Tight Bounds for Adversarially Robust Streams and Sliding Windows via Difference Estimators

David P. Woodruff, Samson Zhou
Carnegie Mellon University
dwoodruf@cs.cmu.edu, samsonzhou@gmail.com

***Abstract*—In the adversarially robust streaming model, a stream of elements is presented to an algorithm and is allowed to depend on the output of the algorithm at earlier times during the stream. In the classic insertion-only model of data streams, Ben-Eliezer *et al.* (PODS 2020, best paper award) show how to convert a non-robust algorithm into a robust one with a roughly $1/\varepsilon$ factor overhead. This was subsequently improved to a $1/\sqrt{\varepsilon}$ factor overhead by Hassidim *et al.* (NeurIPS 2020, oral presentation), suppressing logarithmic factors. For general functions the latter is known to be best-possible, by a result of Kaplan *et al.* (CRYPTO 2021). We show how to bypass this impossibility result by developing data stream algorithms for a large class of streaming problems, *with no overhead in the approximation factor*. Our class of streaming problems includes the most well-studied problems such as the L_2-heavy hitters problem, F_p-moment estimation, as well as empirical entropy estimation. We substantially improve upon all prior work on these problems, giving the first optimal dependence on the approximation factor.**

As in previous work, we obtain a general transformation that applies to any non-robust streaming algorithm and depends on the so-called flip number. However, the key technical innovation is that we apply the transformation to what we call a *difference estimator* for the streaming problem, rather than an estimator for the streaming problem itself. We then develop the first difference estimators for a wide range of problems. Our difference estimator methodology is not only applicable to the adversarially robust model, but to other streaming models where temporal properties of the data play a central role. To demonstrate the generality of our technique, we additionally introduce a general framework for the related sliding window model of data streams and resolve longstanding open questions in that model, obtaining a drastic improvement from the previous $1/\varepsilon^{2+p}$ dependence for F_p-moment estimation for $p \in [1,2]$ and integer $p > 2$ of Braverman and Ostrovsky (FOCS, 2007), to the optimal $1/\varepsilon^2$ bound. We also improve the prior $1/\varepsilon^3$ bound for $p \in [0,1)$, and the prior $1/\varepsilon^4$ bound for empirical entropy, obtaining the first optimal $1/\varepsilon^2$ dependence for both of these problems as well. Qualitatively, our results show there *is no separation* between the sliding window model and the standard data stream model in terms of the approximation factor.

I. Introduction

Efficient computation of statistics over large datasets is increasingly important. Such datasets include logs generated from internet traffic, IoT sensors, financial markets, and scientific observations. To capture these applications, the streaming model defines an underlying dataset through updates that arrive sequentially and describe the evolution of the dataset over time. The goal is to approximate statistics of the input using memory, i.e., space complexity, that is significantly sublinear in the input size n, while only making a single pass over the data.

Adversarially robust streaming model.: In the adversarially robust streaming model, the input is adaptively chosen by an adversary who is given unlimited computational resources and may view the outputs of the streaming algorithm at previous times in the stream. The goal of the adversary is to design the input to the streaming algorithm so that the algorithm eventually outputs an incorrect answer. One application of the model is to recommendation systems, where a large set of possible items arrives in a data stream and the goal is to produce a list of fixed size, i.e., a cardinality constraint, so as to maximize a predetermined function, e.g., a submodular function representing a user's utility [BMSC17], [MBN+17], [AMYZ19]. However, the set of items might subsequently be modified by an honest user based on their personal preferences, e.g., to avoid items they already have. Similar notions of adversarial robustness have been the recent focus of a line of work [KMGG08], [MNS11], [HU14], [OSU18], [BLV19], [NY19], [BY20], [BJWY20], [HKM+20].

Sliding window model.: A related data stream model where temporal properties play a central role is the *sliding window model*. The streaming model does not capture applications in which recent data is considered more accurate and important than data that arrived prior to a certain time. For a number of applications [BBD+02], [MM12], [PGD15], [WLL+16], the unbounded streaming model has performance inferior to the sliding window model [DGIM02], where the underlying dataset consists of only the W most recent updates in the stream, for a parameter $W > 0$ that denotes the window size of the active data. All updates

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00116

before the W most recent updates are expired, and the goal is to aggregate information about the active data using space sublinear in W.

A. Our Contributions

We show that there is no loss in $\frac{1}{\varepsilon}$ factors, up to logarithmic factors, over the standard model of data streams for all of the aforementioned central data stream problems, in either the adversarially robust streaming model or the sliding window model. Our results hold for F_p-moment estimation for $p \in [0, 2]$ and integers $p > 2$, L_2-heavy hitters, and empirical entropy estimation, and we give a general framework that can be applied to other problems as well. Our techniques introduce the following crucial concept, which surprisingly had not been considered for data streams before:

Definition I.1 (Difference Estimator)**.** *Given frequency vectors u and v, an accuracy parameter $\varepsilon > 0$, a failure probability $\delta \in (0, 1)$, and a ratio parameter $\gamma \in (0, 1]$, a $(\gamma, \varepsilon, \delta)$-difference estimator for a function F outputs an additive $\varepsilon \cdot F(u)$ approximation to $F(u+v) - F(u)$ with probability at least $1-\delta$, given $F(u+v)-F(u) \le \gamma \cdot F(u)$ and $F(v) \le \gamma F(u)$.*

It turns out that difference estimators for the frequency moments F_p can be used as building blocks for many other streaming problems, so these will be our focus. We show:

Theorem I.2. *There exist difference estimators for the F_p-moment problem for $p \in [0, 2]$ and integers $p > 2$. In particular, the difference estimator uses:*

(1) $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\log n + \log\frac{1}{\delta}\right) + \log n\right)$ *bits of space for the distinct elements problem, F_0. (See Lemma VI.2.)*

(2) $\mathcal{O}\left(\frac{\gamma \log n}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right)$ *bits of space for F_2. (See Lemma III.3.)*

(3) $\mathcal{O}\left(\frac{\gamma^{2/p}\log n}{\varepsilon^2}(\log\log n)^2\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right)$ *bits of space for F_p with $p \in (0, 2)$. (See Lemma IV.3.)*

(4) $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2} n^{1-2/p}\log^3 n \log\frac{n}{\delta}\right)$ *bits of space for F_p for integer $p > 2$. (See Lemma V.2).*

Using our concept of difference estimators, we develop quite general frameworks for both the adversarially robust streaming model and the sliding window model.

1) Our Results for Adversarially Robust Streams.: We first present a space-efficient framework for adversarially robust streaming algorithms, provided there exists a corresponding difference estimator and strong tracker, i.e., a streaming algorithm that is correct at all times in the stream (see, e.g., [BCIW16], [BCI+17], [BDN17], [Bla20] for examples of strong trackers). For a variety of specific problems, we can further optimize our results, summarized in Figure 1:

Theorem I.3. *There exists an adversarially robust streaming algorithm that outputs a $(1+\varepsilon)$-approximation to:*

(1) *The distinct elements problem, F_0, on insertion-only streams, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2} + \frac{1}{\varepsilon}\log n\right)$ bits of space. (See Theorem VI.4.)*

(2) *The F_p-moment estimation problem for $p \in (0, 2]$ on insertion-only streams, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log n\right)$ bits of space. (See Theorem III.5 and Theorem IV.5.)*

(3) *The Shannon entropy estimation problem on insertion-only streams, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log^3 n\right)$ bits of space. (See Theorem IV.7.)*

(4) *The F_p-moment estimation problem for integer $p > 2$ on insertion-only streams, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2} n^{1-2/p}\right)$ bits of space. (See Theorem V.3.)*

(5) *The L_2-heavy hitters problem on insertion-only streams, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log n\right)$ bits of space. (See Theorem III.6.)*

(6) *The F_p-moment estimation problem on turnstile streams with flip number λ, using $\tilde{\mathcal{O}}\left(\frac{\lambda}{\varepsilon}\log^2 n\right)$ bits of space for $p \in [0, 2]$. (See full version of paper.)*

2) Our Results for the Sliding Window Model.: We next modify the difference estimators from Theorem I.2 to develop a general framework for algorithms in the sliding window model, substantially improving upon the smooth histogram framework, and resolving long-standing questions on moment and entropy estimation algorithms in this model.

