Adversarially Robust Dense-Sparse Tradeoffs via Heavy-Hitters

David P. Woodruff
Department of Computer Science
Carnegie Mellon University
dwoodruf@andrew.cmu.edu

Samson Zhou
Department of Computer Science
Texas A&M University
samsonzhou@gmail.com

Abstract

In the adversarial streaming model, the input is a sequence of adaptive updates that defines an underlying dataset and the goal is to approximate, collect, or compute some statistic while using space sublinear in the size of the dataset. In 2022, Ben-Eliezer, Eden, and Onak showed a dense-sparse trade-off technique that elegantly combined sparse recovery with known techniques using differential privacy and sketch switching to achieve adversarially robust algorithms for L_p estimation and other algorithms on turnstile streams. However, there has been no progress since, either in terms of achievability or impossibility. In this work, we first give improved algorithms for adversarially robust L_p-heavy hitters, utilizing deterministic turnstile heavy-hitter algorithms with better tradeoffs. We then utilize our heavy-hitter algorithm to reduce the problem to estimating the frequency moment of the tail vector. We give a new algorithm for this problem in the classical streaming setting, which achieves additive error and uses space independent in the size of the tail. We then leverage these ingredients to give an improved algorithm for adversarially robust L_p estimation on turnstile streams. We believe that our results serve as an important conceptual message, demonstrating that there is no inherent barrier at the previous state-of-the-art.

1 Introduction

Adversarial robustness for big data models is increasingly important not only for ensuring the reliability and security of algorithmic design against malicious inputs and manipulations, but also to retain guarantees for honest inputs that are nonetheless co-dependent with previous outputs of the algorithm. One such big data model is the streaming model of computation, which has emerged as a central paradigm for studying statistics of datasets that are too large to store. Common examples of datasets that are well-represented by data streams include database logs generated from e-commerce transactions, Internet of Things sensors, scientific observations, social network traffic, or stock markets. To capture these applications, the one-pass streaming model defines an underlying dataset that evolves over time through a number of sequential updates that are discarded irrevocably after processing, and the goal is to compute or approximate some fixed function of the dataset while using space sublinear in both the length m of the data stream and the dimension n of the dataset.

The streaming model of computation. In the classical *oblivious* streaming model, the stream S of updates $u_1, \ldots, u_m$ defines a dataset that is fixed in advance, though the ordering of the sequence of updates may be adversarial. In other words, the dataset is oblivious to any algorithmic design choices, such as instantiations of internal random variables. This is vital for many streaming algorithms, which crucially leverage randomness to achieve meaningful guarantees in sublinear space. For example, the celebrated AMS sketch [AMS99] initializes a random sign vector s and outputs $|\langle s, f\rangle|$ as the estimate for the L_2 norm of the underlying frequency vector f defined by the stream. To show

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0348

correctness of the sketch, we require s to be chosen uniformly at random, independent of the value of f. Similar assumptions are standard across many fundamental sublinear algorithms for machine learning, such as linear regression, low-rank approximation, or column subset selection.

Unfortunately, such an assumption is unreasonable [MNS11, GHS+12, BMSC17, NY19, CN20], as an honest user may need to repeatedly interact with an algorithm, choosing their future actions based on responses to previous questions. For example, in recommendation systems, it is advisable to produce suggestions so that when a user later decides to dismiss some of the items previously recommended by the algorithm, a new high-quality list of suggestions can be quickly computed without solving the entire problem from scratch [KMGG07, MBN+17, KZK18, OSU18, AMYZ19]. Another example is in stochastic gradient descent or linear programming, where each time step can update the eventual output by an amount based on a previous query. For tasks such as linear regression, actions as simple as sorting a dataset have been shown to cause popular machine learning libraries to fail [BHM+21].

Adversarially robust streaming model. In the adversarial streaming model [BY20, HKM+20, ABD+21, BHM+21, KMNS21, WZ21b, BKM+22, BEO22, BJWY22, CGS22, ACGS23, ACSS23, DSWZ23, GLW+24], a sequence of adaptively chosen updates $u_1, \ldots, u_m$ is given as an input data stream to an algorithm. The adversary may choose to generate future updates based on previous outputs of the algorithm, while the goal of the algorithm is to correctly approximate or compute a fixed function at all times in the stream. Formally, the *black-box* adversarial streaming model can be modeled as a two-player game between a streaming algorithm $\mathcal{A}$ and a source $\mathfrak{E}$ that creates a stream of adaptive and possibly adversarial inputs to $\mathcal{A}$. Prior to the game, a fixed statistic $\mathcal{Q}$ is determined, so that the goal of the algorithm is to approximate $\mathcal{Q}$ on the sequence of inputs seen at each time. The game then proceeds across m rounds. In the t-th round:

(1) $\mathfrak{E}$ computes an update u_t for the stream, which possibly depends on all previous outputs from $\mathcal{A}$.

(2) $\mathcal{A}$ uses u_t to update its data structures $\mathcal{D}_t$, acquires a fresh batch R_t of random bits, and outputs a response Z_t to the query $\mathcal{Q}$.

(3) $\mathfrak{E}$ observes and records the response Z_t.

The goal of $\mathfrak{E}$ is to induce from $\mathcal{A}$ an incorrect response Z_t to the query $\mathcal{Q}$ at some time $t \in [m]$ throughout the stream using its control over the sequence $u_1, \ldots, u_m$. By the nature of the game, only a single pass over the stream is permitted. In the context of our paper, each update u_t has the form (i_t, Δ_t), where $i_t \in [n]$ and $\Delta_t \in \{\pm 1\}$. The updates implicitly define a frequency vector $f \in \mathbb{R}^n$, so that u_t changes the value of the (i_t)-th coordinate of f by Δ_t.

Turnstile streams and flip number. In the turnstile model of streaming, updates are allowed to either increase or decrease the weight of elements in the underlying dataset, as compared to insertion-only streams, where updates are only allowed to increase the weight. Whereas various techniques are known for the adversarial robustness on insertion-only streams, significantly less is known for turnstile streams. While near-optimal adversarially robust streaming algorithms for fundamental problem such as L_p estimation have been achieved in polylogarithmic space for $p \leq 2$ by [WZ21b] in the insertion-only model, it is a well-known open question whether there exists a constant $C = \Omega(1)$ such that the same problems require space $\Omega(n^C)$, where n is the dimension of the underlying frequency vector. Indeed, [HW13] showed that the existence of a constant $C = \Omega(1)$ such that no linear sketch with sketching dimension $o(n^C)$ can approximate the L_2 norm of an underlying frequency vector within even a polynomial multiplicative factor, when the adversarial input stream is turnstile and real-valued.

Given an accuracy parameter $(1+\varepsilon)$, the *flip number* λ is the number of times the target function $\mathcal{Q}$ changes by a factor of $(1+\mathcal{O}(\varepsilon))$. It is known that for polynomially-bounded monotone functions $\mathcal{Q}$ on insertion-only streams, we generally have $\lambda = \mathcal{O}\left(\frac{1}{\varepsilon}\log m\right)$, but for turnstile streams that toggle the underlying frequency vector between the all-zeros vector and a nonzero vector with each update, we may have $\lambda = \Omega(m)$. There are various techniques that then implement λ [BJWY22] or even roughly $\sqrt{\lambda}$ [HKM+20, ACSS23] independent instances of an oblivious algorithm, processing all stream updates to all instances. Therefore, the space complexity of these approaches are at least roughly $\sqrt{\lambda}$ times the space required by the oblivious algorithm, which may not be desirable in large setting of $\lambda = \Omega(m)$ for turnstile streams. By considering *dense-sparse tradeoffs*, [BEO22] gave a general

framework that improved upon the $\tilde{\mathcal{O}}\left(\sqrt{\lambda}\right) = \tilde{\mathcal{O}}\left(\sqrt{m}\right)$ space bounds due to the flip number. In particular, their results show that $\tilde{\mathcal{O}}\left(m^{p/(2p+1)}\right)$ space suffices for the goal of L_p norm estimation, where the objective is to estimate $(f_1^p + \ldots + f_n^p)^{1/p}$ for an input vector $f \in \mathbb{R}^n$, which is an important problem that has a number of applications, such as network traffic monitoring [FKSV02, KSZC03, TZ04], clustering and other high-dimensional geometry problems [BIRW16, CJLW22, CCJ+23, CWZ23], low-rank approximation and linear regression [CW09, FMSW10, BDM+20, VVWZ23, WY23], earth-mover estimation [Ind04, AIK08, ABIW09], cascaded norm estimation [JW09, MRWZ20], and entropy estimation [HNO08]. Unfortunately, there has been no progress for L_p estimation on turnstile streams since the work of [BEO22], either in terms of achievability or impossibility. Thus we ask:

> Is there a fundamental barrier for adversarially robust L_p estimation on turnstile streams beyond the dense-sparse tradeoffs?

1.1 Our Contributions

In this paper, we answer the above question in the negative. We show that the techniques of [BEO22] do not realize a fundamental limit for adversarially robust L_p estimation on turnstile streams. In particular, we give an algorithm that uses space $\tilde{\mathcal{O}}\left(m^c\right)$, for some constant $c < \frac{p}{2p+1}$ for $p \in (1,2)$. We first require an adversarially robust algorithm for heavy-hitters.

Heavy hitters. Recall that the ε-L_p-heavy hitter problem is defined as follows.

Definition 1.1 (ε-L_p-heavy hitters)**.** *Given a vector $f \in \mathbb{R}^n$ and a threshold parameter $\varepsilon \in (0,1)$, output a list $\mathcal{L}$ that includes all i such that $f_i \geq \varepsilon \cdot \|f\|_p$ and includes no j such that $f_j < \frac{\varepsilon}{2} \cdot \|f\|_p$.*

Generally, heavy-hitter algorithms actually solve the harder problem of outputting a estimated frequency $\widehat{f}_i$ such that $|\widehat{f}_i - f_i| \leq C \cdot \varepsilon \cdot \|f\|_p$, for each $i \in [n]$, where C is some constant such as $\frac{1}{6}$. Observe that such a guarantee solves the ε-L_p-heavy hitters problem because each i such that $f_i \geq \varepsilon \cdot \|f\|_p$ must have $\widehat{f}_i > \frac{3\varepsilon}{4} \cdot \|f\|_p$ and similarly each j such that $\widehat{f}_j \geq \frac{3\varepsilon}{4}$ must have $f_j \geq \frac{\varepsilon}{2} \cdot \|f\|_p$. We give an adversarially robust streaming algorithm for the L_p-heavy hitters problem on turnstile streams.

