Unbiased Insights: Optimal Streaming Algorithms for ℓ_p Sampling, the Forget Model, and Beyond

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

We study ℓ_p sampling and frequency moment estimation in a single-pass insertion-only data stream. For $p \in (0, 2)$, we present a nearly space-optimal approximate ℓ_p sampler that uses $\widetilde{O}(\log n \log(1/\delta))$ bits of space and for $p = 2$, we present a sampler with space complexity $\widetilde{O}(\log^2 n \log(1/\delta))$. This space complexity is optimal for $p \in (0, 2)$ and improves upon prior work by a $\log n$ factor. We further extend our construction to a continuous ℓ_p sampler, which outputs a valid sample index at every point during the stream.

Leveraging these samplers, we design nearly unbiased estimators for F_p in data streams that include forget operations, which reset individual element frequencies and introduce significant non-linear challenges. As a result, we obtain near-optimal algorithms for estimating F_p for all p in this model, originally proposed by Pavan, Chakraborty, Vinodchandran, and Meel [PODS'24], resolving all three open problems they posed.

Furthermore, we generalize this model to what we call the suffix-prefix deletion model, and extend our techniques to estimate entropy as a corollary of our moment estimation algorithms. Finally, we show how to handle arbitrary coordinate-wise functions during the stream, for any $g \in \mathcal{G}$, where $\mathcal{G}$ includes all (linear or non-linear) contraction functions.

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

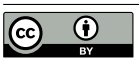
Additional Key Words and Phrases: streaming, sketching, samplers, frequency moments

ACM Reference Format:
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. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 122 (May 2026), 26 pages. https://doi.org/10.1145/3801918

1 Introduction

The *streaming* model of computation, motivated by the need to process massive datasets in applications such as network monitoring and real-time analytics, has been widely studied. In this model, an underlying frequency vector $\boldsymbol{f} \in \mathbb{Z}^n$, initially set to the all-zero vector, is updated over time by a stream of operations. These operations typically consist of coordinate-wise increments (and sometimes decrements). Formally, the stream consists of updates of the form (i, w_t), meaning $f_i \leftarrow f_i + w_t$. When w_t can be both positive and negative, the model is referred to as a turnstile stream. In contrast, when w_t can only be positive, the model is referred to as an insertion-only stream.

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; William Swartworth, Carnegie Mellon University, Pittsburgh, USA, wswartworth@gmail.com; David P. Woodruff, Carnegie Mellon University, Pittsburgh, USA, dwoodruf@cs.cmu.edu.



ACM 2836-6573/2026/5-ART122
https://doi.org/10.1145/3801918

A central problem in this setting is the estimation of *frequency moments*, defined as $F_p = \sum_{i\in[n]} f_i^p$ for a parameter $p \geq 0$. The (common) goal is to compute a multiplicative $(1 \pm \varepsilon)$-approximation to F_p using a constant number of passes over the stream and sublinear space. This problem has been studied in various streaming models, including insertion-only, turnstile, and random-order streams since the work of Alon, Matias, and Szegedy [1]. Frequency moments have a wide range of applications. For instance, F_0—the number of distinct elements—arises in query optimization and database systems [10]; F_1 corresponds to approximate counting [20]; higher moments F_p for $p \geq 2$ capture data skew and have been used in algorithms for data partitioning [8] and error estimation [12].

A closely related and extensively studied problem is ℓ_p *sampling*. Here, given the frequency vector $\boldsymbol{f} \in \mathbb{Z}^n$, the goal is to return an index $i \in \{1, 2, \ldots, n\}$ with probability $|f_i|^p/\|\boldsymbol{f}\|_p^p$. In particular, an approximate ℓ_p sampler is one that returns an index i with probability $(|f_i^p|/\|\boldsymbol{f}\|_p^p)\cdot(1\pm\varepsilon)+\mathrm{poly}(1/n)$ for $\varepsilon \in (0, 1)$, while a perfect ℓ_p corresponds to the case $\varepsilon = 0$. ℓ_p samplers have proved an important tool in the streaming model and have been used in algorithms to estimate the F_p moment, finding heavy hitters, finding duplicates in streams, and cascaded norm estimation [2, 5, 14, 18].

ℓ_p sampling. The work of Jayaram and Woodruff [13] presents perfect ℓ_p samplers for turnstile streams that use $\widetilde{O}(\log^2 n \log(1/\delta))$ bits of space for $p \in (0, 2)$ and $O(\log^3 n \log(1/\delta))$ bits for $p = 2$. On the other hand, Kapralov, Nelson, Pachocki, Wang, and Woodruff [16] establishes a lower bound of $\Omega(\log^2 n \log(1/\delta))$ for ℓ_p sampling in turnstile streams, implying that the sampler of [13] is tight up to a $\mathrm{poly}(\log\log n)$ factor when $p \in (0, 2)$. However, the situation for insertion-only streams remains less clear. A trivial lower bound of $\Omega(\log n)$ is known since that many bits is required to output an index, but no upper bound better than that of [13] has been established even in this restricted model. This gap naturally leads to the following question:

> *Question 1: Is it possible to design a one-pass ℓ_p sampling algorithm that uses $\widetilde{O}(\log n)$ space in a insertion-only stream?*

Forget model. As mentioned, a typical application of ℓ_p sampling is the estimation of the frequency moment F_p, a problem extensively studied in both insertion-only and turnstile streaming models. To better reflect practical needs and to explore the boundaries of sketching-based estimation, in this work we consider a more general class of nonlinear update operations—specifically, *forget* requests.

This model, introduced by Pavan, Chakraborty, Vinodchandran, and Meel [22], permits updates that reset individual coordinates of the frequency vector to zero. Forget updates are motivated by several practical scenarios, including privacy-preserving deletion (e.g., to comply with the European Union's General Data Protection Regulation, which grants individuals the "Right to be Forgotten" [22]), storage optimization via removal of stale data, and the ability to discard faulty or irrelevant information without restarting the stream. Additionally, forget operations enable post hoc analysis on subsets of the data—for example, by zeroing out coordinates that fall outside a region of interest after the stream concludes.

Unlike decrements, forget operations cannot be simulated in the turnstile model without access to the current value of the coordinate being reset. This added nonlinearity introduces substantial challenges: standard linear sketching techniques become too biased and are no longer suitable for accurate F_p estimation. As a result, designing unbiased or low-bias estimators in the presence of such operations requires fundamentally new techniques beyond classical streaming tools.

It is known that no sublinear-space algorithm can provide a $(1 \pm \varepsilon)$-approximation to the F_p moments under unrestricted forgets. In particular, [22] established an $\Omega(n)$ space lower bound for this setting. To overcome this barrier, they proposed the α-RFDS model, in which the algorithm is guaranteed that the final value of the F_p moment is at least a $(1 - \alpha)$ fraction of what it would have

been had no forgets occurred. This permits, for example, the moment to decrease by an arbitrarily large constant factor, as long as the overall loss is bounded.

Under this promise, the work of [22] designed $(1 \pm \varepsilon)$-approximation algorithms for estimating F_0 using $O\left(\frac{1}{\varepsilon^2(1-\alpha)} \log n\right)$ bits of space and for F_1 using $O\left(\frac{1}{\varepsilon^2(1-\alpha)} \log^2 n\right)$ bits of space. They also posed several open questions regarding their optimality and the feasibility of extending such results to other F_p moments, such as F_2:

> *Question 2: What is the tight space complexity for F_0 and F_1 estimation in the α-RFDS model?*
> *Question 3: Can we design space-efficient algorithms for estimating other F_p moments in this model?*

More extensions. Beyond answering the the above three questions, we consider several extensions. We introduce a *prefix-suffix deletion* model, which strictly generalizes forget operations by restricting deletions based on time. A prefix deletion request of the form `prefixdelete`(i, x) removes all occurrences of element x that appeared at or before time i, while a suffix deletion request `suffixdelete`(i, x) removes all occurrences of x from time i up to the current time T when the request is received. This model allows fine-grained control over the retained data and captures a broader range of post hoc filtering operations. For example, at the end of the stream, one can use prefix and suffix deletions to zoom in on a subset of elements in a particular time interval. The model is also closely related to the sliding window paradigm in streaming algorithms [4, 7]. As an illustration, suppose we want to compute a statistic over a sliding window of size w; we can run overlapping sub-streams of length $2w$ every w steps, and apply appropriate prefix deletions to recover the exact window. We believe that prefix and suffix deletions—as well as forget requests—model a wide range of realistic scenarios where data retention is limited by time, privacy, or context-specific relevance.

Moreover, since supporting only forget operations may not adequately capture practical scenarios, we initiate a broader study of non-linear operations in insertion-only streams, significantly generalizing the forget model. Specifically, we consider a family of functions $\mathcal{G}$ such that stream updates may take the form (i, g) for $g \in \mathcal{G}$, which applies the transformation $f_i \leftarrow g(f_i)$. The function family $\mathcal{G}$ includes all contraction functions, encompassing both linear and nonlinear transformations—such as $\sqrt{x}$ and x/c for a constant c—that reduce the magnitude of their input. Many functions in this family are nonlinear (as in the case of forgets), and thus introduce new challenges in designing accurate estimators. As in the α-RFDS model, we assume that these operations do not reduce the final value of the F_p moment by more than a $(1 - \alpha)$ multiplicative factor compared to the version of the stream with only insertions.

1.1 Our Contributions

In this paper, we resolve all of the above open questions.

ℓ_p sampling for $p \in (0, 2]$. We present an ℓ_p sampler for insertion-only streams.

Theorem 1.1 (ℓ_p sampler)**.** *For any constant $p \in (0, 2]$, there is a one-pass streaming algorithm that runs in space $\widetilde{O}(\log^{c(p)} n \log(1/\delta))$, and outputs an index i such that with probability at least $1 - \delta$ we have for every $j \in [n]$,*

$$\Pr[i = j] = \frac{|f_j|^p}{\|f\|_p^p} \pm \frac{1}{\mathrm{poly}(n)} .$$

Here $c(p) = 1$ when $0 < p < 2$ and $c(p) = 2$ when $p = 2$.

We note that Theorem 1.1 improves upon the results of [13] by a factor of $\log n$ for insertion-only streams. For $p \in (0, 2)$, our ℓ_p sampler achieves *optimal* space complexity up to a $\mathrm{poly}(\log \log n)$ factor. Unlike the *perfect* sampler from [13], ours does not explicitly output FAIL upon failure. However, when the required failure probability is $\delta = \mathrm{poly}(1/n)$ for both samplers, our sampler retains all the desirable properties of the perfect sampler in [13] while still shaving a $O(\log n)$ factor in the space.

We next extend our ℓ_p sampler to the continuous case where we are required to have a sampled index at each time step during the stream.

Theorem 1.2 (Continuous ℓ_p sampler)**.** *Consider a stream of length m (and therefore with m time steps). Let $F_{p,[0,t]}$ be the F_p moment of $\boldsymbol{f}$ after the first t time steps, and let $f_{i,[0,t]}$ be the frequency of coordinate i after the first t time steps. There is an algorithm $\mathcal{A}$ that produces a series of m outputs, $z_1, \ldots, z_m$ such that each $z_j \in \{1, \ldots, n\}$, one at each time step with the following properties:*

- $\mathbf{Pr}(z_t = i) = \frac{f_{i,[0,t]}^p}{F_{p,[0,t]}} \pm \frac{1}{\mathrm{poly}(n)}$.
- *Only $\widetilde{O}(\log^{c(p)} n \cdot \log(1/\delta))$ of the z_j's are unique.*
- *$\mathcal{A}$ uses $\widetilde{O}(\log^{c(p)} n \cdot \log(1/\delta))$ bits of space.*
- *$\mathcal{A}$ succeeds with probability at least $1 - \delta$.*

Here $c(p) = 1$ from $0 < p < 2$ and $c(p) = 2$ for $p = 2$.

F_p estimation with forgets. We give nearly-optimal algorithms for estimating F_p moment in the α-RFDS model for all range of p.

Theorem 1.3 (Main result for F_p estimation with forgets)**.** *Given $0 < p < \infty$. There is a one-pass streaming algorithm that uses m bits of space and with high constant probability outputs a $(1 \pm \varepsilon)$-approximation to the value of F_p in the α-RFDS model, where*

$$m = \begin{cases} \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)} \cdot \log n\right), & p \in (0, 2), \\ \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)} \cdot \log^2 n\right), & p = 2, \\ \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}} \cdot n^{1-2/p} \cdot \log^2 n\right), & p \in (2, \infty). \end{cases}$$

We also provide matching lower bounds.

Theorem 1.4 (Lower bounds for F_p estimation with forgets)**.** *Any streaming algorithm that gives a $(1 \pm \varepsilon)$-approximation to the F_p moment in the α-RFDS model requires $\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)} \cdot \log n\right)$ bits of space for $p \in [0, 2]$ and $\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}} \cdot n^{1-2/p}\right)$ bits of space for $p > 2$.*

We summarize our results in Table 1. In addition, we give a surprisingly simple algorithm that achieves the optimal space, up to $\log \log$ factors, when $p = 1$. Finally, one might wonder why along with [22] we only consider insertion-only streams rather than allow for turnstile updates. We address this by giving an $\widetilde{\Omega}(n)$ lower bound against F_p moment estimation for $p = 0$ and $p \geq 1$ in turnstile streams (including strict turnstile streams), even for obtaining constant-factor approximations when $1 - \alpha = O(1)$.

