A Strong Separation for Adversarially Robust ℓ_0 Estimation for Linear Sketches

Elena Gribelyuk
Department of Computer Science
Princeton University
Princeton, NJ, USA
eg5539@princeton.edu

Honghao Lin
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
honghaol@andrew.cmu.edu

David P. Woodruff
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
dwoodruf@andrew.cmu.edu

Huacheng Yu
Department of Computer Science
Princeton University
Princeton, NJ, USA
hy2@cs.princeton.edu

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

***Abstract*—The majority of streaming problems are defined and analyzed in a static setting, where the data stream is any worst-case sequence of insertions and deletions which is fixed in advance. However, many real-world applications require a more flexible model, where an adaptive adversary may select future stream elements after observing the previous outputs of the algorithm. Over the last few years, there has been increased interest in proving lower bounds for natural problems in the adaptive streaming model. In this work, we give the first known adaptive attack against linear sketches for the well-studied ℓ_0-estimation problem over turnstile, integer streams. For any linear streaming algorithm $\mathcal{A}$ which uses sketching matrix $\mathbf{A} \in \mathbb{Z}^{r\times n}$, this attack makes $\tilde{\mathcal{O}}(r^8)$ queries and succeeds with high constant probability in breaking the sketch. Additionally, we give an adaptive attack against linear sketches for the ℓ_0-estimation problem over finite fields $\mathbb{F}_p$, which requires a smaller number of $\tilde{\mathcal{O}}(r^3)$ queries. Finally, we provide an adaptive attack over $\mathbb{R}^n$ against linear sketches $\mathbf{A} \in \mathbb{R}^{r\times n}$ for ℓ_0-estimation, in the setting where $\mathbf{A}$ has all nonzero subdeterminants at least $\frac{1}{\text{poly}(r)}$. Our results provide an exponential improvement over the previous number of queries known to break an ℓ_0-estimation sketch.**

***Index Terms*—streaming, sketching, adversarial robustness.**

Elena Gribelyuk and Huacheng Yu are supported in part by an NSF CAREER award CCF-2339942. Honghao Lin was supported in part by a Simons Investigator Award, NSF CCF-2335412, and a CMU Paul and James Wang Sercomm Presidential Graduate Fellowship. 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 authors would like to acknowledge the Sketching and Algorithm Design workshop held at the Simons Institute for the Theory of Computing for contributions to the development of this work.

I. Introduction

In the classical streaming model, updates to an underlying dataset arrive sequentially and the goal is to compute or approximate some predetermined statistic of the dataset while using space sublinear in m, the length of the stream and n, the dimension of the underlying dataset; ideally, the algorithm should provide this estimate after making only a single pass over the data. The streaming model of computation captures key memory and resource requirements of algorithms in many big data applications, and has therefore emerged as a central paradigm for applications where the size of the data is significantly larger than the available storage, such as logs generated from either virtual or physical traffic monitoring, stock market transactions, scientific observations, and machine and sensor data, e.g., Internet of Things (IoT) sensors, financial markets, and scientific observations.

Observe that in many of these applications, intermediate outputs of the algorithm may impact the distribution of future inputs to the algorithm. For example, in database systems, future queries to the database may be dependent on the full history of responses by the database algorithm to previous queries. In optimization procedures such as stochastic gradient descent, the update at each time step can be based on the history of previous outputs. In recommendation systems, a user may choose to remove some suggestions based on personal preference and then query for a new list of recommendations. Additionally, statistics aggregated from financial markets on the current day could result in algorithmic decisions that impact certain enterprises, thereby affecting their future evaluations, which form a small but nonzero component of the information collected by the algorithm on the next day.

Unfortunately, the classical *oblivious* streaming model assumes that the input is fixed in advance to be the worst possible permutation of elements. Moreover, since the algorithm only provides an estimate *once* at the end of the stream, we may assume that the input stream is independent of the internal randomness of the streaming algorithm. Indeed, the analyses of many randomized streaming algorithms crucially utilize the independence between the internal randomness of the algorithm and the data stream. However, as discussed previously, this may not be a reasonable assumption for the above applications and many additional settings [MNS11], [BMSC17], [NY19], [AMYZ19], [CN20], [CSW+23], [CLN+22], [CNSS23], [DSWZ23], [WZZ23], [CA24]. This motivates the *adversarially robust* streaming model, which we discuss next.