Theorem 1.2. *Let $p \in [1,2]$. There exists an algorithm that uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{2.5}} m^{(2p-2)/(4p-3)}\right)$ bits of space and solves the ε-L_p-heavy hitters problem at all times in an adversarial stream of length m.*

Though not necessarily obvious, our result in Theorem 1.2 improves on the dense-sparse framework of [BEO22] across all $p \in [1,2)$. Specifically, the result of [BEO22] uses space $\tilde{\mathcal{O}}\left(m^\alpha\right)$ for $\alpha = \frac{p}{2p+1}$, while our result uses space $\tilde{\mathcal{O}}\left(m^\beta\right)$ for $\beta = \frac{2p-2}{4p-3}$. It can be shown that $\alpha - \beta = \frac{2-p}{(4p-3)(2p+1)}$, which is at positive for all $p \in [1,2)$. Thus our result is an important conceptual contribution showing that the true nature of the heavy-hitter problem lies beyond the techniques of [BEO22].

A particular regime of interest is $p = 1$, where the previous dense-sparse framework of [BEO22] achieves $\tilde{\mathcal{O}}\left(m^{1/3}\right)$ bits of space, but our result in Theorem 1.2 only requires polylogarithmic space.

Moment estimation. Along the way to our main result, we also give a new algorithm for estimating the residual of a frequency vector up to some tail error. More precisely, given a frequency vector f that is defined implicitly through a data stream and a parameter $k > 0$, let g be a tail vector of f, which omits the k entries of f largest in magnitude, breaking ties arbitrarily. Similarly, let h be a tail vector of f that omits the $(1-\varepsilon)k$ entries of f largest in magnitude, where $\varepsilon \in (0,1)$ serves as an error parameter. Then we give a one-pass streaming algorithm that outputs an estimate for $\|g\|_p^p$ up to additive $\varepsilon \cdot \|h\|_p^p$, using space $\operatorname{poly}\left(\frac{1}{\varepsilon}, \log n\right)$. In particular, our space is independent of the tail parameter k. We defer the full guarantees of our algorithm as well as a more formal discussion to Section 3. We then give our main result:

Theorem 1.3. *Let $p \in [1,2]$ and $c = \frac{24p^2-23p+4}{(4p-3)(12p+3)}$. There exists a streaming algorithm that uses $\mathcal{O}\left(m^c\right) \cdot \operatorname{poly}\left(\frac{1}{\varepsilon}, \log(nm)\right)$ bits of space and outputs a $(1+\varepsilon)$-approximation to the L_p norm of the underlying vector at all times of an adversarial stream of length m.*

https://doi.org/10.52202/079017-0348

It can again be shown that our result in Theorem 1.3 again improves on the dense-sparse framework of [BEO22] across all $p \in (1,2)$. For example, for $p = 1.5$, the previous result uses space $\tilde{\mathcal{O}}\left(m^{3/8}\right) = \tilde{\mathcal{O}}\left(m^{0.375}\right)$, while our algorithm uses space $\tilde{\mathcal{O}}\left(m^{47/126}\right) \approx \tilde{\mathcal{O}}\left(m^{0.373}\right)$. Although our quantitative improvement is mild, we believe it nevertheless illustrates an important message which shows that the dense-sparse technique does not serve as an impossibility barrier.

1.2 Technical Overview

Recall that the *flip number* λ to be the number of times the F_p moment changes by a factor of $(1 + \mathcal{O}(\varepsilon))$, given a target accuracy $(1 + \varepsilon)$. Given a stream with flip number λ, the standard *sketch-switching* technique [BJWY22] for adversarial robustness is to implement λ independent instances of an oblivious streaming algorithm for F_p estimation, iteratively using the output of each algorithm only when it differs from the output of the previous algorithm by a $(1 + \varepsilon)$-multiplicative factor. Subsequently, [HKM+20, ACSS23] showed that by using differential privacy, it suffices to use roughly $\sqrt{\lambda}$ independent instances of an oblivious streaming algorithm for F_p estimation to achieve correctness at all times for an adaptive input stream. Unfortunately, the flip number for a stream of length m can be as large as $\Omega(m)$, such as in the case where the underlying frequency vector alternates between the all zeros vector and a nonzero vector.

The dense-sparse framework of [BEO22] observes that the only case where the flip number can be large is when there are a large number of times in the stream where the corresponding frequency vector is somewhat sparse. For example, in the above scenario where the underlying frequency vector alternates between the all zeros vector and a nonzero vector, all input vectors are 1-sparse. In fact, they notice that for F_p estimation, that once the frequency vector has at least m^C nonzero entries for any fixed constant $C \in (0,1)$, then since all entries must be integral and all updates only change each entry by 1, at least $\Omega_\varepsilon(m^{C/p})$ updates are necessary before the p-th moment of the resulting frequency vector can differ by at least $(1 + \varepsilon)$-multiplicative factor. Hence in the stream updates where the frequency vector has at least m^C nonzero entries, the flip number can be at most $\mathcal{O}\left(m^{1-C/p}\right)$, for $\varepsilon = \Omega(1)$. Thus it suffices to run $\tilde{\mathcal{O}}\left(m^{1/2-C/2p}\right)$ independent instances of the oblivious algorithm, using the differential privacy technique of [HKM+20, ACSS23]. Moreover, in the case where the vector is m^C-sparse, there are sparse recovery techniques that can exactly recover all the nonzero coordinates using $\tilde{\mathcal{O}}\left(m^C\right)$ space, even if the input is adaptive. Hence by balancing $\tilde{\mathcal{O}}\left(m^C\right) = \tilde{\mathcal{O}}\left(m^{(1/2-C/2p}\right)$ at $C = \frac{1}{3}$, [BEO22] achieves $\tilde{\mathcal{O}}\left(m^{p/(2p+1)}\right)$ overall space for F_p estimation for adaptive turnstile streams.

Our key observation is that for $p \in (1,2)$, if the frequency vector has at least m^C nonzero entries, a sequence of $\mathcal{O}_\varepsilon(m^{C/p})$ updates may not always change the p-th moment of the underlying vector. For example, if the updates are all to separate coordinates, then the p-th moment may actually change very little. In fact, a sequence of $\mathcal{O}_\varepsilon(m^{C/p})$ updates may *only* change the p-th moment of the underlying vector by $(1 + \varepsilon)$ if most of the updates are to a small number of coordinates. As a result, most of the updates are to some coordinate that was either initially a heavy-hitter or subsequently a heavy-hitter. Then by tracking the heavy-hitters of the underlying frequency vector, we can handle the hard input for [BEO22], thus demanding a larger number of stream updates before the p-th moment of the vector can change by $(1 + \varepsilon)$. Consequently, the number of independent instances decreases, which facilitates a better balancing and allows us to achieve better space bounds. Unfortunately, there are multiple challenges to realizing this intuition.

Heavy-hitters. First, we need a streaming algorithm for accurately reporting the frequencies of the L_p-heavy hitters at all times in the adaptive stream. However, such a subroutine is not known and naïvely, one might expect an estimate of the L_p norm might be necessary to identify the L_p heavy-hitters. Moreover, algorithms for finding L_p heavy-hitters are often used to estimate the L_p norm of the underlying frequency, e.g., [IW05, WZ12, BBC+17, LSW18, BWZ21, WZ21a, MWZ22, BMWZ23, JWZ24]. Instead, we use a turnstile streaming algorithm DETHH for L_p heavy-hitters [GM07] that uses sub-optimal space $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2} \cdot n^{2-2/p}\right)$ bits of space for $p \in (1,2]$, rather than the optimal COUNTSKETCH, which uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log^2 n\right)$ bits of space. However, the advantage of DETHH is that the algorithm is deterministic, so we can utilize the previous intuition from the dense-sparse framework of [BEO22]. In particular, if the universe size is small, then we can run DETHH, and if the universe size is large, then we collectively handle these cases using an ensemble

of CountSketch algorithms via differential privacy. We provide the full details of the robust L_p-heavy hitter algorithm in Section 2, ultimately achieving Theorem 1.2.

Residual estimation. It remains to estimate the contribution of the elements that are not L_p heavy-hitters, i.e., the residual vector, toward the overall p-th moment. More generally, given a tail parameter $k > 0$ and an error parameter $\varepsilon \in (0,1)$, let g be a tail vector of f that omits the k entries of f largest in magnitude, breaking ties arbitrarily and let h be a tail vector of f that omits the $(1-\varepsilon)k$ entries of f largest in magnitude. We define the level sets of the p-th moment so that level set Λ_ℓ roughly consists of the coordinates of g with magnitude $[(1+\varepsilon)^\ell, (1+\varepsilon)^{\ell+1})$. We then estimate the contribution of each level set to the p-th moment of the residual vector using the subsampling framework introduced by [IW05].

Namely, we note that any "significant" level set has either a small number of items with large magnitude, or a large number of items that collectively have significant contribution to the p-th moment. In the former case, we can use CountSketch to identify the items with large magnitude, while in the latter case, it can be shown that after subsampling the universe, there will be a large number of items in the level set that remain. Moreover, these items will now be heavy with respect to the p-th moment of the resulting frequency vector after subsampling with high probability. Thus, these items can be identified by CountSketch on the subsampled universe. Furthermore, after rescaling inversely by the sampling probability, the total number of such items in the level set can be estimated accurately by rescaling the number of the heavy-hitters in the subsampled universe. Hence in both cases, we can estimate the number of items in the significant level sets and subtract off the largest k such items. We provide the full details of the residual estimation algorithm in Section 3, culminating in Theorem 3.4.

2 Adversarially Robust L_p-Heavy Hitters

In this section, we give an adversarially robust algorithm for L_p-heavy hitters on turnstile streams. We first recall the following deterministic algorithm for L_p-heavy hitters on turnstile streams.

Theorem 2.1. *[GM07] For $p \in [1,2)$, there exists a deterministic algorithm* DetHH *that solves the ε-L_p heavy-hitters on a universe of size t and a stream of length m and uses $\frac{1}{\varepsilon^2} t^{2-2/p} \operatorname{polylog} \frac{tm}{\varepsilon}$ bits of space.*

We also recall the following variant of CountSketch for answering a number of rounds of adaptive queries, as well as a more general framework for answering adaptive queries.

