On Sketching Trimmed Statistics

HONGHAO LIN, Carnegie Mellon University, USA
HOAI-AN NGUYEN, Carnegie Mellon University, USA
DAVID P. WOODRUFF, Carnegie Mellon University, USA

We study sketching trimmed statistics of a frequency vector, including the F_p moment of the top-k coordinates and of the trimmed-k vector. Despite their natural role in robust analytics, this is the first time these problems have been studied in any sublinear space setting. For $p \in [0, 2]$, we obtain $\mathrm{poly}(\log n/\varepsilon)$-space algorithms for both tasks when k is moderately large, and for general k we identify a sharp structural threshold that characterizes exactly when sublinear space is possible: in particular, it is actually determined by the ratio between a_k^2 and $\|\mathbf{x}_{-k}\|_2^2/k$. We extend these results to $p > 2$ and present several applications including algorithms for thresholded F_p estimation and generalized impact indices. Notably, we improve the space bounds of Govindan, Monemizadeh, and Muthukrishnan (PODS '17) for computing the h-index.

CCS Concepts: • **Theory of computation → Sketching and sampling**.

Additional Key Words and Phrases: sketching; streaming; frequency moments

ACM Reference Format:
Honghao Lin, Hoai-An Nguyen, and David P. Woodruff. 2026. On Sketching Trimmed Statistics. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 111 (May 2026), 27 pages. https://doi.org/10.1145/3801907

1 Introduction

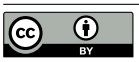
The data-stream model has become a central abstraction for analyzing massive datasets that are too large to store explicitly. Examples include internet traffic logs, financial transactions, database logs, and scientific data streams (e.g., large-scale experiments in fields such as particle physics, genomics, and astronomy). Formally, we assume an underlying frequency vector $\boldsymbol{x} \in \mathbb{Z}^n$ that receives a sequence of positive and negative updates to its coordinates.

Classical work beginning with Alon, Matias, and Szegedy [1] introduced streaming algorithms for estimating global statistics such as the frequency moments $F_p = \sum_i |x_i|^p$, leading to an extensive line of research [4, 6, 10, 11, 24, 25, 31, 32, 35, 39, 45]. Computing F_p moments remains fundamental in database systems, powering tasks in query optimization, data mining, and network monitoring.

However, many analytics tasks in databases and data science rely not on global norms but on *order-based* or *trimmed* statistics—quantities that depend only on the largest, smallest, or central portion of the data. Such measures arise in robust statistics (where extremes are treated as outliers), in descriptive analytics, in regression and optimization (e.g., via Ky-Fan norms [13]), and in impact metrics such as the h-index. Despite their ubiquity, essentially nothing was known about whether such order-restricted statistics can be approximated in sublinear space.

We study two fundamental trimmed statistics. The first is the *top-k F_p moment*, defined as the F_p moment of the k largest coordinates of $\boldsymbol{x}$. This statistic is also known as the Ky-Fan-Norm and has applications in generalized ℓ_p regression [13], composite super-quantile optimization [43], frequency

Authors' Contact Information: Honghao Lin, Carnegie Mellon University, Pittsburgh, USA, honghaol@andrew.cmu.edu; Hoai-An Nguyen, Carnegie Mellon University, Pittsburgh, USA, hnnguyen@andrew.cmu.edu; David P. Woodruff, Carnegie Mellon University, Pittsburgh, USA, dwoodruf@cs.cmu.edu.



ACM 2836-6573/2026/5-ART111
https://doi.org/10.1145/3801907

analysis, and descriptive statistics. Surprisingly, even for natural settings such as estimating the F_p moment of the largest $n/2$ coordinates, no prior sublinear-space streaming algorithms were known.

The second statistic is the *trimmed-k F_p moment*, obtained by discarding both the largest and smallest k coordinates before computing the moment. Trimmed moments are classical tools for mitigating outliers and heavy-tailed noise, and they appear in robust estimation, exploratory analysis, and trimmed regression. Their practical importance is illustrated by the fact that statistical libraries (e.g., in R) include built-in functions for trimmed means and variances, which correspond to trimmed-k F_p norms for $p = 1$ and $p = 2$.

At first glance, these problems might appear solvable using standard tools such as Count-Sketch. However, such approaches require $\Theta(k)$ space to identify the top k items, which is much larger than the $\mathrm{poly}(\log n, 1/\varepsilon)$-space bounds typical for F_p-moment estimation (for $p \in [0, 2]$). Moreover, without structural assumptions on the frequency vector, heavy tails make it difficult to isolate top entries with sufficient accuracy. These limitations motivate the central question of this work:

> *Can we design sublinear-space, one-pass streaming algorithms that approximate trimmed F_p statistics—specifically, the F_p moments of the top-k and trimmed-k versions of the frequency vector?*

We answer this question affirmatively. We design the first sublinear-space *linear sketches* for approximating top-k and trimmed-k F_p statistics, which can be applied directly to a stream of updates. Moreover, linearity ensures that our algorithms extend naturally to distributed and mergeable settings, while also supporting fast update times.

1.1 Our Contributions

We note that the update time of all of our algorithms, or the time that our algorithm takes to process an update, is $\mathrm{poly}(\log n, 1/\varepsilon)$.

First sublinear-space algorithms for trimmed statistics. We design the first linear sketches for approximating both the top-k and trimmed-k frequency moments for $p \in [0, 2]$ up to a $(1 + \varepsilon)$ multiplicative approximation in space $\mathrm{poly}(\log n/\varepsilon)$ whenever k is moderately large (e.g., $k \geq n/\mathrm{polylog} n$).

A sharp structural characterization. For general k, we identify an explicit condition involving the tail of $\boldsymbol{x}$ under which sublinear space is possible. Where $\boldsymbol{a}$ is $\boldsymbol{x}$ sorted in decreasing order by absolute value and $\boldsymbol{x}_{-k}$ is $\boldsymbol{x}$ without the top k coordinates, we show that approximating the top-k F_p moment using $\mathrm{poly}(\log n/\varepsilon)$ space is possible *if and only if*

$$a_k^2 \gtrsim \mathrm{poly}\left(\frac{\varepsilon}{\log n}\right) \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}.$$

Above this threshold, our sketches give $(1 \pm \varepsilon)$-approximations; on the other hand, when $a_k^2 \leq \frac{k}{n^c} \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$, we prove $n^{\Omega(1)}$ space is necessary. This yields a sharp complexity characterization whose threshold is determined by the ratio between a_k^2 and $\frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$. An analogous characterization holds for the trimmed-k statistic, though we show that an additive error term $\varepsilon k \cdot |a_{k-\varepsilon k}|^p$ is unavoidable.

To the best of our knowledge, no prior work has established a connection between trimmed statistics and conditions on the k-residual error.

We also regard our analysis as a concrete contribution. Our algorithms employ a multi-level sub-sampling scheme, where we identify heavy hitters at each level and leverage this information to estimate the value of the desired frequencies. Unlike prior work using similar frameworks, we require a refined analysis of the level set structure, specifically by comparing the residual norms at different levels. We give a more comprehensive view in our technical overview.

Extending to $p > 2$ and applications. We extend our results to $p > 2$, incurring a $\Theta(n^{1-2/p})$ factor that is provably necessary when estimating F_p globally [35].

We also remark on two direct applications of our algorithms.

- Li, Lin, and Woodruff [34] give an $O(k^{2/p}n^{1-2/p}\,\mathrm{poly}(\log n/\varepsilon))$-space linear sketch for estimating the k-residual error $\|\boldsymbol{x}-\boldsymbol{x}_k\|_p^p$ for $p > 2$, where $\boldsymbol{x}_k$ is the best k-sparse approximation of $\boldsymbol{x}$. This residual quantifies the benefit of using a more computationally expensive sparse approximation (i.e., increasing k). When $\|\boldsymbol{x}_k\|_p^p = \Theta(1)\|\boldsymbol{x}_{-k}\|_p^p$, a $(1\pm\varepsilon)$ estimate of the top-k moment yields a $(1\pm\varepsilon)$ estimate of the k-residual error, and in this regime our algorithm achieves a $k^{2/p}$ improvement over [34], which is substantial for large k.
- Estimating the F_p moment of a trimmed-k vector extends naturally to trimmed regression. Given a matrix $\mathbf{A}$ and vector $\mathbf{b}$, trimmed regression minimizes $\|\mathbf{A}x-\mathbf{b}\|_p^p$ after removing the top and bottom k residuals. A linear sketch for trimmed norms allows efficient estimation of $\|\mathbf{A}x-\mathbf{b}\|_{p,\mathrm{trim}}$ for any x. This enables the computation of the trimmed norm across all possible $\boldsymbol{x}$ by leveraging a well-constructed net argument.

We also obtain more involved applications that require not only estimating the contribution of the top-k coordinates but also identifying the appropriate value of k itself.

- Thresholded F_p estimation: approximating $\sum_{i\in\mathcal{B}_\mathcal{T}}|x_i|^p$ where $\mathcal{B}_\mathcal{T}$ is the set of coordinates larger than input $\mathcal{T}$. This is closely related to heavy hitters which is a well-studied problem in databases and streaming [5, 7, 8, 46]. We note that we can take the threshold to be $\varepsilon\|\boldsymbol{x}\|_p$ and in parallel calculate $\|\boldsymbol{x}\|_p$ to get the F_p moment of the ℓ_p heavy hitters.
- Moment based thresholding: approximating k, where k is the largest integer such that $\sum_{i=0}^{k}|a_i|^p \geq k^{p+1}$, and $\boldsymbol{a}$ denotes $\boldsymbol{x}$ sorted in decreasing order. For $p = 1$ this is equivalent to computing the g-index [18].
- Value based thresholding: approximate the top-k F_p moment where k is the largest integer such that each of the top k frequencies is at least k. For $p = 0$, this corresponds to the popular h-index [23] and for $p = 1$ dividing this by k corresponds to the a-index [2].

The h-index, g-index, and a-index can all be viewed as impact indices. Beyond measuring academic publishing impact, the h-index has been used to identify influential users in social networks [42] and to reveal structural properties of large networks, such as approximating the network's degree distribution, computing node coreness, and supporting truss and nucleus decomposition for identifying dense subgraphs [17, 37, 44].

Our algorithm improves upon the h-index algorithm of Govindan, Monemizadeh, and Muthukrishnan [21], which requires a parameter β that lower bounds the h-index and uses $O(\mathrm{poly}\log n\cdot\frac{n}{\beta\varepsilon^2})$ bits of space. Setting $\beta = 1$ yields linear space, and even with a strong lower bound the savings remain limited. For instance, when $h = n^{1/3}$ their algorithm still requires $O(n^{2/3})$ bits, whereas our algorithm achieves a $(1\pm\varepsilon)$ approximation using only $O(\mathrm{poly}(\log n/\varepsilon))$ bits.

Experimental results. We illustrate the practicality of our algorithms by running experiments on two real world datasets and one synthetic dataset. Specifically, we compare our algorithm for computing the F_p moment for the top-k frequencies against the classical Count-Sketch and show that our algorithm achieves better accuracy when given the same space allotment.

1.2 Extended Related Work

There has been a long line of work on computing the frequency moment, or F_p, of a dataset. See Table 1 for a picture of previous work. There is also a long line of work on ℓ_p sampling, which is closely related. In this problem, we are given frequency vector $\boldsymbol{x}\in\mathbb{Z}^n$ and the goal is to return an index $i\in\{1,2,\ldots,n\}$ with probability $|x_i|^p/\|\boldsymbol{x}\|_p^p$. There are generally two samplers that are

F_p	Stream	Upper bound	Lower bound
$p = 0$	insertion-only	$O(\varepsilon^{-2} + \log n)$ [32]	$\Omega(\varepsilon^{-2} + \log n)$ [45]
$p = 0$	turnstile	$O(\varepsilon^{-2} \log n \log\log n)$ [32]	$\Omega(\varepsilon^{-2} \log n \log\log n)$ [16]
$p = 1$	insertion-only	$O(\log\log n + \log \varepsilon^{-1})$ [39]	$\Omega(\log\log n + \log \varepsilon^{-1})$ [39]
$0 < p < 2$	insertion-only, turnstile	$O(\varepsilon^{-2} \log n)$ [31]	$\Omega(\varepsilon^{-2} \log n)$ [31]
$p = 2$	insertion-only, turnstile	$O(\varepsilon^{-2} \log n)$ [1]	$\Omega(\varepsilon^{-2} \log n)$ [6]
$p > 2$	turnstile	$O(n^{1-2/p}(\varepsilon^{-2} + \varepsilon^{-4/p} \log n))$ [20]	$\Omega(\varepsilon^{-2} n^{1-2/p} \log n)$ [35]

Table 1. Upper and lower bounds for F_p estimation. All space bounds are in bits.

studied: approximate [3, 29, 38] and perfect [19, 26, 47]. An approximate sampler is one that returns an index i with probability $(|x_i^p|/\|\boldsymbol{x}\|_p^p) \cdot (1 \pm \varepsilon) + \text{poly}(1/n)$ for $\varepsilon \in (0, 1)$. When we have an approximate sampler where $\varepsilon = 0$, we call these "perfect" samplers. The best known lower bound is $O(\log^3 n)$ (including a $\log n$ factor from the failure probability) for turnstile streams by Kapralov, Nelson, Pachocki, Wang, Woodruff, and Yahyazadeh [33]. See [26] for a more thorough discussion and table on previous work.

We also discuss prior work on techniques and statistics related to the ones we study. We again emphasize that we are not aware of any work on the top-k and trimmed-k statistics that we study in this paper. Daliri, Freire, Musco, Santos, and Zhang [15] study inner product estimation via linear sketches, using priority sampling to preserve large coordinates. Although their focus is different, the underlying sampling idea is related to techniques for trimmed statistics, where one must isolate the contribution of heavy coordinates. Cohen and Kaplan [14] studied bottom-k sketches. Here, there is a ground set where each item in the set is given an independent random rank dependent on the weight of the item. Then, for some input subset, a bottom-k sketch contains the k items with the lowest ranks. This relates to our setting, where we also study approximations of statistics defined by restricting our attention to subsets of the data.

1.3 Road Map

In Section 2 we have our preliminaries. In Section 3 we give a technical overview. In Section 4 we present our algorithm for computing the F_p for $p \in [0, 2]$ for the top-k vector. In Section 5 we present our algorithm for computing the F_p for $p \in [0, 2]$ of the trimmed-k vector. In Section 6 we present the extension of our trimmed statistic algorithms to F_p for $p > 2$. In Section 7 we present a few applications of our algorithms (which include algorithms for impact indices h, g, and a). In Section 8 we give our hardness results. In Section 9 we give our experiments.