Entropy estimation with forgets. We extend our results for F_p estimation to entropy estimation in the forget model. Formally, assuming that $|H(\boldsymbol{f}) - H(\widetilde{\boldsymbol{f}})| \leq \alpha H(\widetilde{\boldsymbol{f}})$ and $F_1(\boldsymbol{f}) \geq (1 - \alpha) F_1(\widetilde{\boldsymbol{f}})$, we present a one-pass streaming algorithm that estimates the entropy of $\boldsymbol{f}$ to within an additive error of ε, using $\widetilde{O}\left(\frac{1}{\varepsilon^2(1-t\alpha)} \cdot \mathrm{poly} \log n\right)$ bits of space, where $t = 1 + O(1/\log n)$.

F_p	Prior Work	Our Work	Lower Bound
$p = 0$	$O\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$ [22]	–	$\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$
$p = 1$	$O\left(\frac{1}{\varepsilon^2(1-\alpha)}\log^2 n\right)$[22]	$\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$	$\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$
$p \in (0,2)$	–	$\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$	$\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$
$p = 2$	–	$\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\log^2 n\right)$	$\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)}\log n\right)$
$p > 2$	–	$\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}}\cdot n^{1-2/p}\right)$	$\Omega\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}}\cdot n^{1-2/p}\right)$

Table 1. Upper Bounds and Lower Bounds for F_p estimation with forgets. $\widetilde{O}(\cdot)$ hides poly log factors.

Heavy-hitters with general operations. We develop an algorithm for estimating F_p norms with $p > 0$ over data streams that support general update operations. Specifically, updates are of the form (i, g), where $g \in \mathcal{G}$ is a contracting function applied as $f_i = g(f_i)$. A crucial component of our approach is the design of an ℓ_2-heavy-hitters data structure capable of handling such updates efficiently in a streaming setting. This structure allows us to identify ℓ_2-heavy hitters using $\widetilde{O}\left(\frac{1}{(1-\alpha)\varepsilon^2}\cdot\log n\right)$ bits of space. Building on this as a black box, we further construct an algorithm to find ℓ_p-heavy hitters, requiring $\widetilde{O}\left(\frac{1}{(1-\alpha)^{2/p}\varepsilon^2}n^{1-2/p}\right)$ bits of space. Importantly, our method not only identifies the indices of the heavy hitters but also approximates their values within an additive error of $\varepsilon\|\boldsymbol{f}\|_2$.

F_p estimation with general operations. For F_p estimation for $p > 0$ on a stream with general operations from $\mathcal{G}$, we design an algorithm that given $\varepsilon \in (0,1)$ obtains a $(1\pm\varepsilon)$-approximation to F_p with constant probability using $\widetilde{O}\left(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^{2+4/p}}n^{1-2/p}\right)$ bits of space for $p > 2$ and $\widetilde{O}\left(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^{2+4/p}}\cdot\operatorname{poly}\log n\right)$ bits of space for $0 < p \le 2$.

Prefix-suffix deletion model. For the F_1 moment, we design an algorithm that, given parameters $\varepsilon, \delta \in (0,1)$, computes a $(1\pm\varepsilon)$ approximation to F_1 with probability at least $1-\delta$, using $\widetilde{O}\left(\frac{1}{(1-\alpha)\varepsilon^2}\log n\log\left(\frac{1}{\delta}\right)\right)$ bits of space.

We also present an algorithm that, given $\varepsilon \in (0,1)$, estimates all coordinates of the frequency vector $\boldsymbol{f}$ to within $\varepsilon\|\boldsymbol{f}\|_2$ additive error using $\widetilde{O}\left(\frac{1}{(1-\alpha)\varepsilon^2}\cdot\log^2 n\right)$ bits of space. This coordinate-wise estimator serves as a key subroutine in our final algorithm for approximating F_p for any $p \ge 2$. Specifically, for $\varepsilon \in (0,1)$ and $p \ge 2$, our algorithm outputs a $(1\pm\varepsilon)$ approximation to F_p using $\widetilde{O}\left(\frac{n^{1-2/p}}{(1-\alpha)^{2/p}\varepsilon^{2+4/p}}\right)$ bits of space.

1.2 Our Techniques

ℓ_p sampling for $p \in (0,2]$. At a high level, we adopt a similar approach to that of [13], which leverages the order statistics of exponential random variables. Specifically, define $z_i = f_i/e_i^{1/p}$, where e_i are i.i.d. exponential random variables, and let $D(1) = \arg\max_i |z_i|$. Then it holds that

$$\mathbf{Pr}[D(1) = i] = \frac{|f_i|^p}{\|\boldsymbol{f}\|_p^p}\ .$$

Moreover, with probability $\Omega(1)$ (or $\Omega(1/\log n)$ when $p = 2$), we have $|z_{D(1)}| = \Theta(1)\cdot\|\boldsymbol{z}_{-D(1)}\|_2$. The core idea in [13] is to generate multiple independent groups of exponential random variables

and identify the maximal index whenever this event occurs. To achieve this, [13] employs a CountSketch-style data structure, which requires at least $\Omega(\log^2 n)$ bits of space.

Recall that we are working in an insertion-only stream. We now turn our attention to another data structure, BPTree [3], which identifies (ε, ℓ_2)-heavy hitters using a more space-efficient approach, requiring only $O(\varepsilon^{-1} \log n)$ bits of space. The hope is to use BPTree to identify the maximal index. However, this introduces a new challenge. To achieve the desired probabilistic guarantee, the exponential random variables need to exhibit full randomness. Since BPTree is not a linear sketch, we cannot directly apply the derandomization technique from [13]. Moreover, standard derandomization approaches would increase the $\log n$ factor in the space complexity.

To overcome this barrier, we open the procedure of BPTree. At a high level, the key sub-routine of the BPTree is the following: assuming there is a single coordinate f_i that constitutes an $O(1)$-fraction of the total mass of the frequency vector $\boldsymbol{f}$, the goal is to identify this $O(1)$-heavy hitter, To solve this problem, BPTree takes $O(\log n)$ rounds and in each round, it decides one bit of the heavy coordinate's identity. We instead give an alternative approach to solve the same heavy-hitter problem in the same $\widetilde{O}(\log n)$ space via inspiration from adaptive sparse recovery [11]. The key advantage of our method is that it requires only $O(\log \log n)$ rounds of linear sketches, compared to the $\log n$ rounds required by BPTree. This structural difference enables us to apply the derandomization technique from [13], while incurring only an $O(\text{poly}(\log \log n))$ space overhead.

In particular, consider a fixed $O(\log \log n)$ branching points. That is, we fix the $O(\log \log n)$ branching time $t_1, t_2, \ldots, t_r$ and the corresponding subset $S_1, S_2, \ldots, S_r$ it goes into. We refer to this fixed sequence as a *path*. The key observation here is that once these branching points are fixed, our algorithm will behave as a linear sketch. This allows us to apply the derandomization result from [13] to show that

$$\left|\mathbf{Pr}(\texttt{path}) - \mathbf{Pr}'(\texttt{path})\right| \leq \frac{1}{n^{O(\log \log n)}}$$

where $\mathbf{Pr}(\texttt{path})$ denotes the true probability of taking a path in the random oracle model and $\mathbf{Pr}'(\texttt{path})$ denotes the probability of taking a path after derandomizing. Note that there are at most $n^{O(\log \log n)}$ paths. Hence, we can take a union bound over all possible paths to obtain the desired derandomization guarantee.

F_p estimation with forgets ($0 < p \leq 2$). Let $\boldsymbol{f}$ denote the frequency vector without the forget operations, and let $\boldsymbol{g}$ denote the actual frequency vector after the forget operations. Our estimator is based on the ℓ_p sampling algorithm. At a high level, let j be an index sampled from the ℓ_p sampler, where

$$\mathbf{Pr}[j = i] = \frac{|f_i|^p}{\|\boldsymbol{f}\|_p^p}.$$

Suppose we can construct an unbiased estimator $\widehat{g_j^p}$ for g_j^p, and an unbiased estimator $\widehat{\frac{1}{p_j}}$ for $\frac{1}{p_j}$, where $p_j = |f_j|^p / \|\boldsymbol{f}\|_p^p$ is the sampling probability of index j. Then $\widehat{\frac{1}{p_j}} \cdot \widehat{g_j^p}$ serves as an unbiased estimator for $F_p(\boldsymbol{g}) = \sum_{i=1}^n g_i^p$. Moreover, if we can show that the variance of this estimator is bounded, we can reduce the overall variance by averaging over multiple independent samples. However, several technical challenges remain:

- How can we obtain an unbiased estimator of g_j^p? This is non-trivial because the stream allows forget operations, which may make it difficult to estimate g_j accurately.
- How can we obtain an unbiased estimator of the inverse sampling probability $1/p_j$? Note that even if an estimator h_j satisfies $\mathbf{E}[h_j] = f_j$, it does not follow that $\mathbf{E}[1/h_j] = 1/f_j$.

We address these challenges in the following sections.

Unbiased estimator of g_j. Our estimator is motivated by the following crucial observation. Suppose that $g_j < (1-\varepsilon)f_j$. Then there must have been a forget operation that occurred after the frequency of item j had reached at least εf_j in $\boldsymbol{f}$. Since j is a sampled coordinate from the ℓ_p sampler, it follows that j is an $O(1)$-heavy hitter in the rescaled frequency vector $\boldsymbol{z}$. Therefore, if we track the $\varepsilon/10$-heavy hitters of $\boldsymbol{z}$ and maintain exact counters for those items, we will be able to detect this forget operation. In this case, we can recover the exact value of g_j, since updates prior to the forget event can be ignored. On the other hand, if $g_j \geq (1-\varepsilon)f_j$, then f_j itself provides a valid $(1\pm\varepsilon)$-approximation to g_j, which suffices for our purposes.

However, within each sampler, maintaining the ε-heavy hitter requires at least $\Omega(1/\varepsilon^2)$ space. If we use $1/\varepsilon^2$ independent samplers to reduce variance, the total space usage becomes $\Omega(1/\varepsilon^4)$, which is not optimal for our setting. To address this space issue, we propose the following alternative estimator for g_j. For each sampler, we uniformly sample a random threshold $w \in (\varepsilon, 1)$ and track the $w/10$-heavy hitters of the stream. Our final estimate for g_j is defined as

$$\mathbf{1}\left(\frac{g_j}{\widehat{f_j}} \geq 1-w\right)\cdot \widehat{f_j'},$$

where $\widehat{f_j}$ and $\widehat{f_j'}$ are two independent estimators of f_j. The first question is whether we can compute this expression exactly. The key observation is the following:

- If $g_j < (1-w/3)f_j$, then—because we are tracking the $w/10$-heavy hitters—we will be able to recover the exact value of g_j.
- Otherwise, if $g_j \geq (1-w/3)f_j$, then the indicator variable evaluates to 1, and we simply return $\widehat{f_j'}$.

Clearly, for the true value of f_j, we have $\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{f_j} \geq 1-w\right)\right] = \frac{g_j}{f_j(1-\varepsilon)}$, since w is uniformly sampled from the interval $(\varepsilon, 1)$. We now explain why the estimator remains (almost) unbiased when we use $\widehat{f_j}$ in place of the true f_j. When $g_j \geq (1-w/3)f_j$, and $\widehat{f_j}$ is a $(1\pm O(w))$-approximation to f_j, then with high probability, the ratio $g_j/\widehat{f_j}$ stays below 1. Conditioning on this, we perform a careful probabilistic analysis and show that the estimator remains unbiased as long as

$$\mathbf{E}\left[\frac{1}{\widehat{f_j}}\right] = \frac{1}{f_j}.$$

We will address how to ensure this property in a later section. For further technical details, we refer the reader to Section 4.

Reducing to $1/\varepsilon^2$ dependence. However, since each w_i is uniformly sampled from $(\varepsilon, 1)$, taking $1/\varepsilon^2$ independent samples results in an expected space usage of $\varepsilon^{-2}\cdot\mathbf{E}[w^{-2}] = O\left(\varepsilon^{-3}\right)$, which is still sub-optimal. To further reduce the space complexity, we observe that we obtain more samples than necessary in each subrange of w_i values. This suggests a sub-sampling strategy within each level. Specifically, suppose we have $1/\varepsilon^2$ independent ℓ_p samplers. Consider a fixed level $[\varepsilon\cdot 2^\ell, \varepsilon\cdot 2^{\ell+1}]$ for some $\ell \in \{0, 1, \ldots, \log(1/\varepsilon)\}$. The expected number of w_i values falling into this level is roughly $N_\ell = O\left(\frac{2^\ell}{\varepsilon}\right)$. We then randomly subsample $N'_\ell = O(4^\ell)$ of them and assign each corresponding estimator a weight of N_ℓ/N'_ℓ. This preserves unbiasedness of the overall estimator. Although the variance may increase significantly at certain levels due to sub-sampling, we perform a careful analysis and show that the total variance across all levels increases by at most an $O(\log(1/\varepsilon))$ factor.