We obtain the following results for important streaming problems:

Theorem I.4. *Let $\varepsilon, \delta > 0$ be given. There exist sliding window algorithms that output a $(1+\varepsilon)$-approximation to:*

(1) *The F_p-moment estimation problem for $p \in (0, 2]$, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log^3 n\right)$ bits of space. (See Theorem VII.5.)*

(2) *The F_p-moment estimation problem for integers $p > 2$, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2} n^{1-2/p}\right)$ bits of space. (See Theorem VII.7.)*

(3) *The Shannon entropy estimation problem, using $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log^5 n\right)$ bits of space. (See Theorem VII.6.)*

Thus, our results show that no loss in $\frac{1}{\varepsilon}$ factors is necessary in the sliding window model, bypassing previous algorithms that were limited by the smooth histogram framework. The previous framework of [BO07]

Problem	[BJWY20] Space	[HKM+20] Space	Our Result
Distinct Elements	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}+\frac{\log n}{\varepsilon}\right)$
F_p Estimation, $p \in (0,2]$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2}\right)$
Shannon Entropy	$\tilde{\mathcal{O}}\left(\frac{\log^6 n}{\varepsilon^5}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{3.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_2-Heavy Hitters	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^4 n}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2}\right)$
F_p Estimation, integer $p > 2$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2.5}}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^2}\right)$
F_p Estimation, $p \in (0,2]$, flip number λ	$\tilde{\mathcal{O}}\left(\frac{\lambda \log^2 n}{\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n\sqrt{\lambda \log n}}{\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\lambda \log^2 n}{\varepsilon}\right)$

Fig. 1: Adversarially robust streaming algorithms.

has an $\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^3}\right)$ space dependence for $p \in (0,1]$, a $\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^{2+p}}\right)$ space dependence for $p \in (1,2]$, an $\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2+p}}\right)$ space dependence for $p > 2$, and an $\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^4}\right)$ space dependence for entropy estimation using known techniques [HNO08]. We note that there are specialized algorithms for the distinct elements and L_2-heavy hitters prolems in the sliding window model that also achieve the optimal dependence on the approximation factor [BGL+18], so we do not state our results for those problems in this model. We summarize our sliding window model results in Figure 2.

II. Framework for Adversarially Robust Streaming Algorithms

In this section, we describe a general framework for adversarially robust streaming algorithms, using the sketch stitching and granularity changing techniques. We first require the following specific form of a difference estimator.

Definition II.1 (Fixed-Prefix Difference Estimator)**.** *Given a stream $\mathcal{S}$, a fixed time t_1, and a splitting time t_2 that is only revealed at time t_2, let frequency vector v be induced by the updates of $\mathcal{S}$ from time t_1 to t_2 and frequency vector w_t be induced by updates from time t_2 to t exclusive. Given an accuracy parameter $\varepsilon > 0$ and a failure probability $\delta \in (0,1)$, a streaming algorithm $\mathcal{B}(t_1, t_2, t, \gamma, \varepsilon, \delta)$ is a $(\gamma, \varepsilon, \delta)$-*difference estimator *for a function F if, with probability at least $1-\delta$, it outputs an additive $\varepsilon \cdot F(v)$ approximation to $F(v+w_t) - F(v)$ simultaneously for all $t \geq t_2$ with $F(v+w_t) - F(v) \leq \gamma \cdot F(v)$ and $F(w_t) \leq \gamma F(v)$ for a ratio parameter $\gamma \in (0,1]$.*

We shall use the shorthand "difference estimator" terminology to refer to Definition I.1 until Section VII; in Section VII we will introduce an additional notion of a difference estimator for when $F(v)$ can change.

We first describe a simplified version of our adversarially robust framework that adapts the usage of difference estimators. To achieve a robust $(1+\mathcal{O}(\varepsilon))$-approximation, [BJWY20] used a "switch-a-sketch" technique that maintains (ε, m)-flip number λ independent subroutines that each provide a $(1+\varepsilon)$ to a function F evaluated on the frequency vector induced by the stream, with high probability. For the remainder of the discussion, we assume $\lambda = \Omega\left(\frac{1}{\varepsilon}\log n\right)$, which is true for many important functions F, especially the important F_p moments. To prevent an adversary from affecting the output of the algorithm, each subroutine is effectively only used once. The output of the i-th subroutine is only used the first time the true output of the subroutine is at least $(1+\varepsilon)^i$. The algorithm of [BJWY20] then repeatedly outputs this value until the $(i+1)$-st subroutine is at least $(1+\varepsilon)^{i+1}$, at which point the algorithm switches to using the output of the $(i+1)$-st subroutine instead. Hence, the adversary information-theoretically knows nothing about the internal randomness of the i-th subroutine until the output is at least $(1+\varepsilon)^i$. However, due to monotonicity of F and correctness of the oblivious $(i+1)$-st instance, whatever knowledge the adversary gains about the i-th instance does not impact the internal randomness of future instances. Intuitively, the switch-a-sketch approach uses a sketch once and switches to another sketch once the estimated F has increased by $(1+\varepsilon)$. F can only increase λ times by definition of the (ε, m)-flip number. Since $\lambda = \Omega\left(\frac{1}{\varepsilon}\log n\right)$, this approach generally achieves $\frac{1}{\varepsilon^3}$ space dependency.

We first observe that if we instead use the switch-a-sketch technique each time F increases by a power of 2, then we only need to switch $\mathcal{O}(\log n)$ sketches. Effectively, this follows from setting $\varepsilon = \mathcal{O}(1)$ in the value of λ. Let t_i be the first time F of the stream surpasses 2^i. The challenge is then achieving a $(1+\varepsilon)$-approximation to F at the times between each 2^i and 2^{i+1}. Let u be the underlying frequency vector at time t_i, so that $F(u) \geq 2^i$. If v is the underlying frequency vector at some time between t_i and t_{i+1}, then we can decompose $F(v) = F(u) + \sum_{j=1}^{\beta}(F(u_j) - F(u_{j-1}))$,

Problem	[BO07] Space	Our Result
L_p Estimation, $p \in (0,1)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, $p \in (1,2]$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^3 n}{\varepsilon^2}\right)$
L_p Estimation, integer $p > 2$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^{2+p}}\right)$	$\tilde{\mathcal{O}}\left(\frac{n^{1-2/p}}{\varepsilon^2}\right)$
Entropy Estimation	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^4}\right)$	$\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^2}\right)$

Fig. 2: Sliding window algorithms.

Algorithm 1 Framework for Robust Algorithms on Insertion-Only Streams

Input: Stream $u_1, \ldots, u_m \in [n]$ of updates to coordinates of an underlying frequency vector, accuracy parameter $\varepsilon \in (0,1)$, $(\gamma, \varepsilon, \delta)$-difference estimator $\mathcal{B}$ for F with space dependency $\frac{\gamma^C}{\varepsilon^2}$ for $C \geq 1$, oblivious strong tracker $\mathcal{A}$ for F

Output: Robust $(1+\varepsilon)$-approximation to F

$\delta \leftarrow \frac{1}{\text{poly}\left(\frac{1}{\varepsilon}, \log n\right)}$, $\zeta \leftarrow \frac{2}{2^{(C-1)/4}-1}$, $\eta \leftarrow \frac{\varepsilon}{64\zeta}$, $\beta \leftarrow \lceil \log \frac{8}{\varepsilon} \rceil$

$a \leftarrow 0$, $\varphi \leftarrow 2^{(C-1)/4}$, $\gamma_j \leftarrow 2^{j-1}\eta$

For $j \in [\beta]$, $\eta_j \leftarrow \frac{\eta}{\beta}$ if $C = 1$, $\eta_j \leftarrow \frac{\eta}{\varphi^{\beta-j}}$ if $C > 1$. ▷Accuracy for each difference estimator

for each update $u_t \in [n]$, $t \in [m]$ **do**

  $X \leftarrow \mathcal{A}_{a+1}(1, t, \eta, \delta)$

  if $X > 2^a$ **then** ▷Switch sketch at top layer

    $a \leftarrow a+1$, $b \leftarrow 0$, $Z_a \leftarrow X$, $t_{a,j} \leftarrow t$ for $j \in [\beta]$.

  $X \leftarrow$ EstimateF ▷Compute estimator X for F using unrevealed sketch

  if $X > \left(1 + \frac{(b+1)\varepsilon}{8}\right) \cdot Z_a$ **then** ▷Switch sketch at lower layer

    $b \leftarrow b+1$, $k \leftarrow \text{lsb}(b,1)$, $j \leftarrow \left\lfloor \frac{b}{2^k} \right\rfloor$

    $Z_{a,k} \leftarrow \mathcal{B}_{a,j}(1, t_{a,k}, t, \gamma_k, \eta_k, \delta)$ ▷Freeze old sketch

    $t_{a,j} \leftarrow t$ for $j \in [k]$. ▷Update difference estimator times

  return $\left(1 + \frac{b\varepsilon}{8}\right) \cdot Z_a$ ▷Output estimate for round t

where we use the convention that $u_0 = u$ and $u_\beta = j$. Moreover, we assume that $F(u_j) - F(u_{j-1}) \leq \gamma \cdot F(v)$ for $\gamma \leq \frac{1}{2^j}$.

Our key observation is that because we only care about a $(1+\varepsilon)$-approximation to $F(v)$, we do not need a $(1+\varepsilon)$-approximation to each of the differences $F(u_j) - F(u_{j-1})$, which may be significantly smaller than $F(v)$. For example, note that a $(2^j \cdot \varepsilon)$-approximation to $F(u_j) - F(u_{j-1})$ only equates to an additive $\mathcal{O}(\varepsilon \cdot F(v))$ error, since $F(u_j) - F(u_{j-1}) = \mathcal{O}\left(\frac{1}{2^j}\right) \cdot F(v)$. We require $\frac{1}{\gamma}$ instances of algorithms with such accuracies, to account for the various possible vectors v. Thus if there exists an algorithm that uses $\frac{\gamma}{\varepsilon^2} S(n)$ bits of space to output additive $\varepsilon \cdot F(v)$ error, then the space required across the level j estimators is $\frac{1}{\varepsilon^2} S(n)$ bits of space. Since there are at most $\mathcal{O}\left(\log \frac{1}{\varepsilon}\right)$ levels, then we do not incur any additional factors in $\frac{1}{\varepsilon}$. Recall that a difference estimator (Definition I.1) to F serves exactly this purpose!