Theorem 2.2. *[CLN+22] For $p \in [1,2)$, there exists a randomized algorithm* RobustCS *that uses $\tilde{\mathcal{O}}\left(\frac{\sqrt{\lambda}}{\varepsilon^2} \log n \log \frac{nm\lambda}{\delta}\right)$ bits of space, and for λ different times t on an adaptive stream of length m on a universe of size n, reports for all $i \in [n]$ an estimate $\widehat{f_i^{(t)}}$ such that $|\widehat{f_i^{(t)}} - f_i^{(t)}| \le \frac{\varepsilon}{100} \cdot \|f^{(t)}\|_2$, where $f^{(t)}$ is the induced frequency vector at time t.*

Theorem 2.3. *[HKM+20, BKM+22, ACSS23, CSW+23] Given a streaming algorithm $\mathcal{A}$ that uses S space and answers a query with constant failure probability $\delta_0 < \frac{1}{2}$, there exists a data structure that answers Q adaptive queries, with probability $1-\delta$ using space $\mathcal{O}\left(S\sqrt{Q} \log^2 \frac{Q}{\delta}\right)$.*

While RobustCS has better space guarantees than DetHH, determinism nevertheless serves an important purpose for us. Namely, adversarial input can induce failures on randomized algorithms but cannot induce failures on deterministic algorithms. On the other hand, the space usage of DetHH grows with the size of the universe. Thus, we now use insight from the dense-sparse framework of [BEO22]. If the universe size is small, then we shall use DetHH. On the other hand, if the universe size is large, then shall use the following robust version of CountSketch, requiring roughly $\sqrt{\lambda}$ number of independent instances, where λ is the flip number. The key observation is that because the universe size is large, then the flip number will be much smaller than in the worst possible case. Moreover, we can determine which case we are in, i.e., the large universe case or the small universe case, by using the following L_0 estimation algorithm:

Theorem 2.4. *[KNW10] There exists an insertion-deletion streaming algorithm* LZeroEst *that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n \log \frac{1}{\delta} \left(\log \frac{1}{\varepsilon} + \log\log m\right)\right)$ bits of space, and with probability at least $1-\delta$, outputs a $(1+\varepsilon)$-approximation to L_0.*

https://doi.org/10.52202/079017-0348

Algorithm 1 RobustHH: Adversarially robust L_p-heavy hitters

Input: Turnstile stream of length m for a frequency vector of length n
Output: Adversarially robust heavy-hitters

$t \leftarrow \mathcal{O}\left(m^{p/(4p-3)}\right)$, $\ell \leftarrow \frac{\varepsilon}{100} \cdot t^{1/p}$, $b \leftarrow \frac{m}{\ell}$,STATE $\leftarrow$ SPARSE
Initialize DetHH with threshold $\frac{\varepsilon}{16}$
Initialize RobustCS robust to b queries, with threshold $\frac{\varepsilon}{16}$ for $r = \mathcal{O}\left(\frac{m}{\varepsilon t^{1/p}}\right)$ rounds
Initialize $\tilde{\mathcal{O}}\left(\sqrt{b}\right)$ copies LZeroEst with accuracy 2 robust to b queries
for each block of ℓ updates **do**
    Update DetHH, RobustCS, and all copies of LZeroEst
    if STATE $=$ SPARSE at the beginning of the block **then**
        Return the output of DetHH
    else
        Return the output of RobustCS at the beginning of the block ▷Theorem 2.2
    Let Z be the output of robust LZeroEst ▷Theorem 2.3 and Theorem 2.4
    if $Z > 100t$ **then**
        STATE $\leftarrow$ DENSE
    else
        STATE $\leftarrow$ SPARSE

We give our algorithm in full in Algorithm 1. Because DetHH is a deterministic algorithm, it will always be correct in the case where the universe size is small. Thus, we first prove that in the case where the universe size is large, then RobustCS ensures correctness within each sequence of ℓ updates.

Lemma 2.5. *Suppose the number of distinct elements at the beginning of a block is at least $50t$. Let S be the output of* RobustCS *at the beginning of a block. Then conditioned on the correctness of* RobustCS*, S solves the L_p-heavy hitter problem on the entire block.*

Next, we show that RobustCS ensures correctness in between blocks as well. We also analyze the space complexity of our algorithm.

Lemma 2.6. *With high probability,* RobustCS *is correct at the beginning of each block of length ℓ.*

Lemma 2.7. *The total space by the algorithm is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{2.5}} m^{(2p-2)/(4p-3)}\right)$ bits of space.*

Given our proof of correctness in Lemma 2.5 and Lemma 2.6, as well as the space analysis in Lemma 2.7, then we obtain Theorem 1.2.

3 Oblivious Residual Estimation Algorithm

In this section, we consider norm and moment estimation of a residual vector, permitting bicriteria error by allowing some slack in the size of the tail. Specifically, suppose the input vector f arrives in the streaming model. Given a tail parameter $k > 0$ and an error parameter $\varepsilon \in (0, 1)$, let g be a tail vector of f that omits the k entries of f largest in magnitude, breaking ties arbitrarily and let h be a tail vector of f that omits the $(1-\varepsilon)k$ entries of f largest in magnitude. We give an algorithm that estimates $\|g\|_p^p$ up to additive $\varepsilon \cdot \|h\|_p^p$, using space poly $\left(\frac{1}{\varepsilon}, \log n\right)$, which is independent of the tail parameter k. It should be noted that our algorithm is imprecise on $\|g\|_p^p$ in two ways. Firstly, it incurs additive error proportional to ε. Secondly, the additive error has error with respect to h, which is missing the top $(1-\varepsilon)k$ entries of f in magnitude, rather than the top k. Nevertheless, the space bounds that are independent of k are sufficiently useful for our subsequent application of L_p estimation. We first define the level sets of the p-th moment and the contribution of each level set.

Definition 3.1 (Level sets and contribution)**.** *Let $\eta > 0$ be a parameter and let m be the length of the stream. Let M be the power of two such that $m^p \le M < (1+\eta)m^p$ and let $\zeta \in [1, 2]$. Then for each integer $\ell \ge 1$, we define the level set $\Gamma_\ell := \left\{ i \in [n] \mid f_i \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^{\ell}} \right) \right\}$. We also define the contribution C_ℓ of level set Γ_ℓ to be $C_\ell := \sum_{i \in \Gamma_\ell} (f_i)^p$.*

For a residual vector g of f with the top k coordinates set to be zero, we similarly define the level sets Λ_ℓ and D_ℓ of g in the natural way, i.e., $D_\ell := \sum_{i\in\Lambda_\ell}(g_i)^p$ for $\Lambda_\ell := \left\{i \in [n] \mid g_i \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^{\ell}}\right)\right\}$.

Algorithm 2 RESIDUALEST: residual F_p approximation algorithm, $p \in [1,2]$

Input: Stream $s_1, \ldots, s_m$ of items from $[n]$, accuracy parameter $\varepsilon \in (0,1)$, $p \in [1,2]$
Output: $(1+\varepsilon)$-approximation to F_p

$\eta \leftarrow \frac{\varepsilon}{100}$, $L \leftarrow \tilde{\mathcal{O}}\left(\frac{\log(nm)}{\eta}\right)$, $P = \tilde{\mathcal{O}}\left(\log(nm)\right)$, $R \leftarrow \tilde{\mathcal{O}}\left(\log\frac{\log n}{\eta}\right)$, $\gamma \leftarrow 2^{20}$
for $t = 1$ to $t = m$ **do**
  for $(i, r) \in [P] \times [R]$ **do**
    Let $U_i^{(r)}$ be a (nested) subset of $[n]$ subsampled at rate $p_i := \min(1, 2^{1-i})$
    if $s_t \in U_i^{(r)}$ **then**
      Send s_t to COUNTSKETCH$_i^{(r)}$ with accuracy η^3
Let $M = 2^i$ for some integer $i \geq 0$, such that $m^p \leq M < 2m^p$
$c \leftarrow k$
for $\ell = 1$ to $\ell = L$ **do**
  $i \leftarrow \max\left(1, \left\lfloor \log(1+\eta)^\ell - \log\frac{\gamma^2\log(nm)}{\eta^3}\right\rfloor\right)$
  Let $H_i^{(r)}$ be the outputs of COUNTSKETCH at level i
  Let $S_i^{(r)}$ be the set of ordered pairs $(j, \widehat{f_j})$ of $H_i^{(r)}$ with $\left(\widehat{f_j}\right)^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^{\ell}}\right]$
  $\widehat{|\Gamma_\ell|} \leftarrow \frac{1}{p_i}\,\text{median}_{r\in[R]}\,|S_i^{(r)}|$, $T_\ell \leftarrow \max(0, \widehat{\Gamma_\ell} - c)$
  $c \leftarrow \max(c - \widehat{\Gamma_\ell}, 0)$
  $\widehat{|\Lambda_\ell|} \leftarrow T_\ell \cdot (1+\eta)^\ell$
Return $\widehat{F_{p,\mathrm{Res}(k)}} = \sum_{\ell\in[L]} \widehat{|\Lambda_\ell|}(1+\eta)^\ell$

Our algorithm attempts to estimate the contribution of each level set. Some of these level sets contribute a "significant" amount to the p-th moment of f, whereas other level sets do not. It can be seen that the number of items in each level set that is contributing can be estimated up to a $(1+\mathcal{O}(\varepsilon))$-approximation. In particular, either a contributing level set has a small number of items with large mass, or a large number of items that collectively have significant mass. We use the heavy-hitter algorithm COUNTSKETCH to detect the level sets with a small number of items with large mass, and count the number of items in these level sets. For the large number of items that collectively have significant mass, it can be shown that after subsampling the universe, there will be a large number of these items remaining, and those items will be identified by COUNTSKETCH on the subsampled universe. Moreover, the total number of such items in the level set can be estimated accurately by rescaling the number of the heavy-hitters in the subsampled universe inversely by the sampling probability. We can thus carefully count the number of items in the contributing level sets and subtract off the largest k such items. Because we only have $(1+\varepsilon)$-approximations to the number of such items, it may be possible that we subtract off too many, hence the bicriteria approximation.

Finally, we note that for the insignificant level sets, we can no longer estimate the number of items in these level set up to $(1+\varepsilon)$-factor. However, we note that the number of such items is only an ε fraction of the number of items in the lower level sets that are contributing. Therefore, we can show that it suffices to set the contribution of these level sets to zero. Our algorithm appears in full in Algorithm 2.

We now show that the number of items (as well as their contribution) in each "contributing" level set with a small number of items with large mass will be estimated within a $(1+\varepsilon)$-approximation.

Lemma 3.2. *Let $r \in [R]$ be fixed. Then with probability at least $\frac{9}{10}$, we have that simultaneously for all $j \in U_i^{(r)}$ for which $(f_j)^p \geq \frac{\eta^3 \cdot F_p(U_i^{(r)})}{2^7\gamma\log^2(nm)}$, $H_\ell^{(r)}$ outputs $\widehat{f_j}$ with $\left(1 - \frac{\eta}{8\log(nm)}\right)\cdot(f_j)^p \leq (\widetilde{f_j})^p \leq \left(1 + \frac{\eta}{8\log(nm)}\right)\cdot(f_j)^p$.*

https://doi.org/10.52202/079017-0348

We now show that the number of items in each "contributing" level set is estimated within a $(1+\varepsilon)$-approximation, including the level sets that contain a large number of small items.

Lemma 3.3. *Given a fixed $\varepsilon \in (0,1)$, let Λ_ℓ be a fixed level set and let $r \in [R]$ be fixed. Let $i = \max\left(1, \left\lfloor \log(1+\eta)^\ell - \log \frac{\gamma^2 \log(nm)}{\eta^3} \right\rfloor\right)$. Define the events $\mathcal{E}_1$ to be the event that $|U_i^{(r)}| \le \frac{32n}{2^i}$ and $\mathcal{E}_2$ to be the event that $F_p(U_i^{(r)}) \le \frac{32F_p}{2^i}$. Then conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$, for each $j \in \Lambda_\ell \cap U_i^{(r)}$, there exists $(j, \widetilde{f}_j)$ in $S_i^{(r)}$ such that with probability at least $\frac{9}{10}$, $\left(1 - \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p \le (\widetilde{f}_j)^p \le \left(1 + \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p$.*

Putting things together, we have the following full guarantees for our algorithm.