Unbiased estimator of $1/f_j$. The remaining task is to give an unbiased estimation of $1/f_j$ as well as the reverse sampling probability $\|\boldsymbol{f}\|_p^p/|f_j|^p$. Here we consider the Taylor expansion of $f(x) = 1/x$. In particular, suppose that T is a $(1 \pm \frac{1}{2})$-approximation to f_j, we can expand $1/f_j$ as

$$\frac{1}{f_j} = \sum_{i=0}^{\infty} \frac{(-1)^i \cdot (f_j - T)^i}{T^{i+1}} = \frac{1}{T} - \frac{(f_j - T)^1}{T^2} + \frac{(f_j - T)^2}{T^3} - \dots$$

Since we only require a $(1 \pm \varepsilon)$-approximation in expectation, and T is a $(1 \pm \frac{1}{2})$ approximation to f_j, it suffices to truncate this series after the first $O(\log(1/\varepsilon))$ terms. To estimate each term of the form $(f_j - T)^q$, we generate q independent estimators $\widehat{f_j}^{(1)}, \widehat{f_j}^{(2)}, \dots, \widehat{f_j}^{(q)}$ of f_j and compute the product $\prod_{i=1}^{q}(\widehat{f_j^{(i)}} - T)$, which is an unbiased estimator for $(f_j - T)^q$ assuming each $\widehat{f_j}^{(i)}$ is unbiased. This allows us to construct an (approximately) unbiased estimator for $1/f_j$ using the truncated Taylor expansion.

For the reverse sampling probability $\|\boldsymbol{f}\|_p^p/|f_j|^p$, we adopt a similar strategy to estimate $1/|f_j|^p$. For the numerator $\|f\|_p^p$, we apply a standard technique based on p-stable distributions, using the geometric mean of several independent sketches. By multiplying these two components—the estimator for $\|\boldsymbol{f}\|_p^p$ and the estimator for $1/|f_j|^p$—we obtain an (approximately) unbiased estimator for the inverse sampling probability $\|\boldsymbol{f}\|_p^p/|f_j|^p$.

F_p estimation with forgets $(p > 2)$. We consider a similar algorithm to the one used for the case $0 < p \le 2$, but replace the ℓ_p sampling with ℓ_2 sampling. By leveraging the standard relationship between the ℓ_p and ℓ_2 norms, we show that it suffices to take $O\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}} \cdot n^{1-2/p}\right)$ samples to achieve the desired accuracy.

F_1 estimation with forgets. We show that there exists a surprisingly simple algorithm for estimating F_1 in the α-RFDS model. In the absence of forget requests, this task would be straightforward—we could simply maintain a count of the number of insertion operations. While this naïvely requires $O(\log m)$ space (where m is the stream length), Morris counters can reduce this to an $O(\log\log m)$ dependence.

To handle forget requests, we estimate the number of insertion operations into the stream that are later erased by a forget request. A random sample of insertion operations of size $O(\frac{1}{1-\alpha}\frac{1}{\varepsilon^2})$ would suffice for this, and we could simply store the identities of the corresponding insertion indices using $O((\frac{1}{1-\alpha}\frac{1}{\varepsilon^2})\log n)$ bits total and check if they are deleted after each operation that we sampled.

Since we cannot sample insertions in advance, we use reservoir sampling to construct the sample online. A technical challenge is that we do not know the exact stream length, as it is estimated using a Morris counter. This introduces slight noise into the sampling probabilities, which might be expected to accumulate over time. However, by carefully bounding the estimation error, we show that the impact on overall space remains negligible. We also note that one could alternatively use the reservoir sampling method of Gronemeier and Sauerhoff [9], though our analysis is simpler using the Morris+ counter [20].

Heavy hitters with general operations. Here we are allowing a group of more general functions including the forget operation. For a class of functions $\mathcal{G}$ which we will describe shortly, at each time t in the stream, we allow an update of the normal addition $(i, +)$ or of the type $(i, g_t \in \mathcal{G})$ which performs $f_i = g_t(f_i)$. For ease of presentation, here we consider $\mathcal{G}$ as the class of contracting functions, or functions with $g : \mathbb{N} \to \mathbb{N}$ with the following conditions: $|g(x) - g(y)| \le |x - y|$.

We now consider the ℓ_2 heavy hitter problem, which can be naturally extended to the ℓ_p heavy hitter problem for any $p \geq 2$. Let $\widetilde{f}$ denote the frequency vector excluding the operations in $\mathcal{G}$, and let f be the true frequency vector that includes the operations in $\mathcal{G}$. The key observation is that any ε-heavy hitter of f is guaranteed to be an ε'-heavy hitter of $\widetilde{f}$, where $\varepsilon' = (1-\alpha)^{1/2}\varepsilon$. Motivated by this, we track the $\varepsilon'/10$-heavy hitters of $\widetilde{f}$ and maintain exact counters (initialized to zero) for these items. When a true heavy hitter of f appears and is added to this list, its frequency may initially be underestimated by up to $\varepsilon'\left\|\widetilde{f}\right\|_2 = \varepsilon\|f\|_2$, due to the counter starting at zero. However, since the operations in the stream are either insertions or contractive transformations, the error in the estimate for this coordinate can only decrease or remain unchanged over time.

F_p estimation with general operations. Our algorithms are inspired by the level-set arguments in the literature (see, e.g., Section 5 in [17]). Take M to be an upper bound on the stream length. At a high level, we divide our vector's entries into level sets and let S_j denote the set of coordinates f_i where $f_i \in [\frac{M}{2^{j+1}}, \frac{M}{2^j}]$ at the end of the stream, we then estimate the $s_j = \sum_{i\in S_j} |f_i|^p$ individually. Then the final estimation becomes

$$s = \sum_i |f_i|^p = \sum_j \sum_{i\in S_j} |f_i|^p = \sum_j s_j \ .$$

A crucial observation is we only need to estimate s_j which takes up at least ε-fraction of F_p. To get a good approximation to s_j, we can do the following. We sample the coordinates of the underlying vector with probability 2^{-i} for $i = 0, 1, \cdots, \log M$. That way, we can find the proper sampling rate where the coordinates in S_j are heavy hitters in the subsampled stream. Therefore, we can use our heavy hitters structure to find the coordinates. By averaging and rescaling, we then can get a good approximation to s_j.

Lower bounds. Our lower bounds for F_p estimation for $p \in [0, 2]$ are derived via a variant of the GapHamming communication problem, known as *GapAndIndex*, which was also utilized by [22]. In the standard GapAndIndex problem, Alice and Bob each hold n-bit strings and wish to estimate the number of indices where both bits are 1, up to an additive error of $\sqrt{n}$. We consider a *one-way* variant where Alice's vector is sparse—containing exactly one 1 in each of r blocks, each of size n/r—and the approximation must be within additive error $\sqrt{r}$[1]. We show that solving this variant requires $\Omega(r \log(n/r))$ bits of communication. Setting $r = 1/\varepsilon^2$ yields an instance that can be solved by a $(1 \pm \varepsilon)$-approximation algorithm for F_p in the $O(1)$-RFDS model. Specifically, Alice inserts items corresponding to the 1-bits in her input, while Bob issues forget operations on the indices where his input has 1s. The resulting F_p moment accurately approximates the number of shared 1-bits between Alice and Bob.

To extend our lower bound beyond the regime where $1-\alpha$ is constant, we take a direct sum of $1/(1-\alpha)$ hard instances with $1-\alpha$ constant. After Alice sends her sketch to Bob, Bob can choose to delete all but one instance of his choice, so in fact he must be able to solve each one individually with good probability. By applying direct sum theorem, we then establish an $\Omega\left(\frac{1}{1-\alpha}\frac{1}{\varepsilon^2}\log n\right)$ lower bound.

For our F_p lower bounds with $p \geq 2$, we use an off-the-shelf communication lower bound for F_p moment estimation. This lower bound is also structured to admit a direct sum lower bound as above, for a sum of $\frac{1}{1-\alpha}$ instances. Applying this idea similarly as for $F_p,\ p \leq 2$ yields our lower bound of $\Omega\left(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^2}n^{1-2/p}\right)$.

[1]To facilitate the reduction, we actually work with a slightly modified version of this problem.

Finally, our lower bound for strict turnstile streams follows from a simple reduction from the well-studied set-disjointness communication problem, under the promise that the intersection size is either 0 or 1. Interestingly, our reduction from set-disjointness results in a three-round protocol, so we use a hardness result for set-disjointness for multi-way protocols.

1.3 Road map

In Section 2 we present the necessary preliminaries. In Section 3 we present our nearly-optimal ℓ_p sampler for $p \in (0, 2]$. In Section 4 we give our algorithm for F_p estimation for $p \in (0, 2]$ in the forget model. In Section 5 we give our algorithm for F_p estimation for $p > 2$ in the forget model. In Section 6 we show how to incorporate general operations for heavy hitters and F_p estimation. Finally, in Section 7 we present our lower bounds. Due to the page limit of the proceedings version, we focus here on these core technical developments. Additional extensions and deferred proofs are given in the full version of this work, available at https://arxiv.org/pdf/2508.07067.

2 Preliminaries

Frequency moments. We take the following definition from [1]. Let $A = (a_1, \ldots, a_m)$ be a sequence of elements, where each a_i is a member of $N = \{1, 2, \ldots, n\}$. Let $v_i = |\{j : a_j = i\}|$ denote the number of occurrences of i in the sequence A, and define, for each $p \geq 0$, $F_p = \sum_{i=1}^{n} v_i^p$.

Heavy hitters. Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, we say that an index $i \in [n]$ is an (ε, ℓ_p)-heavy-hitter for $\boldsymbol{v}$ if $|v_i| \geq \varepsilon \|\boldsymbol{v}\|_p$. The most fundamental heavy-hitters problem is that of identifying a set of size $O(1/\varepsilon^2)$ containing the set of ℓ_2 heavy hitters. An optimal solution was given in the seminal work of Charikar, Chen, and Farach-Colton [6] who used a CountSketch data structure to solve the heavy-hitters problem. Roughly, CountSketch works by partitioning the coordinates of $\boldsymbol{v}$ into approximately $\Omega(1/\varepsilon^2)$ buckets so that each heavy hitter is likely to end up in its own bucket. Simply summing up the items in a bucket would result in an additive approximation to each heavy hitter that depends on $\|\boldsymbol{v}\|_1$. This is improved by first randomly flipping the signs of elements in each bucket before summing, resulting in a much better additive error of $\varepsilon\|\boldsymbol{v}\|_2$ for each heavy hitter. Importantly for us, CountSketch gives estimates of the heavy hitter values, along with their identities. More specifically, by taking $O(\log \frac{1}{\delta})$ independent CountSketches, we obtain an additive approximation of $\varepsilon\|\boldsymbol{v}\|_2$ to a given coordinate v_i with probability at least $1 - \delta$.

3 ℓ_p Sampling for $p \in (0, 2]$

In this section, we will give the construction of our ℓ_p sampler in an insertion-only stream. We first consider the case $0 < p < 2$ and give an overview of the perfect ℓ_p sampler in [13]. Later, we will extend it to the case when $p = 2$ in Section 3.4. At a high level, the algorithm utilizes the order statistics of the exponential random variables.

Definition 3.1 (Exponential random variables)**.** We say e is a exponential random variable with scale parameter $\lambda > 0$, if the probability density function for e is $p(x) = \lambda e^{-\lambda x}$. In particular, we say e is a standard exponential random variable if $\lambda = 1$.

Fact 3.1 (Scaling of exponentials)**.** *Let t be exponentially distributed with rate λ, and let $\alpha > 0$. Then αt is exponentially distributed with rate λ/α.*

Definition 3.2 (Anti-rank)**.** Let $(t_1, \ldots, t_n)$ be independent exponential random variables. For $k = 1, 2, \ldots, n$, we define the k-th anti-rank $D(k) \in [n]$ of $(t_1, \ldots, t_n)$ to be the values $D(k)$ such that $t_{D(1)} \leq t_{D(2)} \leq \cdots \leq t_{D(n)}$.

Fact 3.2 ([19])**.** *For any $i = 1, 2, \ldots, n$, we have* $\Pr[D(1) = i] = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}$.

Corollary 3.3. *Let $f \in \mathbb{R}^n$ be a frequency vector. Let $(e_1, \ldots, e_n)$ be i.i.d. exponential random variables with rate 1. Define $z_i = f_i/e_i^{1/p}$ and let $(D(1), \ldots, D(n))$ be the anti-rank vector of the vector $\left(z_1^{-p} \cdots z_n^{-p}\right)$. Then we have $\mathbf{Pr}[D(1) = i] = \frac{|f_i|^p}{\|f\|_p^p}$.*

Definition 3.3 implies that the ℓ_p sampler only needs to output the maximal index $i^\star$ of z. To do this, the algorithm in [13] needs some gap between $z_{D(1)}$ and $z_{D(2)}$ to detect the maximal index $i^\star$. In particular, it runs a statistical test at the end and outputs FAIL if the test is not satisfied. To ensure the probability of failing the statistical test does not depend on $i^\star$, [13] duplicates each coordinate of for n^c times for some constant c to remove all the heavy items. They show that after this step, the following holds.

Lemma 3.3 ([13])**.** *For constant $p \in (0, 2]$ and $\nu \geq n^{-c/60}$, $\mathbf{Pr}[\neg\texttt{FAIL} \mid D(1)] = \mathbf{Pr}[\neg\texttt{FAIL}] \pm \tilde{O}(\nu)$ for every possible $D(1) \in [n]$.*

Throughout this section, we will assume the frequency vector is after the duplication and that Lemma 3.3 therefore holds. We will also need the following fact.