It is not obvious how to obtain a difference estimator for various functions F. However, for the purposes of a general framework, the theoretical assumption of such a quantity suffices; we shall give explicit difference estimators for specific functions F of interest. The framework appears in full in Algorithm 1.

Algorithm 2 Subroutine EstimateF of Algorithm 1

$X \leftarrow Z_a$, $k \leftarrow \text{numbits}(b+1)$, $z_i \leftarrow \text{lsb}(b+1, k+1-i)$ for $i \in [k]$.

▷$z_1 > \ldots > z_k$ are the nonzero bits in the binary representation of $b+1$.

for $1 \leq j \leq k-1$ **do** ▷Compile previous frozen components for estimator X

  $X \leftarrow X + Z_{a,j}$

$j \leftarrow \left\lfloor \frac{b+1}{2^{z_k}} \right\rfloor$

$X \leftarrow X + \mathcal{B}_{a,j}\left(1, t_{a,z_k}, t, \gamma_{z_k}, \eta_{z_k}, \delta\right)$ ▷Use unrevealed sketch for last component

return X

Theorem II.2 (Framework for adversarially robust algorithms on insertion-only streams)**.** *Let $\varepsilon, \delta > 0$ and F be a monotonic function with (ε, m)-flip* *number* $\lambda = \mathcal{O}\left(\frac{\log n}{\varepsilon}\right)$ *on a stream of length* m*, with* $\log m = \mathcal{O}(\log n)$*. Suppose there exists a* $(\gamma, \varepsilon, \delta)$*-difference estimator for* F *that uses* $\mathcal{O}\left(\frac{\gamma^C}{\varepsilon^2} \cdot S_1(n, \delta, \varepsilon) + S_2(n, \delta, \varepsilon)\right)$ *bits of space for some constant* $C \geq 1$ *and a strong tracker for* F *that use* $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot S_1(n, \delta, \varepsilon) + S_2(n, \delta, \varepsilon)\right)$ *bits of space and functions* S_1, S_2 *that depend on* F*. Then there exists an adversarially robust streaming algorithm that outputs a* $(1+\varepsilon)$*-approximation for* F *that succeeds with constant probability. The algorithm uses*

$$\tilde{\mathcal{O}}\left(\frac{\log n}{\varepsilon^2} S_1(n, \delta', \varepsilon) + \frac{\log n}{\varepsilon} S_2(n, \delta', \varepsilon) + \frac{\log^2 n}{\varepsilon^2}\right)$$

bits of space.

III. Robust F_2 Estimation

In this section, we use the previous framework of Section II to give an adversarially robust streaming algorithm for F_2 moment estimation. Recall that to apply Theorem II.2, we require an F_2 strong tracker and an F_2 difference estimator. We present these subroutines in this section. We further optimize our algorithm beyond the guarantees of Theorem II.2 specifically for F_p moments, so that our final space guarantees in Theorem III.5 matches the best known F_2 algorithm on insertion-only streams, up to lower order $\text{polylog}\,\frac{1}{\varepsilon}$ terms. Finally, we show that our algorithm naturally extends to the problem of finding the L_2-heavy hitters, along with producing an estimate for the frequency of each heavy-hitter up to an additive $\mathcal{O}(\varepsilon) \cdot L_2$ error.

We first recall the following F_2 strong tracker.

Theorem III.1 (Oblivious F_2 strong tracking)**.** *[BDN17] Given an accuracy parameter* $\varepsilon > 0$ *and a failure probability* $\delta \in (0, 1)$*, let* $d = \mathcal{O}\left(\frac{1}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta} + \log\log n\right)\right)$*. There exists an insertion-only streaming algorithm* Estimator *that uses* $\mathcal{O}(d \log n)$ *space to provide* (ε, δ)*-strong* F_2 *tracking of an underlying frequency vector* f*.*

To define our difference estimator, we first note that "good" F_2 approximation to two vectors u and v also gives a "good" approximation to their inner product $\langle u, v\rangle$.

Lemma III.2 (Strong tracking of AMS inner product approximation)**.** *Given vectors* $0^n \preceq u_1 \preceq u_2 \preceq \ldots \preceq u_m \in \mathbb{R}^n$ *whose entries are bounded by a polynomial in* n*, there exists an algorithm that uses a sketching matrix* $M \in \mathbb{R}^{d \times n}$ *with* $d = \mathcal{O}\left(\frac{1}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta} + \log\log n\right)\right)$ *such that for* $m = \text{poly}(n)$ *and a fixed* $v \in \mathbb{R}^n$ *with* $v \succeq 0^n$*,*

$$|\langle u_i, v\rangle - \langle Mu_i, Mv\rangle| \leq \varepsilon \|u_i\|_2 \|v\|_2,$$

simultaneously for all $i \in [m]$ *with probability at least* $1 - \delta$*.*

We now give our F_2 difference estimator using the inner product approximation property.

Lemma III.3 (F_2 difference estimator)**.** *There exists a* $(\gamma, \varepsilon, \delta)$-difference estimator *for* F_2 *that uses space*

$$\mathcal{O}\left(\frac{\gamma \log n}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right).$$

Theorem III.4. *Given* $\varepsilon > 0$*, there exists an adversarially robust streaming algorithm that outputs a* $(1 + \varepsilon)$*-approximation for* F_2 *that uses* $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log^2 n \log^3 \frac{1}{\varepsilon}\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$ *bits of space and succeeds with probability at least* $\frac{2}{3}$*.*

Moreover, the algorithm can be optimized as follows:

Theorem III.5 (Adversarially robust F_2 streaming algorithm)**.** *Given* $\varepsilon > 0$*, there exists an adversarially robust streaming algorithm that outputs a* $(1 + \varepsilon)$*-approximation for* F_2 *that succeeds with probability at least* $\frac{2}{3}$ *and uses* $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n \log^4 \frac{1}{\varepsilon}\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$ *bits of space.*

Heavy-hitters.: As a simple corollary, note that our framework also solves the L_2-heavy hitters problem. By running separate L_2-heavy hitters algorithms corresponding to each difference estimator $\mathcal{A}$ and strong tracker $\mathcal{B}$, with the heavy-hitter threshold corresponding to the accuracy of each procedure, we obtain a list containing all the possible heavy-hitters along with an estimated frequency of each item in the list.

Theorem III.6 (Adversarially robust L_2-heavy hitters streaming algorithm)**.** *Given* $\varepsilon > 0$*, there exists an adversarially robust streaming algorithm* HeavyHitters *that solves the* L_2*-heavy hitters problem with probability at least* $\frac{2}{3}$ *and uses* $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n \log^4 \frac{1}{\varepsilon}\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$ *bits of space.*

IV. Robust F_p Estimation for $0 < p < 2$

In this section, we use the framework of Section II to give an adversarially robust streaming algorithm for F_p moment estimation, where $p \in (0, 2)$. We first require the following definition for p-stable distributions, which will be integral to both our F_p strong tracker and our F_p difference estimator.

Definition IV.1 (p-stable distribution)**.** *[Zol89] For $0 < p \le 2$, there exists a probability distribution $\mathcal{D}_p$ called the p-stable distribution so that for any positive integer n with $Z_1, \ldots, Z_n \sim \mathcal{D}_p$ and vector $x \in \mathbb{R}^n$, then $\sum_{i=1}^n Z_i x_i \sim \|x\|_p Z$ for $Z \sim \mathcal{D}_p$.*

The probability density function f_X of a p-stable random variable X satisfies $f_X(x) = \Theta\left(\frac{1}{1+|x|^{1+p}}\right)$ for $p < 2$, while the normal distribution corresponds to $p = 2$. Moreover, [Nol03] details standard methods for generating p-stable random variables by taking θ uniformly at random from the interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, r uniformly at random from the interval $[0, 1]$, and setting

$$X = f(r, \theta) = \frac{\sin(p\theta)}{\cos^{1/p}(\theta)} \cdot \left(\frac{\cos(\theta(1-p))}{\log \frac{1}{r}}\right)^{\frac{1}{p}-1}.$$

These p-stable random variables are crucial to obtaining a strong F_p tracking algorithm.