a) The adversarially robust streaming model: To address these shortcomings of the classical oblivious streaming model, the adversarially robust streaming model was recently proposed [BJWY22] to capture settings where the sequence of inputs to the streaming algorithm can be adaptive or even adversarial. At each time $t \in [m]$, the streaming algorithm $\mathcal{A}$ receives an update $u_t = (a_t, \Delta_t)$, where each $a_t \in [n]$ is an index and $\Delta_t \in \mathbb{Z}$ denotes an increment or decrement to index a_t in the underlying frequency vector $\mathbf{x}$, i.e., the i^{th} coordinate of the frequency vector is given by $\mathbf{x}_i = \sum_{t:a_t=i} \Delta_t$. Similarly, let $\mathbf{x}^{(t)}$ denote the state of the frequency vector restricted to the first t updates, i.e., $\mathbf{x}_i^{(t)} = \sum_{s \le t:a_s=i} \Delta_s$. We consider the setting where $m = \text{poly}(n)$ and $|\Delta_t| \le \text{poly}(n)$ for all $t \in [m]$. Note that by scaling, we could have also assumed that each Δ_t is an integer multiple of $\frac{1}{\text{poly}(n)}$. Then, $\mathcal{A}$ is an *adversarially robust* streaming algorithm for some estimation function $g : \mathbb{Z}^n \to \mathbb{R}$ if $\mathcal{A}$ satisfies the following requirement.

Definition I.1. *[BJWY22] Let $g : \mathbb{Z}^n \to \mathbb{R}$ be a fixed function. Then, for any $\varepsilon > 0$ and $\delta > 0$, at each time $t \in [m]$ for $m = \text{poly}(n)$, we require our algorithm $\mathcal{A}$ to return an estimate z_t for $g(\mathbf{x}^{(t)})$ such that*

$$\Pr\left[\left|z_t - g(\mathbf{x}^{(t)})\right| \le \varepsilon g(\mathbf{x}^{(t)})\right] \ge 1 - \delta$$

The above definition is also known as the *strong tracking guarantee* for adversarial robustness, as defined in [BJWY22]. Moreover, we may view the adversarial setting as a two-player game between a randomized streaming algorithm $\mathcal{A}$ and an unbounded adversary. In particular, the adversary aims to construct a hard sequence of adaptive[1] updates $\{u_1^*, ..., u_m^*\}$ such that any streaming algorithm $\mathcal{A}$ that produces (ε, δ)-approximate responses $\{z_t\}_{t=1}^m$ will fail to estimate $g(\mathbf{x}_{t^*})$ with constant probability at some step $t^* \in [m]$ during the stream. For a chosen function g, the game proceeds as follows:

[1]Note that we use "adaptive" and "adversarial" interchangeably: both terms indicate that future updates or queries may depend on previous updates and responses of the algorithm.

(1) In each round $t \in [m]$, the adversary selects an update u_t to append to the stream to implicitly define the underlying dataset $\mathbf{x}^{(t)}$ at time t. Importantly, note that $\mathbf{x}^{(t)}$ may depend on all previous updates $u_1, ..., u_{t-1}$, as well as the corresponding responses $z_1, ..., z_{t-1}$ of the streaming algorithm $\mathcal{A}$.
(2) $\mathcal{A}$ receives update u_t and updates its internal state.
(3) Then, $\mathcal{A}$ returns an estimate $z_t(\mathbf{x}^{(t)})$ for $g(\mathbf{x}^{(t)})$ based on the stream observed until time t, and progresses to the next round.

Observe that this sequential game only permits a single pass over the data stream. Alternatively, the adversary may choose to only query the streaming algorithm at specific times during the stream. In future sections, we will let $\mathbf{x}^{(t)}$ denote the query vector at time t, which may have been formed by a sequence of $\mathcal{O}(n)$ insertions or deletions to various indices of the previous query vector $\mathbf{x}^{(t-1)}$.

b) Insertion-only streams: In the insertion-only streaming model, each update $u_t = (a_t, \Delta_t)$ represents an *insertion* of an element $a_t \in [n]$ into the stream $\Delta_t > 0$ times. This corresponds to incrementing the (a_t)-th coordinate of the underlying frequency vector $x_{a_t} = x_{a_t} + \Delta_t$. In the special case that the increments $\Delta_t = 1$ in each step $t \in [m]$, the (a_t)-th coordinate of $\mathbf{x}$ is simply the number of times that element a_t appeared in the stream.