2 Preliminaries

Notation. We use x_i to denote the i^{th} entry of input vector $\boldsymbol{x}$ or equivalently the frequency of element i. $\text{rank}(v)$ denotes the rank of item v in vector $\boldsymbol{x}$, or the number of entries in $\boldsymbol{x}$ with value *greater* than v. $\boldsymbol{x}_{-k}$ denotes the vector $\boldsymbol{x}$ excluding the top k frequencies by absolute value. $|\boldsymbol{x}|$ denotes the length/dimension of vector $\boldsymbol{x}$. In general, we boldface vectors and matrices.

Count-Sketch and heavy hitters. We review the Count-Sketch algorithm [11]. We have q distinct hash functions $h_i : [n] \to [B]$ and an array $\boldsymbol{C}$ of size $q \times B$. Additionally, we have q sign functions

$g_i : [n] \to \{-1, 1\}$. The algorithm maintains $\boldsymbol{C}$ (a Count-Sketch structure) such that $C[\ell, b] = \sum_{j:h_\ell(j)=b} g_\ell(j) \cdot x_j$. The frequency estimation $\widehat{x_i}$ of x_i is defined to be the median of $\{g_\ell(i) \cdot C[\ell, h_\ell(i)]\}_{\ell \le q}$. Here, the parameter B is the number of the buckets we use in this data structure. Formally, when $q = O(\log n)$, we have with probability at least $1 - 1/\text{poly}(n)$, $|\widehat{x_i} - x_i| \le O\left(\frac{\|\boldsymbol{x}_{-B}\|_2}{\sqrt{B}}\right)$. Based on this, we obtain the following Heavy Hitter data structure.

Definition 2.1. Given parameters θ, k, a HeavyHitter data structure $\mathcal{D}$ that receives a stream of updates to the frequency vector $\boldsymbol{f}$ and provides a set $T \in [n]$ of heavy hitters of $\boldsymbol{f}$, where

(1) $i \in T$ if $f_i \ge \theta ||\boldsymbol{f}_{-k}||_2$,
(2) For every $i \in T$, we have $f_i \ge \frac{9}{10} \cdot \theta ||\boldsymbol{f}_{-k}||_2$

where $\boldsymbol{f}_{-k}$ is the frequency vector excluding the top k frequencies. Moreover, for every $i \in T$, $\mathcal{D}$ can estimate f_i up to $\frac{1}{k}\left\|\boldsymbol{f}_{-k}\right\|_2$ additive error.

Turnstile streaming model. Our input is an n-dimensional frequency vector $\boldsymbol{x}$, initially all zeros. The algorithm processes a stream of updates of the form $(i, \pm 1)$, which increment or decrement $\boldsymbol{x}_i$. Updates arrive in arbitrary order, and we assume the stream has length at most $\text{poly}(n)$. The goal is to process the stream using sublinear space and only one pass. As is standard, we assume each coordinate is bounded above by an integer m throughout the stream.

Linear sketches. We can compress an n-dimensional vector $\boldsymbol{x}$ by multiplying it with a matrix $\boldsymbol{S} \in \mathbb{R}^{r \times n}$. A matrix $\boldsymbol{S}$ is a linear sketch if it is drawn independently of $\boldsymbol{x}$ and if $\boldsymbol{S}(x + c_i) = \boldsymbol{S}x + \boldsymbol{S}c_i$ for any update c_i that changes a single entry of $\boldsymbol{x}$ by ± 1. Independence allows $\boldsymbol{S}$ to be generated without knowledge of $\boldsymbol{x}$, and linearity ensures that updates can be incorporated without storing $\boldsymbol{x}$ explicitly. In practice, $\boldsymbol{S}$ is stored implicitly using pseudorandom hash functions, enabling efficient updates. The goal is to minimize the required space, ideally making $r << n$.

Subsampling scheme P. In our algorithms we require a subsampling scheme P such that the i-th level has subsampling probability $r_i = 2^{-i}$ to each coordinate of the underlying vector. A hash function is used to remember which coordinates are subsampled in each level. If coordinates $j_1, j_2, \cdots, j_w$ are subsampled in the i^{th} subsampling level, we are not storing these coordinates explicitly. Rather, this subsampling level only looks at updates which involve $j_1, j_2, \cdots, j_2$ to update its structures (in our case, its heavy hitters structure).

Derandomization. Throughout our paper, we assume that our hash functions exploit full randomness. Such an assumption can be removed with an additional $\text{poly}(\log n)$ factor, e.g., the use of Nisan's pseudorandom generator [40] or its variants. We refer the readers to a more detailed discussion in [26] (Theorem 5).

3 Technical Overview

We begin with the problem of estimating the F_p norm of the top k frequencies, starting with the simple case where each of the top k frequencies has value a_k. Suppose that the condition

$$a_k^2 \ge \text{poly}(\varepsilon/\log n) \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k} \tag{1}$$

holds. A naive approach is to run Count-Sketch (Section 2) with $O(k)$ buckets, whose tail error is $O\left(\frac{\|\boldsymbol{x}_{-k}\|_2}{\sqrt{k}}\right) = O(a_k)$. This captures each of the top-k items but uses $\Theta(k)$ buckets, which is suboptimal when k is large. Our goal is to use only $\text{poly}(1/\varepsilon, \log n)$ bits of space.

To reduce the space, we first sub-sample the stream. Let $\widehat{\boldsymbol{x}}$ be the sub-sampled stream obtained by sampling each coordinate independently with probability $p = \min(1, c/(\varepsilon^2 k))$, for a constant

c. Then $pk = \Theta(1/\varepsilon^2)$, so in expectation the sub-stream contains $\Theta(1/\varepsilon^2)$ of the top-k items. A Chernoff bound shows that after rescaling by $1/p$, the number of surviving coordinates in $\widehat{\boldsymbol{x}}$ that were in the top k is $(1 \pm \varepsilon) \cdot k$ with high probability.

The key point here is that sub-sampling greatly reduces the tail mass while preserving enough of the coordinates in the top k. This allows us to use a much smaller Count-Sketch. Specifically, let $k' = \mathrm{poly}(1/\varepsilon, \log n)$ and apply Count-Sketch with k' buckets to the sub-stream $\widehat{\boldsymbol{x}}$. The tail error in this sub-stream is $O\left(\frac{\|\widehat{\boldsymbol{x}}_{-k'}\|_2}{\sqrt{k'}}\right)$. Since only a p-fraction of the tail survives, $\|\widehat{\boldsymbol{x}}_{-k'}\|_2$ shrinks by approximately $\sqrt{p}$, and thus the tail error is

$$O\left(\frac{\|\widehat{\boldsymbol{x}}_{-k'}\|_2}{\sqrt{k'}}\right) = \mathrm{poly}(\varepsilon/\log n) \cdot \frac{\|\boldsymbol{x}_{-k}\|_2}{\sqrt{k}}.$$

Combined with condition (1), this error is small enough to correctly isolate every surviving coordinate of the top k in the sub-stream. After rescaling by $1/p$, we obtain a $(1 \pm \varepsilon)$ approximation to the F_p norm of the top-k. On the other hand, when condition (1) fails, a reduction to a variant of the 2-SUM problem of [12] shows that any such approximation requires $n^{\Omega(1)}$ bits of space.

We now turn to the general case. Here we use the level-set framework introduced by [25]. We divide the coordinates of $\boldsymbol{x}$ into exponentially growing intervals

$$[0, 1+\varepsilon),\ [1+\varepsilon, (1+\varepsilon)^2),\ \ldots,$$

up to the maximum coordinate called level sets. At $O(\log n)$ decreasing subsampling rates, we maintain ℓ_2 heavy hitters for each subsampled stream. For every level set that contributes significantly to the top-k portion of the F_p norm (i.e., whose size multiplied by its level value contributes non-negligibly to the F_p mass), we locate a subsampling rate at which there are $\Theta(1/\varepsilon^2)$ expected survivors from that level. Using the stored heavy hitters for this sub-stream, we estimate the number of coordinates belonging to that level set. Summing these estimated contributions over all relevant levels yields a good approximation to the top-k F_p moment.

The algorithmic framework is standard, but our analysis is not. Prior work such as [25] analyzes the tail norm at a single truncation level and does not need to compare multiple truncation levels simultaneously. In contrast, our setting requires understanding how the residual norms $\|x_{-t}\|_2$ behave across many adjacent values of t. A naive approach fails because the sketching noise can mask the small differences between $\|x_{-t}\|_2$ and $\|x_{-(t+1)}\|_2$. Our main technical contribution is a refined multi-level analysis that ensures these comparisons are stable under sketching noise.

To explain this structure, consider a contributing level set $[v = (1+\varepsilon)^i,\ (1+\varepsilon)^{i+1})$, and let s_i be the number of coordinates in this interval. Let $\mathrm{rank}(v)$ be the number of coordinates greater than v. We show that condition (1) implies the multi-level analogues

$$v^2 s_i \geq \mathrm{poly}(\varepsilon/\log n) \cdot \|x_{-\mathrm{rank}(v)}\|_2^2, \qquad s_i \geq \mathrm{poly}(\varepsilon/\log n) \cdot \mathrm{rank}(v).$$

These inequalities state that the coordinates in this level set carry enough ℓ_2 mass to dominate the tail after $\mathrm{rank}(v)$. This allows us to run a sub-sampled Count-Sketch with sampling probability $p = \min(1, c/(\varepsilon^2 s_i))$, so that the expected number of survivors from this level is $\Theta(1/\varepsilon^2)$. The Count-Sketch tail error under this sampling is

$$O\left(\frac{\|x_{-\mathrm{rank}(v)}\|_2}{\sqrt{s_i}}\right) \cdot \mathrm{poly}(\varepsilon/\log n),$$

which is small compared to v, allowing us to identify all important coordinates in this level. After rescaling, this gives a $(1 \pm \varepsilon)$ approximation of s_i. To our knowledge, this type of multi-level residual comparison has not appeared before and is the core analytical novelty of our work.

We next consider estimating the F_p moment of the trimmed-k vector, obtained by removing the largest and smallest k coordinates. Naively, we could estimate this by taking the difference between the F_p moment of the top $(n-k)$ frequencies and the F_p moment of the top k frequencies which has error $O(\varepsilon)\cdot(\sum_{i=1}^{n-k}|a_i|^p)$. To achieve better error, we rely on a key observation: we can compute both the top-$(n-k)$ and top-k moments using the *same* level-set decomposition and the same estimates $\tilde{s}_j$ for the sizes of all level sets. This means that when the estimator identifies the position where the first k items end, both computations use the *same approximate cutoff*. Because of this, the large contribution of the top k coordinates to the error cancels when we subtract the two estimates. In particular, what we are actually estimating is $\sum_{i=u}^{u+n-2k}|a_i|^p$ for some $u = k \pm O(\varepsilon k)$, corresponding to the middle portion of the vector after discarding the top k items up to the available accuracy. Since each contributing level set is estimated to within a $(1\pm\varepsilon)$ factor, the resulting estimate of this block incurs a total error of $\varepsilon\left(\sum_{i=k}^{n-k}|a_i|^p \;+\; k|a_{k-\varepsilon k}|^p\right)$. We also give a hard instance to show that this extra additive error term $\varepsilon\cdot k|a_{k-\varepsilon k}|^p$ is unavoidable.

4 F_p Moment for $p\in[0,2]$ of Top-k Frequencies

Here, we prove the following theorem with an algorithm that estimates the F_p moment for $p\in[0,2]$ of the top k frequencies.

Theorem 4.1 (F_p of top-k vector for $0\le p\le 2$). *Given $\boldsymbol{x}\in\mathbb{Z}^n$, $p\in[0,2]$, $k\ge 0$, and $\varepsilon\in(0,1)$, suppose that we have*

$$a_k^2 \ge (\varepsilon/\log n)^c\cdot\frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}. \tag{2}$$

for some constant c. Then there exists a linear sketch that uses $\mathrm{poly}(\log n/\varepsilon)$ bits of space (where the exponent depends on the value of c) and estimates $\sum_{i=1}^{k}|a_i|^p$ up to a $(1\pm\varepsilon)$ multiplicative factor with high constant probability.

Remark 4.1. *When $k\ge\mathrm{polylog}n$, condition (2) is always met. Therefore, this directly gives us the result for moderately large k.*

Before presenting the full algorithm, we first give our Algorithm 1 which estimates the size of each "contributing" level set that contains at least one top-k item up to a $(1\pm\varepsilon)$-factor. We then give the formal definition of what it means for a level set to "contribute" and show that accurately estimating the sizes of contributing level sets gives a good final approximation.

Definition 4.2. Suppose that m is an upper bound on $\|\boldsymbol{x}\|_\infty$ and $t = 1+\log_{1+\varepsilon}m$. Let ζ be a uniform random variable in $[1/2,1]$. Define the level set S_j for $j\in[1,\log_{1+\varepsilon}m]$ to be

$$S_j = \{i\in[n] : |x_i|\in[\zeta(1+\varepsilon)^{t-j-1},\zeta(1+\varepsilon)^{t-j})\}$$

and $s_j = |S_j|$. We say the level set S_j "contributes" if

$$\sum_{i\in S_j}|x_i|^p \ge \frac{\varepsilon^2}{\log^2 n}\left(\sum_{i=1}^{k}|a_i|^p\right).$$

Lemma 4.3. *Suppose that $a_k^2 \ge (\varepsilon/\log n)^c\cdot\frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$. For any j such that $\zeta(1+\varepsilon)^{t-j}\ge|a_k|$ and S_j "contributes", taking $v=\zeta(1+\varepsilon)^{t-j-1}$ we have $v^2\cdot s_j\ge(\varepsilon/\log n)^{c+5}\cdot\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$ where $\mathrm{rank}(v)$ is the rank of v in array $\boldsymbol{x}$ (i.e., the number of entries in $\boldsymbol{x}$ with value greater than v).*

Proof. Since S_j contributes, we have

$$s_j v^p \ge (\varepsilon/\log n)^2 k a_k^p.$$

Algorithm 1 Level-Set-Estimator ($\varepsilon \in (0,1]$)

Require: (i) A subsampling scheme P such that the i-th level has subsampling probability $r_i = 2^{-i}$ to each coordinate of the underlying vector; (ii) an upper bound on each coordinate m; (iii) $L+1$ HeavyHitter structures $\mathcal{D}_0, \ldots, \mathcal{D}_L$ with parameter $\theta = (\varepsilon/\log n)^{c+9}$ where $L = \log n$ and $\mathcal{D}_i$ corresponds to the i-th substream.

