the former case we use the output of the sampling algorithm based on perfect hashing just described. In the latter case we use the output of the sampling algorithm described previously. If $C < \|a\|_0 < 2C$, we can use either sampling algorithm.

It can be easily checked that our overall algorithm is 1-pass, the space is $\text{poly}(\varepsilon^{-1} \log n)$ bits, and the time is $\text{poly}(\varepsilon^{-1} \log n)$. ♣

4 Extensions

4.1 $O(\log n)$ -Pass L_p -Sampler: We use a binary-search-inspired scheme. Let F_p -Estimation be the optimal-space algorithm for estimating the F_p -value of a vector due to Kane *et al* [28]. Given a stream \mathcal{S} , we think of it as two interleaved streams \mathcal{S}_L and \mathcal{S}_U , where \mathcal{S}_L consists of the subsequence of updates to the first $n/2$ coordinates of the vector a , and \mathcal{S}_U the subsequence consisting of the updates to the remaining coordinates. Denote the vector consisting of the lower $n/2$ coordinates of a by a^L , and the vector consisting of the upper $n/2$ coordinates of a by a^U .

We run F_p -Estimation($\mathcal{S}_L, a^L, \eta, \delta$) and F_p -Estimation($\mathcal{S}_U, a^U, \eta, \delta$) independently and in parallel with error parameter $\eta = \Theta(1/\log n)$ and failure probability $\delta = n^{-C}$. Assuming both algorithms succeed, we obtain numbers L, U with

$$L \in [(1 - \eta) \|a^L\|_p^p, (1 + \eta) \|a^L\|_p^p],$$

$$U \in [(1 - \eta) \|a^U\|_p^p, (1 + \eta) \|a^U\|_p^p].$$

We then recurse on the lower $n/2$ coordinates with probability $L/(L+U)$, and recurse on the upper $n/2$ coordinates with probability $U/(L+U)$. After $\log n$ recursive steps, an individual coordinate $i \in [n]$ will be sampled. Assuming F_p -Estimation never fails, fixing any $i \in [n]$, the probability that it is sampled is a telescoping product, putting it in the interval

$$\left[\frac{(1 - \eta)^{\log n} |a_i|^p}{F_p}, \frac{(1 + \eta)^{\log n} |a_i|^p}{F_p} \right],$$

which is contained in the interval

$$\left[\frac{(1 - 1/4) |a_i|^p}{F_p}, \frac{(1 + 1/4) |a_i|^p}{F_p} \right]$$

for sufficiently small η . We can then use rejection sampling to turn this into an exact sampler. We then repeat the procedure $C \log n$ times, and only output FAIL if in every repetition we do not find a sample. The total space is $O(\log^5 n)$ since the space of F_p -Estimation is $O(\eta^{-2} \log n \log 1/\delta)$ bits, where η is the relative error and δ the failure probability. In our application $\eta = \Theta(1/\log n)$ and $\delta = n^{-C}$ for a constant $C > 0$, so this space is $O(\log^4 n)$, and we incur an extra $O(\log n)$ due to the independent repetitions.

$O(\log n)$ -Pass- L_p -Sampler(Stream \mathcal{S}, ϵ):

1. Initialize $a = [n]$, $\delta = O(n^{-C})$, $\eta = \Theta(\frac{1}{\log n})$, and $\beta = 1$.
2. In the first pass, compute $\tilde{F}_p(a) = F_p\text{-Estimation}(\mathcal{S}, a, \frac{1}{36}, \delta)/(1 - 1/36)$.
3. If $\tilde{F}_p(a) = 0$, output FAIL.
4. For $j = 1, 2, \dots, \log_2 n$, in the j -th pass do:
 - (a) Let a_L be the first $|a|/2$ items of a , and let $a_U = a \setminus a_L$.
 - (b) Let \mathcal{S}_L and \mathcal{S}_U be the interleaved streams consisting of the subsequence of updates to a_L and a_U respectively.
 - (c) $L \leftarrow F_p\text{-Estimation}(\mathcal{S}_L, a_L, \eta, \delta)$, $U \leftarrow F_p\text{-Estimation}(\mathcal{S}_U, a_U, \eta, \delta)$
 - (d) If $L = U = 0$, output FAIL.
 - (e) With probability $\frac{L}{L+U}$, assign $a \leftarrow a_L$, $\mathcal{S} \leftarrow \mathcal{S}_L$, and $\beta \leftarrow \beta \cdot \frac{L}{L+U}$, else assign $a \leftarrow a_U$, $\mathcal{S} \leftarrow \mathcal{S}_U$ and $\beta \leftarrow \beta \cdot \frac{U}{L+U}$.
5. Let i be such that $a = \{i\}$. Compute a_i in an extra pass. Let $q = |a_i|^p / \tilde{F}_p(a)$.
6. If $|\beta - q| > \frac{q}{3}$, then output FAIL.
7. Compute, in parallel, $V = F_p\text{-Estimation}(\mathcal{S}, a, \Theta(1), \delta)$ such that $F_p(a) \leq V \leq 2F_p(a)$.
8. With probability $\frac{|a_i|^p}{V\beta}$, let the sample be (i, a_i) .
9. Repeat steps 2 through 8 independently $C \log n$ times. If in every repetition, nothing is chosen in step 8 as the sample, output FAIL. Else, output the sample from the first repetition for which one was chosen.