Theorem IV.2 (Oblivious F_p strong tracking for $0 < p < 2$)**.** *[BDN17] For $0 < p < 2$, there exists an insertion-only streaming algorithm* PSTABLE$(1, t, \varepsilon, \delta)$ *that uses $\mathcal{O}\left(\frac{\log n}{\varepsilon^2}\left(\log\log n + \log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right)$ bits of space and provides (ε, δ)-strong F_p tracking.*

Unfortunately, PSTABLE is based on the p-stable sketch of [Ind06], which offers a conceptual promise for the existence of the quantile estimators needed to guarantee provable bounds, but not an explicit computation. Thus adapting the analysis of PSTABLE in [Ind06], [BDN17] for the purposes of our difference estimator seems to be a challenge. Even for the case $p = 1$, it does not seem evident how to adapt the median estimator of PSTABLE to obtain a difference estimator for F_p.

Instead, we describe a formulation of Li's geometric mean estimator [Li08], which also provides a streaming algorithm for F_p, but was not previously known to offer strong tracking. For a positive integer $q \ge 3$, let d be a multiple of q and let $A \in \mathbb{R}^{d \times n}$ have independent p-stable random variables for the entries of A. Then for a vector $x \in \mathbb{R}^n$ and $y = Ax$, let $z_i := C_{q,p} \cdot \left(\prod_{j=q(i-1)+1}^{qi} |y_j|^{p/q}\right)$ be the geometric mean of the inner products of q random p-stable vectors with the vector x, where

$$C_{q,p} = \left[\frac{2}{\pi} \cdot \Gamma\left(1 - \frac{1}{q}\right) \cdot \Gamma\left(\frac{p}{q}\right) \cdot \sin\left(\frac{\pi p}{2q}\right)\right]^{-q}.$$

A. F_p Difference Estimator for $0 < p < 2$

We now describe our F_p difference estimator and give the high-level details of the analysis. We use Li's geometric mean estimator to maintain $A(v + w_t)$ and Av, where A is the sketching matrix for Li's geometric mean estimator, and v and w_t are frequency vectors. Observe that if we computed $A(v + w_t) - Av$, then we would obtain $A(w_t)$, which is a sketch that allows us to recover $F_p(w_t)$. However, we want to estimate $F_p(v + w_t) - F_p(v)$ rather than $F_p(w_t)$. Instead, we use the sketches $A(v + w_t)$ and Av to compute terms $z_1, z_2, \ldots, z'_1, z'_2, \ldots$, where each z_i is the geometric mean of q consecutive entries in $A(v+w_t)$ and similarly z'_i is the geometric mean of q consecutive entries in Av. Since z_i is an unbiased estimator of $F_p(v + w_t)$ and z_i is an unbiased estimator of $F_p(v)$, it follows that $z_i - z'_i$ is an unbiased estimator of $F_p(v+w_t) - F_p(v)$. We take the average of the values $z_i - z'_i$ across $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$ indices of i to obtain a single estimate and take the median of all estimates. The challenge is achieving both the variance bounds on $z_i - z'_i$ while also obtaining a strong tracking property. To bound the variance, we expand $z_i - z'_i$ to be a sum of $2^q - 1$ geometric means of q terms, each with at least one term $(\langle A_j, w_t\rangle)^{p/q}$. Since A_j is a vector of p-stable entries, then $(\langle A_j, w_t\rangle)^{p/q}$ has the same distribution as $(\|w_t\|_p \cdot X)^{p/q}$ for a p-stable random variable X. We also have $F_p(w_t) = \gamma \cdot F(v)$, where γ is bounded by some absolute constant. Thus we can bound the probability that $(\langle A_j, w_t\rangle)^{p/q} \ge \|v\|_p^{p/q}$.

Lemma IV.3 (F_p difference estimator for $0 < p < 2$)**.** *For $0 < p < 2$, there exists a $(\gamma, \varepsilon, \delta)$*-difference estimator *for F_p that uses space*

$$\mathcal{O}\left(\frac{\gamma^{2/p}\log n}{\varepsilon^2}(\log\log n)^2\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right).$$

B. F_p Estimation Algorithm

We now give an adversarially robust streaming algorithm for F_p moment estimation with $p \in (0, 2)$ by using Theorem II.2.

Theorem IV.4. *Given $\varepsilon > 0$ and $p \in (0, 2)$, there exists an adversarially robust streaming algorithm that outputs a $(1+\varepsilon)$-approximation for F_p that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2}\log^2 n(\log\log n)^2\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$ bits of space and succeeds with probability at least $\frac{2}{3}$.*

The algorithm can further be optimized for the following guarantees:

Theorem IV.5 (Adversarially robust F_p streaming algorithm for $p \in (0, 2)$)**.** *Given $\varepsilon > 0$ and $p \in (0, 2)$, there exists an adversarially robust streaming algorithm that outputs a $(1 + \varepsilon)$ for F_p that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2}\log n(\log\log n)^2\log\frac{1}{\varepsilon}\left(\log\log n + \log\frac{1}{\varepsilon}\right)\right)$ bits of space and succeeds with probability at least $\frac{2}{3}$.*

Finally, we remark that our analysis for Lemma IV.3 can be repeated to show that Li's geometric mean

(1) Let A be a $d \times n$ random matrix whose entries are i.i.d from the p-stable distribution $\mathcal{D}_p$, for $d = \mathcal{O}\left(\frac{\gamma^{2/p}}{\varepsilon^2}\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$
(2) For a parameter $q = 3$, let each $z_i = \prod_{j=q(i-1)+1}^{qi}(Av + Aw_t)_j^{p/q}$ and $z'_i = \prod_{j=q(i-1)+1}^{qi}(Av)_j^{p/q}$.
(3) Output the arithmetic mean of $(z_1 - z'_1), (z_2 - z'_2), \ldots, (z_{d/q} - z'_{d/q})$.

Fig. 3: Difference estimator for $F_p(v + w_t) - F_p(w_t)$ with $0 < p < 2$

estimator provides strong L_p tracking for $p \in (0, 2)$.

Theorem IV.6. *For $p \in (0, 2)$, there exists a one-pass streaming algorithm that uses total space $\mathcal{O}\left(\frac{1}{\varepsilon^2}\log n(\log\log n)^2\left(\log\log n + \log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right)$ bits and provides (ε, δ)-strong tracking for the F_p moment estimation problem.*

Applications to Entropy Estimation.: Finally, we give an application to adversarially robust entropy estimation, similar to Theorem VII.6.

Theorem IV.7 (Adversarially robust entropy streaming algorithm)**.** *Given $\varepsilon > 0$, there exists an adversarially robust streaming algorithm that outputs an additive ε-approximation to Shannon entropy and uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\log^3 n\right)$ bits of space and succeeds with probability at least $\frac{2}{3}$.*

V. Robust F_p Estimation for Integer $p > 2$

In this section, we use the framework of Section II to give an adversarially robust streaming algorithm for F_p moment estimation where $p > 2$ is an integer. We again require both an F_p strong tracker and an F_p difference estimator to use Theorem II.2. Recall that for integer $p > 2$, the dominant space factor is $n^{1-2/p}$ and thus we will not try to optimize the $\log n$ factors. Hence we obtain an F_p strong tracker by adapting an F_p streaming algorithm and a union bound over m points in the stream. The main challenge of the section is to develop the F_p difference estimator, since we have the following F_p strong tracker:

Theorem V.1 (Oblivious F_p strong tracking for integer $p > 2$)**.** *[Gan11], [GW18] For integer $p > 2$, there exists an insertion-only streaming algorithm $\mathrm{GHSS}(1, t, \varepsilon, \delta)$ that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2}n^{1-2/p}\log\frac{n}{\delta}\log^2 n\right)$ bits of space and provides (ε, δ)-strong F_p tracking.*

We use the expansion $F_p(v + u) - F_p(v) = \sum_{k=1}^{p}\binom{p}{k}\langle v^k, u^{p-k}\rangle$ for our difference estimator for F_p moment estimation for integer $p > 2$, where u^k denotes the coordinate-wise k-th power of u. Suppose we use a perfect L_p sampler to sample a coordinate $a \in [n]$ with probability $\frac{v_a^k}{\|v\|_k^k}$. We then set Z to be the a-th coordinate of the frequency vector v^{p-k}. Observe that u arrives completely after the splitting time denotes the end of the updates to frequency vector v. Thus we can sample a at the splitting time and then explicitly compute Z. We can then obtain an unbiased estimate Y to $\|v\|_k^k$ with low variance, so that the expected value of YZ would be exactly $\langle v^k, u^{p-k}\rangle$ and moreover, YZ is a "good" approximation to $\langle v^k, u^{p-k}\rangle$.

The downfall of this approach is that it requires a perfect L_p-sampler for $p > 2$, which is not known. Perfect L_p-samplers are known for $p \le 2$ [JW18], but their constructions are based on duplicating each stream update $\mathrm{poly}(n)$ times. Hence adapting these constructions to build perfect L_p-samplers for $p > 2$ would be space-inefficient, since the space dependence is $\Omega(n^{1-2/p})$ for $p > 2$, rather than $\mathrm{polylog}(n)$ for $p \le 2$. Thus after duplication the space required would be $\Omega((\mathrm{poly}(n))^{1-2/p})$ rather than $\mathrm{polylog}(\mathrm{poly}(n))$. Note that the former requires space larger than n while the latter remains $\mathrm{polylog}(n)$. One possible approach would be to use approximate L_p samplers and their variants [MW10], [JST11], [AKO11], [MRWZ20], but these algorithms already have $\frac{1}{\varepsilon^2}$ space dependency for each instance, which prohibits using $\Omega\left(\frac{1}{\varepsilon}\right)$ instances to reduce the variance of each sampler. Instead, we use the following perfect L_2-sampler of [JW18] to return a coordinate $a \in [n]$ with probability $\frac{v_a^2}{\|v\|_2^2} \pm \frac{1}{\mathrm{poly}(n)}$.