Lemma 3.4 (Proposition 1, [13])**.** *For any $s = 2^j \leq nc^{-2}$ for some $j \in \mathbb{N}$, we have $\sum_{i=4s}^{N} z_{D(i)}^2 = O(\|F\|_p^2/s^{2/p-1})$ if $p \in (0, 2)$ is a constant bounded below 2, and $\sum_{i=4s}^{N} z_{D(i)}^2 = O(\log(n)\|F\|_p^2)$ if $p = 2$, with probability $1 - 3e^{-s}$.*

3.1 Warm-up: BPTree

In this section, we will first assume our algorithm has access to a random oracle. Then in the later section we shall give a way to derandomize it. Our ℓ_p sampler will be based on the following data structure.

Theorem 3.1. *(BPTree, [3]) For any $\varepsilon \in (0, 1)$, there is 1-pass insertion-only streaming algorithm* BPTree *that, with probability at least $1 - \delta$, returns a set of $(\frac{\varepsilon}{2}, \ell_2)$ heavy hitters containing every (ε, ℓ_2)-heavy hitter. The algorithm uses $O(\varepsilon^{-2} \log n \log \frac{1}{\varepsilon\delta})$ bits of space and has $O(\log \frac{1}{\varepsilon\delta})$ update time and $O(\varepsilon^{-2} \log \frac{1}{\varepsilon\delta})$ retrieval time.*

Our algorithm is as follows.

(1) Let z be the vector defined in Definition 3.3.
(2) Using BPTree (Theorem 3.1), keep track of z during the stream. In parallel, we use a CountSketch with $O(1)$ buckets and an AMS sketch ([1]) for ℓ_2 norm to keep track of z.
(3) At the end of the stream, let the $\mathcal{S}$ be the output set of $O(1)$-heavy hitters from the BPTree. For every $i \in \mathcal{S}$, let v_i be the estimation of z_i using the Count-Sketch. Let $\tilde{F}_2$ be an estimation of $\|z\|_2^2$ from the AMS sketch.
(4) Let $v_{D(1)}$ and $v_{D(2)}$ be the first and second largest value in v_i ($i \in \mathcal{S}$). If $v_{D(1)} - v_{D(2)} \geq c_1 \cdot \widetilde{F}_2$ for some constant c_1, output $\arg\max_{i \in s} v_i$.

At a high level, we will run $O(\log\log n + \log(1/\varepsilon))$ independent trials of the above subroutine and do the statistical test at the end. Then we output the maximal index i from an arbitrary trial where the statistical test succeeds. If none of the trials succeed, the algorithm outputs FAIL.

Theorem 3.2. *For any constant $p \in (0, 2)$, there is a one-pass streaming algorithm that runs in space $\widetilde{O}(\log n \log(1/\varepsilon))$, and outputs an index i such that with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$ we have for every $j \in [n]$, $\Pr[i = j] = \frac{|f_j|^p}{\|f\|_p^p} \pm \frac{1}{\mathrm{poly}(n)}$.*

Proof. Condition on the CountSketch for $v_{D(1)}$ and $v_{D(2)}$, and AMS sketch success for a constant approximation in all of the independent trials by taking a union bound. Then, consider an arbitrary

trial where the statistical test succeeds. From the condition we have that using the CountSketch and AMS sketch we can find the maximal index $i^\star$ in $\mathcal{S}$ that returned by the BPTree. On the other hand, from Lemma 3.4 we have that with probability $\Omega(1)$ we have $z^2_{D(1)} > 0.6F_2$, this means that if we run $O(\log\log n + \log(1/\varepsilon))$ independent trials, with probability at least $1 - \text{poly}(\varepsilon/\log n)$ one of the trials will pass the statistical test, and the output of this trial is what we need.

Now we analyze the space complexity of our algorithm. For each independent trial, we set the accuracy parameter $\varepsilon' = O(1)$ and error probability $\delta = \text{poly}(\varepsilon/\log n)$ for the CountSketch, AMS sketch and BPtree. This means the space we need for each trial is $\widetilde{O}(\log n \log(1/\varepsilon))$ and the total amount of space is $\widetilde{O}(\log n \log(1/\varepsilon))$. □

3.2 Heavy-Hitters via Adaptive Sparse Recovery

The difficulty with derandomizing the above ℓ_p sampling algorithm without introducing additional $\log n$ factors is that the BPTree is not a linear sketch. This means we cannot apply the derandomization technique from [13]. At a high level, the key sub-routine of the BPTree is the following: given a 2-approximation of $F_2(\boldsymbol{f})$, find the single $O(1)$-heavy hitter of $\boldsymbol{f}$. To solve this problem, BPTree takes $O(\log n)$ rounds and in each round, it decides one bit of the heavy coordinate. In this section, we will give another way to solve this heavy-hitter problem in just $\widetilde{O}(\log n)$ space via inspiration from [11]. Compared to BPTree, the advantage of the algorithm here is that we only require $\log\log n$ rounds of linear sketches rather than $\log n$ for BPTree, which enables us to apply the derandomization technique from [13] while only blowing up our space by $O(\text{poly}(\log\log n))$ factors. We recall the following lemma from [11] to show correctness of their sparse recovery algorithm.

Lemma 3.5. *(Lemma 3.2, [11]) Suppose that there exists j with $|x_j| \geq C\frac{B^2}{\delta^2}\left\|\boldsymbol{x}_{S\setminus\{j\}}\right\|_2$ for a constant C and parameters B and δ. Then with two non-adaptive linear measurements, with probability $1-\delta$, we can find a set $S \subseteq n$ such that $|S| \leq 1 + n/B^2$ and $\left\|\boldsymbol{x}_{S\setminus\{j\}}\right\|_2 \leq \frac{1}{B}\left\|\boldsymbol{x}_{[n]\setminus\{j\}}\right\|_2$.*

First, we assume that we know $F_2(\boldsymbol{f})$ within a factor of two. Let $\widehat{F}$ be that estimate of $F_2(\boldsymbol{f})$. We will make use of the following F_2 tracker, and mark the times t_i at which the tracker's F_2 estimate becomes at least $i\widehat{F}/(\log\log n)$.

Lemma 3.6 ([3])**.** *There is an algorithm that estimates F_2 at all times i to within a constant factor with high constant probability, using $O(\log n(\log\log n + \log\frac{1}{\delta}))$ space.*

We next note that without loss of generality we can further assume the heavy hitter f_i satisfies $|f_i| \geq (1-1/\log\log n)\left\|\boldsymbol{f}_{[n]\setminus\{i\}}\right\|_2$, as we can randomly divide the universe into $10(\log\log n)^2$ groups and run our procedure in each groups. By Markov's inequality we have with probability at least $1 - 1/\log\log n$, f_i will be $(1-1/\log\log n)$-heavy in its group. This will increase the space by only a $(\log\log n)^2$ factor.

Lemma 3.7. *Suppose that some universe item j satisfies $|f_j| \geq (1-1/\log\log n)\left\|\boldsymbol{f}_{[n]\setminus\{j\}}\right\|_2$. Then there is an algorithm to recover j with probability larger than $1/2$, using $\widetilde{O}(\log n)$ space. Moreover this algorithm runs in only $O(\log\log n)$ rounds and uses only $O(1)$ row linear sketches in each round.*

Proof. We consider a similar strategy to that of [11]. Let $B_0 = 2$ and $B_i = B_{i-1}^{3/2}$ for $i \geq 1$ and define $\delta_i = 2^{-i}/4$ for $i = 0$. Start with $S_0 = [n]$. Consider the stream of updates during time t_i and t_{i+1}, from the condition we have f_j will be $(1-1/C)$-heavy during the stream between t_i and t_{i+1} for some sufficiently large constant C. Applying Lemma 3.5 with parameters $B = 4B_i$ and $\delta = \delta_i$, we can identify a subset $S_{i+1} \subset S_i$ with $j \in S_{i+1}$, ending with $i = r$ where $r = O(\log\log n)$ so $B_r \geq n$.

Similarly to [11], we prove by induction that Lemma 3.5 applies at the i-th iteration. We choose C to match the base case. For the inductive step, suppose $\|x_{S\setminus\{j\}}\|_2 \leq |x_j|/(C'16\frac{B_i^2}{\delta_i^2})$. Then by Lemma 3.5, $\|\boldsymbol{x}_{S\setminus\{j\}}\|_2 \leq |x_j|/(C'64\frac{B_i^3}{\delta_i^2}) = |x_j|/(C'16\frac{B_{i+1}^2}{\delta_{i+1}^2})$. This means that the lemma applies in the next iteration as well. Note that after r-th iteration we have $S_r < 2$, which means that we can identify $j \in S_r$. Taking a union bound, the overall failure probability is at most $\sum_i \delta_i < 1/2$. □

3.3 Derandomization

We next derandomize the ℓ_p sampler. We can see from Lemma 3.7 that the procedure has $O(\log\log n)$ rounds or branching points. For some fixed $O(\log\log n)$ branching points (here we fix the $O(\log\log n)$ branching time $t_1, t_2, \ldots, t_r$ and the subset $S_1, S_2, \ldots, S_r$ it goes into), let us call this a path. Note that there are at most $n^{O(\log\log n)}$ paths.

The main thing we wish to derandomize is the exponential re-scalings. Once we derandomize these scalings, it is possible that the probability of taking a path will differ than the probability of taking that path in the random oracle model. Therefore, we aim to also show that

$$\left|\mathbf{Pr}(\texttt{path}) - \mathbf{Pr}'(\texttt{path})\right| \leq \frac{1}{n^{O(\log\log n)}}$$

where $\mathbf{Pr}(\texttt{path})$ denotes the true probability of taking a path in the random oracle model and $\mathbf{Pr}'(\texttt{path})$ denotes the probability of taking a path after derandomizing. If we show this, then we are done since we can take a union bound over all the paths, giving us total variation distance $1/n^{O(\log\log n)}$.

Note that the sketching matrix in Lemma 3.7 only needs $O(1)$-wise independence, so the only part we need to derandomize is the exponential random variables part. This means that we can fix the value of the sparse recovery part and consider one fixed path. The advantage here is after this step, we can treat the sparse recovery part in Lemma 3.7 as a linear sketching with $O(\log\log n)$ rows. We use the following result from [13].

Lemma 3.8. *([13], Theorem 5) Let* ALG *be any steaming algorithm which, on stream vector* $\boldsymbol{f} \in \{-M, \ldots, M\}^n$ *and fixed matrix* $\boldsymbol{X} \in \mathbb{R}^{t\times nk}$ *with entries contained within* $\{-M, \ldots, M\}$*, for some* $M = \text{poly}(n)$*, outputs a value that only depends on sketches* $\boldsymbol{A}\cdot\boldsymbol{f}$ *and* $\boldsymbol{X}\cdot\texttt{vec}(\boldsymbol{A})$*. Assume that the entries of the random matrix* $\boldsymbol{A} \in \mathbb{R}^{k\times n}$ *are i.i.d. and can be sampled using* $O(\log n)$ *bits. Let* $\sigma : \mathbb{R}^k \times \mathbb{R}^t \to \{0,1\}$ *be any tester which measures the success of* ALG*, namely* $\sigma(\boldsymbol{Af}, \boldsymbol{X}\cdot\texttt{vec}(A)) = 1$ *whenever* ALG *succeeds. Fix any constant* $c \geq 1$*. Then* ALG *can be implemented using a random matrix* $\boldsymbol{A}'$ *using random seed of length* $O((k+t)\log n(\log\log n)^2)$*, such that:*

$$\left|\mathbf{Pr}[\sigma(\boldsymbol{Af}, \boldsymbol{X}\cdot\texttt{vec}(\boldsymbol{A})) = 1] - \mathbf{Pr}[\sigma(\boldsymbol{A}'\boldsymbol{f}, \boldsymbol{X}\cdot\texttt{vec}(\boldsymbol{A}')) = 1]\right| < n^{-c(k+t)}.$$

Here, we take $\boldsymbol{A}$ to be an n-dimensional row vector of i.i.d. exponential random variables. , and $\boldsymbol{X} = \boldsymbol{S}\texttt{diag}(\boldsymbol{f})$ to be a $O(\log\log n)\times n$ fixed matrix where $\boldsymbol{S}$ is the sketching matrix from Lemma 3.7. Therefore, we have that for a fixed path, the probability difference $\left|\mathbf{Pr}(\texttt{path}) - \mathbf{Pr}'(\texttt{path})\right|$ is at most $n^{-c\log\log n}$. This is sufficient for us as then we can take a union bound over all $n^{O(\log\log n)}$ paths.

3.4 Extending to ℓ_2 Sampling

We next consider the case when $p = 2$. By Lemma 3.4, we have that with high constant probability $\left\|z_{-D(1)}\right\|_2^2 = O(\log n)\cdot F_2(\boldsymbol{z})$. Hence, with probability $\Omega\left(\frac{1}{\log n}\right)$ we have $|Z_{D(1)}|^2 > 20\left\|z_{-D(1)}\right\|_2^2$. This means that we will need to run the sub-routine in Section 3.1 $\widetilde{O}(\log(n)\log(1/\varepsilon))$ times.

Corollary 3.4. *There is a one-pass streaming algorithm that runs in space $\widetilde{O}(\log^2 n \log(1/\varepsilon))$, and outputs an index i such that with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$ we have for every $j \in [n]$, $\Pr[i = j] = \frac{|f_j|^2}{\|f\|_2^2} \pm \frac{1}{\mathrm{poly}(n)}$.*

3.5 Continuous ℓ_p Sampler

In some scenarios, providing a sample index i at the end of the stream is not enough. Instead, we may require the sampler to give a sample index at each time step during the stream. We next give a continuous ℓ_p sampler which is based on the ℓ_p sampler we presented in Theorem 3.2.