$t_0 \leftarrow K \log(\varepsilon^{-1} \log m)$ where K is a large constant.
$\zeta \leftarrow$ uniform random variable between $[1/2, 1]$.
$t \leftarrow \log_{1+\varepsilon}(m) + 1$.
for $j = 0, \ldots, L$ **do**
  $\Lambda_j \leftarrow \mathrm{poly}(\varepsilon/L)$ heavy hitters from $\mathcal{D}_j$.
end for
for $j = 0, \ldots, t_0$ **do**
  Let $\tilde{s_j}$ be the number of elements contained in Λ_0 in $[\zeta(1+\varepsilon)^{t-j-1}, \zeta(1+\varepsilon)^{t-j}]$.
end for
for $j = t_0 + 1, ..., t-1$ **do**
  Find the largest ℓ s.t. Λ_ℓ contains $z = \Theta(\log n/\varepsilon^2)$ elements in $[\zeta(1+\varepsilon)^{t-j-1}, \zeta(1+\varepsilon)^{t-j}]$.
  if such ℓ exists **then**
    $\tilde{s_j} \leftarrow z \cdot 2^\ell$.
  else
    $\tilde{s_j} \leftarrow 0$.
  end if
end for
Output each $\tilde{s_j}$ for all $j \in [t]$.

Algorithm 2 Ky-Fan-k-Norm-Estimator ($\varepsilon \in (0,1]$, $p \in (0,2]$, $k \geq 0$)

Run Level-Set-Estimator (ε).
Find the i such that $\sum_{j=0}^{i-1} \tilde{s}_j < k$ and $\sum_{j=0}^{i} \tilde{s}_j \geq k$.
Return $\left(\sum_{j=0}^{i-1} \tilde{s_j} \cdot \zeta^p (1+\varepsilon)^{p(t-j)}\right) + \left(k - \sum_{j=0}^{i-1} \tilde{s}_j\right) \zeta^p (1+\varepsilon)^{p(t-i)}$.

Multiplying both sides by v^{2-p} gives

$$s_j v^2 \geq (\varepsilon/\log n)^2 k a_k^p v^{2-p} \geq (\varepsilon/\log n)^2 k a_k^2 \tag{3}$$

where the last inequality is because we have $v^{2-p} \geq a_k^{2-p}$ since $v \geq a_k$ and $p \leq 2$. Now using the lemma assumption $a_k^2 \geq (\varepsilon/\log n)^c \cdot \|\boldsymbol{x}_{-k}\|_2^2/k$ and multiplying by k gives

$$k a_k^2 \geq (\varepsilon/\log n)^c \|\boldsymbol{x}_{-k}\|_2^2.$$

Substituting this into (3) gives us

$$s_j v^2 \geq (\varepsilon/\log n)^{c+2} \|\boldsymbol{x}_{-k}\|_2^2. \tag{4}$$

We next measure the difference between $\|\boldsymbol{x}_{-k}\|_2^2$ and $\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$. Define the level set T_ℓ for $\ell \in [0, q-1]$ where

$$T_\ell = \{i \in [n] : |x_i| \in [v/2^{\ell+1}, v/2^\ell)\},$$

for $v/2^{\ell+1} \geq |a_k|$ and

$$T_q = \{i \in [n] : |x_i| \in [|a_k|, v/2^q)\}\ .$$

Let $t_\ell = |T_\ell|$, $t_q = |T_q|$. Now, we claim that we have for $w \in [0, q]$ that $t_w \le O(\log^2 n/\varepsilon^2) \cdot s_j \cdot 2^{wp}$. Otherwise we would have

$$\left(\sum_{i=1}^{k} |a_i|^p\right) \ge \left(\frac{v}{2^{w+1}}\right)^p \cdot t_w \ge \left(\frac{v}{2^{w+1}}\right)^p \cdot O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j 2^{wp} \ge O\left(\frac{\log^2 n}{\varepsilon^2}\right)\left(\sum_{i\in S_j} |x_i|^p\right)$$

which contradicts the fact that S_j contributes. So now, taking a sum we get

$$\sum_{\mathrm{rank}(v)\le i\le k} a_i^2 \le \sum_{i\in[q]} t_i \left(\frac{v}{2^i}\right)^2 \le \sum_{i\in[q]} \left(\frac{v}{2^i}\right)^2 O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j 2^{ip} \le v^2 s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right) .$$

This implies that

$$\|\boldsymbol{x}_{-k}\|_2^2 \ge \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2 - v^2 s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right) . \tag{5}$$

Combining (4) and (5) we immediately get that

$$v^2 \cdot s_j \ge (\varepsilon/\log n)^{c+5} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2 . \qquad \square$$

Lemma 4.4. *Consider some level set S_j with $s_j = |S_j|$. Consider the subsampling stream $\mathcal{P}$ where each coordinate is sampled with probability $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2}\right)$ for some constant C and suppose that S_j has z survivors in this stream. Let $\tilde{s}_j = z/r$, then with probability $1 - 1/\mathrm{poly}(n)$ we have $(1-\varepsilon)s_j \le \tilde{s}_j \le (1+\varepsilon)s_j$.*

Proof. Denote the coordinates in S_j as u_i for $i \in [s_j]$. Let X_i for $i \in [s_j]$ denote an indicator random variable which is 1 if u_i was sampled in $\mathcal{P}$ and 0 otherwise. We have that $\mathbf{E}[X_i] = r$ and $\mathbf{E}\left[\sum_i X_i\right] = s_j r$. Then, from Chernoff's bound we have that

$$\mathbf{Pr}\left[\left|\sum_i X_i - s_j r\right| \ge \varepsilon s_j r\right] \le 2\exp(-\varepsilon^2 s_j r/3) \le 2\exp(-C\log n/3) \le 1/\mathrm{poly}(n). \qquad \square$$

Lemma 4.5. *Consider some level set S_j with $s_j = |S_j|$. Consider the subsampling stream $\mathcal{P}/2$ where each coordinate is sampled with probability $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2} \cdot \frac{1}{2}\right)$ for some constant C. With probability at least $1 - 1/\mathrm{poly}(n)$ there are less than $c\log n/\varepsilon^2$ survivors from S_j.*

Proof. Denote the coordinates in S_j as u_i for $i \in [s_j]$. Let X_i for $i \in [s_j]$ denote an indicator random variable which is 1 if u_i was sampled in $\mathcal{P}/2$ and 0 otherwise. We have that $\mathbf{E}[X_i] = r$ and $\mathbf{E}[X] = s_j r$ for $X = \sum_i X_i$. From Chernoff's bound we have that

$$\mathbf{Pr}\left[X \ge \frac{c\log n}{\varepsilon^2}\right] = \mathbf{Pr}[X > 2\cdot\mathbf{E}[X]] \le 2\exp(-C\log n/6) \le 1/\mathrm{poly}(n). \qquad \square$$

Lemma 4.6. *Suppose that $a_k^2 \ge (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ and S_j contributes where $\zeta(1+\varepsilon)^{t-j} \ge |a_k|$. Then, with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$, we have $\tilde{s}_j \in [(1-\varepsilon)s_j, (1+\varepsilon)s_j]$.*

Proof. Consider a level set S_j with a value range in $[v, (1+\varepsilon)v]$ where $\zeta(1+\varepsilon)^{t-j} \ge |a_k|$. Consider the sub-stream $\mathcal{P}$ with corresponding sampling rate $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2}\right)$ for a sufficiently large constant C and assume there are z survivors of S_j in $\mathcal{P}$. Since we have a random threshold ζ in the boundary of the level set, we have with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$, a $(1-\varepsilon)$ fraction of them is within $[v\left(1 + \mathrm{poly}(\varepsilon/\log n)\right), (1+\varepsilon)v\left(1 - \mathrm{poly}(\varepsilon/\log n)\right)]$.

We next analyze the tail error of the heavy hitter data structure. First we have that

$$s_j \geq (\varepsilon^2/\log^2 n) \cdot \mathrm{rank}(v). \tag{6}$$

Suppose that we had $s_j < (\varepsilon^2/\log^2 n) \cdot \mathrm{rank}(v)$. This would mean that $\mathrm{rank}(v) \cdot v^p \geq s_j(\log^2 n/\varepsilon^2) \cdot v^p$, which contradicts the fact that S_j contributes. Therefore, combining (6) with the fact that $\mathcal{P}$ has sampling rate $r = \min\left(1, \frac{C \log n}{s_j \varepsilon^2}\right)$ gives us that with probability $1 - \mathrm{poly}(\varepsilon/\log n)$ using Chernoff's bound that the number of survivors of the top $\mathrm{rank}(v)$ coordinates of x surviving in $\mathcal{P}$ is at most $O((\log n/\varepsilon)^4)$.

Recall that we use $(\log n/\varepsilon)^{c+9}$ buckets in our heavy hitter data structure (Section 2). Hence, with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$, the tail error of the heavy hitter data structure will be at most $\frac{1}{\sqrt{s_j}}(\varepsilon/\log n)^{c/2+9/2} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2$.

Recall that we have condition $a_k^2 \geq (\varepsilon/\log n)^c \cdot \frac{\|x_{-k}\|_2^2}{k}$ and therefore from Lemma 4.3 have $v^2 s_j \geq (\varepsilon/\log n)^{c+5} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$. Combining this with the above tail error we immediately have with high probability the heavy hitter data structure can identify every survivor in $[v\left(1 + (\varepsilon/\log n)^2\right), (1 + \varepsilon)v\left(1 - (\varepsilon/\log n)^2\right)]$.

Combining this with Lemma 4.4, the remaining thing is to show that with high probability that a level with a smaller sampling rate than r does not have $\Theta(\log n/\varepsilon^2)$ survivors of S_j. Recall that in the algorithm, for each level set it finds the sub-stream with the smallest sampling rate such that there are $\Theta(\log n/\varepsilon^2)$ coordinates in the set. Then to get an estimate of the size of the level set we re-scale by the sampling probability. It follows from Lemma 4.5 that we identify the right sampling rate.

So, we have with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$ that $\tilde{s}_j \in [(1 - \varepsilon)s_j, (1 + \varepsilon)s_j]$. □

Remark 4.2. *Lemma 4.6 suggests that our current algorithm requires $(\log n/\varepsilon)^{c+9}$ buckets in the CountSketch data structure. However, this dependence can be further improved. For example, (1) for $p < 2$, we can save an $\log n$ when summing over the different level sets in Lemma 4.3. (2) Our current algorithm estimates each level-set size to $(1 \pm \varepsilon)$ accuracy. However, if a level is "tiny" (e.g., contributes only an ε-fraction of total F_p), such precision is unnecessary: an $O(1)$ constant-factor estimate suffices. Using this idea, we can first obtain a constant-factor approximation of the size of the level set and use it to choose an appropriate sampling level. This idea has appeared in Section 5.1.3 of [36]. This refinement can reduce the final exponent by two. (3) Our current algorithm partitions the coordinates into levels in the power of $(1 + \varepsilon)$. We can instead define level sets using powers of 2 and estimate the average contribution within each such level. This further reduces the exponent by one.*

Lemma 4.7. *Consider some S_j which does not contribute where $\zeta(1 + \varepsilon)^{t-j} \geq |a_k|$. With probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$, $\tilde{s}_j \in [0, (1 + \varepsilon)s_j]$.*

Proof. Consider some level set S_j which does not contribute. In this case, the algorithm may not find enough number of survivors in any subsampling stream. So, our lower bound for the size estimate is 0. If there are enough survivors, then by the same argument as Lemma 4.6, $\tilde{s_j}$ will be a $(1 \pm \varepsilon)$-approximation of s_j. Therefore, for any level set that does not contribute, the size estimate can be an under-estimate but never an estimate by more than a $(1 + \varepsilon)$ factor with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$. □

Lemma 4.8. *The F_p norm of all coordinates in non-contributing level sets is at most $O(\varepsilon) \cdot (\sum_{i=1}^k |a_i|^p)$.*

Proof. There are $\frac{\log m}{O(\varepsilon)}$ level sets, and therefore at most that many non-contributing set. The F_p norm of all the coordinates in each non-contributing set by definition is at most $\frac{\varepsilon^2}{\log^2 n}\left(\sum_{i=1}^k |a_i|^p\right)$.

Therefore, the F_p norm of all coordinates in non-contributing level sets is at most $\frac{\log m}{O(\varepsilon)} \cdot \frac{\varepsilon^2}{\log^2 n} \left(\sum_{i=1}^{k} a_i^p \right)$. □

Our full algorithm is given in Ky-Fan-k-Norm-Estimator (Algorithm 2). We note that in Line 8, this is Λ_0 and not Λ_j. For a contributing level which has fewer than $O(1/\varepsilon^2)$ coordinates, we will need to find all of the coordinates in that level set as opposed to obtaining a subsample of them. This means that we need to look at the entire stream instead of the sub-stream. It is also the reason that we divide the iterations into two parts $(0, t_0)$ and $(t_0 + 1, t - 1)$. We are now ready to prove our Theorem 4.1.

Proof of Theorem 4.1. We have that with probability at least $1 - \text{poly}(\varepsilon/\log n)$ that for a contributing level set S_j, we have that the estimator $\tilde{s}_j \in (1 \pm \varepsilon)s_j$ from Lemma 4.6. For level sets that do not contribute, we have from Lemma 4.7 that the size estimate can be an under-estimate but never an estimate by more than a $(1 + \varepsilon)$ factor with probability at least $1 - \text{poly}(\varepsilon/\log n)$.

Note that we have $t = \log m/(2\varepsilon)$ level sets where $m = \text{poly}(n)$ by assumption. Therefore, we can take a union bound over all the level sets for all the above events and get that they happen with constant probability. Now, we upper bound the output of our algorithm.