Theorem 3.4. *There exists a one-pass streaming algorithm* ResidualEst *that takes an input parameter $k \ge 0$ (possibly upon post-processing the stream) and uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6} \cdot \log^3(nm)\right)$ bits of space to output an estimate $\widehat{F_{p,\mathrm{Res}(k)}}$ with $\mathbf{Pr}\left[\left|\widehat{F_{p,\mathrm{Res}(k)}} - F_{p,\mathrm{Res}(k)}\right| \le \varepsilon \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}\right] \ge \frac{2}{3}$.*

4 Adversarially Robust L_p Estimation

In this section, we give an adversarially robust algorithm for F_p moment estimation on turnstile streams. Due to the relationship between the F_p moment and the L_p norm, our result similarly translates to a robust algorithm for L_p norm estimation. We first require an algorithm to recover all the coordinates of the underlying frequency vector if it is sparse.

Theorem 4.1. *[GSTV07] There exists a deterministic algorithm* SparseRecover *that recovers a k-sparse frequency vector defined by an insertion-deletion stream of length n. The algorithm uses $k \cdot \mathrm{polylog}(n)$ bits of space.*

Algorithm 3 Adversarially robust L_p-estimation

Input: Turnstile stream of length m for a frequency vector of dimension n
Output: Adversarially robust heavy-hitters

$c \leftarrow \frac{24p^2-23p+4}{(4p-3)(12p+3)}$, $\gamma \leftarrow \frac{2c}{5} - \frac{(4p-4)}{(20p-15)}$, $\eta \leftarrow \frac{\varepsilon^2}{100m^\gamma}$, $k \leftarrow \mathcal{O}\left(\frac{1}{\eta^p}\right)$, $\ell \leftarrow \mathcal{O}\left(\varepsilon \cdot m^{c/p} k^{1-1/p}\right)$, STATE $\leftarrow$ SPARSE
Initialize SparseRecover with sparsity $\mathcal{O}(m^c)$
Initialize RobustHH with threshold $\varepsilon\eta$
Initialize LZeroEst with accuracy 2 robust to $b := \frac{m}{\ell}$ queries
Initialize ResidualEst with parameter k and accuracy $\mathcal{O}(\varepsilon)$ robust to b queries
for each block of ℓ updates **do**
    Update RobustHH, LZeroEst, and ResidualEst
    if STATE $=$ SPARSE at the beginning of the block **then**
        Let g be the vector output by SparseRecover
        $\widehat{G} \leftarrow \|g\|_p^p$
        Return $\widehat{G}$
    else
        Let g be the vector output by RobustHH at the beginning of the block
        Let $\widehat{H}$ be the output of ResidualEst
        $\widehat{G} \leftarrow \|g\|_p^p$
        Return $\widehat{G} + \widehat{H}$
    Let Z be the output of robust LZeroEst
    if $Z > 100t$ **then**
        STATE $\leftarrow$ DENSE
    else
        STATE $\leftarrow$ SPARSE

We remark that SparseRecover is deterministic and guarantees correctness on a turnstile stream, even if the frequency vector is not sparse at some intermediate step of the stream. On the other hand,

if the frequency vector is not sparse, then a query to SPARSERECOVER could be erroneous. Hence, our algorithm thus utilizes robust LZEROEST to detect whether the underlying frequency vector is dense or sparse. Similar to [BEO22], the intuition is that due to the sparse case always succeeding, the adversary can only induce failure if the vector is dense, which in turn decreases the flip number. However, because we also accurately track the heavy-hitters, then the adversary must spread the updates across a multiple number of coordinates, resulting in a larger number of updates necessary to double the residual vector. Since the number of updates is larger, then the flip number is smaller, and so our algorithm can use less space. Unfortunately, even though the residual vector may not double in its p-th moment, the p-th moment of entire frequency vector f may change drastically. This is a nuance for the analysis because our error guarantee can no longer be relative to the $\|f\|_p^p$. Indeed, $\varepsilon \cdot \|f\|_p^p$ additive error may induce $(1+\varepsilon)$-multiplicative error at one point, but at some later point we could have $\|f'\|_p^p \ll \|f\|_p^p$, so that the same additive error could even be polynomial multiplicative error. Hence, we require the RESIDUALEST subroutine from Section 3, whose guarantees are in terms of the residual vector. We give our algorithm in full in Algorithm 3.

We upper bound the amount that the p-th moment of the residual vector can change, given a bounded number of updates.

Lemma 4.2. *Let f be a frequency vector and g be the residual vector omitting the k coordinates of f largest in magnitude. Let v be any arbitrary vector such that $\|v\|_1 \leq \frac{\varepsilon}{100} \cdot \|g\|_p \cdot k^{1-1/p}$ and $\|v\|_1 \leq \frac{1}{2}\|g\|_1$. Let u be the residual vector omitting the k coordinates of $f+v$ largest in magnitude. Then we have $|\|g\|_p^p - \|u\|_p^p| \leq \frac{\varepsilon}{4} \cdot \|g\|_p^p$.*

We now show correctness and space complexity of Algorithm 3, after which Theorem 1.3 follows.

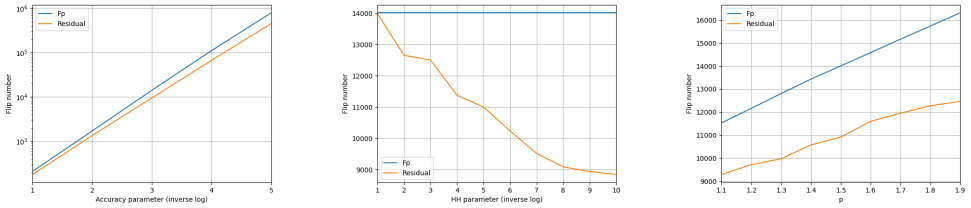
Lemma 4.3. *For $\log n = \Theta(\log m)$, Algorithm 3 uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{7.5}} \cdot m^c\right)$ bits of space in total. Moreover, for any fixed time during a stream, let f be the induced frequency vector and let $\widehat{F}$ be the output of Algorithm 3. Then we have that with high probability, $(1-\varepsilon)\|f\|_p^p \leq \widehat{F} \leq (1+\varepsilon)\|f\|_p^p$.*

5 Empirical Evaluations

In this section, we describe our empirical evaluations for comparing the flip number of the entire vector and the flip number of the residual vector on real-world datasets. Note that these quantities parameterize the space used by the algorithm of [BEO22] and by our algorithm, respectively.

CAIDA traffic monitoring dataset. We used the CAIDA dataset [CAI16] of anonymized passive traffic traces from the 'equinix-nyc' data center's high-speed monitor. The dataset is commonly used for empirical evaluations on frequency moments and heavy-hitters. We extracted the sender IP addresses from 12 minutes of the internet flow data, which contained 2,9922,873 total events.

Experimental setup. Our empirical evaluations were performed Python 3.10 on a 64-bit operating system on an AMD Ryzen 7 5700U CPU, with 8GB RAM and 8 cores with base clock 1.80 GHz. We compare the flip number of the entire data stream versus the flip number of the residual vector across various values of the algorithm error $\varepsilon \in \{10^{-1}, 10^{-2}, \ldots, 10^{-5}\}$, values of the heavy-hitter threshold $\alpha \in \{4^{-1}, 4^{-2}, \ldots, 4^{-10}\}$, and the frequency moment parameter $p \in \{1.1, 1.2, \ldots, 1.9\}$. We describe the results in Figure 1.



(a) Flip number across $-\log_{10} \varepsilon$ (b) Flip number across $-\log_4 \alpha$ (c) Flip number across p

Fig. 1: Empirical evaluations on the CAIDA dataset, comparing flip number of the p-th frequency moment and the residual, for $\varepsilon = \alpha = 0.001$ and $p = 1.5$ when not variable. Smaller flip numbers indicate less space needed by the algorithm.

https://doi.org/10.52202/079017-0348

Results and discussion. Our empirical evaluations serve as a simple proof-of-concept demonstrating that adversarially robust algorithm can use significantly less space than existing algorithms. In particular, existing algorithms use space that is an increasing function of the flip number of the p-th frequency moment, while our algorithms use space that is an increasing function of the flip number of the residual, which is significantly less across all settings in Figure 1. While the ratio does increase as the exponent p increases in Figure 1c, there is not a substantial increase, i.e., 1.24 to 1.31 from $p = 1.1$ to $p = 1.9$. On the other hand, as α decreases in Figure 1b, the ratio increases from 1.002 for $\alpha = 4^{-1}$ to 1.6 for $\alpha = 4^{-10}$. Similarly, in Figure 1a, the ratio of these quantities begins at 1.17 for $\varepsilon = 10^{-1}$ and increases to as large as 1.75 for $\varepsilon = 10^{-5}$. Therefore, even in the case where the input is not adaptive, our empirical evaluations demonstrate that these flip number quantities can be quite different, and consequently, our algorithm can use significantly less space than previous existing algorithms.

Acknowledgements

David P. Woodruff was supported in part by a Simons Investigator Award and NSF CCF-2335412. Samson Zhou is supported in part by NSF CCF-2335411. The work was conducted in part while David P. Woodruff and Samson Zhou were visiting the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.

References

[ABD+21] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *STOC: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 447–455, 2021.

[ABIW09] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. Efficient sketches for earth-mover distance, with applications. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 324–330, 2009.

[ACGS23] Sepehr Assadi, Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Coloring in graph streams via deterministic and adversarially robust algorithms. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, pages 141–153, 2023.

[ACSS23] Idan Attias, Edith Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. In *14th Innovations in Theoretical Computer Science Conference, ITCS*, pages 8:1–8:19, 2023.

[AIK08] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 343–352, 2008.

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

[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.

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

[BDM+20] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 517–528, 2020.

[BEO22] Omri Ben-Eliezer, Talya Eden, and Krzysztof Onak. Adversarially robust streaming via dense-sparse trade-offs. In *5th Symposium on Simplicity in Algorithms, SOSA*, 2022. (to appear).

[BHM+21] Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing, NeurIPS*, pages 3544–3557, 2021.

[BIRW16] Arturs Backurs, Piotr Indyk, Ilya P. Razenshteyn, and David P. Woodruff. Nearly-optimal bounds for sparse recovery in generic norms, with applications to k-median sketching. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 318–337, 2016.

[BJWY22] Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. *J. ACM*, 69(2):17:1–17:33, 2022.

[BKM+22] Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, and Uri Stemmer. Dynamic algorithms against an adaptive adversary: generic constructions and lower bounds. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1671–1684, 2022.

[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*, pages 508–516, 2017.

[BMWZ23] Vladimir Braverman, Joel Manning, Zhiwei Steven Wu, and Samson Zhou. Private data stream analysis for universal symmetric norm estimation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 45:1–45:24, 2023.