Lemma 3.9 (Continuous ℓ_p sampler)**.** *Consider a stream of length m (and therefore with m time steps). Let $F_{p,[0,t]}$ be the F_p moment of $\boldsymbol{f}$ after the first t time steps, and let $f_{i,[0,t]}$ be the frequency of coordinate i after the first t time steps. There is an algorithm $\mathcal{A}$ that produces a series of m outputs $z_1, \ldots, z_m$ such that each $z_j \in \{1, \ldots, n\}$, one at each time step with the following properties:*

- $\mathbf{Pr}(z_t = i) = \frac{f_{i,[0,t]}^p}{F_{p,[0,t]}} \pm \frac{1}{\mathrm{poly}(n)}$.
- *Only $O(\log^{c(p)} n \cdot \log\log n \cdot \log(1/\varepsilon))$ of the z_j's are unique.*
- *$\mathcal{A}$ uses $O(\log^{c(p)} n \cdot \mathrm{polylog}\,(1/\varepsilon) \cdot \mathrm{poly}(\log\log n))$ bits of space.*
- *$\mathcal{A}$ succeeds with probability at least $1 - \mathrm{poly}(\varepsilon/\log n)$.*

Here $c(p) = 1$ from $0 < p < 2$ and $c(p) = 2$ for $p = 2$.

Proof. We first consider the case when $p \in (0, 2)$. Similarly to what we did in Section 3.1, we run $O(\log(1/\varepsilon) \cdot \log\log n)$ independent trials of the sub-routines and in each subroutine we use the BPTree to keep track of the vector $\boldsymbol{z}$ scaled by the exponential random variables. Starting from $t = 1$, suppose that if we have a sample z_{t-1} at time $t - 1$ which corresponds to a particular trial of the exponential variables. Then we will set $z_t = z_{t-1}$ if the statistical test for this specific trial still be satisfied at the time t. Otherwise, we will look at the other trials of the sub-routine and output z_t from an arbitrary sub-routine where the statistical test passes.

From the correctness of Theorem 3.2 we get the correctness of properties (1) and (3). To bound the number of unique $z_i's$, note that every time when we change the value of z_t, the statistical test must have failed again for the corresponding trials. This implies the F_2 moment of the rescaled vectors must increase by a constant factor during this period of time. Since we have only $O(\log(1/\varepsilon) \cdot \log\log(n))$ number of trials, we can get the number of the time z_i changes is at most $O(\log n \cdot \log(1/\varepsilon) \cdot \log\log(n))$. Finally, we consider the overall success probability. The only thing we need is when z_i changes, at least one of the trials of the sub-routines passes the statistical test. Since for each trial, the success probability is $\Omega(1)$, from the fact that z_i can only changes $O(\log n \cdot \log(1/\varepsilon) \cdot \log\log(n))$ time and we have $O(\log(1/\varepsilon) \cdot \log\log(n))$ trials of the sub-routines we have the overall success probability is at least $1 - \mathrm{poly}(\varepsilon/\log n)$.

We next consider the case for $p = 2$. Here, we run $\tilde{O}(\log n \log(1/\delta))$ independent trials of the sub-routines. The remaining argument then goes through. □

3.6 Estimating the Value and Sampling Probability

As mentioned, one application of our ℓ_p sampler is to estimate the F_p moment. In this case in addition to an index i, we may also need an estimation of f_i and the sampling probability $p_i = \frac{|f_i|^p}{\|f\|_p^p}$ (in fact, the inverse sampling probability). We will next first show that we can get good enough estimators of the values of the coordinates that are sampled by the ℓ_p sampler.

Lemma 3.10. *For a sampled index i by an ℓ_p sampler from Theorem 3.2, we can get an estimator $\widehat{f_i}$ to f_i such that $\mathbf{E}[\widehat{f_i}] = f_i$ and $\mathbf{E}[\widehat{f_i}^2] = O(f_i^2)$. The space requirement is $\widetilde{O}(\log n)$ bits.*

Proof. Let $\boldsymbol{z}$ be the vector $\boldsymbol{f}$ recaled by the exponential random variables $u_i^{1/p}$. We use a CountSketch with a constant number of buckets to estimate the frequency of the sampled index z_i (this can be derandomized from the same procedure in [13]). By the property of CountSketch, we have $\mathbf{E}[\widehat{z}_i] = z_i$. To bound the variance, note that we have that $|z_i|^2 \geq \Theta(1)\|\boldsymbol{z}\|_2^2$ since i is the sampled coordinate. So, we have $\mathbf{E}[\widehat{z}_i^2] = O\left(\|\boldsymbol{z}\|_2^2\right) = O\left(z_i^2\right)$. Multiplying by $u_i^{1/p}$ in both side we get the desired result. □

We now consider the estimation of $|f_i|^p$. For $p = 2$, we can simply run two independent estimator of f_i and take their product. For other $p \in (0, 2)$ we will use the following Taylor's series estimator.

Lemma 3.11. *Suppose X is a fixed value in $[(1-\alpha\cdot T, (1+\alpha)\cdot T]$, and suppose that we have access to $Y_1, \ldots, Y_N$ where the Y_j's are independent with $\mathbf{E}[Y_j] = X$ for some constant C and $\mathbf{Var}[Y_j] \leq T^2/4$.*

For $\alpha < \frac{1}{2}$, we can construct an estimator $\widehat{Q}$ of X^p for $p \in [0, 2)$ with $\left|\mathbf{E}[\widehat{Q}] - X^p\right| \leq \varepsilon^{10} T^p$ and $\mathbf{Var}[\widehat{Q}] = O(T^{2p}\log(1/\varepsilon))$ for $N = O(\log^2 \frac{1}{\varepsilon})$ for $\varepsilon \in (0, 1)$.

Proof. Consider the Taylor series expansion for $f(x) = x^p$ for $p \in [0, 2)$ centered at T. For $p = 0$ we have $f(x) = x^0 = 1$. For $p = 1$ we have $f(x) = x = T + (x - T)$. For $p \in [0, 2)$ but not 0 or 1, we have

$$x^p = T^p + pT^{p-1}(x-T) + \frac{p(p-1)}{2!}T^{p-2}(x-T)^2 + \frac{p(p-1)(p-2)}{3!}T^{p-3}(x-T)^3 + \ldots$$

valid for $x \in (0, 2T)$.

Define $P_k(x)$ to be the associated degree k Taylor polynomial. So, we have

$$P_k(x) = T^p \sum_{i=0}^{k} \binom{p}{i} \left(\frac{x-T}{T}\right)^i.$$

Recall that when truncating a Taylor series at degree k for some function $f(x)$, the error is

$$\left|P_k(x) - f(x)\right| = \left|\frac{f^{k+1}(\xi)}{(k+1)!}\cdot(x-T)^{k+1}\right|$$

where ξ is some fixed point between T and x, and $f^{k+1}(\xi)$ denotes the value of the $(k+1)$-th derivative of function f at ξ.

For $f(x) = x^p$, we have that

$$f^{k+1}(\xi) = p(p-1)(p-2)\ldots(p-k)\cdot\xi^{p-k-1}.$$

So, for $X \in [(1-\alpha)\cdot T, (1+\alpha)\cdot T]$, we have error

$$\begin{aligned}
\left|P_k(X) - X^p\right| &= \left|\frac{p(p-1)(p-2)\ldots(p-k)\cdot\xi^{p-k-1}}{(k+1)!}\cdot(X-T)^{k+1}\right| \\
&\leq \frac{(k+1)!\cdot\xi^{p-k-1}}{(k+1)!}\cdot\left|(X-T)^{k+1}\right| = \frac{\left|(X-T)^{k+1}\right|}{\xi^{k+1-p}} \\
&\leq \frac{(\alpha T)^{k+1}}{((1-\alpha)\cdot T)^{k+1-p}} = \frac{\alpha^{k+1}}{(1-\alpha)^{k+1-p}}\cdot T^p. \qquad \text{(since } X \in [(1-\alpha)\cdot T, (1+\alpha)\cdot T]\text{)}
\end{aligned}$$

By the lemma statement, we have access to $Y_1, \ldots, Y_N$ which are independent unbiased estimators of X and therefore to $X - T$ after shifting by T. For each Y_j for $j \in [N]$ we have

$$\mathbf{E}[(Y_j - T)^2] \le \mathbf{E}[2(Y_j - X)^2 + 2(X - T)^2] = 2\,\mathbf{Var}[Y_j] + 2(X - T)^2$$
$$\text{(since } (A+B)^2 \le 2A^2 + 2B^2 \text{ by AM-GM)}$$
$$\le \frac{T^2}{2} + 2(\alpha T)^2 \le T^2. \qquad \text{(since } \mathbf{Var}[Y_j] \le T^2/4 \text{ and } \alpha \le 1/2)$$

Let us take ℓ of the independent estimators $Y_j - T$ and multiply them. Call this estimator Y^ℓ. Since we have that the estimators $Y_j - T$ for $j \in [N]$ are independent, we have that $\mathbf{E}[Y^\ell] = (X - T)^\ell$. Furthermore, we have that

$$\mathbf{E}[(Y^\ell)^2] = \mathbf{E}[((Y_1 - T)(Y_2 - T)\ldots(Y_\ell - T))^2] \le (T^2)^\ell.$$

By simply rescaling, we get an estimator Z_ℓ such that

$$\mathbf{E}[Z_\ell] = \binom{p}{\ell}\frac{(X-T)^\ell}{T^{\ell-p}}, \mathbf{E}[Z_\ell^2] \le \binom{p}{\ell}^2 \frac{(T^2)^\ell}{T^{2(\ell-p)}} \le 4 \cdot T^{2p}.$$

Let us set our estimator $\widehat{Q} = \sum_{i=1}^k Z_i$. By using our upperbound on $\left|P_k(X) - X^p\right|$ above, we have

$$\left|\mathbf{E}[\widehat{Q}] - X^p\right| = \left|\mathbf{E}[Z_1 + \ldots + Z_k] - X^p\right| \le \frac{\alpha^{k+1}}{(1-\alpha)^{k+1-p}} \cdot T^p.$$

We also have

$$\mathbf{Var}[\widehat{Q}] = \mathbf{Var}[Z_1 + \ldots + Z_k] = \mathbf{Var}[Z_1] + \ldots + \mathbf{Var}[Z_k] \le \mathbf{E}(Z_1^2) + \ldots + \mathbf{E}(Z_k^2) \le 4kT^{2p}.$$

The result follows upon setting $k = \Theta(\log \frac{1}{\varepsilon})$. □

By running $O(\log^2(1/\varepsilon))$ independent copies of the estimator from Lemma 3.10 and plugging this into Lemma 3.11, we get the following lemma.

Lemma 3.12. *For a sampled index i by an ℓ_p sampler from Theorem 3.2, we can get an estimator v_i to f_i^p such that $\mathbf{E}[v_i] = (1 \pm \varepsilon^{10})|f_i|^p$ and $\mathbf{E}[v_i^2] = O(|f_i|^{2p})$. The space requirement is $\widetilde{O}(\log n \log^2(1/\varepsilon))$ bits.*

We next consider the estimation of the (reverse) sampling probability $1/p_i$. Recall that the sampling probability for an index i is $f_i^p / \|\boldsymbol{f}\|_p^p$. Here we show how to estimate the reverse sampling probability which will be used in our downstream applications.

We first show how to get a rough estimate of F_p (or $\|\boldsymbol{f}\|_p^p$) for *constant* $p \in (0, 2)$, which is part of the inverse sampling probability. To do this, we will use the geometric mean estimator and p-stable random variables. We first give the definition of p-stable random variables.

Definition 3.5. (Zolotarev [23]) For $0 < p < 2$, there exists a probability distribution $\mathcal{D}_p$ called the p-stable distribution with $\mathbf{E}[e^{itZ}] = e^{-|t|^p}$ for $Z \sim \mathcal{D}_p$. For any n and vector $\boldsymbol{x} \in \mathbb{R}^n$, if $Z_1, \ldots, Z_n \sim \mathcal{D}_p$ are independent, then $\sum_{j=1}^n Z_j \boldsymbol{x}_j \sim \|\boldsymbol{x}\|_p Z$ for $Z \sim \mathcal{D}_p$.

The following fact will also be useful.

Lemma 3.13. *(Nolan [21], Theorem 1.2) For fixed $0 < p < 2$, the probability density function of the p-stable distribution is $\Theta(|x|^{-p-1})$ for large x.*

When considering one p-stable random variable, they have undefined expectation and infinite variance. However, we will show that by using a geometric mean we can control the expectation and variance. We take $\boldsymbol{S}$ to be a matrix of i.i.d. p-stable random variables with $k = \frac{2}{p-\varepsilon'}$ where ε' is

a small constant between 0 and 1. We consider F which we call the *geometric mean estimator* which is based on the same sketch $\boldsymbol{Sx}$ (for a fixed $\boldsymbol{x} \in \mathbb{R}^n$) with k rows. In particular, we have

$$F = |(\boldsymbol{Sx})_a \cdot (\boldsymbol{Sx})_b \cdot (\boldsymbol{Sx})_c \dots|^{\frac{p-\varepsilon'}{2}}.$$

Lemma 3.14. $\mathbf{E}[F] = C \cdot \|\boldsymbol{x}\|_p$ *where* $C > 0$ *is a constant that does not depend on* $\boldsymbol{x}$. $\mathbf{Var}[F] = O(\|\boldsymbol{x}\|_p^2)$.