As proven above, for each level set S_j, the algorithm estimates s_j within a $(1 \pm \varepsilon)$ factor. In addition, each coordinate value in this level is within a $(1 \pm \varepsilon)$ factor due to the range of each level set. Since our estimator takes the sum of the top k values, we have that the output is at most

$$(1 + \varepsilon)^2 \sum_{i=1}^{k} |a_i|^p = \big(1 + O(\varepsilon)\big) \sum_{i=1}^{k} |a_i|^p \ .$$

We now lower bound the output of our algorithm. Recall that besides the error in estimating s_j for contributing sets S_j and the error associated with the level set range, underestimating comes from non-contributing level sets. The sum of all the coordinates in the non-contributing level sets is at most $O(\varepsilon) \left(\sum_{i=1}^{k} |a_i|^p \right)$ by Lemma 4.8. Therefore, we have that the output of the algorithm is at least

$$(1 - \varepsilon)^2 \sum_{i=1}^{k} |a_i|^p - O(\varepsilon) \left(\sum_{i=1}^{k} |a_i|^p \right) = \big(1 - O(\varepsilon)\big) \sum_{i=1}^{k} |a_i|^p \ . \qquad \square$$

5 Estimation of Trimmed-k F_p Moment for $p \in [0, 2]$

We now present our result for the trimmed-k norm.

Theorem 5.1 (F_p of trimmed-k vector for $0 \le p \le 2$). *Given $\boldsymbol{x} \in \mathbb{Z}^n$, $p \in [0, 2]$, $k \ge 0$, and $\varepsilon \in (0, 1)$, if we have*

$$a_k^2 \ge (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$$

for some constant c, then there exists a linear sketch that uses $\text{poly}(\log n/\varepsilon)$ bits of space (where the exponent depends on the value of c) and estimates $\sum_{i=k}^{n-k} |a_i|^p$ with error $\varepsilon \left(\sum_{i=k}^{n-k} |a_i|^p + k|a_{k-\varepsilon k}|^p \right)$ with high constant probability.

Our estimator Trimmed-k-Norm-Estimator (Algorithm 3) is similar to that of the previous section. However, we have an additional difficulty since we need to remove the contribution of the top and bottom k coordinates. We only can estimate the contribution of each level set up to a $(1 \pm \varepsilon)$-factor. Therefore, we are actually estimating $\sum_{i=u}^{u+n-2k} a_i^p$ where $u = k \pm O(\varepsilon)k$.

At a high level our algorithm runs Level-Set-Estimator (Algorithm 1) to divide the universe into level sets and estimate how many coordinates are in each level set. Then it sums up the top $n - k$

Algorithm 3 Trimmed-k-Norm-Estimator ($\varepsilon \in (0,1]$, $p \in (0,2]$, $k \geq 0$)

1: Run Level-Set-Estimator (ε).
2: Find the i_1 such that $\sum_{j=0}^{i_1-1} \tilde{s}_j < k$ and $\sum_{j=0}^{i_1} \tilde{s}_j \geq k$.
3: Find the i_2 such that $\sum_{j=0}^{i_2-1} \tilde{s}_j < n-k$ and $\sum_{j=0}^{i_2} \tilde{s}_j \geq n-k$.
4: top-k $\leftarrow \left(\sum_{j=0}^{i_1-1} \tilde{s}_j \cdot \zeta^p (1+\varepsilon)^{p(t-j)}\right) + \left(k - \sum_{j=0}^{i_1-1} \tilde{s}_j\right) \zeta^p (1+\varepsilon)^{p(t-i_1)}$.
5: top-n-minus-k $\leftarrow \left(\sum_{j=0}^{i_2-1} \tilde{s}_j \cdot \zeta^p (1+\varepsilon)^{p(t-j)}\right) + \left((n-k) - \sum_{j=0}^{i_2-1} \tilde{s}_j\right) \zeta^p (1+\varepsilon)^{p(t-i_2)}$.
6: **Return** top-n-minus-k − top-k.

coordinates and then subtracts off the top k coordinates. Note that the condition (2) always holds for a_{n-k} as the trimmed-k vector of $\boldsymbol{x}$ only makes sense when $k \leq n/2$, which means we have $n-k = \Theta(n)$. We first show that $u = k \pm O(\varepsilon)k$.

Lemma 5.2. $u \geq k - O(\varepsilon)k$.

Proof. Recall that like reasoned in the proof of Theorem 4.1, for each level set associated with the top k coordinates, we either underestimate its size or estimate its size up to a $(1 \pm \varepsilon)$-factor. Therefore, we have $u \geq k - O(\varepsilon)k$ where equality holds when we overestimate each level set associated with the top-k coordinates by a $(1+\varepsilon)$ factor. □

Lemma 5.3. $u \leq k + O(\varepsilon)k$.

Proof. From Lemma 4.6, we know that for each level set that contributes (by Definition 4.2) we can estimate its size up to a $(1 \pm \varepsilon)$-factor. We first bound the number of coordinates in the level sets (associated with coordinates in the top-k) that do not contribute and we therefore do not estimate well.

We claim that for each level set S_j, if it contains at least $\Omega(k\varepsilon^2/\log n)$ coordinates, then the algorithm estimates its size up to a $(1\pm\varepsilon)$ factor. To show this, consider the sub-stream with sampling rate $r = \Theta(\frac{1}{k\mathrm{poly}(\varepsilon/\log n)})$. By Chernoff's bound, with high probability there will be $\Theta(1/\varepsilon^2)$ survivors in this sub-stream. Furthermore, since we are guaranteed that $a_k^2 \geq \mathrm{poly}(\varepsilon/\log n) \cdot \|\boldsymbol{x}_{-k}\|_2^2 / k$, this implies the coordinates in this level set can be identified by the Heavy Hitter data structure. Following the same proof as Lemma 4.6, our algorithm estimates the size of this level set up to a $(1 \pm \varepsilon)$ factor with probability $1 - \mathrm{poly}(\varepsilon/\log n)$.

Therefore, for each level set associated with coordinates in the top-k, we either estimate its size up to a $(1 \pm \varepsilon)$-factor or the level set has $o(k\varepsilon^2/\log n)$ coordinates. Since there are at most $\log n/O(\varepsilon)$ such level sets, we have $u \leq k + O(\varepsilon)k$. □

The above discussion shows that if we can estimate $\sum_{i=u}^{u+n-2k} |a_i|^p$ up to error $\varepsilon \left(\sum_{i=u}^{u+n-2k} |a_i|^p\right) + \varepsilon \cdot k|a_k|^p$, then the overall error of our estimator is

$$O(\varepsilon)\left(\sum_{i=k}^{n-k} |a_i|^p + k|a_{k-\varepsilon k}|^p\right).$$

To achieve this, we shall consider a similar level-set argument as that in Section 4.

We first define a "contributing" level set. Recall Definition 4.2 for the definition of S_j and s_j.

Definition 5.4. We say the level set S_j "contributes" if

$$\sum_{i\in S_j} |x_i|^p \geq \frac{\varepsilon^2}{\log^2 n}\left(\sum_{i=k+1}^{n-k} |a_i|^p + k|a_k|^p\right).$$

Lemma 5.5. *If S_j contributes and we have $v = \zeta(1+\varepsilon)^{t-j} \geq |a_{n-k}|$ then we have $v^2 \cdot s_j \geq (\varepsilon/\log n)^5 \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$ where $\mathrm{rank}(v)$ is the rank of v in array $\boldsymbol{x}$ (i.e., the number of entries in $\boldsymbol{x}$ with value greater than v).*

Proof. Since S_j contributes, we have that

$$s_j v^p \geq (\varepsilon/\log n)^2 \cdot \left(\sum_{i=k+1}^{n-k} |a_i|^p + k|a_k|^p\right) \geq (\varepsilon/\log n)^2 \cdot (n-k)|a_{n-k}|^p.$$

Multiplying by v^{2-p} gives

$$v^2 s_j \geq (\varepsilon/\log n)^2 \cdot (n-k)|a_{n-k}|^p v^{2-p} \geq (\varepsilon/\log n)^2 \cdot (n-k)a_{n-k}^2 \tag{7}$$

where the last inequality is because we have $v \geq |a_{n-k}|$ and $p \leq 2$. Note that we have $n-k \geq k$ since the problem is only properly defined if we have $k \leq \frac{n}{2}$. Therefore, we have

$$(n-k)a_{n-k}^2 \geq k a_{n-k}^2 \geq \|\boldsymbol{x}_{-(n-k)}\|_2^2.$$

Plugging this into (7) gives us

$$v^2 s_j \geq (\varepsilon/\log n)^2 \cdot \|\boldsymbol{x}_{-(n-k)}\|_2^2. \tag{8}$$

We next measure the difference between $\left\|\boldsymbol{x}_{-(n-k)}\right\|_2^2$ and $\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$. For the level set

$$T_\ell = \{i \in [n] : |x_i| \in [v/2^{\ell+1}, v/2^\ell]\},$$

where $v/2^{\ell+1} \geq a_{n-k}$ and

$$T_q = \{i \in [n] : |x_i| \in [|a_{n-k}|, v/2^q)\}.$$

Let $t_\ell = |T_\ell|$, $t_q = |T_q|$. We now claim that for $w \in [0, q]$ we have $t_w \leq O(\log^2 n/\varepsilon^2) \cdot s_j \cdot 2^{wp}$. Otherwise, we would have

$$\sum_{i=k+1}^{n-k} |a_i|^p + k|a_k|^p \geq \left(\frac{v}{2^{w+1}}\right)^p \cdot t_w \geq \left(\frac{v}{2^{w+1}}\right)^p \cdot O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j 2^{wp} \geq O\left(\frac{\log^2 n}{\varepsilon^2}\right)\left(\sum_{i\in S_j} |x_i|^p\right)$$

which contradicts the fact that S_j contributes. Now, taking a sum we get

$$\sum_{\mathrm{rank}(v)\leq i\leq n-k} a_i^2 \leq \sum_{i\in[q]} t_i \left(\frac{v}{2^i}\right)^2 \leq \sum_{i\in[q]} \left(\frac{v}{2^i}\right)^2 O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j 2^{ip} \leq v^2 s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right). \tag{9}$$

This implies that

$$\left\|\boldsymbol{x}_{-(n-k)}\right\|_2^2 \geq \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2 - v^2 s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right).$$

Combining (8) and (9) we get immediately that

$$v^2 \cdot s_j \geq (\varepsilon/\log n)^5 \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2. \qquad \square$$

Lemma 5.6. *Suppose that S_j contributes where $\zeta(1+\varepsilon)^{t-j} \geq |a_{n-k}|$. Then with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$ we have $\tilde{s}_j = (1\pm\varepsilon)s_j$.*

Proof. The proof is similar to that of Lemma 4.6. Again consider some S_j with value range $[v, (1+\varepsilon)v]$ where $\zeta(1+\varepsilon)^{t-j} \geq |a_{n-k}|$. Consider the sub-stream $\mathcal{P}$ with corresponding sampling rate $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2}\right)$ for a sufficiently large constant C and assume that there are z survivors of S_j in $\mathcal{P}$. Since we have a random threshold ζ in the boundary of the level set, we have with probability $1-\mathrm{poly}(\varepsilon/\log n)$ that a $(1-\varepsilon)$ fraction of them is in $[v(1+\mathrm{poly}(\varepsilon/\log n)), (1+\varepsilon)v(1-\mathrm{poly}(\varepsilon/\log n))]$.

We next analyze the tail error of the heavy hitter data structure. Since S_j contributes, we have

$$s_j \geq (\varepsilon^2/\log^2 n)\cdot \mathrm{rank}(v).$$

Recall again that we use $(\log n/\varepsilon)^{c+9}$ buckets in the HeavyHitter data structure. So, by the same logic as in Lemma 4.6, with probability $1-\mathrm{poly}(\varepsilon/\log n)$ the tail error of the HeavyHitter structure is at most $\frac{1}{\sqrt{s_j}}(\varepsilon/\log n)^{c/2+9/2}\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2$.

We have $v^2\cdot s_j \geq (\varepsilon/\log n)^5\cdot\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_2^2$ by Lemma 5.5. The rest of the proof goes through exactly as in Lemma 4.6. □

Lemma 5.7. *Consider some S_j which does not contribute. With probability at least $1-\mathrm{poly}(\varepsilon/\log n)$, $\tilde{s}_j \in [0, (1+\varepsilon)s_j]$.*

Proof. This follows from the same proof as Lemma 4.7. □

Lemma 5.8. *The F_p norm of all coordinates in non-contributing level sets is at most $O(\varepsilon)\cdot(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_k|^p)$.*

Proof. There are $\frac{\log m}{O(\varepsilon)}$ level sets, and therefore at most that many non-contributing sets. Therefore by definition, we have that the F_p norm of all coordinates in non-contributing level sets is at most

$$\frac{\log m}{O(\varepsilon)}\cdot\frac{\varepsilon^2}{\log^2 n}\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_k|^p\right) = O(\varepsilon)\cdot\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_k|^p\right).$$ □

After obtaining Lemma 5.6, similarly to the proof of Theorem 4.1, we can get the correctness of Theorem 5.1.

Proof of Theorem 5.1. We have that with probability at least $1-\mathrm{poly}(\varepsilon/\log n)$ that for contributing level set S_j, we have that the estimator $\tilde{s}_j \in (1\pm\varepsilon)s_j$ from Lemma 5.6. For level sets that do not contribute, we have from Lemma 5.7 that the size estimate can be an under-estimate but never an estimate by more than a $(1+\varepsilon)$ factor with probability at least $1-\mathrm{poly}(\varepsilon/\log n)$.

Note that we have $t = \log m/(2\varepsilon)$ level sets where $m = \mathrm{poly}(n)$ by assumption. Therefore, we can take a union bound over all the level sets for all the above events and get that they happen with constant probability. Now, we upper bound the output of our algorithm.

As proven above, for each contributing level set S_j, the algorithm estimates s_j within a $(1\pm\varepsilon)$ factor. In addition, each coordinate value in this level is within a $(1\pm\varepsilon)$ factor due to the range of each level set. So, the output is at most

$$(1+\varepsilon)^2\sum_{i=u}^{u+n-2k}|a_i|^p = \left(1+O(\varepsilon)\right)\sum_{i=u}^{u+n-2k}|a_i|^p\ .$$

We now lower bound the output of our algorithm. Recall that besides the error in estimating s_j for contributing sets S_j and the error associated with the level set range, underestimating comes from non-contributing level sets. The sum of all the coordinates in the non-contributing level sets

is at most $O(\varepsilon)\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_k|^p\right)$ by Lemma 5.8. Therefore, we have that the output of the algorithm is at least

$$(1-\varepsilon)^2\sum_{i=u}^{u+n-2k}|a_i|^p - O(\varepsilon)\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_k|^p\right) = \left(1-O(\varepsilon)\right)\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_{k-\varepsilon k}|^p\right). \qquad \square$$

6 Trimmed Statistics for $p > 2$

In this section, we give our sketching algorithms for the trimmed statistic of a vector for the case when $p > 2$. We use the same algorithms as Algorithm 2 and Algorithm 3 but instead keep track of the $\mathrm{poly}(\varepsilon/\log n)\cdot\frac{1}{n^{1-2/p}}$- ℓ_2 heavy hitters.