We also obtain unbiased estimates X and Y of v_a^{k-2} and $\|v\|_2^2$, respectively. Given $a \in [n]$, we then track the a-th coordinate of u^{p-k} exactly. We show that the product of these terms X, Y, and u^{p-k} forms an unbiased estimate to $\langle v^k, u^{p-k}\rangle$. We then analyze the variance and show that taking the mean of enough repetitions gives a $(1 + \varepsilon)$-approximation to $\langle v^k, u^{p-k}\rangle$. By repeating the estimator for each summand in $\sum_{k=1}^{p}\binom{p}{k}\langle v^k, u^{p-k}\rangle$, it follows that we obtain a $(\gamma, \varepsilon, \delta)$-difference estimator for F_p.

We use the well-known CountSketch algorithm for identifying heavy-hitters, which consists of a table with $\log\frac{n}{\delta}$ rows, each consisting of $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$ buckets, to identify $\varepsilon \cdot L_2$ heavy hitters. For each row, each item $i \in [n]$ in the universe is hashed to one of the $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$

buckets along with a random sign. The signed sum of all items assigned to each bucket across all rows is tracked by the data structure and the estimated frequency of each item i is the median of the values associated with each bucket that i is hashed to, across all rows.

Unfortunately, perfect L_2 sampling coordinates of v alone is not enough; the variance of the resulting procedure is too high to obtain space dependency $\frac{\gamma}{\varepsilon^2}$. Thus we also run a subroutine that removes a set of "heavy" coordinates $\mathcal{H}$ of v and tracks the corresponding coordinates of u. Although we have the exact values of u_a for $a \in \mathcal{H}$, we still do not have exact values of v_a; instead, we have estimates $\widehat{v_a}$ for each v_a with $a \in \mathcal{H}$. Setting h to be the sparse vector that contains the estimates $\widehat{v_a}$ for each $a \in \mathcal{H}$ and $w := v - h$, our algorithm perfect L_2 samples from L_2 sample from.

Lemma V.2 (F_p difference estimator for integer $p > 2$)**.** *For integer $p > 2$, there exists a $(\gamma, \varepsilon, \delta)$-difference estimator for F_p that uses space $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2} n^{1-2/p} \log^3 n \log \frac{n}{\delta}\right)$.*

Using our difference estimator, we obtain a robust algorithm for F_p moment estimation for integer $p > 2$.

Theorem V.3 (Adversarially robust F_p streaming algorithm for integer $p > 2$)**.** *Given $\varepsilon > 0$ and integer $p > 2$, there exists an adversarially robust streaming algorithm that outputs a $(1 + \varepsilon)$-approximation for F_p that succeeds with probability at least $\frac{2}{3}$ and uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} n^{1-2/p} \log^5 n \log^3 \frac{1}{\varepsilon}\right)$ bits of space.*

VI. Robust F_0 Estimation

In this section, we use the framework of Section II to give an adversarially robust streaming algorithm for the distinct elements problem or equivalently, the F_0 moment estimation.

Theorem VI.1 (Oblivious F_0 strong tracking)**.** *[Bla20] There exists an insertion-only streaming algorithm* F0Estimate$(1, t, \varepsilon, \delta)$ *that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta} + \log n\right)$ bits of space and provides (ε, δ)-strong F_0 tracking.*

Theorem VI.1 and other F_0 approximation algorithms use a balls-and-bins argument with increasing levels of sophistication [BJK+02], [KNW10], [Bla20]. We use a similar balls-and-bins argument, where each item is subsampled at a level k with probability $\frac{1}{2^k}$, to obtain an F_0 difference estimator. By counting the number of items in a level with $\Theta\left(\frac{\gamma}{\varepsilon^2}\right)$ items that survive the subsampling process for the stream u, it follows that the expected number of the distinct items in v but not u is $\Theta\left(\frac{\gamma^2}{\varepsilon^2}\right)$. Thus we obtain a $\left(1 + \frac{\varepsilon}{\gamma}\right)$-approximation to $F_0(v) - F_0(u)$ from this procedure by first running the balls-and-bins experiment on u and counting the number of bins that are occupied at some level k with $\Theta\left(\frac{\gamma}{\varepsilon^2}\right)$ survivors. We then run the same balls-and-bins experiment on $v - u$ by only counting the additional bins that are occupied at level k, and rescaling this number by 2^k. Note that additional bins only correspond to items in v but not u, which is exactly $F_0(v) - F_0(u)$. Although level k does not necessarily give a $(1 + \varepsilon)$-approximation to $F_0(v) - F_0(u)$, it does give a $\left(1 + \frac{\varepsilon}{\gamma}\right)$-approximation to $F_0(v) - F_0(u)$, which translates to an additive $\varepsilon \cdot F_0(u)$ approximation to $F_0(v) - F_0(u)$ and is exactly the requirement for the F_0 difference estimator, since $F_0(v) - F_0(u) \le \gamma F(u)$.

Lemma VI.2 (F_0 difference estimator)**.** *There exists a $(\gamma, \varepsilon, \delta)$-difference estimator for F_0 that uses $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2}\left(\log \frac{1}{\varepsilon} + \log \log n + \log \frac{1}{\delta}\right) + \log n\right)$ bits of space.*

Note that the difference estimator in Lemma VI.2 only requires pairwise independence and thus can be derandomized using a hash function that can be stored using $\mathcal{O}(\log n)$ bits of space. We now use our difference estimator to get an adversarially robust streaming algorithm for the distinct elements problem.

Theorem VI.3. *Given $\varepsilon > 0$, there exists an adversarially robust streaming algorithm that outputs a $(1+\varepsilon)$-approximation to F_0 that succeeds with probability at least $\frac{2}{3}$ and uses space*

$$\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n \log^3 \frac{1}{\varepsilon} \cdot \left(\log \frac{1}{\varepsilon} + \log \log n\right) + \frac{1}{\varepsilon^2} \log^2 n\right).$$

Optimized F_0 Algorithm.: To improve the space requirements, we can again observe that it suffices to maintain sketches $\mathcal{A}_a$ and $\mathcal{B}_{a,c}$ for $\mathcal{O}\left(\log \frac{1}{\varepsilon}\right)$ values of a at a time, instead of maintaining all sketches $\mathcal{A}_a$ and $\mathcal{B}_{a,c}$ simultaneously. By the same argument as before, it suffices to maintain only the most sketches $\mathcal{A}_i$ and $\mathcal{B}_{i,c}$ for only the smallest $\mathcal{O}\left(\log \frac{1}{\varepsilon}\right)$ values of i that are at least a since the output increases by a factor of 2 each time a increases and thus any larger index will have only missed $\mathcal{O}(\varepsilon)$ fraction of the F_0 of the stream. Hence, any larger index still outputs a $(1 + \varepsilon)$-approximation once it becomes initialized.

Theorem VI.4. *Given $\varepsilon > 0$, there exists an adversarially robust streaming algorithm that outputs a $(1+\varepsilon)$-approximation for F_0 that succeeds with probability at least $\frac{2}{3}$ and uses total space*

$$\mathcal{O}\left(\frac{1}{\varepsilon^2} \log^4 \frac{1}{\varepsilon}\left(\log \log n + \log \frac{1}{\varepsilon}\right) + \frac{1}{\varepsilon} \log n \log \frac{1}{\varepsilon}\right).$$

(1) Find a list $\mathcal{H}$ that includes all $i \in [n]$ with $v_i \geq \frac{\gamma^{1/p}}{16}\|v\|_p$.
(2) Using CountSketch, obtain an estimate $\widehat{v_i}$ to v_i with additive error $\frac{\varepsilon\gamma^{1/p}}{64\gamma}\|v\|_p$ for each $i \in \mathcal{H}$ and let $h \in \mathbb{R}^n$ be the vector such that $h_i = \widehat{v_i}$ if $i \in \mathcal{H}$ and zero otherwise.
(3) Perform perfect L_2 sampling on $v - h$ to obtain a set $\mathcal{S}$ of size $k = \mathcal{O}\left(\frac{\gamma}{\varepsilon^2}n^{1-2/p}\right)$.
(4) Obtain an estimate $\widehat{s_i}$ to $v_i - h_i$ for each $i \in \mathcal{S}$.
(5) Let W be a $(1+\varepsilon)$-approximation to $\|v-h\|_2^2$.
(6) Output $W + \sum_{k=1}^{p-1}\binom{p}{k}\left(\sum_{i\in\mathcal{H}}\widehat{v_i}^k, u_i^{p-k} + W \cdot \sum_{i\in\mathcal{S}}\widehat{s_i}^{k-2}, u_i^{p-k}\right)$.