Proof. We have $(\boldsymbol{Sx})_a = \|\boldsymbol{x}\|_p \cdot P_1$, $(\boldsymbol{Sx})_b = \|\boldsymbol{x}\|_p \cdot P_2$, $(\boldsymbol{Sx})_c = \|\boldsymbol{x}\|_p \cdot P_3$, and so on where $P_1, P_2, P_3, \dots$ are independent p-stable random variables by Definition 3.5. Therefore we have that

$$\mathbf{E}[F_i] = \|\boldsymbol{x}\|_p \cdot \mathbf{E}[|P_1 P_2 P_3 \dots|^{\frac{p-\varepsilon'}{2}}].$$

We now show that $\mathbf{E}[|P_1 P_2 P_3 \dots|^{\frac{p-\varepsilon'}{2}}]$ is a finite scalar. Recall that P_1, P_2, P_3, and so on are independent. Therefore we have

$$\mathbf{E}[|P_1 P_2 P_3 \dots|^{\frac{p-\varepsilon'}{2}}] = \int_{\frac{1}{\frac{p-\varepsilon'}{2}}} C \cdot \frac{|z_1 z_2 z_2 \dots|^{\frac{p-\varepsilon'}{2}}}{(z_1 z_2 z_2 \dots)^{p+1}} dz_1 dz_2 dz_3 \dots$$

by Lemma 3.13 which converges to a constant. The variance is similar. We have that

$$\mathbf{Var}[F_i] = \|\boldsymbol{x}\|_p^2 \cdot \mathbf{Var}[|P_2 P_2 P_3 \dots|^{\frac{p-\varepsilon'}{2}}].$$

By definition we have $\mathbf{Var}[|P_2 P_2 P_3 \dots|^{\frac{p-\varepsilon'}{2}}] \le \mathbf{E}[|P_1 P_2 P_3 \dots|^{p-\varepsilon'}]$ and so we again get that this converges to a constant. □

We remark that the geometric mean estimator which we use in Lemma 3.14 to estimate the inverse sampling probability of a coordinate by our ℓ_p sampler was derandomized by [15] (Theorem 12) with only an additive logarithmic factor.

So, we can use Lemma 3.14 to get a unbiased (after dividing by C) estimator for $\|\boldsymbol{f}\|_p$ with variance $O(\|\boldsymbol{f}\|_p^2)$. To turn this into an estimator for $\|\boldsymbol{f}\|_p^p$, we again run $\log^2(1/\varepsilon)$ independent copies and plug into the Taylor's estimator in Lemma 3.11.

To finish estimating the inverse sampling probability, we also need to estimate $1/f_i^p$. We can get an estimator for f_i^p using Lemma 3.10 and Lemma 3.11. Now, we show how to estimate $1/f_i^p$ from this. The following lemma gives the precise guarantee that we will apply. The proof of this lemma is similar to the proof of Lemma 3.11, and is deferred to the full version of the paper.

Lemma 3.15. *Suppose that* X *is a fixed value in* $[(1-\alpha)\cdot T, (1+\alpha)\cdot T]$, *and suppose that we have access to* $Y_1, \dots, Y_N$ *where the* Y_j*'s are independent with* $\mathbf{E}[Y_j] = X$ *and* $\mathbf{Var}[Y_j] \le T^2/4$.

For $\alpha < \frac{1}{2}$, *we can construct an estimator* $\widehat{Q}$ *of* $\frac{1}{X}$ *with* $\left|\mathbf{E}[\widehat{Q}] - 1/X\right| \le \frac{\varepsilon^{10}}{T}$ *and* $\mathbf{Var}[\widehat{Q}] = O(\frac{1}{T^2}\log\frac{1}{\varepsilon})$ *for* $N = O(\log^2\frac{1}{\varepsilon})$ *for* $\varepsilon \in (0,1)$.

Therefore, we can conclude the following by multiplying the estimator for $\|\boldsymbol{f}\|_p^p$ and $1/f_i^p$.

Lemma 3.16. *Using* $O(\log n \log^2(1/\varepsilon))$ *bits of space, for an index* i *sampled by an* ℓ_p *sampler, we have an estimator* $\widehat{p}_i$ *to* $p_i = f_i^p/\|\boldsymbol{f}\|_p^p$ *with* $\left|\mathbf{E}[1/\widehat{p}_i] - 1/p_i\right| \le \varepsilon^8 \cdot 1/p_i$ *and* $\mathbf{E}[1/\widehat{p}_i^2] = O(1/p_i^2 \cdot \log^2(1/\varepsilon))$.

4 F_p Estimation for $p \in (0, 2]$

In this section, we will present algorithms for F_p estimation ($0 < p \le 2$) in the presence of forgets. Let $\boldsymbol{f}$ be the vector without the forget operations and $\boldsymbol{g}$ be the actual frequency vector with forget operations. For the purpose of demonstration, we will first give an algorithm with $1/\varepsilon^3$ dependence

and show how to reduce this to $1/\varepsilon^2$. At a high level, we run $k = \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\right)$ ℓ_p samplers from Section 3. Let $\mathcal{I}$ denote the set of sampled indices. If we are allowed to take a second pass on the data stream, then we can maintain an individual counter for each coordinate in $\mathcal{I}$, which means that we can get the extra value of g_j for every $j \in \mathcal{I}$. On the other hand, we can also get an estimation of the reverse sampling probability $\widehat{\frac{1}{p_i}}$ (Lemma 3.16). One can argue that $\frac{1}{k} \cdot \sum_{j\in\mathcal{I}} g_j^p/\widehat{p}_j$ will be a good estimator of the F_p moment of $\boldsymbol{g}$.

However, we are not allowed a second pass on the stream. To implement the algorithm in a single-pass stream, the observation is that if $g_i \geq (1-\varepsilon)f_i$, then we can just use f_i as an estimation of g_i as we are allowing a $(1\pm\varepsilon)$-approximation. On the other hand, when $g_i < (1-\varepsilon)f_i$, we have that there must be a forget operation after the frequency of coordinate i in $\boldsymbol{f}$ becomes εf_i. Let $\boldsymbol{z}$ denote the corresponding rescaled vector of $\boldsymbol{f}$ in the ℓ_p sampler. Recall that since i is the sampled coordinate, we have $z_i = \Theta(1)\|\boldsymbol{z}_{-i}\|_2$. This implies that if we maintain the ε-heavy hitters of $\boldsymbol{z}$ during the stream and assign a counter to each of them. The corresponding counter can observe such a forget operation and then get the extra value of g_i. The above observation gives a one-pass streaming algorithm, but since for each ℓ_p sampler, we need an extra $\widetilde{O}\left(\frac{1}{\varepsilon^2}\right)$ space for each sampler to keep track of the ε-heavy hitters, which result in a total $O(\varepsilon^{-4})$ dependence. To get a better space, the crucial observation is that it is not necessary to maintain all the ε-heavy hitters in all ℓ_p samplers. For example, if $g_i = 1/2f_i$, then maintaining $\Theta(1)$-heavy hitter during the stream is sufficient.

Our algorithm is presented in Algorithm 1. At a high level, from each sub-routine in each sampler L_i (recall that for each ℓ_p, we have multiple groups of exponential random variables to make sure one of them passes the statistical testing), we uniformly sample $w_i \in (\varepsilon, 1)$ and keep track of the w_i-heavy hitters of the rescaled frequency vector $\boldsymbol{z}$. For sampled vector j, Our estimator for g_j will be $X_j = \mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) \cdot \widehat{f}'_j$ where $\widehat{f}_j$ and $\widehat{f}'_j$ are two independent unbiased estimator of f_j. Intuitively, the expectation of $\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right)$ should be close to $\frac{g_j}{f_j}$ and then the expectation of X_j is close to g_j. To prove the algorithm, we will need the following lemmas.

To prove the algorithm, we will need the following lemmas.

Lemma 4.1 (Continuous F_2 estimator, [3])**.** *Let $0 < \varepsilon < 1$. There is a streaming algorithm that outputs at each time t a value $\widehat{F}_2^{(t)}$ such that*

$$\mathbf{Pr}\left(|\widehat{F}_2^{(t)} - F_2^{(t)}|) \leq \varepsilon F_2^{(t)}, \text{ for all } 0 \leq t \leq m\right) > 1 - \delta\ .$$

The algorithm uses $O(\varepsilon^{-2}\log n \log(1/\delta))$ bits of space.

Lemma 4.2. *Using $O(w^{-2}\log n \log^3(1/\varepsilon))$ bits of space, for an index j sampled by an ℓ_p sampler, we have an estimator $\widehat{f}_j$ to f_j such that $\left|\mathbf{E}[1/\widehat{f}_j] - 1/f_j\right| \leq \varepsilon^8 \cdot 1/f_j$ and with probability at least $1 - \mathrm{poly}(\varepsilon)$, $|f_j - \widehat{f}_j| \leq \frac{w}{100} \cdot f_j$.*

Proof. Let $\boldsymbol{z}$ be the vector $\boldsymbol{f}$ recaled by the exponential random variables $u_i^{1/p}$. We use a CountSketch with $O(w^{-2})$ buckets to estimate the frequency of the sampled index z_j (this can be derandomized from the same procedure in [13]). By the property of CountSketch, we have $\mathbf{E}[\widehat{z}_j] = z_j$. Note that we have that $|z_j|^2 \geq \Theta(1)\|\boldsymbol{z}\|_2^2$ since j is the sampled coordinate. Since we can use $\log(1/\varepsilon)$-wise independence hash function for the random signs of CountSketch, from the standard concentration result, we have with probability $1 - \mathrm{poly}(\varepsilon)$, $|\widehat{z}_j - z_j| \leq \frac{w}{100} z_j$. Multiplying by $u_j^{1/p}$ in both side we get the desired result.

Algorithm 1

1: Initialize $k = \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\right)$ ℓ_p samplers (Section 3) $L_1, L_2, \ldots, L_k$ with parameter w_i where w_i is uniformly sampled in $(\varepsilon, 1)$.
2: For each subroutine in sampler L_i, let $\boldsymbol{z}$ denote the frequency vector $\boldsymbol{f}$ (without forget operations) rescaled by the exponential random variable.
3: **During the stream**: for each sub-routine in L_i, we use a BPTree (Theorem 3.1) to keep track of $(\frac{w_i}{10}, \ell_2)$-heavy hitter of $\boldsymbol{z}$ and a continuous F_2 estimator (Lemma 4.1) to keep track of $F_2(\boldsymbol{z})$. whenever $\widehat{F_2}(\boldsymbol{z})$ increases by a factor of $(1 + O(w_i))$:
   (1) Let J be the set of heavy hitters output by the BPTree.
   (2) Discard all counters C_i that are not associated with one coordinate in J.
   (3) For each $i \in J$, if there is no counter associated with i. Initialize a new counter $C_i = 0$ and use C_i to keep track of the future updates to the coordinate i in the stream. In particular, if there is $(i, +)$ operation, we increase C_i by 1, and if there is a forget operation, we set $C_i = 0$.
4: **After the stream:** for each sampler L_i, let j be the sampled coordinate with rescaled frequency vector $\boldsymbol{z}$.
   (1) Let C_j be the counter associated with j during the stream.
   (2) Let $\widehat{f_j}$ be an estimation of f_j and $\widehat{f_j'}$ be another estimation of f_j.
   (3) Let $\widehat{g_j} = C_j$ if C_j observed a forget operation on j during the stream after it was initialized or we set $\widehat{g_j} = \widehat{f_j}$ otherwise.
   (4) Compute the value $X_j = \mathbf{1}\left(\frac{\widehat{g_j}}{\widehat{f_j}} \geq 1 - w_i\right) \cdot \widehat{f_j'}$ (Lemma 4.3) and the reverse sampling probability $\widehat{\frac{1}{p_j}}$ (Lemma 3.16).
   (5) If $\widehat{g_j}/\widehat{f_j} \leq \frac{1}{2}$, set $\widehat{X_j^p} = \widehat{g_j^p}$, otherwise compute $\widehat{X_j^p}$ using the Taylor series (Lemma 3.11).
5: **Output:** the average of $\widehat{X_j^p} \cdot \widehat{\frac{1}{p_j}}$ for each ℓ_p sampler.

Moreover, by taking $\log^2(1/\varepsilon)$ independent copies and plug into the Taylor's estimator in Lemma 3.15 we can get a $\widehat{f_j}$ such that $\left|\mathbf{E}[1/\widehat{f_j}] - 1/f_j\right| \leq \varepsilon^8 \cdot 1/f_j$. Since with probability at least $1 - \mathrm{poly}(\varepsilon)$, each input here has a distance to f_j within $\frac{w}{100} f_j$, condition on this, we have $|f_j - \widehat{f_j}| \leq \frac{w_i}{100} \cdot f_j$. □

Lemma 4.3. *With probability at least* $1 - \mathrm{poly}(\varepsilon)$*, we have that* $\mathbf{1}\left(\frac{\widehat{g_j}}{\widehat{f_j}} \geq 1 - w_i\right) = \mathbf{1}\left(\frac{g_j}{\widehat{f_j}} \geq 1 - w_i\right)$.

Proof. We first consider the case when $g_j \geq (1 - w_i/3) \cdot f_j$. In this case we have that $\mathbf{1}\left(\frac{g_j}{\widehat{f_j}} \geq 1 - w_i\right) = 1$. Suppose that the counters for heavy hitters observe the last forget operations on j, then we can get the exact value of g_j. On the other hand, if this does not occur we will set $\widehat{g_j} = \widehat{f_j}$, which means that $\mathbf{1}\left(\frac{\widehat{g_j}}{\widehat{f_j}} \geq 1 - w_i\right) = 1$.