In the adversarially robust streaming model with insertion-only updates, it is known that many central streaming problems admit sublinear space algorithms, c.f., [HKMM20], [BHM+21], [WZ21], [ABJ+22], [BJWY22], [CGS22], [BKM+22], [JPW22], [ACGS23], [ACSS23]. In particular, [BHM+21] showed that by using the popular merge-and-reduce framework, adversarial robustness is essentially built into the analysis for a wide class of problems such as clustering, subspace embeddings, linear regression, and graph sparsification. In other words, there exist adversarially robust algorithms for these problems that use the same sampling-based approach as classical streaming algorithms in the case where the inputs must be insertion-only. Similarly, [WZ21] showed that for fundamental problems such as norm and moment estimation, distinct elements estimation, heavy-hitters, and entropy estimation, there exist adversarially robust algorithms that pay a small polylogarithmic overhead over the classical insertion-only streaming algorithms that use sublinear space.

c) Turnstile streams.: There is significantly less known about adversarially robust streaming algorithms with turnstile updates. The work of [BJWY22] gives an algorithm that uses space sublinear in the stream length m in the case that the stream has *bounded deletions*. However, in the general turnstile streaming setting, the best known adaptive upper bounds are still much worse than in the oblivious case. The work of [HKMM20] showed a way to use differential privacy to protect the internal randomness of the streaming algorithm from the adversary: this framework converts an oblivious streaming al-

gorithm for estimation problem f into an adversarially robust streaming algorithm for the same problem, with an $\tilde{\mathcal{O}}(\sqrt{m})$ blow-up in the space complexity for turnstile streams[2]. More recently, the work of [BEO22] gave an adversarially robust streaming algorithm for F_p estimation by combining the differential privacy framework of [HKMM20] with standard results from sparse recovery; for ℓ_0 estimation, this reduced the space blow-up to $\tilde{\mathcal{O}}\left(m^{1/3}\right)$. Still, when the stream length m is a sufficiently large polynomial of the dimension n of the frequency vector, the space complexity of the algorithm is not sublinear in n.

A natural question is whether there is an inherent space-complexity separation between oblivious and adaptive streaming. To this end, [HW13] showed that no linear sketch can approximate the ℓ_2 norm within even a polynomial multiplicative factor against adaptive queries when the sketching matrix and input stream are both real-valued. First, a natural idea is to try to adapt the attack therein to obtain an attack against linear sketches for ℓ_p-estimation in the integer setting. However, the attack of [HW13] crucially relies on Gaussian rotational invariance to argue that the algorithm's observations can be parametrized solely by the norms of the inputs. It is not clear whether it is possible to discretize the Gaussian queries of their attack, as the direction in the sketch space may still reveal some information about the norm. Secondly, we remark that [HW13] also cannot handle the case of ℓ_0-estimation over the reals, since ℓ_0 is not a norm (since the attack requires $\|Cx\| = C\|x\|$ for scalars $C > 0$). Thus, an entirely different approach is needed to handle ℓ_0-estimation over the integers.

Additionally, in 2021, [KMNS21] showed that there exists a streaming problem for which there is an exponential separation in the space complexity needed to solve the problem in the oblivious and adaptive settings; specifically, this lower bound is shown for a streaming version of the adaptive data analysis problem in the bounded-storage model of computation. Later, the work of [CGS22] noted an elegant quadratic separation between oblivious and adaptive streaming for the minimum spanning forest problem over streams with edge insertions and deletions. Subsequently, [CGS22] showed a separation for oblivious and adaptive streams for insertion-only streams for the problem of graph-coloring. Thus, a well-known open problem [KMNS21], [BJWY22], [Wor21], [Wor23] is the following:

> *Is there a separation between oblivious and adaptive turnstile streaming for any natural "statistical" streaming estimation problem?*

We make progress toward answering this question in the affirmative, as we show a lower bound against linear sketches for ℓ_0 estimation in the adversarial streaming model.

[2]We use the notation $\tilde{\mathcal{O}}(f)$ to represent $f \cdot \text{polylog}(f)$.

d) ℓ_0-estimation problem and linear sketching in the adversarial streaming model.: In this work, we study the classical streaming problem of estimating the number of distinct elements in a turnstile stream, also known as the ℓ_0-estimation problem, where $\|x\|_0 = |\{i : x_i \neq 0\}|$. Given a stream of updates $u_1 = (a_1, \Delta_1), ..., u_m = (a_m, \Delta_m)$, let $a_t \in [n]$ be an index and let $\Delta_t \in \mathbb{Z}$ denote an increment or decrement to index a_t of the underlying frequency vector $\mathbf{x} \in \mathbb{Z}^n$, where $|\Delta_t| \leq \text{poly}(n)$. The task of the streaming algorithm $\mathcal{A}$ is to produce an estimate z such that $\Pr\left[|z - \|\mathbf{x}\|_0| \leq \varepsilon\|\mathbf{x}\|_0\right] \geq 1 - \delta$ for any $\varepsilon, \delta > 0$ fixed in advance. The ℓ_0-estimation problem has been studied extensively in the last 40 years, beginning with the seminal work of (Flajolet and Martin, FOCS, 1983) [FM85]. The frequency moment estimation problem has since been studied in many other works [BJK+02], [IW05], [KNW10], [KNPW11], culminating in a nearly optimal algorithm for ℓ_0-estimation in turnstile streams of [KNW10], which succeeds with high constant probability and gives a $(1\pm\varepsilon)$-approximation using $\mathcal{O}\left(\varepsilon^{-2}\log(n)\left(\log\frac{1}{\varepsilon} + \log\log n\right)\right)$ bits of space.