[BNS+21] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan R. Ullman. Algorithmic stability for adaptive data analysis. *SIAM J. Comput.*, 50(3), 2021.

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

[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.

[CAI16] CAIDA. The caida ucsd anonymized internet traces. https://www.caida.org/catalog/datasets/passive_dataset, 2016.

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

[CCJ+23] Xi Chen, Vincent Cohen-Addad, Rajesh Jayaram, Amit Levi, and Erik Waingarten. Streaming euclidean MST to a constant factor. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, pages 156–169, 2023.

[CGS22] Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Adversarially robust coloring for graph streams. In *13th Innovations in Theoretical Computer Science Conference, ITCS*, pages 37:1–37:23, 2022.

[CJLW22] Xi Chen, Rajesh Jayaram, Amit Levi, and Erik Waingarten. New streaming algorithms for high dimensional EMD and MST. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 222–233, 2022.

 https://doi.org/10.52202/079017-0348

[CLN+22] Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, Moshe Shechner, and Uri Stemmer. On the robustness of countsketch to adaptive inputs. In *International Conference on Machine Learning, ICML*, pages 4112–4140, 2022.

[CN20] Yeshwanth Cherapanamjeri and Jelani Nelson. On adaptive distance estimation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

[CSW+23] Yeshwanth Cherapanamjeri, Sandeep Silwal, David P. Woodruff, Fred Zhang, Qiuyi Zhang, and Samson Zhou. Robust algorithms on adaptive inputs from bounded adversaries. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

[CW09] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC*, pages 205–214, 2009.

[CWZ23] Vincent Cohen-Addad, David P. Woodruff, and Samson Zhou. Streaming euclidean k-median and k-means with o(log n) space. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 883–908, 2023.

[DFH+15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 117–126. ACM, 2015.

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

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 51–60, 2010.

[DSWZ23] Itai Dinur, Uri Stemmer, David P. Woodruff, and Samson Zhou. On differential privacy and adaptive data analysis with bounded space. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III*, pages 35–65, 2023.

[FKSV02] Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. An approximate l1-difference algorithm for massive data streams. *SIAM J. Comput.*, 32(1):131–151, 2002.

[FMSW10] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 630–649, 2010.

[GHS+12] Anna C. Gilbert, Brett Hemenway, Martin J. Strauss, David P. Woodruff, and Mary Wootters. Reusable low-error compressive sampling schemes through privacy. In *IEEE Statistical Signal Processing Workshop, SSP*, pages 536–539, 2012.

[GLW+24] Elena Gribelyuk, Honghao Lin, David P. Woodruff, Huacheng Yu, and Samson Zhou. A strong separation for adversarially robust l_0 estimation for linear sketches. *CoRR*, abs/2409.16153, 2024.

[GM07] Sumit Ganguly and Anirban Majumder. Cr-precis: A deterministic summary structure for update data streams. In *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, First International Symposium, ESCAPE*, pages 48–59, 2007.

[GSTV07] Anna C. Gilbert, Martin J. Strauss, Joel A. Tropp, and Roman Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 237–246, 2007.

[HKM+20] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

[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.

[HW13] Moritz Hardt and David P. Woodruff. How robust are linear sketches to adaptive inputs? In *Symposium on Theory of Computing Conference, STOC*, pages 121–130, 2013.

[Ind04] Piotr Indyk. Algorithms for dynamic geometric problems over data streams. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 373–380, 2004.

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

[JW09] T. S. Jayram and David P. Woodruff. The data stream space complexity of cascaded norms. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 765–774, 2009.

[JWZ24] Rajesh Jayaram, David P. Woodruff, and Samson Zhou. Streaming algorithms with few state changes. In *Proceedings of the 43rd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS*, 2024.

[KMGG07] Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Selecting observations against adversarial objectives. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pages 777–784, 2007.

[KMNS21] Haim Kaplan, Yishay Mansour, Kobbi Nissim, and Uri Stemmer. Separating adaptive streaming from oblivious streaming using the bounded storage model. In *Advances in Cryptology - CRYPTO - 41st Annual International Cryptology Conference, CRYPTO Proceedings, Part III*, pages 94–121, 2021.

[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.

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

[KZK18] Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

[LSW18] Roie Levin, Anish Prasad Sevekari, and David P. Woodruff. Robust subspace approximation in a stream. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018.

[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.

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

 https://doi.org/10.52202/079017-0348

[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.

[MWZ22] Sepideh Mahabadi, David P. Woodruff, and Samson Zhou. Adaptive sketches for robust regression with importance sampling. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM)*, pages 31:1–31:21, 2022.

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

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

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

[VVWZ23] Ameya Velingker, Maximilian Vötsch, David P. Woodruff, and Samson Zhou. Fast (1+ϵ)-approximation algorithms for binary matrix factorization. In *International Conference on Machine Learning, ICML*, pages 34952–34977, 2023.

[WY23] David P. Woodruff and Taisuke Yasuda. Online lewis weight sampling. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 4622–4666, 2023.

[WZ12] David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 941–960, 2012.

[WZ21a] David P. Woodruff and Samson Zhou. Separations for estimating large frequency moments on data streams. In *48th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 112:1–112:21, 2021.

[WZ21b] David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1183–1196, 2021.

A Preliminaries

For a positive integer $n > 0$, we use $[n]$ to denote the set of integers $\{1, \ldots, n\}$. We use $\mathrm{poly}(n)$ to denote a fixed polynomial in n whose degree can be set by adjust constants in the algorithm based on various desiderata, e.g., in the failure probability. We use $\mathrm{polylog}(n)$ to denote $\mathrm{poly}(\log n)$. When there exist constants to facilitate an event to occur with probability $1 - \frac{1}{\mathrm{poly}(n)}$, we say that the event occurs with high probability. For a random variable X, we use $\mathbb{E}[X]$ to denote its expectation and $\mathrm{Var}(X)$ to denote its variance.

Recall that for $p > 0$, the L_p norm of a vector $v \in \mathbb{R}^n$ is $\|v\|_p = (v_1^p + \ldots + v_n^p)^{1/p}$. The p-th moment of v is defined as $F_p(v) = \|v\|_p^p$. Note that for a constant $p \geq 1$, a $(1+\varepsilon)$-approximation to the $F_p(v)$ implies a $(1+\varepsilon)$-approximation to $\|v\|_p$. Similarly, for a sufficiently small constant $\varepsilon \in (0, 1)$, a $(1 + \mathcal{O}(\varepsilon))$-approximation to $\|v\|_p$ implies a $(1 + \mathcal{O}(\varepsilon))^p = (1+\varepsilon)$-approximation to $F_p(v)$. We thus use the problems of L_p norm estimation and F_p moment estimation interchangeably in discussion.

We use $F_{p,\mathrm{Res}(k)}(f)$ to denote the p-th moment of a vector g obtained by setting to zero the k coordinates of f largest in magnitude, breaking ties arbitrarily. We also define $\|v\|_0$ to be the number of nonzero coordinates of v, so that $\|v\|_0 = |\{i \in [n] \mid v_i \neq 0\}|$.

We recall the following notions regarding differential privacy.

Definition A.1 (Differential privacy). *[DMNS06] Given $\varepsilon > 0$ and $\delta \in (0,1)$, a randomized algorithm $\mathcal{A} : D \to R$ with domain D and range R is (ε, δ)-differentially private if, for every neighboring datasets S and S' and for all $\mathcal{E} \subseteq R$,*

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

Theorem A.2 (Private median, e.g., [HKM+20]). *Given a database $\mathcal{D} \in X^*$, there exists an $(\varepsilon, 0)$-differentially private algorithm* PrivMed *that outputs an element $x \in X$ such that with probability at least $1-\delta$, there are at least $\frac{|S|}{2} - k$ elements in S that are at least x, and at least $\frac{|S|}{2} - k$ elements in S in S that are at most x, for $k = \mathcal{O}\left(\frac{1}{\varepsilon} \log \frac{|X|}{\delta}\right)$.*

Theorem A.3 (Advanced composition, e.g., [DRV10]). *Let $\varepsilon, \delta' \in (0,1]$ and let $\delta \in [0,1]$. Any mechanism that permits k adaptive interactions with mechanisms that preserve (ε, δ)-differential privacy guarantees $(\varepsilon', k\delta + \delta')$-differential privacy, where $\varepsilon' = \sqrt{2k \ln \frac{1}{\delta'}} \cdot \varepsilon + 2k\varepsilon^2$.*

Theorem A.4 (Generalization of DP, e.g., [DFH+15, BNS+21]). *Let $\varepsilon \in (0, 1/3)$, $\delta \in (0, \varepsilon/4)$, and $n \geq \frac{1}{\varepsilon^2} \log \frac{2\varepsilon}{\delta}$. Suppose $\mathcal{A} : X^n \to 2^X$ is an (ε, δ)-differentially private algorithm that curates a database of size n and produces a function $h : X \to \{0,1\}$. Suppose $\mathcal{D}$ is a distribution over X and S is a set of n elements drawn independently and identically distributed from $\mathcal{D}$. Then*

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}, h \leftarrow \mathcal{A}(S)}\left[\left|\frac{1}{|S|}\sum_{x \in S} h(x) - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}\left[h(x)\right]\right| \geq 10\varepsilon\right] < \frac{\delta}{\varepsilon}.$$

Algorithm 4 Adversarially Robust Framework

Input: Oblivious algorithms $\mathcal{A}$ with failure probability δ_0, number of queries Q, failure probability δ

Output: Algorithm robust to Q queries, with failure probability at most δ

1: $r \leftarrow \mathcal{O}\left(\sqrt{Q} \log^2 \frac{Q}{\delta\delta_0}\right)$
2: Implement $k = \mathcal{O}(r)$ independent instances $\mathcal{A}_1, \ldots, \mathcal{A}_k$ of $\mathcal{A}$ on the input
3: **for** each query q_i, $i \in [Q]$ **do**
4: Let $Z_{i,j}$ be the output of $\mathcal{A}_j$ on q_i
5: Let PrivMed be $\left(\frac{1}{r}, 0\right)$-DP
6: Return PrivMed$(\{Z_{i,j}\}_{j \in [k]})$

We remark that Algorithm 4 is the algorithm corresponding to the statement of Theorem 2.3

B Missing Proofs from Section 2

One reason that DetHH is not commonly utilized is that with the additional power of randomness, significantly better space bounds can be achieved, such as by the following guarantees:

Theorem B.1. *[CCF04] For $p \in [1,2)$, there exists a randomized algorithm* CountSketch *that solves the ε-L_p heavy-hitters on a universe of size n and a stream of length m and uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n \log \frac{nm}{\delta}\right)$ bits of space.*

To achieve the guarantees of Theorem 2.2, a natural approach would be to apply Theorem 2.3 to the guarantees of CountSketch in Theorem B.1. However, this does not achieve the optimal bounds because each round of adaptive queries can require multiple answers, i.e., estimated frequencies for each of the heavy-hitters at that time. Thus, [CLN+22] proposed a slight variation of the algorithm along with intricate analysis to achieve the guarantees of Theorem 2.2.