We next consider the other case $g_j < (1 - w_i/3) \cdot f_j$. This implies that there was a forget operation after the frequency of f_j became $w_i f_j/3$ during the stream. Recall that in the corresponding rescaled frequency vector $\boldsymbol{z}$ we have $z_j = \Theta(1)\left\|\boldsymbol{z}_{-j}\right\|_2$ and we check the $w_i/10$-heavy hitters of $\boldsymbol{z}$ whenever the F_2 norm of $\boldsymbol{z}$ increase by a $(1 + O(w_i))$-factor. This means that condition on the success of the

continuous F_2 norm estimator and the BPTree we have that the coordinate j will be identified as a heavy hitter before the last operation to j, and the counter will not be discarded in the reminder of the stream. This implies that $\widehat{g}_j = g_j$. □

Lemma 4.4. *With probability at least* $1 - \mathrm{poly}(\varepsilon)$, *we have that* $\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right)\right] = \frac{g_j}{f_j}\left(1 \pm O(\varepsilon)\right)$.

Proof. We first consider the case when $g_j \geq (1 - c\varepsilon)f_j$ for some small constant c. Note that since $w_i \in (\varepsilon, 1)$ we have $\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) = \frac{g_j}{f_j}\left(1 \pm O(\varepsilon)\right)$.

We next consider the other case. Assume $g_j = (1 - v)f_j$, and condition on the event that $|f_j - \widehat{f}_j| \leq \frac{w_i}{100} \cdot f_j$. We have that if $w_i \leq 0.9v$, then $g_j/\widehat{f}_j \leq (1 - v) \cdot (1 + \frac{w_i}{100}) < 1 - w_i$ and $g_j/f_j = (1 - v) < 1 - w_i$. This implies $\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) = \mathbf{1}\left(\frac{g_j}{f_j} \geq 1 - w_i\right) = 0$.

Similarly, when $w_i \geq 1.1v$, we have that $g_j/\widehat{f}_j \geq (1 - v) \cdot (1 - \frac{w_i}{100}) > 1 - w_i$ and $g_j/f_j = (1 - v) > 1 - w_i$. This implies $\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) = \mathbf{1}\left(\frac{g_j}{f_j} \geq 1 - w_i\right) = 1$.

We next consider the case when $0.9v < w_i < 1.1v$. Let $\mathcal{D}$ denote this event, $p_{\mathcal{D}}$ denote the probability that $\mathcal{D}$ occurs, and p_v denote the probability that $w_i \geq 1.1v$. Note that since w_i is uniformly sampled in $(\varepsilon, 1)$, we have $\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{f_j} \geq 1 - w_i\right)\right] = \frac{g_j/f_j}{1-\varepsilon}$. We then have

$$\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{f_j} \geq 1 - w_i\right) \mid \mathcal{D}\right] = \frac{(1-\varepsilon)^{-1} g_j/f_j - p_v}{p_{\mathcal{D}}}$$

from a simple probability computation. On the other hand, we have

$$\begin{aligned}
\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) \mid \mathcal{D}\right] &= \mathbf{E}\left[\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) \mid \widehat{f}_j, \mathcal{D}\right]\right] \\
&= \mathbf{E}\left[\frac{(1-\varepsilon)^{-1} g_j/\widehat{f}_j - p_v}{p_{\mathcal{D}}}\right] = \frac{(1-\varepsilon)^{-1} g_j \cdot \mathbf{E}[1/\widehat{f}_j] - p_v}{p_{\mathcal{D}}} \\
&= \frac{(1-\varepsilon)^{-1}(g_j/f_j) \cdot (1 \pm \varepsilon^8) - p_v}{p_{\mathcal{D}}} .
\end{aligned}$$

Here the last equation follows from Lemma 4.2.

Putting everything together, we have that $\mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right)\right] = \frac{g_j}{f_j}\left(1 + O(\varepsilon)\right)$. □

Lemma 4.5. *With probability at least* $1 - \mathrm{poly}(\varepsilon)$, *we have that* $\mathbf{E}\left[X_j\right] = g_j \cdot \left(1 \pm O(\varepsilon)\right)$ *and* $\mathbf{E}\left[X_j^2\right] = O(f_j \cdot g_j)$

Proof. Recall that from Lemma 4.3 we have that $X_j = \mathbf{1}\left(\frac{g_j}{\widehat{f}_j} \geq 1 - w_i\right) \cdot \widehat{f}_j'$ and $\widehat{f}_j$ and $\widehat{f}_j'$ are two independent random variables. Then from Lemma 4.4 and Lemma 3.10 we have that with

probability at least $1 - \text{poly}(\varepsilon)$,

$$\mathbf{E}\left[X_j\right] = \mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f_j}} \geq 1 - w_i\right)\right] \cdot \mathbf{E}\left[\widehat{f_j'}\right] = \frac{g_j}{f_j} \cdot \left(1 \pm O(\varepsilon)\right) \cdot f_j = g_j \cdot \left(1 \pm O(\varepsilon)\right)$$

and

$$\mathbf{E}\left[X_j^2\right] = \mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f_j}} \geq 1 - w_i\right)^2\right] \cdot \mathbf{E}\left[\left(\widehat{f_j'}\right)^2\right] = \frac{g_j}{f_j} \cdot \left(1 \pm O(\varepsilon)\right) \cdot O(f_j^2) = O(f_j \cdot g_j)\ . \qquad \square$$

We are now ready to prove the following Theorem.

Theorem 4.1. *With high constant probability, Algorithm 1 uses space $\widetilde{O}\left(\frac{1}{\varepsilon^3(1-\alpha)} \cdot \log n\right)$ for $0 < p < 2$ and $\widetilde{O}\left(\frac{1}{\varepsilon^3(1-\alpha)} \cdot \log^2 n\right)$ for $p = 2$, and outputs a value $\widetilde{F}$ with $\left|\tilde{F} - F_p(\boldsymbol{g})\right| \leq \varepsilon F_p(\boldsymbol{g})$ in the α-RFDS model.*

Proof. As in Algorithm 1, we run $N = \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\right)$ ℓ_p sampler in Theorem 3.2. At the end of the stream, we get N sampled indices i where the sampling probability is proportional to the value of f_i^p. For each sampled index j, when $\widehat{g_j}/\widehat{f_j} \leq 1/2$, with probability at least $1 - \text{poly}(\varepsilon)$ we can get the exact value of g_j, otherwise from Lemma 4.5 we can get an estimator X_j such that with probability at least $1 - \text{poly}(\varepsilon)$, $\mathbf{E}\left[X_j\right] = g_j \cdot \left(1 \pm O(\varepsilon)\right)$ and $\mathbf{E}\left[X_j^2\right] = O(f_j \cdot g_j)$. Then, condition on this, from Lemma 3.11 we can use it to compute an estimator $\widehat{X_j^p}$ (we can run $O(\log(1/\varepsilon))$ independent copies of X_j and then plug into the Taylor's series) such that $\mathbf{E}\left[\widehat{X_j^p}\right] = g_j^p \cdot \left(1 \pm O(\varepsilon)\right)$ and $\mathbf{E}\left[\widehat{X_j^{2p}}\right] = O(f_j^p \cdot g_j^p)$. Next, from Lemma 3.16 we have that we can compute an unbiased estimator $\widehat{p_j}$ such that $\left|\mathbf{E}[1/\widehat{p_j}] - 1/p_j\right| \leq \varepsilon^8 \cdot 1/p_i$ and $\mathbf{E}[1/\widehat{p_j}^2] = O(1/p_j^2 \cdot \log^2(1/\varepsilon))$, where $p_j = \frac{f_j^p}{\|f\|_p^p}$. This implies that for the estimator $\frac{\widehat{1}}{p_j} \cdot \widehat{X_j^p}$ we have that

$$\mathbf{E}\left(\widehat{\frac{1}{p_j}} \cdot \widehat{X_j^p}\right) = \sum_{i=1}^{n} p_i \cdot \frac{1}{p_i}\left(1 + \varepsilon^{10}\right) \cdot g_i^p \left(1 \pm O(\varepsilon)\right) = \left(1 \pm O(\varepsilon)\right) \cdot \sum_{i=1}^{n} g_i^p = \left(1 \pm O(\varepsilon)\right) \cdot F_p(\boldsymbol{g}).$$

and

$$\mathbf{E}\left(\widehat{\frac{1}{p_j}} \cdot \widehat{X_j^p}\right)^2 \leq \widetilde{O}(1) \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{p_i^2} \cdot (f_i g_i)^p \leq \widetilde{O}(1) \cdot \sum_{i=1}^{n} \frac{\|\boldsymbol{f}\|_p^p}{f_i^p} (f_i g_i)^p = \widetilde{O}(1) \|\boldsymbol{f}\|_p^p \cdot \sum_{i=1}^{n} g_i^p \leq \widetilde{O}\left(\frac{1}{1-\alpha}\right) \|\boldsymbol{g}\|_p^{2p}\ ,$$

where the $\widetilde{O}(\cdot)$ hides an $O(\log(1/\varepsilon))$ factor and the last inequality follows from the assumption of the α-RFDS model that $\|\boldsymbol{g}\|_p^p \geq (1-\alpha)\|\boldsymbol{f}\|_p^p$. Taking a union bound over all of the N samples, and by Chebyshev's inequality we have that after taking an average of all of the N samples, with high constant probability, we have that $\left|\tilde{F} - F_p(\boldsymbol{g})\right| \leq \varepsilon F_p(\boldsymbol{g})$.

We next consider the space complexity of our algorithm. We first consider the case when $0 < p < 2$. There are $N = \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\right)$ ℓ_p samplers, for the i-th sampler L_i, since we need to keep track of the $w_i/10$-heavy hitters for each rescaled frequency vector $\boldsymbol{z}$, the space usage will be

$\widetilde{O}(w_i^{-2}\log n\cdot \mathrm{poly}\log(1/\varepsilon))$. Hence, the total space usage will be

$$\widetilde{O}\left(\log n\cdot \mathrm{poly}\log(1/\varepsilon)\cdot \sum_{i=1}^{N} w_i^{-2}\right).$$

Recall that w_i are uniformly sampled from $(\varepsilon,1)$. Then we have that $\mathbf{E}\left[\sum_{i=1}^{N} w_i^{-2}\right] = O\left(1/\varepsilon^3\cdot\log\left(\frac{1}{\varepsilon(1-\alpha)}\right)\right)$, which implies the expectation of the space usage of our algorithm will be $\widetilde{O}\left(\frac{1}{\varepsilon^3(1-\alpha)}\cdot\log n\right)$. For the case $p=2$, note that the only difference here is for each ℓ_2 sampler, we have $\widetilde{O}(\log n\log(1/\varepsilon))$ sub-routines instead of $O(\log\log n\log(1/\varepsilon))$ for $0<p<2$, which implies that with high probability, the expectation of the total space usage will be $\widetilde{O}\left(\frac{1}{\varepsilon^3(1-\alpha)}\cdot\log^2 n\right)$. □

Reducing to $1/\varepsilon^2$ dependence. Unfortunately, the above algorithm has a $1/\varepsilon^3$ dependence. We next discuss how to achieve a tight $1/\varepsilon^2$ dependence here. The key idea is that for each range of the value of w_i, we actually have more samples than we need. Specifically, consider a fixed level $[2^\ell\varepsilon, 2^{\ell+1}\varepsilon]$ for some $\ell\in\{0,1,\ldots,\log(1/\varepsilon)\}$. Since the total number of samples is $N=\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)}\right)$, with probability at least $1-\mathrm{poly}(\varepsilon)$, we have the number of w_i in this level is $N_\ell = O\left(2^\ell/\varepsilon\cdot\frac{1}{1-\alpha}\right)$. What we do next is we randomly sample $N_\ell' = O(4^\ell\cdot\frac{1}{1-\alpha})$ of them and assign each of these estimators with weight $\frac{N_\ell}{N_\ell'}$. Clearly, after this step, our overall estimator is still unbiased. We next consider the variance of the new estimator. Note that for each of the ℓ_p sampler in this level, condition on its sampled index is j, we have $\mathbf{E}\left[X_j^2\right] = \mathbf{E}\left[\mathbf{1}\left(\frac{g_j}{\widehat{f}_j}\geq 1-w_i\right)^2\right]\cdot\mathbf{E}\left[\left(\widehat{f_j'}\right)^2\right] \leq \mathbf{E}\left[\left(\widehat{f_j'}\right)^2\right] = O(f_j^2) = O(f_j\cdot g_j)$. The reason for the last equation is that we can, without loss of generality, assume $g_j\geq 1/3\cdot f_j$ as otherwise we can get the exact value of g_i directly. Then, similarly to the previous section, we can have $\mathbf{E}\left(\widehat{\frac{1}{p_j}}\cdot\widehat{X_j^p}\right)^2 \leq \widetilde{O}\left(\frac{1}{1-\alpha}\right)\|\boldsymbol{g}\|_p^{2p}$. From this we have that after the uniform sampling and rescaling, the variance of the sum of estimations in this level (without the averaging) will be at most

$$\left(\frac{N_\ell}{N_\ell'}\right)^2\cdot N_\ell'\cdot\widetilde{O}\left(\frac{1}{1-\alpha}\right)\|\boldsymbol{g}\|_p^{2p} = \widetilde{O}\left(\frac{1}{\varepsilon^2}\cdot\frac{1}{(1-\alpha)^2}\right)\|\boldsymbol{g}\|_p^{2p}.$$

Hence, after taking a sum over the $\log(1/\varepsilon)$ levels and averaging by N, we have that the variance of the final estimator will be at most

$$\frac{1}{N^2}\cdot\log(1/\varepsilon)\cdot\widetilde{O}\left(\frac{1}{\varepsilon^2}\cdot\frac{1}{(1-\alpha)^2}\right)\|\boldsymbol{g}\|_p^{2p} = O\left(\varepsilon^2\right)\cdot\|\boldsymbol{g}\|_p^{2p},$$

which is the same order as before. We finally consider the space usage after this modification. For each level, we only need to maintain the ℓ_p samplers that have been uniformly sampled in this level, this implies the total space complexity will be

$$\sum_{\ell=0}^{\log(1/\varepsilon)} N_\ell'\cdot\frac{1}{2^{2\ell}\cdot\varepsilon^2}\cdot\widetilde{O}(\log^{t(p)} n\cdot\mathrm{poly}\log(1/\varepsilon)) = \widetilde{O}\left(\frac{\log^{t(p)} n}{\varepsilon^2(1-\alpha)}\right),$$

where $t(p)=1$ for $0<p<2$ and $t(p)=2$ for $p=2$. Putting everything together, we have the following theorem.