6.1 Top-k

Theorem 6.1 (F_p of top-k vector for $p > 2$). *Given $\boldsymbol{x}\in\mathbb{Z}^n$, $p > 2$, $k \geq 0$, and $\varepsilon\in(0,1)$, suppose that we have $|a_k|^p \geq (\varepsilon/\log n)^c\cdot\frac{\|\boldsymbol{x}_{-k}\|_p^p}{k}$ for some constant c. Then there exists a linear sketch that uses $\mathrm{poly}(\log n/\varepsilon)\cdot n^{1-2/p}$ bits of space and estimates $\sum_{i=1}^k|a_i|^p$ up to a $(1\pm\varepsilon)$ multiplicative factor with high constant probability.*

We first consider the estimation of the F_p moment of the top-k frequencies. We use the same definition of "contribute" as Definition 4.2.

The following lemma is analogous to Lemma 4.3 for the case when $p \leq 2$.

Lemma 6.2. *For any j such that that $\zeta(1+\varepsilon)^{t-j}\geq|a_k|$ and S_j "contributes", taking $v = \zeta(1+\varepsilon)^{t-j-1}$ we have $v^p\cdot s_j \geq (\varepsilon/\log n)^{c+5}\cdot\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$ where $\mathrm{rank}(v)$ is the rank of v in array $\boldsymbol{x}$ (i.e., the number of entries in $\boldsymbol{x}$ with value greater than v).*

Proof. Since S_j contributes by definition we have

$$v^p\cdot s_j \geq (\varepsilon/\log n)^2\cdot|a_k|^p\cdot k \geq (\varepsilon/\log n)^{c+2}\|\boldsymbol{x}_{-k}\|_p^p \tag{10}$$

where the last inequality comes from the condition that $a_k^p \geq (\varepsilon/\log n)^c\cdot\|\boldsymbol{x}_{-k}\|_p^p/k$.

We next measure the difference between $\|\boldsymbol{x}_{-k}\|_p^p$ and $\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$. Define the level set T_ℓ for $\ell\in[0,q-1]$ where

$$T_\ell = \{i\in[n] : |x_i|\in[v/2^{\ell+1}, v/2^\ell)\},$$

for $v/2^{\ell+1}\geq|a_k|$ and

$$T_q = \{i\in[n] : |x_i|\in[|a_k|, v/2^q)\}.$$

Let $t_\ell = |T_\ell|$, $t_q = |T_q|$. Now, we claim that we have for $w\in[0,q]$ that $t_w \leq O(\log^2 n/\varepsilon^2)\cdot s_j\cdot 2^{wp}$. Otherwise we would have

$$\left(\sum_{i=1}^k|a_i|^p\right)\geq\left(\frac{v}{2^{w+1}}\right)^p\cdot t_w \geq \left(\frac{v}{2^{w+1}}\right)^p\cdot O\left(\frac{\log^2 n}{\varepsilon^2}\right)s_j 2^{wp} \geq O\left(\frac{\log^2 n}{\varepsilon^2}\right)\left(\sum_{i\in S_j}|x_i|^p\right)$$

which contradicts the fact that S_j contributes. So now, taking a sum we get

$$\sum_{\mathrm{rank}(v)\leq i\leq k}|a_i|^p \leq \sum_{i\in[q]}t_i\left(\frac{v}{2^i}\right)^p \leq O\left(\frac{\log^2 n}{\varepsilon^2}\right)s_j\sum_{i\in[q]}\left(\frac{v}{2^i}\right)^p 2^{ip} \leq v^p s_j\cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right).$$

So we get

$$\|\boldsymbol{x}_{-k}\|_p^p \geq \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p - v^p s_j\cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right). \tag{11}$$

Combining (10) and (11) we immediately get that

$$v^p \cdot s_j \geq (\varepsilon/\log n)^{c+5} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p . \qquad \square$$

Lemma 6.3. *Suppose that* $|a_k|^p \geq (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_p^p}{k}$ *and* S_j *contributes where* $\zeta(1+\varepsilon)^{t-j} \geq |a_k|$. *Then, with probability at least* $1 - \mathrm{poly}(\varepsilon/\log n)$, *we have* $\tilde{s}_j \in [(1-\varepsilon)s_j, (1+\varepsilon)s_j]$.

Proof. Consider a level set S_j with a value range in $[v, (1+\varepsilon)v]$ where $\zeta(1+\varepsilon)^{t-j} \geq |a_k|$. Consider the sub-stream $\mathcal{P}$ with corresponding sampling rate $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2}\right)$ for a sufficiently large constant C and assume there are z survivors of S_j in $\mathcal{P}$. Since we have a random threshold ζ in the boundary of the level set, we have with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$, a $(1-\varepsilon)$ fraction of them is within $[v\left(1+\mathrm{poly}(\varepsilon/\log n)\right), (1+\varepsilon)v\left(1-\mathrm{poly}(\varepsilon/\log n)\right)]$.

We next analyze the tail error of the heavy hitter data structure. First we have that

$$s_j \geq (\varepsilon^2/\log^2 n) \cdot \mathrm{rank}(v). \tag{12}$$

Suppose that we had $s_j < (\varepsilon^2/\log^2 n)\cdot\mathrm{rank}(v)$. This would mean that $\mathrm{rank}(v)\cdot v^p \geq s_j(\log^2 n/\varepsilon^2)\cdot v^p$, which contradicts the fact that S_j contributes. Therefore, combining (12) with the fact that $\mathcal{P}$ has sampling rate $r = \min\left(1, \frac{C\log n}{s_j\varepsilon^2}\right)$ gives us that with probability $1 - \mathrm{poly}(\varepsilon/\log n)$ using Chernoff's bound that the number of survivors of the top $\mathrm{rank}(v)$ coordinates of x surviving in $\mathcal{P}$ is at most $O((\log n/\varepsilon)^4)$.

Suppose that we use $(\log n/\varepsilon)^{c+9} \cdot n^{1-2/p}$ buckets in our heavy hitter data structure (Section 2). Hence, with probability at least $1-\mathrm{poly}(\varepsilon/\log n)$, the tail error of the heavy hitter data structure will be at most $\frac{1}{\sqrt{s_j}}(\varepsilon/\log n)^{c/2+9/2} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p$ (converting from $\|\cdot\|_2$ to $\|\cdot\|_p$ by Holder's Inequality).

Recall that we have condition $|a_k|^p \geq (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_p^p}{k}$ and therefore from Lemma 6.2 have $v^p\cdot s_j \geq (\varepsilon/\log n)^{c+5}\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$. Combining this with the above tail error we immediately have with high probability the heavy hitter data structure can identify every survivor in $[v\left(1+\mathrm{poly}(\varepsilon/\log n)\right), (1+\varepsilon)v\left(1-\mathrm{poly}(\varepsilon/\log n)\right)]$.

Combining this with Lemma 4.4, the remaining thing is to show that with high probability that a level with a smaller sampling rate than r does not have $\Theta(\log n/\varepsilon^2)$ survivors of S_j. Recall that in the algorithm, for each level set it finds the sub-stream with the smallest sampling rate such that there are $\Theta(\log n/\varepsilon^2)$ coordinates in the set. Then to get an estimate of the size of the level set we re-scale by the sampling probability. It follows from Lemma 4.5 that we identify the right sampling rate.

So, we have with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$ that $\tilde{s}_j \in [(1-\varepsilon)s_j, (1+\varepsilon)s_j]$. $\square$

The rest of the proof of Theorem 6.1 follows from the proof of Theorem 4.1 and from the fact that Holder's inequality gives us $\|\cdot\|_2 \cdot \frac{1}{n^{1/2-1/p}} \leq \|\cdot\|_p$ for $p > 2$.

6.2 Trimmed-k

Theorem 6.4 (F_p for trimmed-k vector for $p > 2$). *Given* $\boldsymbol{x} \in \mathbb{Z}^n$, $p > 2$, $k \geq 0$, *and* $\varepsilon \in (0,1)$, *suppose that we have* $|a_k|^p \geq (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_p^p}{k}$ *for some constant c. Then there exists a linear sketch that uses* $\mathrm{poly}(\log n/\varepsilon) \cdot n^{1-2/p}$ *bits of space and estimates* $\sum_{i=k+1}^{n-k}|a_i|^p$ *with error* $\varepsilon\left(\sum_{i=k+1}^{n-k}|a_i|^p + k|a_{k-\varepsilon k}|^p\right)$ *with high constant probability.*

We next consider the estimation of the F_p moment of the trimmed-k vector. We say a level set S_j "contributes" according to Definition 5.4. The following lemma is analogous to Lemma 5.5 for the case when $p \leq 2$.

Lemma 6.5. *If S_j contributes and we have $v = \zeta(1+\varepsilon)^{t-j} \geq |a_{n-k}|$ then we have $v^p \cdot s_j \geq (\varepsilon/\log n)^{c+5} \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$ where $\mathrm{rank}(v)$ is the rank of v in array $\boldsymbol{x}$ (i.e., the number of entries in $\boldsymbol{x}$ with value greater than v).*

Proof. Since S_j contributes, by definition, we have

$$v^p \cdot s_j \geq (\varepsilon/\log n)^2 \cdot a_{n-k}^p \cdot (n-k) \geq O(\varepsilon/\log n)^2 \left\|\boldsymbol{x}_{-(n-k)}\right\|_p^p \tag{13}$$

given $n-k \geq \frac{n}{2}$. Note that the problem is only properly defined if we have $k \leq \frac{n}{2}$.

We next measure the difference between $\left\|\boldsymbol{x}_{-(n-k)}\right\|_p^p$ and $\left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$. For the level set

$$T_\ell = \{i \in [n] : |x_i| \in [v/2^{\ell+1}, v/2^\ell]\},$$

where $v/2^{\ell+1} \geq |a_{n-k}|$ and

$$T_q = \{i \in [n] : |x_i| \in [|a_{n-k}|, v/2^q)\}.$$

Let $t_\ell = |T_\ell|$, $t_q = |T_q|$. We now claim that for $w \in [0,q]$ we have $t_w \leq O(\log^2 n/\varepsilon^2) \cdot s_j \cdot 2^{wp}$. Otherwise, we would have

$$\sum_{i=k+1}^{n-k} |a_i|^p + k|a_k|^p \geq \left(\frac{v}{2^{w+1}}\right)^p \cdot t_w \geq \left(\frac{v}{2^{w+1}}\right)^p \cdot O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j 2^{wp} \geq O\left(\frac{\log^2 n}{\varepsilon^2}\right)\left(\sum_{i \in S_j} |x_i|^p\right)$$

which contradicts the fact that S_j contributes. Now, taking a sum we get

$$\sum_{\mathrm{rank}(v) \leq i \leq n-k} |a_i|^p \leq \sum_{i \in [q]} t_i \left(\frac{v}{2^i}\right)^p \leq O\left(\frac{\log^2 n}{\varepsilon^2}\right) s_j \sum_{i \in [q]} \left(\frac{v}{2^i}\right)^p 2^{ip} \leq v^p s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right). \tag{14}$$

This implies that

$$\left\|\boldsymbol{x}_{-(n-k)}\right\|_p^p \geq \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p - v^p s_j \cdot O\left(\frac{\log^3 n}{\varepsilon^2}\right).$$

Combining (13) and (14) we get immediately that $v^p \cdot s_j \geq (\varepsilon/\log n)^5 \cdot \left\|\boldsymbol{x}_{-\mathrm{rank}(v)}\right\|_p^p$.

□

The rest of the proof of Theorem 6.4 follows from the proof of Theorem 5.1 and from the fact that Holder's inequality gives us $\|\cdot\|_2 \cdot \frac{1}{n^{1/2-1/p}} \leq \|\cdot\|_p$ for $p > 2$.

7 Applications

In this section, we will consider some variants of the problem we have considered thus far.

7.1 Sum of Large Items

Corollary 7.1 (Thresholded F_p estimation). *Given $\boldsymbol{x} \in \mathbb{Z}^n$, $p \in [0,2]$, threshold $\mathcal{T} \geq 0$, and $\varepsilon \in (0,1)$, if we have $a_k^2 \geq (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$, for some constant c where k is the largest integer such that $|a_k| \geq \mathcal{T}$, then there exists a linear sketch that uses $\mathrm{poly}(\log n/\varepsilon)$ bits of space and estimates $\sum_{i \in \mathcal{B}_\mathrm{T}} |x_i|^p$ for $\mathcal{B}_\mathcal{T} = \{i \in n : |x_i| \geq \mathcal{T}\}$ with error $\varepsilon\left(\sum_{i \in \mathcal{B}_\mathcal{T}} |x_i|^p\right) + (1+\varepsilon)\mathcal{T}^p \cdot |x_{(1-\varepsilon)\mathcal{T},\mathcal{T}}|$ with high constant probability where $|x_{(1-\varepsilon)\mathcal{T},\mathcal{T}}|$ denotes the number of coordinates with value $[(1-\varepsilon)\mathcal{T}, \mathcal{T})$.*

The first application is the estimation of the F_p moment of the frequencies that are larger than a given threshold $\mathcal{T}$ in absolute value. At a high level our algorithm, Summed-HH-Estimator (Algorithm 4), divides the items in level sets. Then we take the sum over all level sets associated with a value that is above the threshold. We note that there is extra difficulty here since level sets

Algorithm 4 Summed-HH-Estimator ($\varepsilon \in (0,1]$, $p \in (0,2]$, $k \geq 0$, $\mathcal{T} \geq 0$)

1: Run Level-Set-Estimator (ε).
2: $t \leftarrow \log_{1+\varepsilon}(m) + 1$.
3: Find the largest i such that $\zeta(1+\varepsilon)^{t-i} \geq \mathcal{T}$.
4: **Return** $\left(\sum_{j=0}^{i} \tilde{s}_j \cdot \zeta^p (1+\varepsilon)^{p(t-j)}\right)$.