Moreover, we focus on the case that $\mathcal{A}$ is a *linear* streaming algorithm, meaning that $\mathcal{A}$ samples a sketching matrix $\mathbf{A} \sim \mathcal{S}$, and for any input $\mathbf{x} \in \mathbb{Z}^n$, $\mathcal{A}$ returns $f(\mathbf{A}, \mathbf{Ax})$, where f is any function. It is important to note that for long enough streams, all known turnstile streaming algorithms are linear sketches, and in fact, it is known that when the stream length is long enough, turnstile streaming algorithms with fixed inputs $\mathbf{x}$ can be captured by maintaining a linear sketch $\mathbf{Ax}$ over the course of the stream [LNW14], [AHLW16], [KP20]. Motivated by the reasons above, we focus on proving lower bounds against linear sketches for the ℓ_0-estimation problem in the adaptive streaming setting.

A. Our Results

We resolve the open problem posed above by giving the first known adaptive attack against linear sketches for the turnstile ℓ_0-estimation problem over the integers. Our results are derived from the following promise problem.

Definition I.2 (ℓ_0 gap norm problem)**.** *Let $0 \leq \alpha < \beta \leq 1$. We say that an algorithm $\mathcal{A}$ solves the (α, β)-ℓ_0 gap norm problem if, for any input $x \in \mathbb{Z}^n$, $\mathcal{A}$ outputs 0 if $\|x\|_0 \leq \alpha n$ and outputs 1 if $\|x\|_0 \geq \beta n$. If $\|x\|_0$ satisfies neither of these conditions, $\mathcal{A}$ may return either 0 or 1.*

Furthermore, we focus our attention on linear streaming algorithms, defined as follows:

Definition I.3 (Linear streaming algorithm)**.** *Let $\mathcal{A}$ be a streaming algorithm for a function g, and let $A \in \mathbb{Z}^{r\times n}$ be a sketching matrix of its choice, sampled from some distribution $A \sim \mathcal{S}$ over sketching matrices. We say that a streaming algorithm $\mathcal{A}$ is linear if, for every update $x \in \mathbb{Z}^n$, $\mathcal{A}$ observes Ax and returns an estimate $f(A, Ax)$, where f is any function.*

In all of our results, we assume that the dimensions of the sketching matrix $\mathbf{A}$ satisfy $r \ll n$.

Theorem I.4 (Informal version of Theorem IV.1)**.** *Suppose that $\mathcal{A}$ is a linear streaming algorithm that solves the $(\alpha + c, \beta - c)$- ℓ_0 gap norm problem for some constants α, β, c. Then there exists a randomized adversary that, with high constant probability can generate a distribution D over $\mathbb{Z}^n$ such that $\mathcal{A}$ fails on D with constant probability. This adaptive attack makes at most $\tilde{\mathcal{O}}\left(r^8\right)$ queries and runs in* $\mathrm{poly}(r)$ *time.*

This result has implications beyond adversarial streaming. In particular, since the existence of a so-called pseudodeterministic streaming algorithm for a particular task implies the existence of adversarially robust streaming algorithm for the same task, our attack implies that any linear pseudodeterministic algorithm for the turnstile ℓ_0-estimation over the integers can be made to fail after $\mathrm{poly}(r)$ adaptive queries. This relates to open questions raised in [GGMW20], which asked whether there can be linear pseudodeterministic streaming algorithms for the ℓ_2 estimation problem.

Next, we give an attack against linear sketches for ℓ_0-estimation where all entries of the sketching matrix $\mathbf{A}$ and input $\mathbf{x}$ are over $\mathbb{F}_p$ for some prime p. We note that known ℓ_0 sketches can also be adapted to work over such fields with minimal changes (see, e.g., footnote 2 of [MRU11]).