Lemma 2.5. *Suppose the number of distinct elements at the beginning of a block is at least $50t$. Let S be the output of* RobustCS *at the beginning of a block. Then conditioned on the correctness of* RobustCS*, S solves the L_p-heavy hitter problem on the entire block.*

Proof. Suppose the number of distinct elements at the beginning of a block is at least $50t$. Let f be the frequency vector at the beginning of the block and let g be the frequency vector at any

https://doi.org/10.52202/079017-0348

intermediate step in the block. Conditioned on the correctness of RobustCS, we have that the estimated frequency $\widehat{f}_i$ of each item i satisfies

$$|\widehat{f}_i - f_i| \leq \frac{t^{1/p}}{100}.$$

Thus if f_i is an $\frac{\varepsilon}{2}$-L_p heavy hitter, then $i \in S$ and conversely if $i \in S$, then $f_i \geq \frac{\varepsilon}{4}\|f\|_p$.

Since each block has length $\ell = \frac{\varepsilon}{100}t^{1/p}$, then $|g_i - f_i| \leq \frac{\varepsilon}{100}t^{1/p}$. Moreover, because the number of distinct elements is at least $50t$, then we have $\|f\|_p \geq 50t^{1/p}$. Therefore if g_i is an ε-L_p heavy hitter, then f_i is an $\frac{\varepsilon}{2}$-L_p heavy hitter, so that $i \in S$. Similarly if $g_i < \frac{\varepsilon}{2}\|g\|_p$, then $f_i < \frac{\varepsilon}{4}\|f\|_p$, so that $i \notin S$. □

Lemma 2.6. *With high probability,* RobustCS *is correct at the beginning of each block of length ℓ.*

Proof. We have $\ell = \frac{\varepsilon}{100} \cdot t^{1/p}$. Note that there are $b = \frac{m}{\ell} = \frac{100m}{\varepsilon t^{1/p}}$ blocks of length ℓ. Thus it suffices to require RobustCS to be robust to b queries in the subroutine AdaptiveHH to achieve correctness at the beginning of each block, with high probability. □

Lemma 2.7. *The total space by the algorithm is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{2.5}}m^{(2p-2)/(4p-3)}\right)$ bits of space.*

Proof. Since DetHH is called with threshold $\frac{\varepsilon}{16}$ for $t = \mathcal{O}\left(m^{p/(4p-3)}\right)$, then the total space by DetHH is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}t^{2-2/p}\right) = \tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}m^{(2p-2)/(4p-3)}\right)$ bits.

We require RobustCS to be robust to b queries in the subroutine AdaptiveHH, thus using space $\tilde{\mathcal{O}}\left(\frac{\sqrt{b}}{\varepsilon^2}\log n \log\frac{nm\lambda}{\delta}\right)$ for $\delta = \frac{1}{\mathrm{poly}(n,m)}$ and

$$\tilde{\mathcal{O}}\left(\sqrt{b}\right) = \tilde{\mathcal{O}}\left(\sqrt{\frac{m}{\varepsilon t^{1/p}}}\right) = \tilde{\mathcal{O}}\left(\varepsilon^{-1/2}m^{(2p-2)/(4p-3)}\right).$$

Similarly, we use $\tilde{\mathcal{O}}\left(\sqrt{b}\right) = \tilde{\mathcal{O}}\left(\varepsilon^{-1/2}m^{(2p-2)/(4p-3)}\right)$ instances of LZeroEst to guarantee robustness against b queries. Each instance of LZeroEst uses $\mathcal{O}\left(\frac{1}{\varepsilon^2}\log^2(nm)\right)$ bits of space. Hence, the overall space is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{2.5}}m^{(2p-2)/(4p-3)}\right)$ bits. □

C Missing Proofs from Section 3

We revisit the guarantee of CountSketch with a different parameterization in this section.

Theorem C.1. *[CCF04] Given $p \in [1,2]$, there exists a one-pass streaming algorithm* CountSketch *that with high probability, reports all $j \in [n]$ for which $(f_j)^p \geq \varepsilon^p F_p$, along with estimations $\widehat{f}_j$, such that $(1-\varepsilon)f_j^p \leq (\widetilde{f}_j)^p \leq (1+\varepsilon)f_j^p$.*

Observe that to provide the guarantees of Theorem C.1, CountSketch would require space $\mathrm{poly}\left(\frac{1}{\varepsilon}, \log n\right)$, rather than quadratic dependency $\frac{1}{\varepsilon}$.

Lemma 3.2. *Let $r \in [R]$ be fixed. Then with probability at least $\frac{9}{10}$, we have that simultaneously for all $j \in U_i^{(r)}$ for which $(f_j)^p \geq \frac{\eta^3 \cdot F_p(U_i^{(r)})}{2^7\gamma\log^2(nm)}$, $H_\ell^{(r)}$ outputs $\widehat{f}_j$ with $\left(1 - \frac{\eta}{8\log(nm)}\right)\cdot(f_j)^p \leq (\widetilde{f}_j)^p \leq \left(1 + \frac{\eta}{8\log(nm)}\right)\cdot(f_j)^p$.*

Proof. The proof follows from Theorem C.1 and the fact that CountSketch is only run on the substream induced by $I_\ell^{(r)}$. □

Proof. Consider casework on $\left\lfloor \log(1+\eta)^\ell - \log\frac{\gamma^2\log(nm)}{\eta^3}\right\rfloor \leq 1$ or $\left\lfloor \log(1+\eta)^\ell - \log\frac{\gamma^2\log(nm)}{\eta^3}\right\rfloor > 1$. Informally, the casework corresponds to whether the frequencies $\left(\widehat{f}_j\right)^p$ in a significant level set are large or not large, i.e., whether they are above the

heavy-hitter threshold before subsampling the universe. Thus if the frequencies are large, then the heavy-hitter algorithm will estimate their frequencies, but if the frequencies are not large, then we must perform subsampling before the items surpass the heavy-hitter threshold.

Suppose $\left\lfloor \log(1+\eta)^\ell - \log \frac{\gamma^2 \log(nm)}{\eta^3} \right\rfloor \leq 1$, so that $\frac{1}{(1+\eta)^{\ell-1}} \geq \frac{\eta^3}{\gamma \log^2(nm)}$. Note that $j \in \Lambda_\ell$ implies $(f_j)^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^\ell}\right)$ and thus $(f_j)^p \geq \frac{\eta^3 \zeta M}{\gamma \log^2(nm)}$. Note that $M \geq F_p$ and thus by Lemma 3.2, we have that with probability at least $\frac{9}{10}$, $H_i^{(r)}$ outputs $\widehat{f_j}$ such that

$$\left(1 - \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p \leq (\widetilde{f_j})^p \leq \left(1 + \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p,$$

as desired.

For the other case, suppose $\left\lfloor \log(1+\eta)^\ell - \log \frac{\gamma^2 \log(nm)}{\eta^3} \right\rfloor > 1$, so that $i = \left\lfloor \log(1+\eta)^\ell - \log \frac{\gamma^2 \log(nm)}{\eta^3} \right\rfloor$. Since $p_i = 2^{1-i}$, then we have that

$$p_i = \frac{2\gamma \log^2(nm)}{(1+\eta)^\ell \eta^3}.$$

Since $j \in \Lambda_i$, we have again $(f_j)^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^\ell}\right)$ and therefore,

$$(f_j)^p \geq \frac{F_p}{4 \cdot (1+\eta)^\ell} \geq \frac{\eta^3}{4\gamma \log^2(nm)} \frac{F_p}{2^{i-1}}.$$

Conditioning on the event $\mathcal{E}_2$, we have $F_p(U_i^{(r)}) \leq \frac{32 F_p}{2^i}$ and thus

$$(f_j)^p \geq \frac{\eta^3}{4\gamma \log^2(nm)} \frac{2F_p}{2^i} \geq \frac{\eta^3}{128\gamma \log^2(nm)} \cdot F_p(U_i^{(r)}).$$

Hence by Lemma 3.2, we have that with probability at least $\frac{9}{10}$, $H_i^{(r)}$ outputs $\widehat{f_j}$ such that

$$\left(1 - \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p \leq (\widetilde{f_j})^p \leq \left(1 + \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p,$$

as desired. □

We now give the correctness guarantees of Algorithm 2.

Lemma C.2. $\mathbf{Pr}\left[\left|\widehat{F_{p,\mathrm{Res}(k)}} - F_{p,\mathrm{Res}(k)}\right| \leq \varepsilon \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}\right] \geq \frac{2}{3}$.

Proof. We would like to show that for each level set ℓ, we accurately estimate its residual contribution D_ℓ. More specifically, we would like to show $|\widehat{D_\ell} - D_\ell| \leq \frac{\eta}{8\log(nm)} \cdot F_p$ for all $\ell \in [L]$. Let g be the residual vector of f with the largest k coordinates in magnitude set to zero. For a level set ℓ, we define the fractional contribution $\phi_\ell := \frac{C_\ell}{\sum_{i \in [n]} (f_i)^p}$. Given an accuracy parameter ε and a stream of length m, we define a level set Λ_ℓ to be *significant* if $\phi_\ell \geq \frac{\varepsilon^2 \eta}{100p \log(nm)}$. Furthermore, we define a level set Λ_ℓ to be *contributing* if $\phi_\ell \geq \frac{\varepsilon \eta}{100p \log(nm)}$. Otherwise, the level set is defined to be $\phi_\ell < \frac{\varepsilon^2 \eta}{100p \log(nm)}$.
.

For a fixed ℓ, we have that $D_\ell = \sum_{j \in \Lambda_\ell} (g_j)^p$, where $j \in \Lambda_\ell$ if $(g_j)^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^\ell}\right)$. On the other hand, for each fixed r, we have that $S_\ell^{(r)}$ is determined using items j whose estimated frequency are in the range $(\widehat{g_j})^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^\ell}\right)$, so it is possible that j could be classified into contributing to Λ_ℓ even if $j \notin \Lambda_\ell$. Hence, we first analyze an "idealized" setting, where each index j is correctly classified across all level sets $\ell \in [L]$. We that we achieve a $(1 + \tilde{\mathcal{O}}(\varepsilon))$-approximation to F_p in the idealized setting and then argue that because we choose ζ uniformly at random, then only approximation guarantee will worsen only slightly but still remain a $(1+\varepsilon)$-approximation to F_p, since only a small number of coordinates will be misclassified and so our approximation guarantee will only slightly degrade.

https://doi.org/10.52202/079017-0348

Idealized setting. For a fixed $r \in [R]$, let $\mathcal{E}_1$ be the event that $|U_i^{(r)}| \le \frac{32n}{2^\ell}$ and let $\mathcal{E}_2$ be the event that $F_p(U_i^{(r)}) \le \frac{32F_p}{2^\ell}$. Note that $M \ge F_p$ and thus conditioned on $\mathcal{E}_1, \mathcal{E}_2$, then by Lemma 3.2, we have that with probability at least $\frac{9}{10}$, $H_i^{(r)}$ outputs $\widehat{f_j}$ such that

$$\left(1 - \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p \le (\widetilde{f_j})^p \le \left(1 + \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p,$$

as desired.