Algorithm 5 Threshold-Norm-Estimator ($\varepsilon \in (0,1]$, $p \in (0,2]$)

1: Run Level-Set-Estimator (ε).
2: Find the largest $\tilde{k}$ such that the estimated F_p moment of the top-k' frequencies by Algorithm 2 $\geq \tilde{k}^{p+1}$.
3: **Return** $\tilde{k}$.

only allow us to estimate coordinates within a $(1+\varepsilon)$-factor. In particular, we cannot distinguish between coordinates with value $(1-\varepsilon)\mathcal{T}$ and $\mathcal{T}$ causing us to incur additional additive error.

Proof of Theorem 7.1. For the purposes of analysis, take k to be the largest integer such that $|a_k| \geq \mathcal{T}$. Let i be the largest integer number such that $\zeta(1+\varepsilon)^{t-i} \geq \mathcal{T}$ and let k' be the largest integer number such that $|a'_k| \geq \zeta(1+\varepsilon)^{t-i-1}$. We can see the output Z of Algorithm 4 will be the same as the output of Algorithm 2 with input parameter k'. From the assumption we have that $a^2_{k'} \geq \mathrm{poly}(\varepsilon/\log n) \cdot \frac{\|\boldsymbol{x}_{-k'}\|_2^2}{k'}$, as $|a_{k'}|$ can only differ from $|a_k|$ by a $(1 \pm \varepsilon)$-factor. From Theorem 4.1 we have that with high constant probability $\left|Z - \sum_{i=i}^{k'} |a_i|^p\right| \leq \varepsilon \left(\sum_{i=1}^{k'} |a_i|^p\right)$

We next measure the difference between $\left(\sum_{i=1}^{k'} |a_i|^p\right)$ and $\left(\sum_{i=1}^{k} |a_i|^p\right)$. Clearly, it can be upper bounded by $\mathcal{T}^p \cdot |x_{(1-\varepsilon)\mathcal{T},\mathcal{T}}|$ where $|\boldsymbol{x}_{(1-\varepsilon)\mathcal{T},\mathcal{T}}|$ denotes the number of coordinates with value $[(1-\varepsilon)\mathcal{T}, \mathcal{T})$. Thus by triangle inequality we get that with high constant probability, Z is an estimation of $\left(\sum_{i=1}^{k} |a_i|^p\right)$ with error at most $\varepsilon \left(\sum_{i \in \mathcal{B}_{\mathcal{T}}} |x_i|^p\right) + (1+\varepsilon)\mathcal{T}^p \cdot |\boldsymbol{x}_{(1-\varepsilon)\mathcal{T},\mathcal{T}}|$. □

7.2 Extensions of Impact Indices

7.2.1 *Extension of the g-index.*

Corollary 7.2. *Given $\boldsymbol{x} \in \mathbb{Z}^n$, $p \in [1,2]$ and $\varepsilon \in (0,1)$, let k be the largest integer number such that $\sum_{i=0}^{k} |a_i|^p \geq k^{p+1}$. If we have $a^2_{(1-\varepsilon)k} \geq (\varepsilon/\log n)^c \cdot \frac{\left\|\boldsymbol{x}_{-(1-\varepsilon)k}\right\|_2^2}{(1-\varepsilon)k}$ for some constant c, then there exists a linear sketch that uses $\mathrm{poly}(\log n/\varepsilon)$ bits of space and outputs a $\tilde{k}$ such that $(1-\varepsilon)\cdot k \leq \tilde{k} \leq (1+\varepsilon)\cdot(k+1)$ with high constant probability.*

Here we prove Theorem 7.2 with our algorithm Threshold-Norm-Estimator (Algorithm 5). We run Level-Set-Estimator and then use that to find our estimate to k.

Proof of Theorem 7.2. Take k to be the largest integer such that $\sum_{i=1}^{k} |a_i|^p \geq k^{p+1}$. We show that the estimate of the algorithm $\tilde{k}$ incurs an appropriate error. In the rest of the proof, we assume the estimation of each level set in Algorithm 1 satisfies the guarantee, which holds with high constant probability. We first lower bound $\tilde{k}$.

Lemma 7.3. $\tilde{k} \geq (1-\varepsilon) \cdot k$.

Proof. It is sufficient to show that the algorithm will find $(1-\varepsilon) \cdot k$ frequencies such that their *estimated* F_p moment is at least $(1-\varepsilon)^{p+1} \cdot k^{p+1}$.

By definition, the top k frequencies have a F_p norm of at least k^{p+1}. So, the top $(1-\varepsilon)\cdot k$ frequencies have a F_p norm of at least $(1-\varepsilon)\cdot k^{p+1}$. Since we have condition $a^2_{(1-\varepsilon)k} \geq \text{poly}(\varepsilon/\log n)\cdot \frac{\|\boldsymbol{x}_{-(1-\varepsilon)k}\|_2^2}{(1-\varepsilon)k}$, by Theorem 4.1, estimating the F_p moment of these top $(1-\varepsilon)\cdot k$ frequencies incurs at most $\varepsilon \cdot \sum_{i=1}^{(1-\varepsilon)\cdot k} |a_i|^p$ error. So, we have that the estimated F_p norm of the top $(1-\varepsilon)\cdot k$ frequencies is at least

$$(1-\varepsilon)\cdot(1-\varepsilon)\cdot k^{p+1} \geq (1-\varepsilon)^{p+1}\cdot k^{p+1}.$$

Recall that we have $p \in [1,2]$. □

Now we upper bound $\tilde{k}$. We first prove the following which is needed to upper bound $\tilde{k}$.

Lemma 7.4. *If the condition $a_j^2 \geq \text{poly}(\varepsilon/\log n)\cdot \frac{\|\boldsymbol{x}_{-j}\|_2^2}{j}$ is not met, with high probability Algorithm 2 only overestimates $\sum_{i=1}^{j}|a_i|^p$ by a $(1+\varepsilon)$ multiplicative factor.*

Proof. Recall that in the proof of Theorem 4.1, we have the property that with high probability the estimation of s_i for each level set is either a $(1+\varepsilon)$-approximation (when the condition for a_j is met and the level set "contribute") or only an over-estimation by at most a $(1+\varepsilon)$-factor. This means that the output of the algorithm is at most

$$(1+\varepsilon)^2\sum_{i=1}^{j}|a_i|^p = \big(1+O(\varepsilon)\big)\sum_{i=1}^{j}|a_i|^p\ .$$ □

Lemma 7.5. $\tilde{k} \leq (1+\varepsilon)\cdot(k+1)$.

Proof. The algorithm finds the largest $\tilde{k}$ such that the *estimated* F_p norm of the top $\tilde{k}$ frequencies is at least $\tilde{k}^{p+1}$. Therefore, we will argue that for every $k' > (1+\varepsilon)(k+1)$, the estimated F_p norm of the top k' frequencies is (strictly) less than $(k')^{p+1}$.

By definition, the top $(k+1)$ frequencies have a F_p norm strictly less than $(k+1)^{p+1}$. Otherwise, this contradicts that k is the correct answer. Therefore, the F_p norm of the top $c\cdot(k+1)$ frequencies for some c is at most $c\cdot(k+1)^{p+1}$.

From Lemma 7.4 we know that in either case $\sum_{i=1}^{j}|a_i|^p$ is overestimated by at most a $(1+\varepsilon)$ factor for $j = c\cdot(k+1)$. So we have that the estimated F_p norm of the top $c\cdot(k+1)$ frequencies is less than $(1+\varepsilon)\cdot c\cdot(k+1)^{p+1} \leq c^{p+1}\cdot k^{p+1}$ given that $c > (1+\varepsilon)$. □

Combining Lemma 7.3 and Lemma 7.5 we get the correctness of Theorem 7.2. □

7.2.2 *Extension of the h-index and a-index.*

Corollary 7.6. *For input $\boldsymbol{x} \in \mathbb{Z}^n$, take k to be the largest integer such that $|a_k| \geq k$. Given $\boldsymbol{x} \in \mathbb{Z}^n$, $p \in [0,2]$ and $\varepsilon \in (0,1)$, if we have $a^2_{(1+\varepsilon)k} \geq (\varepsilon/\log n)^c \cdot \frac{\|\boldsymbol{x}_{-(1+\varepsilon)k}\|_2^2}{(1+\varepsilon)k}$ for some constant c, then there exists a linear sketch that uses $\text{poly}(\log n/\varepsilon)$ bits of space and estimates $\sum_{i=1}^{k}|a_i|^p$ up to a $(1\pm\varepsilon)$ multiplicative factor with high constant probability.*

Here we prove Theorem 7.6 with our algorithm Index-Norm-Estimator (Algorithm 6). We run Level-Set-Estimator and then use that to estimate the value of k. Then we return the F_p norm of the top k' frequencies.

Proof of Theorem 7.6. Let k denote the largest integer such that $|a_k| \geq k$.

Lemma 7.7. *With high probability, we have $(1-\varepsilon/2)\cdot k \leq k' \leq (1+\varepsilon/4)\cdot k$*

Algorithm 6 Index-Norm-Estimator ($\varepsilon \in (0,1]$, $p \in (0,2]$)

1: Run Level-Set-Estimator ($\varepsilon/10$).
2: Take f to be the vector where the first $\tilde{s}_0$ elements are $\zeta(1+\varepsilon)^t$, the next $\tilde{s}_1$ elements are $\zeta^{(1+\varepsilon)^{(t-1)}}$, the next $\tilde{s}_2$ elements are $\zeta(1+\varepsilon)^{(t-2)}$, and so on.
3: Find the largest $\tilde{k}$ such that $f_{\tilde{k}} \geq \tilde{k}$.
4: **Return** $\sum_{i=1}^{\tilde{k}} f_i^p$.

Proof. We first show that with high probability $\tilde{k} \geq (1-\varepsilon/2)\cdot k$. By definition, there are at least k frequencies each having frequencies at least k. From Lemma 4.6 we know that with high probability for each contributing S_j, we have $\tilde{s}_j \geq (1-\varepsilon/10)\cdot s_j$. From Lemma 5.3 we can get the total number of coordinates in a non-contribute level set is at most $\varepsilon/10$. Put these two things together, we can get that with high probability, the level set estimator will find at least $(1-\varepsilon/2)$ coordinates having frequencies at least $(1-\varepsilon/2)\cdot k$. This means that $k' \geq (1-\varepsilon/2)\cdot k$.

We next show that with high probability $k' \leq (1+\varepsilon/4)\cdot k$. By definition, there are at most k frequencies each having value strictly greater than k. From Lemma 4.6 and Lemma 4.7 we know that with high probability for each contributing S_j, we have $\tilde{s}_j \leq (1+\varepsilon/10)\cdot s_j$. This means that with high probability, the level set estimator can find at most $(1+\varepsilon/4)$ coordinates having frequencies at most $(1+\varepsilon/4)\cdot k$, which implies $k' \leq (1+\varepsilon/4)\cdot k$. □

After obtaining the lower bound and upper bound k', recall that the output of Algorithm 6 is the estimation of F_p moment the top-k' frequencies. Given $(1-\varepsilon/2)\cdot k \leq k' \leq (1+\varepsilon/4)\cdot k$, from Theorem 4.1 we have that with high probability, the output Z of Algorithm 6 satisfies

$$(1-\varepsilon/2)(1-\varepsilon/2)\left(\sum_{i=1}^{k}|a_i|^p\right) \leq Z \leq (1+\varepsilon/4)(1+\varepsilon/2)\left(\sum_{i=1}^{k}|a_i|^p\right).$$

This implies Z is an approximation to $\sum_{i=1}^{k}|a_i|^p$ within error at most $\varepsilon\cdot\left(\sum_{i=1}^{k}|a_i|^p\right)$ □

8 Lower Bounds

8.1 Hardness of F_p of Top-k

8.1.1 Main bound. We show that when the condition $a_k^2 \geq \mathrm{poly}(\varepsilon/\log n)\cdot\frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ does not hold, then $n^{O(1)}$ space is required. Formally, we have the following.

Lemma 8.1. *Suppose that $a_k^2 \leq c\cdot\frac{k}{n}\cdot\frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ for some constant c. Assume $k \leq 0.1n$ and $\varepsilon \in (1/\sqrt{k}, 1]$. Then, any $O(1)$-pass streaming algorithm that outputs a $(1+\varepsilon)$-approximation of $\sum_{i=1}^{k}|a_i|^p$ with high constant probability requires $\Omega(\varepsilon^{-2}n/k)$ bits of space.*

We will consider the following variant of the 2-SUM problem in [12].

Definition 8.2. For binary strings $\boldsymbol{x} = (x_1,\ldots,x_L) \in \{0,1\}^L$ and $\boldsymbol{y} = (y_1,\ldots,y_L) \in \{0,1\}^L$, define $\mathrm{INT}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{L} x_i \wedge y_i$ which is the number of indices i where both x_i and y_i are 1 and define $\mathrm{DISJ}(\boldsymbol{x},\boldsymbol{y})$ to be 1 if $\mathrm{INT}(\boldsymbol{x},\boldsymbol{y}) = 0$ and 0 otherwise.

Definition 8.3 ([12, 48]). Suppose Alice has t binary strings $(\boldsymbol{X}^1,\ldots,\boldsymbol{X}^t)$ where each string $\boldsymbol{X}^i \in \{0,1\}^L$ has length L. Likewise, Bob has t strings $(\boldsymbol{Y}^1,\ldots,\boldsymbol{Y}^t)$ each of length L. $\mathrm{INT}(\boldsymbol{X}^i,\boldsymbol{Y}^i)$ is guaranteed to be either 0 or $\alpha \geq 1$ for each pair of strings $(\boldsymbol{X}^i,\boldsymbol{Y}^i)$. Furthermore, at least $1/2$ of the $(\boldsymbol{X}^i,\boldsymbol{Y}^i)$ pairs are guaranteed to satisfy $\mathrm{INT}(\boldsymbol{X}^i,\boldsymbol{Y}^i) = \alpha$. In the 2-SUM$(t,L,\alpha)$ problem, Alice

and Bob attempt to approximate $\sum_{i\in[t]} \mathrm{DISJ}(\boldsymbol{X}^i, \boldsymbol{Y}^i)$ up to an additive error of $\sqrt{t}$ with constant probability.

Lemma 8.4 ([12]). *To solve 2-SUM(t, L, α) with constant probability, the expected number of bits Alice and Bob need to communicate is $\Omega(tL/\alpha)$.*

We remark that the construction of the input distribution in [48] for this problem has the property such that for every pair (X_j^i, Y_j^i), the probability of $(X_j^i + Y_j^i) \geq 1$ is $\Theta(1)$. Hence, we can also add the promise that there is at most a constant fraction of (X_j^i, Y_j^i) over i, j having $(X_j^i + Y_j^i) = 0$. We will use this in our proof.