Theorem I.5 (Informal version of Theorem VI.1)**.** *Suppose $\mathcal{A}$ is a linear streaming algorithm that solves the $(\alpha + c, \beta - c) - \ell_0$ gap norm problem with some constants α, β, and c. There exists an adaptive attack that makes $\tilde{\mathcal{O}}\left(r^3\right)$ queries and with high constant probability outputs a distribution D over $\mathbb{Z}^n$ such that when $\mathbf{x} \sim D$, $\mathcal{A}$ fails to decide the ℓ_0 gap norm problem with constant probability.*

Finally, we give an attack against linear sketches with real entries in the case that sketching matrix $\mathbf{A} \in \mathbb{R}^{r \times n}$ has all nonzero subdeterminants at least $\frac{1}{\mathrm{poly}(r)}$. We remark that this is a natural class of sketching matrices to consider, as the known sketches have this property.

Theorem I.6 (Informal version of Theorem VII.1)**.** *Suppose that $\mathcal{A}$ is a linear streaming algorithm that solves the $(\alpha+c, \beta-c)$-ℓ_0 gap norm problem with some constants α, β and c, where $\mathbf{A} \in \mathbb{R}^{r \times n}$ is the sketching matrix such that $\mathbf{A}$ has all nonzero subdeterminants at least $\frac{1}{\mathrm{poly}(r)}$. Then there exists a randomized algorithm, which after making an adaptive sequence of queries to $f(\mathbf{A}, \mathbf{Ax})$, with high constant probability can generate a distribution D on $\mathbb{R}^n$ such that $f(\mathbf{A}, \mathbf{Ax})$ fails on D with constant probability. Moreover, this adaptive attack algorithm makes at most* $\mathrm{poly}(r)$ *queries and runs in* $\mathrm{poly}(r)$ *time.*

This attack serves as a proof-of-concept and as further motivation for our fingerprinting-based techniques. Additionally, in a recent work on adversarially-robust property-preserving hash functions [BLV18], it was conjectured that there is an efficient adaptive attack against linear sketches for ℓ_0-estimation over the reals; our attack resolves this question for the class of sketching matrices with not-too-small subdeterminants.

B. Technical Overview

In this section, we give a description of the attack against linear sketches for the ℓ_0-estimation problem.

As the "adaptive adversary", the primary goal of our attack is to gradually learn the sketching matrix $\mathbf{A}$, and design "harder" queries as more of $\mathbf{A}$ becomes known to us. A sketching matrix $\mathbf{A}$ may preserve a "significant amount of information" about some coordinates x_i in $\mathbf{Ax}$ (e.g., when there is a row of $\mathbf{A}$ that is nonzero only in column i, $\mathbf{Ax}$ can recover x_i precisely), while it only mildly "depends on" the other coordinates (e.g., when a coordinate i is always "mixed" in a sum of many coordinates). The coordinates that $\mathbf{A}$ preserves a significant information about, or the *significant coordinates*, can be very useful for estimating the ℓ_0-norm when the queries are non-adaptive. For example, one may sample $\mathbf{A}$ in a careful way such that a random set of $\mathcal{O}(1)$ coordinates is significant, and from $\mathbf{Ax}$, one can approximately identify whether each of them is zero. Then, just based on the fraction of non-zeroes among these sampled coordinates, the ℓ_0-norm can already be approximated up to an additive error of, say $0.1n$, solving ℓ_0 gap norm.

Thus, our main strategy is to gradually identify the significant coordinates, and set them to zero in all future queries as soon as we find any.[3] This makes the future queries *harder for* $\mathbf{A}$, since intuitively, $\mathbf{A}$ would be wasting some of its budget on a coordinate that is always zero, effectively reducing its dimension. When the number of rows $r \ll n$, $\mathbf{A}$ cannot simultaneously preserve a significant amount of information for too many x_i's. After we have learned all such coordinates, the query algorithm would have to only rely on the other *insignificant coordinates*, which $\mathbf{A}x$ only mildly depends on.

In order to perform such attacks, there are three main problems to solve:

- define "significance" and show that not too many coordinates are significant when $r \ll n$;
- show that we can learn which coordinates are significant using polynomially many queries;
- show that the query algorithm cannot estimate the ℓ_0-norm accurately when $\mathbf{x}$ is supported only on the insignificant coordinates. In fact, we will design distributions for $\mathbf{x}$ with very different ℓ_0-norms, such that the impact of the insignificant coordinates on the sketch $\mathbf{Ax}$ is nearly identical regardless of the ℓ_0-norm.

In the following, we elaborate on how we solve the above problems.

[3] In fact, zeroing out these coordinates after we learn them is the only type of adaptive move in our attack.

a) Fingerprinting codes: First let us see how we should learn the significant coordinates. While we have not formally defined "significant coordinates" yet, for now let us focus on an *important* extreme case: the sketch $\mathbf{Ax}$ is simply an (unknown) subset of r coordinates of $\mathbf{x}$, i.e., each row of $\mathbf{A}$ is a unit vector with one 1 in some column and zero elsewhere. These r unknown coordinates are (very) significant, and all other coordinates are (completely) insignificant.