The Proof. Let SuccessfulSubroutines be the event that all invocations of $F_p\text{-Estimation}$ in the above algorithm succeed. By our choice of δ and a union bound, with probability at least $1 - O(n^{-C})$, SuccessfulSubroutines occurs.

Let GoodNormEstimation be the event that

$$\tilde{F}_p \leq F_p \leq (1 + \frac{1}{12})\tilde{F}_p.$$

With probability at least $1 - n^{-C}$, we have that

$$\tilde{F}_p \geq \frac{(1 - \frac{1}{36}) F_p}{(1 - \frac{1}{36})} = F_p,$$

as well as

$$\tilde{F}_p \leq \frac{(1 + \frac{1}{36}) F_p}{(1 - \frac{1}{36})} \leq \left(1 + \frac{1}{12}\right) F_p.$$

LEMMA 4.1. Suppose event SuccessfulSubroutines occurs. Then for all i , the probability that coordinate i is chosen in step 5 is $(1 \pm 1/4)|a_i|^p / F_p(a)$.

Proof. Fix an $i \in [n]$. Then there is a unique sequence of assignments $a^0 = [n], a^1, a^2, \dots, a^{\log_2 n} = \{i\}$ to the loop variable a , for which $a = a^j$ after iteration j of step 4, that cause coordinate i to be chosen in step 5. For $j \in \{0, 1, \dots, \log_2 n\}$, let \mathcal{E}_j be the event that $a = a^j$ after iteration j of step 4. Then,

$$\begin{aligned} \Pr[i \text{ chosen in step 4}] &= \bigcap_{j=0}^{\log_2 n} \Pr[\mathcal{E}_j \mid \mathcal{E}_1, \dots, \mathcal{E}_{j-1}] \\ &= \bigcap_{j=1}^{\log_2 n} \Pr[\mathcal{E}_j \mid \mathcal{E}_{j-1}]. \end{aligned}$$

For any j ,

$$\begin{aligned} \Pr[\mathcal{E}_j \mid \mathcal{E}_{j-1}] &= \frac{(1 \pm \eta)F_p(a^j)}{(1 \pm \eta)F_p(a_L^{j-1}) + (1 \pm \eta)F_p(a_U^{j-1})} \\ &= (1 \pm 3\eta) \frac{F_p(a^j)}{F_p(a^{j-1})}, \end{aligned}$$

where the final equality follows for $\eta \leq 1/2$. Hence,

$$\begin{aligned} \Pr[i \text{ chosen in step 4}] &= (1 \pm 3\eta)^{\log_2 n} \prod_{j=1}^{\log_2 n} \frac{F_p(a^j)}{F_p(a^{j-1})} \\ &= (1 \pm 3\eta)^{\log_2 n} \frac{|a_i|^p}{F_p(a)}. \end{aligned}$$