We are now ready to prove our Lemma 8.1. At a high level, we will show a reduction from the estimation of $\sum_{i=1}^k |a_i|^p$ to the 2-SUM problem and derive a $O(\varepsilon^{-2}n/k)$ lower bound.

Proof of Theorem 8.1. Without loss of generality, we assume $\varepsilon^2 k$ is an integer. Consider the 2-SUM$(\varepsilon^{-2}, \varepsilon^2 n, \varepsilon^2 k)$ problem with the input instance be $(\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^{\varepsilon^{-2}})$ given to Alice and $(\boldsymbol{Y}^1, \boldsymbol{Y}^2, \ldots, \boldsymbol{Y}^{\varepsilon^{-2}})$ given to Bob. Let $\boldsymbol{X}, \boldsymbol{Y} \in \{0,1\}^n$ be the concatenation of the $\boldsymbol{X}^i$'s and $\boldsymbol{Y}^i$'s respectively. We construct the input array $\boldsymbol{A} = \boldsymbol{x} + \boldsymbol{y}$ for our top-k sum problem as follows. For $\boldsymbol{x}, \boldsymbol{y} \in \{0, k/2\}^n$, we let $x_i = k/2$ if and only if $X_i = 1$, and $y_i = k/2$ if and only if $Y_i = 1$.

We next consider the entry of the array $\boldsymbol{A} = \boldsymbol{x} + \boldsymbol{y}$. It is clear that the coordinates of $\boldsymbol{A}$ are $0, k/2$, or k. Let c be the fraction of the $\boldsymbol{X}^i, \boldsymbol{Y}^i$ such that $\mathrm{INT}(\boldsymbol{X}^i, \boldsymbol{Y}^i) = \alpha = \varepsilon^2 k$. Recall that from the assumption on the input, we know that $\frac{1}{2} \leq c \leq 1$. We first consider the number of coordinates of $\boldsymbol{A}$ that are equal to k. From the definition of $\boldsymbol{A}$ we get that it is equal to $c \cdot \varepsilon^{-2} \cdot \varepsilon^2 k = ck$. Moreover, from the assumption of the input we have all of the top k entries have a value of either $k/2$ or k. Let $\boldsymbol{a}$ be an array with the decreasing order of the array $\boldsymbol{A} = \boldsymbol{x} + \boldsymbol{y}$. Then we have

$$a_k^2 < c \cdot \frac{k}{n} \cdot \frac{\|\boldsymbol{A}_{-k}\|_2^2}{k} .$$

The reduction is given as follows. Suppose that there is a $q = O(1)$ pass streaming algorithm $\mathcal{A}$ which gives a $(1 \pm \varepsilon)$-approximation to the top-k sum of the input array.

(1) Given Alice's strings $(\boldsymbol{X}^1, \ldots, \boldsymbol{X}^{\varepsilon^{-2}})$ each of length $\varepsilon^2 n$, let $\boldsymbol{X}$ be the concatenation of Alice's strings having total length $\varepsilon^{-2}(\varepsilon^2 n) = n$. Similarly let $\boldsymbol{Y} \in \{0,1\}^n$ be the concatenation of Bob's strings.
(2) Alice constructs the vector $\boldsymbol{x}$ as defined above and performs the updates to $\mathcal{A}$ based on $\boldsymbol{x}$. Then Alice sends the memory of the algorithm to Bob.
(3) Bob constructs the vector $\boldsymbol{y}$ as defined above and performs the updates to $\mathcal{A}$ based on $\boldsymbol{y}$.
(4) If the algorithm $\mathcal{A}$ is a $q \geq 2$-pass algorithm, repeat the above process q times.
(5) Run $\mathcal{A}(\boldsymbol{x} + \boldsymbol{y})$ and compute the solution 2-SUM$(\varepsilon^{-2}, \varepsilon^2 n, \varepsilon^2 k)$ based on $\mathcal{A}(\boldsymbol{x} + \boldsymbol{y})$.

To show the correctness of the above algorithm, recall that the output of $\mathcal{A}(\boldsymbol{x} + \boldsymbol{y})$ is $(1 \pm \varepsilon) \sum_{i=1}^k |a_i|^p$ where $\boldsymbol{a}$ is the array $\boldsymbol{A} = \boldsymbol{x} + \boldsymbol{y}$ in non-increasing order. Note that we have shown that there are ck entries among the top k entries of $\boldsymbol{A}$ having value k as well as the top k entries of $\boldsymbol{A}$ having a value of either k or $k/2$. This implies from the output of $\mathcal{A}(\boldsymbol{x} + \boldsymbol{y})$ we can get a $(1 \pm \varepsilon)$-approximation of c, which yields a approximation of $\sum_{i\in[t]} \mathrm{DISJ}(\boldsymbol{X}^i, \boldsymbol{Y}^i)$ with additive error ε^{-1} (as there are a total number of ε^{-2} of the pair $\boldsymbol{X}^i, \boldsymbol{Y}^i$).

□

Corollary 8.5. *Suppose that $a_k^2 \leq c' \cdot \frac{k}{n^c} \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ for some constant $c', c \in (0,1)$. Assume $k \leq 0.1n$ and $\varepsilon \in (1/\sqrt{k}, 1]$. Then, any $O(1)$-pass streaming algorithm that outputs a $(1 \pm \varepsilon)$-approximation of $\sum_{i=1}^k |a_i|^p$ with high constant probability requires $\Omega(\varepsilon^{-2} n^c / k)$ bits of space.*

Proof. Note that we can let $m = n^c/k$ and put the above input instance in the first n^c coordinates in the input to the streaming algorithm, and let the remaining coordinates to be 0. □

8.1.2 Additional bound. We also prove the following lower bound, which shows that when a_k is small enough compared to the F_2 moment of the tail $\|\boldsymbol{x}_{-k}\|_2^2$, even an $O(k^p)$ approximation is hard.

Lemma 8.6. *Suppose that $a_k^2 \le c \cdot \frac{k^3}{n} \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ for some constant c. Then, any $O(1)$-pass streaming algorithm that outputs a $O(k^p)$ approximation of $\sum_{i=1}^k |a_i|^p$ with high constant probability requires $\Omega(n/k^3)$ bits of space.*

We will consider the following k-player set-disjointess problem.

Lemma 8.7 ([22, 27, 30]). *In the k-players set disjointness problem, there are k players with subset $S^1, S^2, \ldots, S^k$, each drawn from $\{1, 2, \cdots, m\}$, and we are promised that either the sets are (1) pairwise disjoint, or (2) there is a unique element j occuring in all the sets. To distinguish the two cases with high constant probability, the total communication is $\Omega(m/k)$ bits.*

Proof of Theorem 8.6. We shall show that for $m = n/k$, if there is a $O(k^p)$-approximation one-pass streaming for the top-k sum, then the k players can use this to solve the above k-plyaer set disjointness problem. This immediately implies an $\Omega(m/k^2) = \Omega(n/k^3)$ lower bound (the extra $1/k$ factor here is due to the fact that the k players need to send the memory status of the algorithm $k-1$ times).

For a player i, let $\boldsymbol{a}^i \in \mathbb{Z}^{n/k}$ denote the binary indicator vector of S^i and $\boldsymbol{x}^i = (\boldsymbol{a}^i, \boldsymbol{a}^i, \ldots, \boldsymbol{a}^i) \in \mathbb{Z}^n$, which is formed by k copies of $\boldsymbol{a}^i$. Consider the vector $\boldsymbol{a} = \sum_i \boldsymbol{a}^i$, in case (1), we have $\boldsymbol{a} = (1, 1, \ldots, 1)$, while in the case (2), $\boldsymbol{a}$ has one coordinate that equals to k and the remaining coordinates that equals to 1. Similarly we have the vector $\boldsymbol{x} = \sum_i \boldsymbol{x}^i = (1, 1, \ldots, 1)$ in case (1) and $\boldsymbol{x}$ has k values equals to k in case (2). Note that in both case we have $a_k = k$ and $\|\boldsymbol{x}_{-k}\|_2^2 = n - k$, which means that

$$a_k^2 < \frac{k^3}{n} \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k} .$$

On the other hand, the sum $\sum_{i=1}^k a_i^p = k$ or k^{p+1} for the two different cases. This means that use an $O(k^p)$ approximation algorithm for the top-k sum problem, the k-players can solve the k-player set disjointness problem with high constant probability, which yields $\Omega(n/k^3)$ bits of space. □

Corollary 8.8. *Suppose that $a_k^2 \le c' \cdot \frac{k^3}{n^c} \cdot \frac{\|\boldsymbol{x}_{-k}\|_2^2}{k}$ for some constant $c', c \in (0, 1)$. Then, any $O(1)$-pass streaming algorithm that outputs a $O(k^p)$ approximation of $\sum_{i=1}^k |a_i|^p$ with high constant probability requires $\Omega(n^c/k^3)$ bits of space.*

Proof. Note that we can let $m = n^c/k$ and put the above input instance in the first n^c coordinates in the input to the streaming algorithm, and let the remaining coordinates to be 0. □

8.2 Hardness of F_p of Trimmed-k

In this section, we give a lower bound for estimating the $\|\boldsymbol{x}_{-k}\|_p^p$, which shows the $\varepsilon k \cdot |a_{k-\varepsilon k}|^p$ error is necessary. We consider the following Gap-Hamming problem.

Definition 8.9. In the Gap-Hamming problem, Alice gets $\boldsymbol{a} \in \{0,1\}^n$ and Bob get $\boldsymbol{b} \in \{0,1\}^n$, and their goal is to determine the Hamming distance $\Delta(\boldsymbol{x}, \boldsymbol{y})$ satisfies $\Delta(\boldsymbol{x}, \boldsymbol{y}) \ge n(c_1 - c_2\varepsilon)$ or $\Delta(\boldsymbol{x}, \boldsymbol{y}) \le n(c_1 - 2c_2\varepsilon)$ with probability $1 - \delta$ for some constant c_1, c_2.

Lemma 8.10 ([28]). *The one-way communication complexity of Gap Hamming is $\Omega(\varepsilon^{-2} \log n \log(1/\delta))$.*

Lemma 8.11. *Any one-pass streaming algorithm that estimating $\|\boldsymbol{x}_{-k}\|_p^p$ with error $\varepsilon k \cdot |a_{k-\varepsilon k}|^p$ and with high constant probability requires $\Omega(\varepsilon^{-2} \log k)$ bits of space.*

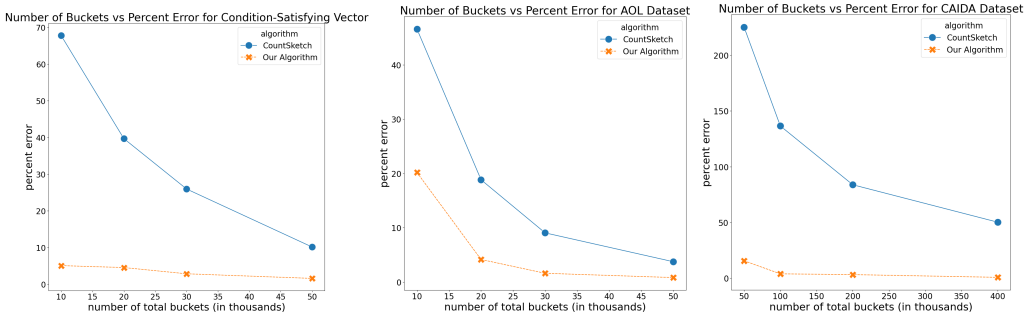


Fig. 1. Number of Buckets vs Error (Synthetic, AOL, CAIDA respectively).

Proof. Given the binary vector $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^{2k}$, let $\boldsymbol{x} \in \mathbb{R}^n = B \cdot \boldsymbol{a}$ for some large value $B = \text{poly}(n)$ and $y = B \cdot \boldsymbol{b}$. Suppose that Alice and Bob want to determine the case where (1) $\Delta(\boldsymbol{a}, \boldsymbol{b}) = k$, and (2) $\Delta(\boldsymbol{a}, \boldsymbol{b}) = k + \varepsilon k$. For the case one, we have $\left\|(\boldsymbol{x} - \boldsymbol{y})_{-k}\right\|_p^p = 0$ whereas for case two we have $\left\|(\boldsymbol{x} - \boldsymbol{y})_{-k}\right\|_p^p = \varepsilon k \cdot B^p$. Hence, if there is a streaming algorithm for the trimmed k-sum problem with at most $\varepsilon k |a_{k-\varepsilon k}|^p$, then Alice and Bob can use it to design a protocol to solve the Gap Hamming problem, from which we get a $\Omega(\varepsilon^{-2} \log k)$ bits of space lower bound. □

Note that we can scale the ε to $\varepsilon' = \text{poly}(\varepsilon / \log n)$, which shows that for any $\text{poly}(\log n / \varepsilon)$ bits of space, the $\text{poly}(\varepsilon / \log n) \cdot |a_{k-\varepsilon k}|^p$ error is best possible.

9 Experiments

In this section, we evaluate the empirical performance of our algorithm on the following three datasets. All of the experiments are conducted on a laptop with a 2.42GHz CPU and 16GB RAM.

- **Synthetic**: we generate a vector of size 10 million where k entries are random integers between 10 and 100 thousand and the rest of the entries are between 1 and 100. This ensures that the condition we set for our top k algorithm is likely met.
- **AOL**[1] [41]: we create the underlying frequency vector based on the query of each entry of the dataset. Specifically, each unique query corresponds to an entry in our underlying frequency vector, and the value of that entry is the number of times the query is in the dataset. The frequency vector had size about 1.2 million with a max frequency of 98,554.
- **CAIDA**[2] [9]: we create the frequency vector based on DNS names. For the subset of the dataset we use, the frequency vector had size about 400 thousand with a max frequency of 48.

Experimental setting. In our implementations of Algorithm 2, we make standard modifications which are done in the practical implementation of streaming algorithms. In particular, we only have a constant number of subsampling levels and level sets in our implementation. In all the three datasets, we consider the F_1 moment of the top $k = 1000$ frequencies for simplicity and interpretability. This can be extended to more general F_p as well.