It turns out that this case is exactly what an *interactive fingerprinting code* can solve. In the interactive fingerprinting code problem [SU15], an algorithm $\mathcal{P}$ selects a set of coordinates $\mathcal{S} \subset [n]$ with $|\mathcal{S}| = k$, which is unknown to the fingerprinting code $\mathcal{F}$.[4] Then, the goal of $\mathcal{F}$ is to discover the set $\mathcal{S}$ by making adaptive queries $c^t \in \{\pm 1\}^n$ at each time t, and enforcing the requirement that $\mathcal{P}$ must return an answer a^t that is consistent with *some* coordinate in c^t, i.e., $a^t = c_i^j$ for some $i \in [n]$. Equivalently, this is for $\mathcal{P}$ to distinguish between $c^t = (-1, \ldots, -1)$ and $(1, \ldots, 1)$. Importantly, we also impose the constraint that $\mathcal{P}$ can only observe the coordinates c_i^t for $i \in \mathcal{S}$. The attack then proceeds by assigning a score s_i^t to each index $i \in [n]$ at every round $t \in [\ell]$, which corresponds to a measure of the *correlation* between values of the i-th index $(c_i^1, \ldots, c_i^t)$ and the responses $(a^1, \ldots, a^t)$ given by $\mathcal{P}$ during the first t rounds. It has been shown in [SU15] that even under the weak requirement of outputting -1 when $c^t = (-1, \ldots, -1)$ and outputting 1 when $c^t = (1, \ldots, 1)$, there is still a nontrivial correlation between the output and *some* coordinates in $\mathcal{S}$. Over time, these correlation scores will accumulate, and are used by $\mathcal{F}$ to correctly detect coordinates $i \in \mathcal{S}$ with high probability. It has been shown [SU15] that this can be done in $\mathcal{O}\left(k^2\right)$ queries.

In the extreme case where each row of $\mathbf{A}$ is a unit vector $\mathbf{e}_i$ with a single 1 in some column i and zero everywhere else, we note that the sketch will precisely observe the value of $\mathbf{x}_i$. Furthermore, the task of ℓ_0 gap norm requires the algorithm to distinguish between the number of non-zeroes $\leq \alpha n$ and $\geq \beta n$, for some constants $0 < \alpha < \beta < 1$. This is a stronger requirement than that of $\mathcal{P}$ in the fingerprinting code problem, which merely has to distinguish between all zero queries and all non-zero queries. Thus, the same attack strategy with the same number of queries applies in this case.

b) Significant coordinates: Next, let us see for a matrix $\mathbf{A}$, which coordinates $\mathbf{Ax}$ can preserve a significant amount of information about the input x. First, if there is a unit vector e_i (as in the above extreme case), then coordinate i is clearly very significant. Also, note that since $\mathbf{Ax}$ is linear, the query algorithm can recover any $\mathbf{w}^\top \mathbf{x}$ for $\mathbf{w}$ in the row span of $\mathbf{A}$ (i.e., $\exists \mathbf{y}^\top$, s.t., $\mathbf{w}^\top = \mathbf{y}^\top \mathbf{A}$). Thus, a relaxation of the unit vector together with the linearity gives the following definition of "significance" of a coordinate i:

$$\exists \mathbf{y}^\top \in \mathbb{R}^r, (\mathbf{y}^\top \mathbf{A})_i^2 \geq \frac{1}{s} \cdot \|\mathbf{y}^\top \mathbf{A}\|_2^2,$$

for some parameter $s \geq 1$. That is, there exists a linear combination of the rows such that the i-th coordinate is ℓ_2-heavy. Equivalently, the *leverage score* of column i is at least $\frac{1}{s}$. It turns out that if the query vectors were allowed to have *coordinates with real numbers*, this definition captures exactly which coordinates are significant, and is sufficient for proving that if the query vector is supported only on the "insignificant coordinates" (in this sense), the query algorithm cannot approximate the ℓ_0-norm.