Using the well known fact that $(1 + t) \leq e^t$ for all $t \in \mathbb{R}$, we have

$$(1 + 3\eta)^{\log_2 n} \leq e^{3\eta \log_2 n} = e^{1/8} \leq 1 + 1/4,$$

where we choose η so that $3\eta \log_2 n = 1/8$, and where the last inequality follows by a Taylor expansion. For the other direction, we appeal to Proposition B.3, part 2, of [30], which states that for all $t, n \in \mathbb{R}$ such that $r \geq 1$ and $|t| \leq r$,

$$e^t(1 - t^2/r) \leq (1 + t/r)^r.$$

Then

$$(1 - 3\eta)^{\log_2 n} = (1 - (3\eta \log_2 n) / \log_2 n)^{\log_2 n}.$$

Thus, setting $t = -3\eta \log_2 n = -1/8$, and $r = \log_2 n$, applying this inequality we have,

$$(1 - 3\eta)^{\log_2 n} \geq e^{-1/8} (1 - (1/8)^2 / \log_2 n) \geq (1 - 1/8)(1 - (1/8)^2) \geq (1 - 1/4).$$

Thus,

$$\beta = \Pr[i \text{ chosen in step 5}] = (1 \pm 1/4)|a_i|^p / F_p(a).$$

♣

Proof. [of Theorem (1.3)] Event SuccessfulSubroutines occurs with probability at least $1 - O(n^{-C})$. In this case, by Lemma 4.1, $O(\log n)$ -Pass- L_p -Sampler never outputs FAIL in step 4d, and for each $i \in [n]$, the probability that coordinate i is chosen in step 5 is $(1 \pm 1/4)|a_i|^p / F_p$.

Event GoodNormEstimation occurs with probability at least $1 - \delta$. We condition on both SuccessfulSubroutines and GoodNormEstimation occurring, which, by a union bound and our choice of $\delta = O(n^{-C})$, happens with probability at least $1 - n^{-C}$.

Since GoodNormEstimation occurs and, w.l.o.g., $F_p(a) > 0$, the algorithm does not output FAIL in step 3.

Notice that $q = |a_i|^p / \tilde{F}_p$, and we have that

$$\frac{|a_i|^p}{F_p(a)} \leq q \leq (1 + \frac{1}{12}) \cdot \frac{|a_i|^p}{F_p(a)}.$$

Hence,

$$|\beta - q| \leq \left(\frac{1}{4} + \frac{1}{12} \right) \frac{|a_i|^p}{F_p(a)} \leq \frac{q}{3}.$$

So the algorithm does not output FAIL in step 6.

The probability that coordinate i is sampled in Step 8 is thus $\beta \cdot \frac{|a_i|^p}{V\beta} = \frac{|a_i|^p}{V}$. Hence the probability some coordinate in Step 8 is sampled is

$$\sum_i \frac{|a_i|^p}{V} \geq \frac{1}{2}.$$

Otherwise, we output FAIL. By repeating steps 2 through 8 a total of $C \log n$ times, we amplify the probability to $1 - n^{-\Omega(C)}$.

For the efficiency, the number of passes is always $O(\log n)$. The space used is dominated by that of F_p -Estimation on a set of size at most n with parameters η' and δ . The space is $O(\log^5 n)$ since the space of F_p -Estimation is $O(\eta^{-2} \log n \log 1/\delta)$ bits, where η is the relative error and δ the failure probability. In our application $\eta = \Theta(1/\log n)$ and $\delta = n^{-C}$ for a constant $C > 0$, so F_p -Estimation consumes $O(\log^4 n)$ bits of space. Hence, across all $C \log n$ repetitions, we use $O(\log^5 n)$ bits of space. ♣

4.2 F_k -Estimation from Sampling: For $k > 2$, the following is our F_k -estimation algorithm, which succeeds with probability at least $3/4$. To amplify the success probability to $\geq 1 - n^{-C}$, repeat the algorithm $O(C \log n)$ times and take the median of the outputs. There are two variants of it: (1) a 1-pass $n^{1-2/k} \text{poly}(\varepsilon^{-1} \log n)$ -space algorithm, and (2) an $O(\log n)$ -pass algorithm with $O(n^{1-2/k} k^2 \varepsilon^{-2} \log^5 n)$ bits of space, depending on whether we use a 1-pass or an $O(\log n)$ -pass L_2 -sampler.