We first show that when $(\widehat{f_j})^p$ is correctly classified for all j into level sets $\ell \in [L]$, then with probability $1 - \frac{1}{\mathrm{poly}(nm)}$, we have that simultaneously for each fixed level set ℓ, $|\widehat{D_\ell} - D_\ell| \le \frac{\eta}{8\log(nm)} \cdot F_p$.

We define $\widehat{D_\ell} = T_\ell \cdot (1+\eta)^\ell$, where T_ℓ is the estimated size of Λ_ℓ, formed by attempting to truncate the top k coordinates across the level sets Γ_ℓ. In particular, we define $\widehat{|\Gamma_\ell|} = \frac{1}{p_i} \operatorname{median}_{r\in[R]} |S_i^{(r)}|$ for $i = \max\left(1, \left\lfloor \log(1+\eta)^\ell - \log\frac{\gamma^2\log(nm)}{\eta^3} \right\rfloor\right)$.

We analyze the expectation and variance of $\widehat{|\Gamma_\ell|}$. Firstly, let $r \in [R]$ be fixed and for each $j \in \Gamma_\ell$, let Y_j be the indicator variable for whether $Y_j \in S_i^{(r)}$. We have

$$\mathbb{E}\left[\frac{1}{p_i} \cdot |\Gamma_\ell \cap S_i^{(r)}|\right] = \frac{1}{p_i} \cdot \sum_{j\in\Gamma_\ell} \mathbb{E}[Y_j] = \frac{1}{p_i} \cdot (p_i \cdot |\Gamma_\ell|) = |\Gamma_\ell|.$$

Similarly, we have

$$\begin{aligned}\mathrm{Var}\left(\frac{1}{p_i} \cdot |\Gamma_\ell \cap S_i^{(r)}|\right) &\le \frac{1}{p_i^2} \cdot \sum_{j\in\Gamma_\ell} \mathbb{E}[Y_j] \\ &= \frac{1}{p_i^2} \cdot (p_i \cdot |\Gamma_\ell|) = \frac{|\Gamma_\ell|}{p_i}.\end{aligned}$$

Because $p_i \ge \min\left(1, \frac{\gamma^2\log^2(nm)}{(1+\eta)^\ell\eta^3}\right)$, then we have that by Chebyshev's inequality,

$$\mathbf{Pr}\left[\left|\frac{1}{p_i} \cdot |\Gamma_\ell \cap S_i^{(r)}| - |\Gamma_\ell|\right| \ge\right] |\Gamma_\ell| \cdot \sqrt{(1+\eta)^\ell\eta^3} \le \frac{1}{10},$$

conditioned on the events $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$.

To analyze the probability of the events $\mathcal{E}_1$ and $\mathcal{E}_2$, recall that in $U_i^{(r)}$, each item is sampled with probability 2^{-i+1}. Hence,

$$\mathbb{E}\left[|U_i^{(r)}|\right] \le \frac{n}{2^{i-1}}, \qquad \mathbb{E}\left[F_p(U_i^{(r)})\right] \le \frac{F_p}{2^{i-1}}.$$

We define $\mathcal{E}_1$ to be the event that $|U_i^{(r)}| \le \frac{32n}{2^i}$. By Markov's inequality, we have $\mathbf{Pr}[E_1] \ge \frac{15}{16}$. Similarly, we define $\mathcal{E}_2$ to be the event that $F_p(U_i^{(r)}) \le \frac{32F_p}{2^i}$. By Markov's inequality, we also have $\mathbf{Pr}[E_2] \ge \frac{15}{16}$. We have that $\Pr \mathcal{E}_3 \mid \mathcal{E}_1 \wedge \mathcal{E}_2 \ge \frac{9}{10}$. Thus by a union bound,

$$\mathbf{Pr}\left[\left|\frac{1}{p_i} \cdot |\Gamma_\ell \cap S_i^{(r)}| - |\Gamma_\ell|\right| \ge\right] |\Gamma_\ell| \cdot \sqrt{(1+\eta)^\ell\eta^3} \le \frac{1}{3}.$$

By Chernoff bounds, we thus have

$$\mathbf{Pr}\left[\left|\widehat{|\Gamma_\ell|} - |\Gamma_\ell|\right| \ge\right] |\Gamma_\ell| \cdot \sqrt{(1+\eta)^\ell\eta^3} \le \mathrm{poly}\left(\frac{\varepsilon}{\log(nm)}\right).$$

Moreover, if level ℓ is significant, then either $p_i = 0$ or $|\Gamma_\ell| \ge \frac{(1+\eta)^\ell}{2\eta^3}$. If $p_i = 0$, then $\widehat{|\Gamma_\ell|} = |\Gamma_\ell|$. Otherwise if $|\Gamma_\ell| \ge \frac{(1+\eta)^\ell}{2\eta^3}$, then with probability at least $1 - \mathrm{poly}\left(\frac{\varepsilon}{\log(nm)}\right)$, we have that simultaneously for all significant levels $\ell \in [L]$, $(1-\eta)|\Gamma_\ell| \le \widehat{|\Gamma_\ell|} \le (1+\eta)|\Gamma_\ell|$ or in other words,

$$\left|\widehat{|\Gamma_\ell|} - |\Gamma_\ell|\right| \le \eta|\Gamma_\ell|.$$

Since we subtract off the top k coordinates in Γ_ℓ to form $\widehat{|\Lambda_\ell|}$ then we also have $\widehat{|\Lambda_\ell|} - \Lambda_\ell \leq \eta|\Gamma_\ell|$. It follows that since $j \in \Lambda_\ell$ for $(g_j)^p \in \left[\frac{\zeta M}{(1+\eta)^{\ell-1}}, \frac{\zeta M}{(1+\eta)^\ell}\right)$, then for $\widehat{D_\ell} = |\Lambda_\ell| \cdot (1+\eta)^\ell$, we have that $|\widehat{D_\ell} - D_\ell| \leq \eta(1+\eta)C_\ell$. Taking the sum over all the significant levels, we see that the error is at most $\sum_{\ell \in [L]} 2\eta C_\ell \leq \frac{\varepsilon^2}{2} \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}$.

Note that the same guarantee holds if level ℓ is insignificant, provided that $|\Gamma_\ell| < \frac{(1+\eta)^\ell}{1000\eta^3}$. On the other hand, if level ℓ is insignificant and $|\Gamma_\ell| < \frac{(1+\eta)^\ell}{1000\eta^3}$. Thus with probability at least $1 - \mathrm{poly}\left(\frac{\varepsilon}{\log(nm)}\right)$, we have that simultaneously for all insignificant levels $\ell \in [L]$, [

$$\widehat{|\Gamma_\ell|} \leq \frac{1}{200\eta^3}.$$

Then we set $\widehat{D_\ell} = 0$, so that $|\widehat{D_\ell} - D_\ell| = D_\ell$. In fact, we observe that the number of items in insignificant level sets can only be at most an η fraction of the items in the contributing level sets beneath them. Since the sum of the contributions of contributing level sets is at most $F_{p,\mathrm{Res}((1-\varepsilon)k)}$, then taking the sum over all the significant levels, we see that the error is at most the contribution of the tail of the insignificant levels, which by definition is at most $\frac{\varepsilon}{2} \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}$.

Randomized boundaries. By Lemma 3.2, we have that conditioned on $\mathcal{E}_3$,

$$\left(1 - \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p \leq (\widetilde{f_j})^p \leq \left(1 + \frac{\eta}{8\log(nm)}\right) \cdot (f_j)^p,$$

independently of the choice of ζ. Because we drawn $\zeta \in [1,2]$ uniformly at random, then the probability that $j \in [n]$ is misclassified is at most $\frac{\eta}{2\log(nm)}$.

If $j \in [n]$ is indeed misclassified, then it can only be classified into either level set $\Gamma_{\ell+1}$ or level set $\Gamma_{\ell-1}$, since $\left(\widehat{f_j}^p\right)$ is a $\left(1 + \frac{\eta}{8\log(nm)}\right)$-approximation to $(f_j)^p$. As a result, a misclassified index induces at most $\eta(f_j)^p$ additive error to the contribution of level set Γ_ℓ and hence at most $\eta(f_j)^p$ additive error to the contribution of level set Λ_ℓ in the residual vector. Therefore, the total additive error across all $j \in [n]$ due to misclassification is at most $\eta \cdot F_p$ in expectation. By Markov's inequality, the total additive error due to misclassification is at most $\frac{\varepsilon}{2} \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}$ with probability at least 0.95. □

Theorem 3.4. *There exists a one-pass streaming algorithm* ResidualEst *that takes an input parameter $k \geq 0$ (possibly upon post-processing the stream) and uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6} \cdot \log^3(nm)\right)$ bits of space to output an estimate $\widehat{F_{p,\mathrm{Res}(k)}}$ with $\mathbf{Pr}\left[\left|\widehat{F_{p,\mathrm{Res}(k)}} - F_{p,\mathrm{Res}(k)}\right| \leq \varepsilon \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}\right] \geq \frac{2}{3}$.*

Proof. Consider Algorithm 1. By Lemma C.2, we have that

$$\mathbf{Pr}\left[\left|\widehat{F_{p,\mathrm{Res}(k)}} - F_{p,\mathrm{Res}(k)}\right| \leq \varepsilon \cdot F_{p,\mathrm{Res}((1-\varepsilon)k)}\right] \geq \frac{2}{3}.$$

It thus remains to analyze the space complexity.

Note that Algorithm 1 implements $P \cdot R$ instances of CountSketch with accuracy η^3, for $P = \tilde{\mathcal{O}}(\log(nm))$, $R = \tilde{\mathcal{O}}\left(\log \frac{\log n}{\eta}\right)$, and $\eta = \frac{\varepsilon}{100}$. By Theorem C.1, each instance of CountSketch with threshold η^3 uses $\mathcal{O}\left(\frac{1}{\eta^6} \cdot \log^2(nm)\right)$ bits of space. Therefore, the total space usage of Algorithm 1 is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6} \cdot \log^3(nm)\right)$ bits. □

D Missing Proofs from Section 4

We first show that the p-th moment of f can be essentially split by looking at the p-th moment of the vector consisting of the largest k coordinates and the remaining tail vector.

https://doi.org/10.52202/079017-0348

Lemma D.1. *Let $\varepsilon \in (0,1)$ be a fixed accuracy parameter and let $p > 0$ be fixed. Let $f \in \mathbb{R}^n$ be any fixed vector and let $k \geq 0$ be any fixed parameter. Let g be the vector consisting of the k coordinates of f largest in magnitude and let h be the residual vector, so that $f = g + h$. Suppose $\widehat{G}$ and $\widehat{H}$ satisfy*

$$\|g\|_p^p - \frac{\varepsilon}{4}\|f\|_p^p \leq \widehat{G} \leq \|g\|_p^p + \frac{\varepsilon}{4}\|f\|_p^p$$
$$\|h\|_p^p - \frac{\varepsilon}{4}\|f\|_p^p \leq \widehat{H} \leq \|h\|_p^p + \frac{\varepsilon}{4}\|f\|_p^p.$$

Then

$$\left(1 - \frac{\varepsilon}{2}\right)\|f\|_p^p \leq \widehat{G} + \widehat{H} \leq \left(1 + \frac{\varepsilon}{2}\right)\|f\|_p^p.$$

Proof. The claim follows immediately from the fact that $\|f\|_p^p = \|g\|_p^p + \|h\|_p^p$ since $f = g + h$ but g and h have disjoint support. □

In fact, we show the estimation is relatively accurate even if the tail vector does not quite truncate the k entries largest in magnitude.