We compare our algorithm to the classical Count-Sketch across a range of total bucket sizes, using relative error with respect to the ground truth as the evaluation metric. In this context, the bucket size refers to the *total* number of buckets used in the implementation. For Count-Sketch, vector items are hashed into buckets and recovered across multiple independent repetitions. The

[1] https://www.kaggle.com/datasets/dineshydv/aol-user-session-collection-500
[2] https://publicdata.caida.org/datasets/topology/ark/

final estimate for each item is obtained by taking the median of these repetitions. The total number of buckets is therefore the number of buckets per repetition multiplied by the number of repetitions (i.e., the number of medians taken). We vary the number of repetitions from 1 to 10 and report the lowest error achieved, as there is a trade-off between the number of repetitions and the number of buckets per repetition.

Our algorithm incorporates multiple subsampling levels, each containing a smaller Count-Sketch structure with several repetitions. As a result, the total number of buckets is given by the product of the number of subsampling levels, the number of repetitions, and the number of buckets in each Count-Sketch structure.

Results. The comparison result is presented in Figure 1, which suggests that our algorithm consistently outperforms the classical Count-Sketch across a range of total bucket sizes.

For the AOL dataset, as shown in Figure 1, both Count-Sketch and our algorithm exhibit decreasing error as the total number of buckets increases, which aligns with expectations. However, our algorithm consistently outperforms Count-Sketch across all tested bucket sizes. Specifically, for bucket sizes of 10,000, 20,000, 30,000, and 50,000, Count-Sketch yields relative errors of 46.59%, 18.83%, 9.05%, and 3.74%, respectively. In contrast, our algorithm achieves significantly lower errors of 20.18%, 4.15%, 1.61%, and 0.82%. These results indicate that Count-Sketch requires substantially more space to match the accuracy of our method. This performance gap is expected, as Count-Sketch has a linear dependence on k to achieve a provable guarantee, whereas our algorithm does not. For the CAIDA dataset, the overall trends are similar to those observed for the AOL dataset. However, we note that both algorithms required a larger number of total buckets to achieve reasonable error rates. This is likely due to the underlying frequency vector being flatter, with less distinction between the top-k entries and the remainder. Despite this increased difficulty, our algorithm continues to outperform Count-Sketch across all tested bucket sizes. Specifically, for bucket sizes of 50,000, 100,000, 200,000, and 400,000, Count-Sketch yields relative errors of 225.14%, 136.50%, 83.75%, and 50.18%, respectively, while our algorithm achieves substantially lower errors of 15.57%, 3.85%, 3.16%, and 0.71%.

For the synthetic dataset, our algorithm again outperforms Count-Sketch and achieves comparable or better accuracy with significantly less space. For bucket sizes of 10,000, 20,000, 30,000, and 50,000, Count-Sketch yields errors of 67.81%, 39.68%, 25.93%, and 10.11%, respectively, while our algorithm achieves much lower errors of 5.05%, 4.52%, 2.82%, and 1.56%. Notably, the number of buckets required to obtain low error is relatively small compared to the size of the underlying frequency vector. This aligns with our expectations, as the synthetic dataset was constructed such that the top-k entries are significantly larger than the rest, making them easier to identify accurately.

Acknowledgments

The authors were supported in part by Office of Naval Research award number N000142112647 and a Simons Investigator Award. Honghao Lin was supported in part by a CMU Paul and James Wang Sercomm Presidential Graduate Fellowship. Hoai-An Nguyen was supported in part by NSF GRFP award number DGE2140739.

References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. 1999. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.* 58, 1 (1999), 137–147. doi:10.1006/JCSS.1997.1545

[2] Sergio Alonso, Francisco Javier Cabrerizo, Enrique Herrera-Viedma, and Francisco Herrera. 2009. h-Index: A Review Focused in its Variants, Computation and Standardization for Different Scientific Fields. *J. Informetrics* 3, 4 (2009), 273–289. doi:10.1016/J.JOI.2009.04.001

[3] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. 2011. Streaming Algorithms via Precision Sampling. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE, 363–372.

[4] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. 2004. An Information Statistics Approach to Data Stream and Communication Complexity. *J. Comput. System Sci.* 68, 4 (2004), 702–732. doi:10.1016/j.jcss.2003.11.006

[5] Arnab Bhattacharyya, Palash Dey, and David P. Woodruff. 2016. An Optimal Algorithm for l1-Heavy Hitters in Insertion Streams and Related Problems. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Tova Milo and Wang-Chiew Tan (Eds.). ACM, 385–400. doi:10.1145/2902251.2902284

[6] Mark Braverman and Or Zamir. 2024. Optimality of Frequency Moment Estimation. *CoRR* abs/2411.02148 (2024). arXiv:2411.02148 doi:10.48550/ARXIV.2411.02148

[7] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. 2017. 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 2017, Chicago, IL, USA, May 14-19, 2017*, Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts (Eds.). ACM, 361–376. doi:10.1145/3034786.3034798

[8] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, and David P. Woodruff. 2016. Beating CountSketch for Heavy Hitters in Insertion Streams. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 740–753. doi:10.1145/2897518.2897558

[9] Center for Applied Internet Data Analysis (CAIDA). 2008–ongoing. The CAIDA UCSD IPv4 Routed /24 DNS Names Dataset - May 6, 2025. https://www.caida.org/catalog/datasets/ipv4_dnsnames_dataset/. Accessed: 2025-05-06.

[10] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. 2003. Near-Optimal Lower Bounds on the Multi-Party Communication Complexity of Set Disjointness. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*. IEEE Computer Society, 107–117. doi:10.1109/CCC.2003.1214414

[11] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. 2002. Finding Frequent Items in Data Streams. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2380)*, Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan J. Eidenbenz, and Ricardo Conejo (Eds.). Springer, 693–703. doi:10.1007/3-540-45465-9_59

[12] Yu Cheng, Max Li, Honghao Lin, Zi-Yi Tai, David P. Woodruff, and Jason Zhang. 2024. Tight Lower Bounds for Directed Cut Sparsification and Distributed Min-Cut. *Proc. ACM Manag. Data* 2, 2 (2024), 85. doi:10.1145/3651148

[13] Kenneth L. Clarkson and David P. Woodruff. 2015. Sketching for M-Estimators: A Unified Approach to Robust Regression. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, Piotr Indyk (Ed.). SIAM, 921–939. doi:10.1137/1.9781611973730.63

[14] Edith Cohen and Haim Kaplan. 2007. Bottom-k sketches: better and more efficient estimation of aggregates. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2007, San Diego, California, USA, June 12-16, 2007*, Leana Golubchik, Mostafa H. Ammar, and Mor Harchol-Balter (Eds.). ACM, 353–354. doi:10.1145/1254882.1254926

[15] Majid Daliri, Juliana Freire, Christopher Musco, Aécio S. R. Santos, and Haoxiang Zhang. 2024. Sampling Methods for Inner Product Sketching. *Proc. VLDB Endow.* 17, 9 (2024), 2185–2197.

[16] Elbert Du, Michael Mitzenmacher, David Woodruff, and Guang Yang. 2023. Separating k-Player from t-Player One-Way Communication, with Applications to Data Streams. *Theory of Computing* 19, 10 (2023), 1–44. doi:10.4086/toc.2023.v019a010

[17] Talya Eden, Shweta Jain, Ali Pinar, Dana Ron, and C. Seshadhri. 2018. Provable and Practical Approximations for the Degree Distribution using Sublinear Graph Samples. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 449–458. doi:10.1145/3178876.3186111

[18] Leo Egghe. 2006. Theory and Practise of the g-index. *Scientometrics* 69, 1 (2006), 131–152. doi:10.1007/S11192-006-0144-7

[19] Gereon Frahling, Piotr Indyk, and Christian Sohler. 2008. Sampling in Dynamic Data Streams and Applications. *Internat. J. Comput. Geom. Appl.* 18, 1-2 (2008), 3–28. doi:10.1142/S0218195908002520

[20] Sumit Ganguly. 2011. Polynomial Estimators for High Frequency Moments. *CoRR* abs/1104.4552 (2011).

[21] Priya Govindan, Morteza Monemizadeh, and S. Muthukrishnan. 2017. Streaming Algorithms for Measuring H-Impact. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts (Eds.). ACM, 337–346. doi:10.1145/3034786.3056118

[22] André Gronemeier. 2009. Asymptotically Optimal Lower Bounds on the NIH-Multi-Party Information Complexity of the AND-Function and Disjointness. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings (LIPIcs, Vol. 3)*, Susanne Albers and Jean-Yves Marion (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 505–516. doi:10.4230/LIPICS.STACS.2009.1846

[23] Jorge E. Hirsch. 2005. An Index to Quantify an Individual's Scientific Research Output. *Proc. Natl. Acad. Sci. USA* 102, 46 (2005), 16569–16572. doi:10.1073/PNAS.0507655102

[24] Piotr Indyk. 2006. Stable Distributions, Pseudorandom Generators, Embeddings, and Data Stream Computation. *J. ACM* 53, 3 (2006), 307–323. doi:10.1145/1147954.1147955

[25] Piotr Indyk and David Woodruff. 2005. Optimal Approximations of the Frequency Moments of Data Streams. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 202–208. doi:10.1145/1060590.1060621

[26] Rajesh Jayaram and David Woodruff. 2021. Perfect L_p Sampling in a Data Stream. *SIAM J. Comput.* 50, 2 (2021), 382–439. doi:10.1137/18M1229912

[27] T. S. Jayram. 2009. Hellinger Strikes Back: A Note on the Multi-Party Information Complexity of AND. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings (Lecture Notes in Computer Science, Vol. 5687)*, Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P. Rolim (Eds.). Springer, 562–573. doi:10.1007/978-3-642-03685-9_42

[28] T. S. Jayram and David P. Woodruff. 2011. Optimal Bounds for Johnson-Lindenstrauss Transforms and Streaming Problems with Sub-Constant Error. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, Dana Randall (Ed.). SIAM, 1–10. doi:10.1137/1.9781611973082.1

[29] Hossein Jowhari, Mert Saglam, and Gábor Tardos. 2011. Tight Bounds for Lp Samplers, Finding Duplicates in Streams, and Related Problems. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, Maurizio Lenzerini and Thomas Schwentick (Eds.). ACM, 49–58. doi:10.1145/1989284.1989289

[30] Akshay Kamath, Eric Price, and David P. Woodruff. 2021. A Simple Proof of a New Set Disjointness with Applications to Data Streams. In *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference) (LIPIcs, Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 37:1–37:24. doi:10.4230/LIPICS.CCC.2021.37

[31] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. 2010. 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 2010, Austin, Texas, USA, January 17-19, 2010*, Moses Charikar (Ed.). SIAM, 1161–1178. doi:10.1137/1.9781611973075.93

[32] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. 2010. An Optimal Algorithm for the Distinct Elements Problem. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*, Jan Paredaens and Dirk Van Gucht (Eds.). ACM, 41–52. doi:10.1145/1807085.1807094

[33] Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P. Woodruff, and Mobin Yahyazadeh. 2017. Optimal Lower Bounds for Universal Relation, and for Samplers and Finding Duplicates in Streams. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, Chris Umans (Ed.). IEEE Computer Society, 475–486. doi:10.1109/FOCS.2017.50

[34] Yi Li, Honghao Lin, and David P. Woodruff. 2024. Optimal Sketching for Residual Error Estimation for Matrix and Vector Norms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=RsJwmWvE6Q

[35] Yi Li and David P. Woodruff. 2013. 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 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8096)*, Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim (Eds.). Springer, 623–638. doi:10.1007/978-3-642-40328-6_43

[36] Honghao Lin, Hoai-An Nguyen, William Swartworth, and David P. Woodruff. 2026. Unbiased Insights: Optimal Streaming Algorithms for ℓ_p Sampling, the Forget Model, and Beyond. In *Symposium on Principles of Database Systems 2026*.

[37] Linyuan Lü, Tao Zhou, Qian-Ming Zhang, and H. Eugene Stanley. 2016. The H-Index of a Network Node and its Relation to Degree and Coreness. *Nature Communications* 7, 1 (April 2016), 1–7. doi:10.1038/ncomms10168

[38] Morteza Monemizadeh and David P. Woodruff. 2010. 1-Pass Relative-Error L_p-Sampling with Applications. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, Moses Charikar (Ed.). SIAM, 1143–1160. doi:10.1137/1.9781611973075.92

[39] Jelani Nelson and Huacheng Yu. 2022. Optimal Bounds for Approximate Counting. In *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Leonid Libkin and Pablo Barceló (Eds.). ACM, 119–127. doi:10.1145/3517804.3526225

[40] Noam Nisan. 1992. Pseudorandom Generators for Space-Bounded Computation. *Combinatorica* 12, 4 (1992), 449–461. doi:10.1007/BF01305237

[41] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A Picture of Search. In *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1, 2006 (ACM International Conference Proceeding Series, Vol. 152)*, Xiaohua Jia (Ed.). ACM, 1. doi:10.1145/1146847.1146848

[42] Fabián Riquelme. 2015. Measuring User Influence on Twitter: A survey. *CoRR* abs/1508.07951 (2015). arXiv:1508.07951 http://arxiv.org/abs/1508.07951

[43] Jake Roth and Ying Cui. 2023. On $O(n)$ Algorithms for Projection onto the Top-k-sum Constraint. *CoRR* abs/2310.07224 (2023). arXiv:2310.07224 doi:10.48550/ARXIV.2310.07224

[44] Ahmet Erdem Sariyüce, C. Seshadhri, and Ali Pinar. 2018. Local Algorithms for Hierarchical Dense Subgraph Discovery. *Proc. VLDB Endow.* 12, 1 (2018), 43–56. doi:10.14778/3275536.3275540

[45] David Woodruff. 2004. Optimal Space Lower Bounds for all Frequency Moments. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 167–175.

[46] David P. Woodruff. 2016. New Algorithms for Heavy Hitters in Data Streams. In *19th International Conference on Database Theory, ICDT 2016, Bordeaux, France, March 15-18, 2016 (LIPIcs, Vol. 48)*, Wim Martens and Thomas Zeume (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 4:1–4:12. doi:10.4230/LIPICS.ICDT.2016.4

[47] David P. Woodruff, Shenghao Xie, and Samson Zhou. 2025. Perfect Sampling in Turnstile Streams Beyond Small Moments. arXiv:2504.07237 [cs.DS] https://arxiv.org/abs/2504.07237

[48] David P. Woodruff and Qin Zhang. 2014. An Optimal Lower Bound for Distinct Elements in the Message Passing Model. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 718–733. doi:10.1137/1.9781611973402.54

Received December 2025; accepted February 2025