F_k -Estimation:

1. Initialize $\epsilon'' = \epsilon/(4k)$, and $T = O(n^{1-2/k})/(\epsilon'')^2$.
2. Run an ϵ'' -relative-error augmented L_2 -Sampler algorithm $4T$ times in parallel, and let the first T output frequencies (possibly negative) be $a_{i_1}, a_{i_2}, \dots, a_{i_T}$. If more than $3T$ of the outputs of L_2 -Sampler are FAIL, then output FAIL.
3. In parallel, run F_2 -Estimation($[n], \epsilon'', 1/8$) of $[1, 28]$. Denote the output by \tilde{F}_2 .
4. Output $\frac{\tilde{F}_2}{T} \cdot \sum_{j=1}^T |a_{i_j}|^{k-2}$.

In what follows we will assume that the input parameter $\epsilon > 16/n$, as otherwise we can compute F_k exactly in $O(\log(n)/\epsilon)$ bits of space by keeping a counter for each coordinate.

THEOREM 4.1. *For any $k > 2$ and $\epsilon < 1$, instantiating F_k -Estimation with our 1-pass L_2 -Sampler algorithm results in a $(1 \pm \epsilon)$ -approximation to F_k with probability at least $3/4$. The space complexity is $n^{1-2/k} \cdot \text{poly}(\varepsilon^{-1} \log n)$ bits.*

Proof. We condition on the event \mathcal{E} that F_2 -Estimation succeeds, which occurs with probability at least $7/8$. We can thus assume that $F_k \neq 0$, since if $F_k = 0$ we will have $\tilde{F}_2 = 0$ and we will correctly output 0 in step 4. In the remainder of the proof, assume that $F_k \geq 1$.

We also condition on the event \mathcal{F} that at most $3T$ of the outputs of L_2 -Sampler are FAIL. By Theorem 1.1 and a Chernoff bound, event \mathcal{F} occurs with probability $1 - e^{-\Omega(T)}$.

For $i \in [n]$, let q_i be the probability that coordinate i is returned by an invocation of L_2 -Sampler, given that L_2 -Sampler does not output FAIL. By Theorem 1.1, $q_i = (1 \pm \epsilon'')|a_i|^2 / F_2$. So for any

$j \in [T]$, we have

$$\begin{aligned}\mathbf{E}[|a_{i_j}|^{k-2}] &= \sum_{i=1}^n q_i |a_i|^{k-2} (1 \pm \varepsilon'')^{k-2} \\ &= \sum_{i=1}^n (1 \pm \varepsilon'')^{k-1} \frac{|a_i|^2}{F_2} \cdot |a_i|^{k-2} \\ &= (1 \pm \varepsilon'')^{k-1} \frac{F_k}{F_2}.\end{aligned}$$

Let $G_j = |a_{i_j}|^{k-2}$, and $G = \frac{\tilde{F}_2}{T} \cdot \sum_{j=1}^T G_j$. Then $\mathbf{E}[G_j] = (1 \pm \varepsilon'')^{k-1} \frac{F_k}{F_2}$. Thus, since event \mathcal{E} occurs,

$$\begin{aligned}\mathbf{E}[G] &= T(1 \pm \varepsilon'')^{k-1} (1 \pm \varepsilon'') \frac{F_k F_2}{T F_2} \\ &= (1 \pm \varepsilon/2) F_k,\end{aligned}$$

for sufficiently small ε . By independence of the G_j and the fact that $\tilde{F}_2 \leq 2F_2$,

$$\begin{aligned}\mathbf{Var}[G] &\leq \frac{4F_2^2}{T^2} \sum_{j=1}^T \mathbf{Var}[G_j] \\ &= \frac{4F_2^2}{T^2} \cdot T \sum_{i=1}^n q_i |a_i|^{2k-4} (1 \pm \varepsilon'')^{2k-4} \\ &\leq \frac{4F_2^2 e^{\varepsilon(2k-4)/(4k)}}{T} \sum_{i=1}^n \left(\frac{2|a_i|^2}{F_2} \right) |a_i|^{2k-4} \\ &= O\left(\frac{F_2 F_{2k-2}}{T} \right).\end{aligned}$$

To bound $F_2 F_{2k-2}$, we use Hölder's inequality in the same way as in previous work [1, 11, 25]. Namely,

$$\sum_{i=1}^n |a_i b_i| \leq \left(\sum_{i=1}^n |a_i|^p \right)^{1/p} \left(\sum_{i=1}^n |b_i|^q \right)^{1/q}$$

for any reals p, q with $p, q > 1$ and $1/p + 1/q = 1$. Taking a_i in this inequality to be our a_i^2 , and taking $b_i = 1, p = k/2, q = k/(k-2)$, we have $F_2 \leq F_k^{2/k} n^{1-2/k}$. Moreover, $F_{2k-2}^{1/(2k-2)} \leq F_k^{1/k}$ using the fact that the L_k -norms of a vector are non-increasing in k , and $k \leq 2k-2$ for $k \geq 2$. So $F_{2k-2} \leq F_k^{2-2/k}$. Taking the product of the inequalities, $F_2 F_{2k-2} \leq n^{1-2/k} F_k^2$.