Fig. 4: F_p difference estimator for $F_p(v+u) - F_p(v)$ with integer $p > 2$.

(1) Use a set of exponential random variables to form a vector V of duplicated and scaled coordinates of v.
(2) Hash the coordinates of V into a CountSketch data structure with $\mathcal{O}(\log n)$ buckets.
(3) Use the same exponential random variables to perform perfect L_2 sampling on u to obtain a coordinate (i,j) and an unbiased estimate $\widehat{u_{i,j}}$ to $u_{i,j}$.
(4) Let $\widehat{U}$ be an unbiased estimate of $\|u\|_2^2$ with second moment $\mathcal{O}\left(\|u\|_2^4\right)$.
(5) Query CountSketch for an unbiased estimate $\widehat{v_{i,j}}$ to $v_{i,j}$ and set an estimator as $\widehat{U} \cdot \widehat{v_{i,j}}\left(\widehat{u_{i,j}}\right)^{p-3}$.
(6) Output the mean of $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2} \cdot n^{1-2/p}\right)$ such estimators.

Fig. 5: F_p difference estimator for $\langle v, u^{p-1}\rangle$ with integer $p > 2$.

VII. Framework for Sliding Window Algorithms

In this section, we describe a general framework for norm estimation in the sliding window model, using the sketch stitching and granularity changing technique. We first require the following background on sliding windows algorithms.

Smooth Histograms.: Braverman and Ostrovsky introduced the *smooth histogram*, an elegant framework that solves a large number of problems in the sliding window model. The smooth histogram data structure maintains a number of timestamps throughout the data stream, along with a streaming algorithm for each timestamp that stores a sketch of all the elements seen from the timestamp. The timestamps maintain the invariant that at most three checkpoints produce values that are within $(1-\beta)$ of each other, since any two of the sketches would always output values that are within $(1-\alpha)$ afterwards. Hence if the function is polynomially bounded, then the smooth histogram data structure only needs a logarithmic number of timestamps.

Definition VII.1 (Suffix-Pivoted Difference Estimator)**.** *Given a stream $\mathcal{S}$ and fixed times t_1, t_2, and t_3, let frequency vectors u and v be induced by the updates of $\mathcal{S}$ between times $[t_1, t_2)$ and $[t_2, t_3)$. Given an accuracy parameter $\varepsilon > 0$ and a failure probability $\delta \in (0,1)$, a streaming algorithm $\mathcal{C}(t_1, t_2, t, \gamma, \varepsilon, \delta)$ is a $(\gamma, \varepsilon, \delta)$-*suffix difference estimator *for a function F if, with probability at least $1-\delta$, it outputs an additive $\varepsilon \cdot F(v+w_t)$ approximation to $F(u+v+w_t) - F(v+w_t)$ for all frequency vectors w_t induced by $[t_3, t)$ for times $t > t_3$, given $\min(F(u), F(u+v) - F(v)) \leq \gamma \cdot F(v)$ for a ratio parameter $\gamma \in (0,1]$.*

Observe that the difference between Definition II.1 and Definition VII.1 is that the fixed-prefix difference estimator approximates $F(v+w_t) - F(v)$ when the contribution to F of the first frequency vector v that arrives in the stream is much larger than that of w_t, while the suffix-pivoted difference estimator approximates $F(v+w_t) - F(w_t)$ when $F(w_t)$ is larger than $F(v)$.

We adapt our sketch stitching and granularity changing technique to the sliding window model by focusing on the suffix of the stream, since prefixes of the stream may expire. We thus run the highest accuracy algorithms, the separate streaming algorithms $\mathcal{A}$, on various suffixes of the stream similar to the smooth histogram framework. It follows from smoothness that we maintain an instance of $\mathcal{A}$ starting at some time $t_0 \leq m - W + 1$, whose output is within a factor 2 of the value of F on the sliding window. Our task is then to remove the extraneous contribution of the updates between times t_0 and $m-W+1$, i.e., the starting time of the sliding window. We partition the substream of these updates into separate blocks based on their contribution to the value of F by guessing $\mathcal{O}(\log n)$ values for the final value of F on the sliding window and forming new difference estimators at level j when the value of F on each block has exceeded a $\frac{1}{2^j}$ fraction of the corresponding guess. We terminate a guess when there are more than $100 \cdot 2^j$ blocks in that level, indicating that the guess is too low. We can maintain separate sketches for these blocks, with varying granularities, and stitch these sketches together at the end. We give our algorithm in full in Algorithm 3.

Interpretation of Algorithm 3.: We now translate between the previous intuition and the pseudocode of Algorithm 3. At each time, Algorithm 3 only runs two subroutines: GuessAndUpdate and MergeSW. The first subroutine GuessAndUpdate creates new instances of each algorithm (both a streaming algorithm approximating F on a suffix of the stream and a difference estimator starting at each time) at each time in the stream. Moreover, GuessAndUpdate partitions the stream into blocks for the difference estimator by using exponentially increasing guesses for the value of F at the end of the stream to assist with appropriate granularity for each block. The second subroutine MergeSW performs maintenance on the data structure to ensure that there are not too many instances that have been created by the first subroutine that are simultaneously running. Namely, MergeSW deletes algorithms running on suffixes of the stream that output a similar value, so that the number of remaining algorithms is logarithmic rather than linear. Similarly, MergeSW merges two difference estimators when it is clear their combined contribution is still too small. Finally at the end of the stream, StitchSW creates an estimate for the value of F on the stream by stitching together the estimates output by each difference estimator. Although Algorithm 1 is notationally heavy, each timestamp $t^{(k)}_{i,j,\ell}$ should be associated with (1) a guess $k \in [C \log n]$ for the value of F at the end of the stream, (2) an index $i \in \mathcal{O}(\log n)$ roughly associated with the number of times F has *actually* doubled in the stream so far, (3) a granularity j, and (4) the number of the block ℓ in granularity j.

The main intuition behind Lemma VII.2 is that there are two sources of error. The first source of error originates from the boundaries of the blocks corresponding to the difference estimator not aligning with the beginning of the sliding window. This error, resulting from no difference estimator being assigned to compute the exactly correct value, cannot be accounted for even if all difference estimators have zero error. On the other hand, this error is upper bounded by the contribution of a difference estimator at the bottom level. The second source of error stems from the approximation error caused by each of the difference estimators. Since we can bound the total number of difference estimators being used in our output, we can also upper bound the total approximation error due to the difference estimators.

Algorithm 3 Moment Estimation in the Sliding Window Model

Input: Stream $u_1, \ldots, u_m$ of updates, an $(\varepsilon, \varepsilon^q)$-smooth function F, accuracy parameter $\varepsilon \in (0,1)$, window parameter $W > 0$

Output: Robust $(1+\varepsilon)$-approximation to F

- $\delta \leftarrow \frac{1}{\mathrm{poly}(m)}$, $\eta \leftarrow \frac{\varepsilon}{2^{20} q \log \frac{1}{\varepsilon}}$, $\varphi \leftarrow \sqrt{2}$ be a parameter
- $\beta \leftarrow \left\lceil \log \frac{100 \cdot 4^q}{\varepsilon^q} \right\rceil$, $\gamma_j \leftarrow 2^{3-j}$ for all $j \in [\beta]$
- **for** each update $u_t \in [n]$, $t \in [m]$ **do**
  - GuessAndUpdate ▷Create new subroutines for each update
  - MergeSW ▷Removes extraneous subroutines
- **return** $Z \leftarrow$ StitchSW ▷Estimate F on sliding window

Algorithm 4 Subroutine GuessAndUpdate of Algorithm 3: create new subroutines for each update

- Let s be the number of instances of $\mathcal{A}$.
- $t_{s+1} \leftarrow t$
- Start a new instance $\mathcal{A}(t_{s+1}, t, \eta, \delta)$.
- **for** $j \in [\beta]$ **do** ▷Maintain instances of each granularity
  - **for** $k \in [C \log n]$ **do** ▷n^C is upper bound on value of F
    - Let r_k be the number of instances of timestamps $t^{(k)}_{s+1,j,*}$.
    - **if** $\mathcal{A}(t^{(k)}_{s+1,j,r_k}, t-1, 1, \delta) \in [n^C/2^{j+k+11}, n^C/2^{j+k+10}]$, $\mathcal{A}(t^{(k)}_{s+1,j,r_k}, t, 1, \delta) > n^C/2^{j+k+10}$, and $r < 100 \cdot 2^{j+10}$ **then**
      - **for** $\ell > k$ **do**
        - $t^{(\ell)}_{s+1,j,r_{\ell+1}} \leftarrow t$.
        - Demarcate SDiffEst$(t^{(\ell)}_{s+1,j,r_\ell}, t^{(\ell)}_{s+1,j,r_{\ell+1}}, t, \gamma_j, \eta, \delta)$. ▷Update splitting time
        - Start a new instance $\mathcal{A}(t^{(\ell)}_{s+1,j,r_{\ell+1}}, t, 1, \delta)$.