Theorem 4.2. *Let $0 < p \leq 2$. Let $\boldsymbol{f}$ be the vector without the forget operations and $\boldsymbol{g}$ be the actual frequency vector with forget operations. There is an algorithm that uses space $\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)} \cdot \log n\right)$ for $0 < p < 2$ and $\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)} \cdot \log^2 n\right)$ for $p = 2$, and with high constant probability outputs a value $\widetilde{F}$ with $\left|\tilde{F} - F_p(\boldsymbol{g})\right| \leq \varepsilon F_p(\boldsymbol{g})$ in the α-RFDS model.*

5 F_p Estimation for $p > 2$

We next consider the case $p > 2$. Here our strategy will be based on ℓ_2 sampling. In particular, we consider the same algorithm as shown in Algorithm 1 for $p = 2$ but using the Taylor series to estimate $\widehat{X_j^p}$. Then, from a single estimator $\widehat{\frac{1}{p_j}} \cdot \widehat{X_j^p}$, it is easy to verify that the expectation is still almost unbiased and

$$\mathbf{E}\left(\widehat{\frac{1}{p_j}} \cdot \widehat{X_j^p}\right)^2 \leq \widetilde{O}(1) \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{p_i^2} \cdot (f_i g_i)^p \leq \widetilde{O}(1) \cdot \sum_{i=1}^{n} \frac{\|\boldsymbol{f}\|_2^2}{f_i^2} (f_i g_i)^p$$

To bound this, note that from Hölder's inequality we have $\|\boldsymbol{f}\|_2^2 \leq n^{1-2/p} \|\boldsymbol{f}\|_p^2$ and we can assume $g_i \geq 1/3 f_i$ as otherwise we can get the exact value of g_i. Then we have

$$\begin{aligned}
\mathbf{E}\left(\widehat{\frac{1}{p_j}} \cdot \widehat{X_j^p}\right)^2 &\leq \widetilde{O}(1)\|\boldsymbol{f}\|_2^2 \cdot \sum_{i=1}^{n} f_i^{p-2} g_i^p \leq \widetilde{O}\,(1) \cdot 3^p \cdot \|\boldsymbol{f}\|_2^2 \cdot \|\boldsymbol{g}\|_{2p-2}^{2p-2} \\
&\leq \widetilde{O}\,(1) \cdot 3^p \cdot n^{1-2/p} \|\boldsymbol{f}\|_p^2 \cdot \|\boldsymbol{g}\|_{2p-2}^{2p-2} \leq \widetilde{O}\left(\frac{3^p}{(1-\alpha)^{2/p}}\right) \cdot n^{1-2/p} \cdot \|\boldsymbol{g}\|_p^2 \cdot \|\boldsymbol{g}\|_{2p-2}^{2p-2} \\
&\leq \widetilde{O}\left(\frac{3^p}{(1-\alpha)^{2/p}}\right) \cdot n^{1-2/p} \cdot \|\boldsymbol{g}\|_p^{2p} \quad .
\end{aligned}$$

This implies we only need to set the number of ℓ_2 samplers to be $N = \widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}} \cdot n^{1-2/p}\right)$. Formally, we have the following theorem.

Theorem 5.1. *Let $p > 2$ be a constant. Let $\boldsymbol{f}$ be the vector without the forget operations and $\boldsymbol{g}$ be the actual frequency vector with forget operations. There is an algorithm that uses $\widetilde{O}\left(\frac{1}{\varepsilon^2(1-\alpha)^{2/p}} \cdot n^{1-2/p} \cdot \log^2 n\right)$ bits of space, and with high constant probability outputs a value $\widetilde{F}$ with $\left|\tilde{F} - F_p(\boldsymbol{g})\right| \leq \varepsilon F_p(\boldsymbol{g})$ in the α-RFDS model.*

6 Extensions to General Operations

In this section, we will first present our heavy-hitters data structure. Then, we will give our algorithm for F_p estimation for $p > 0$ based on this heavy-hitters data structure.

6.1 Heavy Hitters

In particular, here we are allowing a group of more general functions including the forget operation. For a class of functions $\mathcal{G}$ which we will describe, at each time t in the stream, we allow an update of the normal addition $(i, +)$ or of the type $(i, g_t \in \mathcal{G})$ which performs $f_i = g_t(f_i)$. In this proceedings version, we consider $\mathcal{G}$ as the class of contracting functions, or functions with $g : \mathbb{N} \to \mathbb{N}$ with the following conditions:

(1) $|g(x) - g(y)| \leq |x - y|$
(2) $g(0) = 0$.

We note that the actual class of functions $\mathcal{G}$ that our algorithm can accommodate is more general and we will formally define it in the full version of this paper. We first consider the case of ℓ_2-heavy hitters.

Theorem 6.1. *There is an algorithm that estimates all coordinates of $\boldsymbol{f}$ to within $\varepsilon\|\boldsymbol{f}\|_2$ additive error using $O\left(\frac{1}{1-\alpha}\frac{1}{\varepsilon^2}\log n\log(1/\delta)\right)$ space in the α-RFDS model with general arbitrary contraction operations.*

Next we show how to solve the ℓ_p heavy-hitters problem for other value of p using essentially the same procedure as for ℓ_2.

Theorem 6.2. *Let $p > 2$. In the α-RFDS model with general operations from $\mathcal{G}$, there is a streaming algorithm to find all indices i with $f_i \geq \varepsilon\|\boldsymbol{f}\|_p$ with high probability using $\widetilde{O}(\frac{1}{\varepsilon^2}\frac{1}{(1-\alpha)^{2/p}}n^{1-2/p})$ space with high probability. Moreover this algorithm gives an additive $\varepsilon\|\boldsymbol{f}\|_p$ approximation to the values of all heavy hitters.*

Theorem 6.3. *Let $0 < p < 2$. In the α-RFDS model with general operations from $\mathcal{G}$, there is a streaming algorithm to find all indices i with $f_i \geq \varepsilon\|\boldsymbol{f}\|_p$ with high probability using $\widetilde{O}(\frac{1}{\varepsilon^2}\frac{1}{(1-\alpha)^{2/p}}\cdot\log n)$ bits space. Moreover this algorithm gives an additive $\varepsilon\|\boldsymbol{f}\|_p$ approximation to the values of all heavy hitters.*

6.2 F_p Estimation for $p > 0$

Based on the heavy hitters algorithm above, we have the following F_p estimation algorithm for $p > 0$.

Theorem 6.4. *There is an algorithm to obtain a $(1 \pm \varepsilon)$ approximation to F_p for $p > 0$ in the α-RFDS model with general operations from $\mathcal{G}$ using $\widetilde{O}\left(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^{2+4/p}}n^{1-2/p}\right)$ bits of space for $p > 2$ and $\widetilde{O}\left(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^{2+4/p}}\cdot\operatorname{poly}\log n\right)$ bits of space for $0 < p \leq 2$.*

7 Lower bounds for the forget model

In this section, we present our lower bounds for the forget model.

Theorem 7.1 (Lower Bound for F_p Moment for $0 \leq p \leq 2$)**.** *In the α-RFDS model, computing a $1 \pm \varepsilon$ approximation to the F_p moment for $p \in [0, 2]$ requires $\Omega(\frac{1}{1-\alpha}\frac{1}{\varepsilon^2}\log n)$ space.*

Theorem 7.2 (Lower Bound for F_p Moments for $p \geq 2$)**.** *A streaming algorithm for solving solving F_p moment estimation to $(1\pm\varepsilon)$ multiplicative error in the α-RFDS model requires $\Omega(\frac{1}{(1-\alpha)^{2/p}}\frac{1}{\varepsilon^2}n^{1-2/p})$ space.*

Theorem 7.3 (Lower Bound for Turnstile Stream)**.** *In the $\frac{2}{3}$-forget model with strict turnstile updates, approximating either the F_0 moment, or any F_p moment with $p \geq 1$ to relative error $1 \pm \frac{1}{10}$ with $9/10$ probability requires $\Omega(n/\log n)$ space.*

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

[3] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. 2017. BPTree: An $_2$ 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

[4] Vladimir Braverman, Elena Grigorescu, Harry Lang, David P. Woodruff, and Samson Zhou. 2018. Nearly Optimal Distinct Elements and Heavy Hitters on Sliding Windows. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*. 7:1–7:22.

[5] Vladimir Braverman, Rafail Ostrovsky, and Carlo Zaniolo. 2012. Optimal Ssampling from Sliding Windows. *J. Comput. System Sci.* 78, 1 (2012), 260–272. doi:10.1016/j.jcss.2011.04.004

[6] 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

[7] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 2002. Maintaining Stream Statistics over Sliding Windows. *SIAM J. Comput.* 31, 6 (2002), 1794–1813. https://doi.org/10.1137/S0097539701398363

[8] David J. DeWitt, Jeffrey F. Naughton, Donovan A. Schneider, and S. Seshadri. 1992. Practical Skew Handling in Parallel Joins. In *18th International Conference on Very Large Data Bases, August 23-27, 1992, Vancouver, Canada, Proceedings*, Li-Yan Yuan (Ed.). Morgan Kaufmann, 27–40. http://www.vldb.org/conf/1992/P027.PDF

[9] André Gronemeier and Martin Sauerhoff. 2009. Applying Approximate Counting for Computing the Frequency Moments of Long Data Streams. *Theory Comput. Syst.* 44, 3 (2009), 332–348. doi:10.1007/S00224-007-9048-Z

[10] Peter J. Haas, Jeffrey F. Naughton, S. Seshadri, and Lynne Stokes. 1995. Sampling-Based Estimation of the Number of Distinct Values of an Attribute. In *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio (Eds.). Morgan Kaufmann, 311–322. http://www.vldb.org/conf/1995/P311.PDF

[11] Piotr Indyk, Eric Price, and David P. Woodruff. 2011. On the Power of Adaptivity in Sparse Recovery. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, Rafail Ostrovsky (Ed.). IEEE Computer Society, 285–294. doi:10.1109/FOCS.2011.83

[12] Yannis E. Ioannidis and Viswanath Poosala. 1995. Balancing Histogram Optimality and Practicality for Query Result Size Estimation. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, USA, May 22-25, 1995*, Michael J. Carey and Donovan A. Schneider (Eds.). ACM Press, 233–244. doi:10.1145/223784.223841

[13] 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

[14] 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*, Maurizio Lenzerini and Thomas Schwentick (Eds.). ACM, 49–58.

[15] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. 2011. Fast Moment Estimation in Data Streams in Optimal Space. In *STOC'11—Proceedings of the 43rd ACM Symposium on Theory of Computing*. ACM, New York, 745–754. doi:10.1145/1993636.1993735

[16] 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*. 475–486.

[17] Yi Li, David P. Woodruff, and Taisuke Yasuda. 2021. Exponentially Improved Dimensionality Reduction for l1: Subspace Embeddings and Independence Testing. In *Conference on Learning Theory, COLT*. 3111–3195.

[18] Morteza Monemizadeh and David P. Woodruff. 2010. 1-Pass Relative-Error Lp-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.

[19] H. N. Nagaraja. 2006. Order Statistics from Independent Exponential Random Variables and the Sum of the Top Order Statistics. In *Advances in distribution theory, order statistics, and inference*. Birkhäuser Boston, Boston, MA, 173–185. doi:10.1007/0-8176-4487-3_11

[20] 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

[21] John P. Nolan. [2020] ©2020. *Univariate Stable Distributions: Models for Heavy Tailed Data.* Springer, Cham. xv+333 pages. doi:10.1007/978-3-030-52915-4

[22] Aduri Pavan, Sourav Chakraborty, N. V. Vinodchandran, and Kuldeep S. Meel. 2024. On the Feasibility of Forgetting in Data Streams. *Proc. ACM Manag. Data* 2, 2 (2024), 102.

[23] V. M. Zolotarev. 1986. *One-Dimensional Stable Distributions.* Translations of Mathematical Monographs, Vol. 65. American Mathematical Society, Providence, RI. x+284 pages. doi:10.1090/mmono/065 Translated from the Russian by H. H. McFaden, Translation edited by Ben Silver.

Received December 2025; accepted February 2026