Thus, by our choice of T , $\mathbf{Var}[G] = O(\varepsilon^2 \mathbf{E}[G])$. It follows by Chebyshev's inequality that

$$\Pr[|G - \mathbf{E}[G]| > \frac{\varepsilon}{4} \mathbf{E}[G]] \leq 1/16,$$

for an appropriate choice of constant in the big-Oh defining T . In this case, G is at least $(1 -$

$\varepsilon/4)(1 - \varepsilon/2)F_k \geq (1 - \varepsilon)F_k$. Moreover, G is at most $(1 + \varepsilon/4)(1 + \varepsilon/2)F_k \leq (1 + \varepsilon)F_k$ for sufficiently small ε .

It follows that with probability at least

$$7/8 - e^{-\Omega(t)} - 1/16 \geq 3/4,$$

the output of the algorithm is a $(1 \pm \varepsilon)$ -approximation to F_k . The space complexity is dominated by step 2 and is $n^{1-2/k} \cdot \text{poly}(\varepsilon^{-1} \log n)$. ♣

We can also instantiate F_k -Estimation with the $O(\log n)$ -pass augmented L_2 -Sampler algorithm of Theorem 1.3. Correctness remains the same since the sampler of Theorem 1.3 is exact, whereas the proof of Theorem 4.1 only needs a sampler with relative error. The difference, though, is in the complexity analysis. The space is dominated by step 2 of the algorithm. Using Theorem 1.3, this gives $O(n^{1-2/k} k^2 \varepsilon^{-2} \log^5 n)$ bits. We note that the probability of error of L_2 -sampler is at most n^{-C} , and by increasing the space of F_2 -Estimation in step 3 by a logarithmic factor, that step also fails with probability at most n^{-C} . Hence, we obtain an $O(\log n)$ -pass $(1 \pm \varepsilon)$ -approximation algorithm with $O(n^{1-2/k} k^2 \varepsilon^{-2} \log^5 n)$ bits of space with $1/\text{poly}(n)$ probability of error. To achieve $1/\text{poly}(n)$ probability of error, the previous algorithm of [7] (improving $\text{poly}(\varepsilon^{-1} \log n)$ factors of [25]), needs $O(k^2 \varepsilon^{-2-2/k} n^{1-2/k} \log^4 n)$ bits. Thus, we improve the complexity by an $\varepsilon^{-2/k}/\log^2 n$ factor, which is interesting for small ε . We need to divide by $\log^2 n$ since our computed space was per each of the $O(\log n)$ passes. This improvement is interesting in the two-party communication setting, where the communication improvement is the same, but allowing more rounds is more natural.

Acknowledgments: We would like to thank T.S. Jayram and Jelani Nelson for helpful comments.

References

- [1] N. Alon, Y. Matias, and M. Szegedy, *The space complexity of approximating the frequency moments*, In *STOC*, 1996, pp. 20–29.
- [2] A. Andoni, K. DoBa, P. Indyk, and D. Woodruff, *Efficient sketches for earth-mover distances with applications* In *FOCS*, 2009.
- [3] A. Andoni, K. Do Ba, and P. Indyk, *Block heavy hitters*, In *MIT-CSAIL-TR-2008-024*, 2008.
- [4] A. Andoni, P. Indyk, and R. Krauthgamer, *Overcoming the L_1 non-embeddability barrier: algorithms for product metrics*, In *SODA*, 2009, pp. 865–874.