Lemma VII.2 (Correctness of framework)**.** *With high probability, Algorithm 3 gives a $(1+\varepsilon)$-approximation to the value of $F(W)$.*

Algorithm 5 Subroutine MERGESW of Algorithm 3: removes extraneous subroutines

Let s be the number of instances of $\mathcal{A}$.
for $i \in [s]$, $j \in [\beta]$, and $k \in [C \log n]$ **do** ▷Difference estimator maintenance
  Let r_k be the number of instances of timestamps $t^{(k)}_{s+1,j,*}$.
  for $\ell \in [r_k - 1]$ **do** ▷Merges two algorithms with "small" contributions
    if $\mathcal{A}(t^{(k)}_{i,j,\ell-1}, t^{(k)}_{i,j,\ell+1}, 1, \delta) \le n^C/2^{k+j+10}$ **then**
      Merge (add) the sketches for $\mathcal{A}(t_{i,j,k-1}, t_{i,j,k}, 1, \delta)$ and $\mathcal{A}(t_{i,j,k}, t_{i,j,k+1}, 1, \delta)$.
      Merge (add) the sketches for $\text{SDIFFEST}(t_{i,j,k-1}, t_{i,j,k}, t, \gamma_j, \eta, \delta)$ and $\text{SDIFFEST}(t_{i,j,k}, t_{i,j,k+1}, t, \gamma_j, \eta, \delta)$.
      Relabel the times $t_{i,j,*}$.
  for $i \in [s-2]$ **do** ▷Smooth histogram maintenance
    if $\mathcal{A}(t_{i+2}, t, \eta, \delta) \ge (1 - 1/8^q)\mathcal{A}(t_i, t, \eta, \delta)$ **then**
      for $j \in [\beta]$ and $k \in [C \log n]$ **do**
        Append the times $t^{(k)}_{i+1,j,*}$ to $\{t^{(k)}_{i,j,*}\}$.
      Delete t_{i+1} and all times $t_{i+1,*,*}$.
      Relabel the times $\{t_i\}$ and $\{t^{(k)}_{i,j,*}\}$.

Algorithm 6 Subroutine STITCHSW of Algorithm 3: output estimate of F_p on the sliding window

Let i be the largest index such that $t_i \le m - W + 1$.
Let k be the smallest integer such that $n^C/2^k \le \mathcal{A}(t_i, m, 1, \delta)$.
$c_0 \leftarrow t_i$
$X \leftarrow \mathcal{A}(t_i, m, \eta, \delta)$
for $j \in [\beta]$ **do** ▷Stitch sketches
  Let a be the smallest index such that $t^{(k)}_{i,j,a} \ge c_{j-1}$
  Let b be the largest index such that $t^{(k)}_{i,j,b} \le m - W + 1$
  $c_j \leftarrow t^{(k)}_{i,j,b}$
  $Y_j \leftarrow \sum_{k=a}^{b-1} \text{SDIFFEST}(t^{(k)}_{i,j,k}, t^{(k)}_{i,j,k+1}, m, \gamma_j, \eta, \delta)$
return $Z := X - \sum_{j=1}^{\beta} Y_j$

Theorem VII.3 (Framework for sliding window algorithms). *Let $\varepsilon, \delta \in (0,1)$ be constants and F be a monotonic and polynomially bounded function that is $(\varepsilon, \varepsilon^q)$-smooth for some constant $q \ge 0$. Suppose there exists a $(\gamma, \varepsilon, \delta)$-suffix pivoted difference estimator that uses space $\frac{\gamma}{\varepsilon^2} S_F(m, \delta, \varepsilon)$ and a streaming algorithm for F that uses space $\frac{1}{\varepsilon^2} S_F(m, \delta, \varepsilon)$, where S_F is a monotonic function in m, $\frac{1}{\delta}$, and $\frac{1}{\varepsilon}$. Then there exists a sliding window algorithm that outputs a $(1+\varepsilon)$ approximation to F that succeeds with constant probability and uses $\frac{1}{\varepsilon^2} \cdot S_F(m, \delta', \varepsilon) \cdot \text{poly}\left(\log m, \log \frac{1}{\varepsilon}\right)$ space, where $\delta' = \mathcal{O}\left(\frac{1}{\text{poly}(m)}\right)$.*

A. Moment Estimation for $p \in (0, 2]$

To improve Algorithm 3 for F_p moment estimation with $p \in (0, 2]$, we remove the additional $\mathcal{O}(\log n)$ overhead associated with making $\mathcal{O}(\log n)$ guesses for the value of F at the end of the stream. Instead, we note that since we maintain a constant factor approximation to the value of F_p due to the smooth histogram, it suffices to partition the substream into blocks for the difference estimators based on the ratio of the difference to the constant factor approximation to the value of F_p. We also recall the following useful characterization of the smoothness of the F_p moment function.

Lemma VII.4. *[BO07] The F_p function is $\left(\varepsilon, \frac{\varepsilon^p}{p}\right)$-smooth for $p \ge 1$ and $(\varepsilon, \varepsilon)$-smooth for $0 < p \le 1$.*

Moreover, we require constructions of suffix-pivoted difference estimators for F_p moment estimation with $p \in (0, 2]$. However, we claim that the previous constructions, i.e., the fixed-prefix difference estimator based on Li's geometric estimator for $p \in (0, 2)$ and the fixed-prefix difference estimator based on the inner product sketch for $p = 2$ are already valid suffix-pivoted difference estimators. This is because the key property to approximating the difference $F_p(v + u_1) - F_p(u_1)$ in "small" space is that $F_p(v)$ is small. We can express the variance of both Li's geometric estimator and the inner product sketch in terms of $F_p(v)$ so that smaller values of $F_p(v)$ correspond to smaller variance for the estimators. Hence even if $F_p(v + u_2) - F_p(u_2)$ is much larger than $F_p(v + u_1) - F_p(u_1)$ for some $u_2 \succeq u_1$, as long as $F_p(v + u_1) - F_p(u_1) \le \gamma \cdot F_p(v + u_1)$, then $F_p(v) \le \gamma \cdot F_p(v + u_1) \le F_p(v + u_2)$. Therefore, the variance for our difference estimators is at most $\gamma F_p(v + u_2)^2$, so we only need to run $\mathcal{O}\left(\frac{\gamma}{\varepsilon^2}\right)$ independent copies of the difference estimators.

Theorem VII.5. *Given $\varepsilon > 0$ and $p \in (0, 2]$, there exists a one-pass algorithm in the sliding window model that outputs a $(1 + \varepsilon)$-approximation to the L_p norm with probability at least $\frac{2}{3}$. The algorithm uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log^3 n \log^3 \frac{1}{\varepsilon}\right)$ bits of space for $p = 2$. For $p \in (0, 2)$, the algorithm uses space $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log^3 n (\log \log n)^2 \log^3 \frac{1}{\varepsilon}\right)$.*

(1) Find a list $\mathcal{H}$ that includes all $a \in [n]$ with $u_a \geq \frac{\varepsilon}{100p^{p+1}\log^2 n}\|u\|_p$.
(2) Using CountSketch, obtain an estimate $\widehat{u_a}$ to u_a with additive error $\frac{\varepsilon}{100p^{p+1}\log^2 n}\|u\|_p$ for each $a \in \mathcal{H}$ and let $h \in \mathbb{R}^n$ be the vector such that $h_a = \widehat{u_a}$ if $a \in \mathcal{H}$ and zero otherwise.
(3) Perform perfect L_2 sampling on $w := u - h$ to obtain a set $\mathcal{S}$ of size $k = \mathcal{O}\left(\frac{\gamma}{\varepsilon^2}n^{1-2/p}\right)$.
(4) Obtain an estimate $\widehat{w_a}$ to w_a for each $a \in \mathcal{S}$.
(5) Let $\widehat{W}$ be a $(1+\varepsilon)$-approximation to $\|w\|_2^2$.
(6) Output $W + \sum_{k=1}^{p-1}\binom{p}{k}\left(\sum_{a\in\mathcal{H}}\widehat{v_a}^k, u_a^{p-k} + W \cdot \sum_{a\in\mathcal{S}}\widehat{w_a}^{k-2}, u_a^{p-k}\right)$.

(1) Use a set of exponential random variables to form a vector W of duplicated and scaled coordinates of w.
(2) Hash the coordinates of W into a CountSketch data structure with $\mathcal{O}(\log n)$ buckets.
(3) Use the same exponential random variables to perform perfect L_2 sampling on v to obtain a coordinate (i,j) and an unbiased estimate $\widehat{v_{i,j}}$ to $v_{i,j}$.
(4) Let $\widehat{V}$ be an unbiased estimate of $\|v\|_2^2$ with second moment $\mathcal{O}\left(\|v\|_2^4\right)$.
(5) Query CountSketch for an unbiased estimate $\widehat{w_{i,j}}$ to $w_{i,j}$ and set an estimator as $\widehat{V} \cdot \widehat{w_{i,j}}\left(\widehat{v_{i,j}}\right)^{p-3}$.
(6) Output the mean of $\tilde{\mathcal{O}}\left(\frac{\gamma}{\varepsilon^2}\cdot n^{1-2/p}\right)$ such estimators.