However, when x is restricted to having integer coordinates bounded by $\text{poly}(n)$, it turns out that the leverage scores are not sufficient. Consider the matrix $\mathbf{A}$ with just one row of the form $(C, C, C, \ldots, C, 1)$ that has C in every coordinate except that the last coordinate is 1, for some integer $C \geq 2$. Every column has a leverage score of only $\mathcal{O}\left(\frac{1}{n}\right)$. On the other hand, when x can only have integer coordinates, $\mathbf{A}x$ tells us the value of x_n *modulo* C (when C is large, this may even completely reveal the coordinate). This phenomenon can be explained by considering the vector $\mathbf{w}^\top = \left(1, 1, 1, \ldots, 1, \frac{1}{C}\right)$, which is in the row span of $\mathbf{A}$. If we look at the *fractional part* of the inner product $\mathbf{w}^\top \mathbf{x}$, the first $n-1$ coordinates never contribute to the value regardless of $\mathbf{x}$. In other words, in the *fractional parts* of $\mathbf{w}^\top$, $\left(0, 0, 0, \ldots, 0, \frac{1}{C}\right)$, the last coordinate is in fact very heavy.

This suggests that in general, we should focus on the fractional parts of the vectors in the row span, which motivates us to define the significance of a coordinate i in the following way:

$$\exists \mathbf{y}^\top \in \mathbb{R}^r, |\textsc{Frac}((\mathbf{y}^\top \mathbf{A})_i)|^2 \geq \frac{1}{s} \cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{A})\|_2^2, \quad (1)$$

where $\textsc{Frac}(\cdot)$ is the fractional part, and when applied on a vector, it is applied coordinate-wise. It turns out that this definition captures our needs, and is what we will use for our main result over the integers.

c) Matrix pre-processing: To facilitate the analysis of the attack, we will first "pre-process" the sketching matrix $\mathbf{A}$ to a new matrix $\mathbf{A}'$ that separates the significant coordinates and the insignificant coordinates, while not weakening the sketch $\mathbf{Ax}$.

Let us consider the following pre-processing procedure on $\mathbf{A}$: while there exists a column $i \in [n]$ such that (1) holds, we zero out the i-th column of $\mathbf{A}$ and add i to the set of significant coordinates $\mathcal{S}$. Note that new columns may become significant as we zero out a column, and the procedure is applied iteratively on the remaining matrix until no column satisfies (1). Finally, for each coordinate $i \in \mathcal{S}$, we add a new row $\mathbf{e}_i$. Thus, the overall pre-processing can be viewed as follows: we find the significant coordinates; since $\mathbf{Ax}$ may preserve a significant amount of their information, we might as well just strengthen the

[4] $\mathcal{P}$ is referred to as the adversary in the original fingerprinting code problem, which would be the opposite for our application. To avoid confusion, we renamed it according to the standpoint here.

sketch so that it actually stores them precisely; then the rest of the sketch is made independent of them by zeroing out the corresponding columns.

Let $\mathbf{A}'$ denote the matrix after these operations. Without loss of generality, we can assume that the actual sketching matrix is $\mathbf{A}'$ instead of $\mathbf{A}$, since $\mathbf{A}'\mathbf{x}$ can recover $\mathbf{A}\mathbf{x}$ (as we can just add the new rows $\mathbf{e}_i$, with the correct weights, back to each row where column i was zeroed out), it makes the algorithm at least as powerful as it was. The new sketching matrix $\mathbf{A}'$ has the following form:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{D} \\ \mathbf{S} \end{bmatrix},$$

where we note that no column is significant in the sense of (1) for $\mathbf{D}$, and $\mathbf{S}$ has at most one non-zero entry 1 in each row and column. Moreover, the non-zero columns of $\mathbf{D}$ and $\mathbf{S}$ are disjoint. We refer to $\mathbf{D}$ as the dense part and $\mathbf{S}$ as the sparse part, and note that the set of significant coordinates $\mathcal{S}$ is precisely the set of non-zero columns in the sparse part $\mathbf{S}$.

Note that the sparse part is exactly the extreme case that we discussed earlier, and $\mathcal{S}$ can be learned using the fingerprinting code *if there were no dense part*. Moreover, we show that the definition of significant coordinates and the pre-processing procedure guarantee that the sparse part is small, $|\mathcal{S}| \ll n$, so that after learning $\mathcal{S}$ and zeroing out these coordinates in the query, we will not be left with a trivial problem. Roughly speaking, this is shown by proving that under the uniform distribution of $\mathbf{x} \in \{-1, 0, 1\}^n$, if a column i satisfies (1), then $\mathbf{A}\mathbf{x}$ must have a nontrivial mutual information with $\mathbf{x}_i$, $I(\mathbf{A}\mathbf{x}; \mathbf{x}_i) \geq \Omega\left(\frac{1}{s}\right)$. Then if the pre-processing algorithm removes T columns iteratively, by applying the chain rule for mutual information, we can argue that the mutual information between $\mathbf{A}\mathbf{x}$ and all these T corresponding coordinates is at least $\Omega\left(\frac{T}{s}\right)$. On the other hand, it can be at most $\mathcal{O}(r \log n)$, as $\mathbf{A}\mathbf{x}$ can be encoded in $\mathcal{O}(r \log n)$ bits. Hence, we can add at most $T = \mathcal{O}(rs \log n)$ rows to the sparse part.