- [5] Z. Bar-Yossef, T.S. Jayram, R. Kumar, and D. Sivakumar, *An Information Statistics Approach to Data Stream and Communication Complexity*, In *FOCS*, 2002, pp. 209–218.
- [6] M. Bellare and J. Rompel, *Randomness-efficient oblivious sampling*, In *FOCS*, 1994, pp. 276–287.
- [7] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha, *Simpler algorithm for estimating frequency moments of data streams*, In *SODA*, 2006, pp. 708–713.
- [8] V. Braverman and R. Ostrovsky, *Measuring independence of datasets*, *CoRR*, abs/0903.0034, 2009.
- [9] M. Charikar, K. Chen, and M. Farach-Colton, *Finding frequent items in data streams*, In *Theor. Comput. Sci.*, 312(1):3–15, 2004.
- [10] K. L. Clarkson and D. P. Woodruff, *Numerical linear algebra in the streaming model*, In *STOC*, 2009, pp. 205–214.
- [11] D. Coppersmith and R. Kumar, *An improved data stream algorithm for frequency moments*, In *SODA*, 2004, pp. 151–156.
- [12] G. Cormode and S. Muthukrishnan, *An improved data stream summary: the count-min sketch and its applications*, *J. Algorithms*, 55(1):58–75, 2005.
- [13] G. Cormode and S. Muthukrishnan, *Space efficient mining of multigraph streams*, In *PODS*, 2005, pp. 271–282.
- [14] G. Cormode, S. Muthukrishnan, and I. Rozenbaum, *Summarizing and mining inverse distributions on data streams via dynamic inverse sampling*, In *VLDB*, 2005, pp. 25–36.
- [15] D. Feldman, M. Monemizadeh, C. Sohler, and D. Woodruff, *Coresets and sketches for high dimensional subspace problems*, In *SODA*, 2010.
- [16] P. Flajolet and G. N. Martin, *Probabilistic counting algorithms for data base applications*, *Journal of Computer and System Sciences*, 31:182–209, 1985.
- [17] G. Frahling, P. Indyk, and C. Sohler, *Sampling in dynamic data streams and applications*, In *SoCG*, 2005, pp. 142–149.
- [18] E. Friedgut, *Hypergraphs, Entropy, and Inequalities*, *The American Mathematical Monthly*, 111(9):749–760, 2004.
- [19] S. Ganguly, A. N. Singh, and S. Shankar, *Finding frequent items over general update streams*, In *SS-DBM*, 2008.
- [20] N. J.A. Harvey, J. Nelson, and K. Onak, *Sketching and streaming entropy via approximation theory*, In *FOCS*, 2008, pp. 489–498.
- [21] P. Indyk, *A small approximately min-wise independent family of hash functions*, In *SODA*, 1999, pp. 454–456.
- [22] P. Indyk, *Stable distributions, pseudorandom generators, embeddings, and data stream computation*, *J. ACM*, 53(3):307–323, 2006.
- [23] P. Indyk, *Sketching, streaming and sublinear-space algorithms*, Graduate course notes available at <http://stellar.mit.edu/S/course/6/fa07/6>. 895/, 2007.
- [24] P. Indyk and A. McGregor, *Declaring independence via the sketching of sketches*, In *SODA*, 2008, pp. 737–745.
- [25] P. Indyk and D. P. Woodruff, *Optimal approximations of the frequency moments of data streams*, In *STOC*, 2005, pp. 202–208.
- [26] T. S. Jayram and D. P. Woodruff, *The data stream space complexity of cascaded norms*, In *FOCS*, 2009.
- [27] T. Johnson, S. Muthukrishnan, and I. Rozenbaum, *Sampling algorithms in a stream operator* In *SIGMOD Conference*, 2005, pp. 1–12.
- [28] D. M. Kane, J. Nelson, and D. P. Woodruff, *On the exact space complexity of sketching and streaming small norms*, In *SODA*, 2010.
- [29] P. Li, *Estimators and tail bounds for dimension reduction in L_p ($0 < p \leq 2$) using stable random projections*, In *SODA*, 2008, pp. 10–19.
- [30] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge Univ. Press, 1995.
- [31] J. Ian Munro and M Paterson, *Selection and sorting with limited storage*, *Theoretical Computer Science*, 12(3):315–323, 1980.
- [32] S. Muthukrishnan, *Data streams: algorithms and applications*, 2003.
- [33] T. Sarlós, *Improved approximation algorithms for large matrices via random projections*, In *FOCS*, 2006, pp. 143–152.