Using connections between Shannon entropy and L_p estimation for $p \in (0,2]$, we obtain the following corollary:

Theorem VII.6. *Given $\varepsilon > 0$, there exists a sliding window algorithm that outputs an additive ε-approximation to Shannon entropy and uses $\tilde{\mathcal{O}}\left(\frac{\log^5 n}{\varepsilon^2}\right)$ bits of space and succeeds with probability at least $\frac{2}{3}$.*

B. Moment Estimation for Integer $p > 2$

In this section, we describe our algorithm to estimate the F_p moment in the sliding window model, for integer $p > 2$. Suppose that we have vectors u and v such that the vector u arrives before the vector v and $F_p(u) \leq \gamma F_p(v) \leq 2^p F_p(u)$, for some $\gamma \leq 1$. A crucial subroutine in Algorithm 3 is the MergeSW subroutine, which controls the number of blocks in which the substream is partitioned into, at each granularity, and thus gives efficient bounds on the space of the algorithm. We will again create $\mathcal{O}(\log n)$ parallel instances as in Algorithm 5, corresponding to exponentially increasing guesses 2^i for the value of $F_p(v)$ and incur the additional $\mathcal{O}(\log n)$ overhead in space. We will then partition blocks based on their F_p values in comparison to the guess of $F_p(v)$. Now if each block u is required to satisfy $F_p(u) \leq \gamma F_p(v) \leq 2^p F_p(u)$ for some $\gamma \approx \frac{1}{2^j}$, then we can assume our guess for $F_p(v)$ is incorrect if the partitioning creates more than 2^j blocks. Hence, we can again assume that each block u satisfies $F_p(u) \leq \gamma F_p(v) \leq 2^p F_p(u)$, at the cost of an additional $\mathcal{O}(\log n)$ overhead in space.

As in Section VII-A, it remains to find a difference estimator for $F_p(u+v) - F_p(v)$, i.e., an algorithm with space dependency $\tilde{\mathcal{O}}\left(\frac{\gamma}{\varepsilon^2}n^{1-2/p}\right)$. Observe that $F_p(u+v) - F_p(v) = \sum_{k=1}^{p}\binom{p}{k}\langle u^k, v^{p-k}\rangle$. We estimate each term $\langle u^k, v^{p-k}\rangle$ separately.

We first use a heavy-hitter algorithm HHEst to simultaneously find all heavy-hitters $a \in [n]$ such that $u_a \geq \varepsilon\gamma^{1/p}L_p(v)$ across all vectors u induced by the blocks. These coordinates form a set $\mathcal{H}$ and we read off the corresponding coordinates of v to estimate $\sum_{a\in\mathcal{H}}\langle u^k, v^{p-k}\rangle$ for each k. For $a \notin \mathcal{H}$, we analyze separate algorithms for estimating $\sum_{a\notin\mathcal{H}}\langle u, v^{p-1}\rangle$ and $\sum_{a\notin\mathcal{H}}\langle u^k, v^{p-k}\rangle$ for $k \geq 2$, though the analysis is similar for both algorithms.

We first run a heavy-hitters algorithm to find all indices $a \in [n]$ such that

$$u_a \geq \frac{\varepsilon}{100p^{p+1}\log^2 n}\gamma^{1/p}L_p(v) \geq \frac{\varepsilon}{100p^{p+1}\log^2 n}L_p(u),$$

which takes $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}n^{1-2/p}\right)$ space, since $F_p(u) \leq \gamma F_p(v)$. Thus we bound the variance for a significant level set $L_{i,j}$ with $2^i \geq \frac{\varepsilon}{100p^{p+1}\log^2 n}$.

Theorem VII.7. *Given $\varepsilon > 0$ and integer $p > 2$, there exists a one-pass algorithm in the sliding window model that outputs a $(1+\varepsilon)$-approximation to the L_p norm with probability at least $\frac{2}{3}$ and uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}n^{1-2/p}\right)$ bits of space.*

References

[AKO11] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms via precision sampling. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, pages 363–372. IEEE Computer Society, 2011. 7

[AMYZ19] Dmitrii Avdiukhin, Slobodan Mitrovic, Grigory Yaroslavtsev, and Samson Zhou. Adversarially robust submodular maximization under knapsack constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, pages 148–156, 2019. 1

[BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 1–16, 2002. 1

[BCI+17] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. Bptree: An ℓ_2 heavy hitters algorithm using constant memory. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 361–376, 2017. 2

[BCIW16] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, and David P. Woodruff. Beating countsketch for heavy hitters in insertion streams. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 740–753, 2016. 2

[BDN17] Jaroslaw Blasiok, Jian Ding, and Jelani Nelson. Continuous monitoring of ℓ_p norms in data streams. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 32:1–32:13, 2017. 2, 5, 6

[BGL+18] Vladimir Braverman, Elena Grigorescu, Harry Lang, David P. Woodruff, and Samson Zhou. Nearly optimal distinct elements and heavy hitters on sliding windows. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 7:1–7:22, 2018. 3

[BJK+02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Randomization and Approximation Techniques, 6th International Workshop, RANDOM, Proceedings*, pages 1–10, 2002. 8

[BJWY20] Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 63–80, 2020. 1, 3

[Bla20] Jaroslaw Blasiok. Optimal streaming and tracking distinct elements with high probability. *ACM Trans. Algorithms*, 16(1):3:1–3:28, 2020. 2, 8

[BLV19] Elette Boyle, Rio LaVigne, and Vinod Vaikuntanathan. Adversarially robust property-preserving hash functions. In *10th Innovations in Theoretical Computer Science Conference, ITCS*, pages 16:1–16:20, 2019. 1

[BMSC17] Ilija Bogunovic, Slobodan Mitrovic, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70, pages 508–516, 2017. 1

[BO07] Vladimir Braverman and Rafail Ostrovsky. Smooth histograms for sliding windows. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Proceedings*, pages 283–293, 2007. 2, 4, 11

[BY20] Omri Ben-Eliezer and Eylon Yogev. The adversarial robustness of sampling. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 49–62, 2020. 1

[DGIM02] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining stream statistics over sliding windows. *SIAM J. Comput.*, 31(6):1794–1813, 2002. 1

[Gan11] Sumit Ganguly. Polynomial estimators for high frequency moments. *CoRR*, abs/1104.4552, 2011. 7

[GW18] Sumit Ganguly and David P. Woodruff. High probability frequency moment sketches. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 107, pages 58:1–58:15, 2018. 7

[HKM+20] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. *CoRR*, abs/2004.05975, 2020. 1, 3

[HNO08] Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 489–498, 2008. 3

[HU14] Moritz Hardt and Jonathan R. Ullman. Preventing false discovery in interactive data analysis is hard. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 454–463, 2014. 1

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

[JST11] Hossein Jowhari, Mert Sauglam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 49–58, 2011. 7

[JW18] Rajesh Jayaram and David P. Woodruff. Perfect l_p sampling in a data stream. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 544–555. IEEE Computer Society, 2018. 7

[KMGG08] Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008. 1

[KNW10] Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS*, pages 41–52, 2010. 8

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

[MBN+17] Slobodan Mitrovic, Ilija Bogunovic, Ashkan Norouzi-Fard, Jakub Tarnawski, and Volkan Cevher. Streaming robust submodular maximization: A partitioned thresholding approach. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 4557–4566, 2017. 1

[MM12] Gurmeet Singh Manku and Rajeev Motwani. Approximate frequency counts over data streams. *PVLDB*, 5(12):1699, 2012. 1

[MNS11] Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. *SIAM J. Comput.*, 40(6):1845–1870, 2011. 1

[MRWZ20] Sepideh Mahabadi, Ilya P. Razenshteyn, David P. Woodruff, and Samson Zhou. Non-adaptive adaptive sampling on turnstile streams. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1251–1264, 2020. 7

[MW10] Morteza Monemizadeh and David P. Woodruff. 1-pass relative-error l_p-sampling with applications. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1143–1160, 2010. 7

[Nol03] John Nolan. *Stable distributions: models for heavy-tailed data*. Birkhauser New York, 2003. 6

[NY19] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. *ACM Trans. Algorithms*, 15(3):35:1–35:30, 2019. 1

[OSU18] James B. Orlin, Andreas S. Schulz, and Rajan Udwani. Robust monotone submodular function maximization. *Math. Program.*, 172(1-2):505–537, 2018. 1

[PGD15] Odysseas Papapetrou, Minos N. Garofalakis, and Antonios Deligiannakis. Sketching distributed sliding-window data streams. *VLDB J.*, 24(3):345–368, 2015. 1

[WLL+16] Zhewei Wei, Xuancheng Liu, Feifei Li, Shuo Shang, Xiaoyong Du, and Ji-Rong Wen. Matrix sketching over sliding windows. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference*, pages 1465–1480, 2016. 1

[Zol89] Vladimir M. Zolotarev. One-dimensional stable distributions. *Bull. Amer. Math. Soc*, 20:270–277, 1989. 6