Lemma D.2. *Let f be a frequency vector, g be the residual vector omitting the k coordinates of f largest in magnitude, and h be the residual vector omitting the $\left((1 - \frac{\varepsilon}{4})\right) k$ coordinates of f largest in magnitude. Then we have $|\|g\|_p^p - \|h\|_p^p| \leq \frac{\varepsilon}{4} \cdot \|f\|_p^p$.*

Proof. Note that the smallest $\frac{\varepsilon}{4}$ coordinates of the top k coordinates is only nonzero when $k \geq \frac{4}{\varepsilon}$. Thus they can only contribute $\frac{\varepsilon}{4}$ fraction to the entire moment. It follows that $|\|g\|_p^p - \|h\|_p^p| \leq \frac{\varepsilon}{4} \cdot \|f\|_p^p$, as desired. □

Lemma 4.2. *Let f be a frequency vector and g be the residual vector omitting the k coordinates of f largest in magnitude. Let v be any arbitrary vector such that $\|v\|_1 \leq \frac{\varepsilon}{100} \cdot \|g\|_p \cdot k^{1-1/p}$ and $\|v\|_1 \leq \frac{1}{2}\|g\|_1$. Let u be the residual vector omitting the k coordinates of $f + v$ largest in magnitude. Then we have $|\|g\|_p^p - \|u\|_p^p| \leq \frac{\varepsilon}{4} \cdot \|g\|_p^p$.*

Proof. Let $\|g\|_p^p = M$. Since $\|v\|_1 \leq \frac{1}{2}\|g\|_1$, then by an averaging argument $|u_i|$ can be at most $\left(\frac{8M}{k}\right)^{1/p}$ before i is in the top k coordinates of $f + v$. Similarly, if $i \in [n]$ is in the top k coordinates of f for $|v_i|$ less than $\left(\frac{8M}{k}\right)^{1/p}$, and i is no longer in the top k coordinates of $f + v$, then we must have $|u_i| \leq \left(\frac{16M}{k}\right)^{1/p}$. Otherwise by an averaging argument, $|u_i|$ would be too large and i would be in the top k coordinates of $f + v$.

Thus the contribution to $|\|g\|_p^p - \|u\|_p^p|$ is at most the contribution in the case where the number of coordinates i with $|v_i| = \left(\frac{8M}{k}\right)^{1/p}$ is maximized. Because $\|v\|_1 \leq \frac{\varepsilon}{100} \cdot M \cdot k^{1-1/p}$, then there can be at most $\frac{\varepsilon}{100} \cdot k$ coordinates $i \in [n]$ such that $|v_i| \geq \left(\frac{8M}{k}\right)^{1/p}$. By the above argument, for each i, we have $||g_i|^p - |u_i|^p| \leq \frac{16M}{k}$. Since there can be at most $\frac{\varepsilon}{100} \cdot k$ such coordinates, then the total change in the p-th moment of residual is at most $16M \cdot \frac{\varepsilon}{100} \cdot k \leq \frac{\varepsilon}{4} \cdot M$. The desired result then follows from the recollection that $\|g\|_p^p = M$. □

We now show the correctness of Algorithm 3.

Lemma D.3. *For any fixed time during a stream, let f be the induced frequency vector and let $\widehat{F}$ be the output of Algorithm 3. Then we have that with high probability,*

$$(1 - \varepsilon)\|f\|_p^p \leq \widehat{F} \leq (1 + \varepsilon)\|f\|_p^p.$$

Proof. Consider the first time t in a block of ℓ updates and let f be the frequency vector induced by the stream up to that point. We first observe that RobustHH with threshold $\varepsilon\eta$ will return any coordinates $i \in [n]$ such that $f_i \geq \varepsilon^p \eta^p \cdot \|f\|_p^p$ up to $(1 + \varepsilon)$-approximation. For the remaining coordinates in the k-sparse vector returned by RobustHH, any k of them can contribute at most $\varepsilon^p \cdot \|f\|_p^p$. Therefore, we have by Lemma D.1 that conditioned on the correctness of RobustHH

and ResidualEst, we have $\widehat{G}+\widehat{H}$ is a $(1+\mathcal{O}(\varepsilon))$-approximation to $\|f\|_p^p$. For the purposes of notation, let h denote the residual vector of f at time t, omitting the k coordinates of f largest in magnitude.

Now, consider some later time t' in the same block of ℓ updates and let v be the frequency vector induced by the updates in the block, i.e., the updates from t to t'. Let u be the residual vector omitting the k coordinates of $f+v$ largest in magnitude. Since $\|v\|_1 \leq \ell$ for $\ell = \mathcal{O}\left(\varepsilon \cdot m^{c/p} k^{1-1/p}\right)$, then by Lemma 4.2, we have that $|\|h\|_p^p - \|u\|_p^p| \leq \frac{\varepsilon}{4} \cdot \|h\|_p^p$. Thus provided that $\widehat{H}$ is a $(1+\mathcal{O}(\varepsilon))$-approximation to $\|h\|_p^p$, then it remains a $\left(1+\frac{\varepsilon}{4}\right)$-approximation to $\|u\|_p^p$. Hence conditioned on the correctness again of RobustHH at time t', we have that $\widehat{H}+\widehat{H}$ remains a $(1+\varepsilon)$-approximation to $\|f\|_p^p$ at time t.

As correctness of RobustHH follows from Theorem 1.2, it remains to show correctness of ResidualEst on an adaptive stream. Because each block has size ℓ, then the stream has at most $\frac{m}{\ell}$ such blocks. Hence by the adversarial robustness of differential privacy, i.e., Theorem 2.3, it suffices to run $\tilde{\mathcal{O}}\left(\frac{\sqrt{m}}{\ell}\right)$ copies of ResidualEst to guarantee correctness with high probability at all times. □

Finally, we analyze the space complexity of our algorithm.

Lemma D.4. *For $\log n = \Theta(\log m)$, Algorithm 3 uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{7.5}} \cdot m^c\right)$ bits of space in total.*

Proof. We observe that Algorithm 3 uses a few main subroutines. Firstly, it runs SparseRecover with sparsity $\mathcal{O}(m^c)$, which requires $m^c \cdot \operatorname{polylog}(nm)$ bits of space, by Theorem 4.1. Next, it runs RobustHH with threshold $\varepsilon\eta$, for $\eta = \frac{\varepsilon^2}{100m^\gamma}$. By Theorem 1.2, RobustHH uses $\tilde{\mathcal{O}}\left(\frac{1}{(\varepsilon\eta)^{2.5}} m^{(2p-2)/(4p-3)}\right)$ bits of space. Note that for our choice of $\gamma = \frac{2c}{5} - \frac{(4p-4)}{(20p-15)}$, we have $\tilde{\mathcal{O}}\left(\frac{1}{(\varepsilon\eta)^{2.5}} m^{(2p-2)/(4p-3)}\right) = \tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{7.5}} \cdot m^c\right)$ bits of space. Finally, it runs $\tilde{\mathcal{O}}\left(\frac{\sqrt{m}}{\ell}\right)$ copies of LZeroEst and ResidualEst. By Theorem 3.4, each instance of ResidualEst uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6} \cdot \log^3(nm)\right)$ bits of space. By Theorem 2.4, each instance of LZeroEst uses $\mathcal{O}\left(\log^2(nm) \log\log m\right)$ bits of space. Since $\ell = \mathcal{O}\left(\varepsilon \cdot m^{c/p} k^{1-1/p}\right)$, then we have $\tilde{\mathcal{O}}\left(\frac{\sqrt{m}}{\ell}\right) = \tilde{\mathcal{O}}(m^c)$ and thus the total space usage by these subroutines is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^6} \cdot m^c\right)$ bits The desired claim then follows by noting that across all procedures, the space usage is $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{7.5}} \cdot m^c\right)$, due to our balancing choices of ℓ, γ, and c. □

Lemma 4.3. *For $\log n = \Theta(\log m)$, Algorithm 3 uses $\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{7.5}} \cdot m^c\right)$ bits of space in total. Moreover, for any fixed time during a stream, let f be the induced frequency vector and let $\widehat{F}$ be the output of Algorithm 3. Then we have that with high probability, $(1-\varepsilon)\|f\|_p^p \leq \widehat{F} \leq (1+\varepsilon)\|f\|_p^p$.*

Proof. Note that Lemma 4.3 follows from Lemma D.3 and Lemma D.4. □

Putting things together, we get the full guarantees of our main result:

Theorem 1.3. *Let $p \in [1,2]$ and $c = \frac{24p^2-23p+4}{(4p-3)(12p+3)}$. There exists a streaming algorithm that uses $\mathcal{O}(m^c) \cdot \operatorname{poly}\left(\frac{1}{\varepsilon}, \log(nm)\right)$ bits of space and outputs a $(1+\varepsilon)$-approximation to the L_p norm of the underlying vector at all times of an adversarial stream of length m.*

Proof. Observe that the correctness stems from Lemma D.3, while the space complexity follows form Lemma D.4. □

Broader Impact Statement

As adversarial robustness can have applications to many applications in machine learning, a potential broader impact of our work is the advancement of the theoretical foundations of trustworthy machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

https://doi.org/10.52202/079017-0348

NeurIPS Paper Checklist

(1) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Yes, the abstract and introduction accurately reflect the claims made, including the contributions made in the paper, as well as important assumptions and limitations.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(2) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Yes, the theorem statements in the paper formally describe the limitations of our theoretical results.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(3) **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, the paper provides the complete proofs in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(4) **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, the paper fully discloses the information needed to reproduce the main experimental results of the paper, including information about the code and datasets in the full version.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(5) **Open access to data and code**

https://doi.org/10.52202/079017-0348

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, open access to the data and code are provided in the full version of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(6) **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, all the testing parameters are described in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

(7) **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, the paper provides the significant statistics of our experiments, which are deterministic.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(8) **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes, the computing resources are described in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(9) **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Yes, the paper conforms in every respect with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(10) **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes, we address the potential broader impacts in the appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-0348

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(11) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our dataset was acquired from a publicly available repository and we do not introduce new datasets in this paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(12) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All original assets used in this paper have been referenced appropriately.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(13) **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Yes, the provided code is well-documented.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(14) **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(15) **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0348