d) Description of the attack: The last piece is to show that the dense part (insignificant coordinates) cannot be useful, by carefully picking query distributions. More specifically, we will design a family of distributions $\mathcal{D}$ over $\{-R, -(R-1), \ldots, R\}$ for some integer $R = \text{poly}(n)$ bounded by a small polynomial in n, with the following properties:

(1) For $D_p \in \mathcal{D}$ where $p \in [\alpha, \beta]$ for some constant $0 < \alpha < \beta < 1$, we have $\mathop{\mathbf{Pr}}_{X \sim D_p}[X = 0] = p$;
(2) For any $p, q \in [\alpha, \beta]$, we have $d_{\text{tv}}(\mathbf{D}\mathbf{x}_p, \mathbf{D}\mathbf{x}_q) \leq \frac{1}{\text{poly}(n)}$ for $\mathbf{x}_p \sim D_p^n$ and $\mathbf{x}_q \sim D_q^n$.

We will give more details about how to construct such a family of distributions later in this section. Given such a family, consider query vectors $\mathbf{x} \sim D_p^n$ for different p, we can express $\mathbf{A}'\mathbf{x} = \begin{bmatrix} \mathbf{D}\mathbf{x} \\ \mathbf{S}\mathbf{x} \end{bmatrix}$. From the property of the distribution family $\mathcal{D}$, we know that the (marginal) distribution of $\mathbf{D}\mathbf{x}$ is almost identical regardless of the value of p. Moreover, since $\mathbf{D}$ and $\mathbf{S}$ have disjoint nonzero columns, $\mathbf{D}\mathbf{x}$ and $\mathbf{S}\mathbf{x}$ are independent conditioned on p. This allows us to conclude that if we sample the queries from these distributions, then the algorithm must approximate $\|\mathbf{x}\|_0$ by only looking at the sparse part $\mathbf{S}\mathbf{x}$. It turns out that these distributions D_p can be "integrated" into the fingerprinting code, so that the dense part cannot help the algorithm when we attack the sparse part. This allows us to gradually identify $\mathcal{S}$, and eventually zero out all these coordinates. When we make one more query with all coordinates in $\mathcal{S}$ zeroed out, the algorithm must not produce a correct output with high probability based only on $\mathbf{D}\mathbf{x}$.

e) Constructing hard distributions for the insignificant coordinates: We wish to construct a family of distributions such that the total variation distance between $\mathbf{D}\mathbf{x}_p$ and $\mathbf{D}\mathbf{x}_q$ for $\mathbf{x}_p \sim D_p^n, \mathbf{x}_q \sim D_q^n$ is small. We will rely on the following property of $\mathcal{D}$: for every pair $D_p, D_q \in \mathcal{D}$, we have that

$$\mathop{\mathbb{E}}_{X \sim D_p}\left[X^k\right] = \mathop{\mathbb{E}}_{X \sim D_q}\left[X^k\right]$$

for all $k \in [K]$, i.e. the first $K = \mathcal{O}(r \log n)$ moments of D_p and D_q match. In fact, we will make all distributions $D_p \in \mathcal{D}$ *symmetric*, i.e., $D_p(t) = D_p(-t)$. Thus, all odd moments are zero, and hence, equal. Then for the even moments, the condition is equivalent to $\sum_{i=0}^{R} i^k \cdot (D_p(i) - D_q(i)) = 0$ for $k \leq K$. Our construction is based on the following fact (e.g., see Claim 1 in [LWY20]): there exists a polynomial Q with degree at most $R - \Omega(\sqrt{R})$ such that

$$|Q(0)| = \Omega(1) \qquad \text{and} \qquad \sum_{i=0}^{R} \left| \binom{R}{i} \cdot Q(i) \right| = \mathcal{O}(1).$$

The degree bound on Q further implies that

$$\sum_{i=0}^{R} (-1)^i \binom{R}{i} \cdot Q(i) \cdot i^t = 0$$

for all non-negative integers $t < R - \deg(Q)$, since $Q(i) \cdot i^t$ is a polynomial of degree strictly less than R, and $\sum_{i=0}^{R} (-1)^i \binom{R}{i} \cdot P(i) = 0$ holds for any polynomial P of degree $< R$.

Hence, we will set $R = \Theta(K^2)$ for a sufficiently large leading constant, and define the distribution family $\mathcal{D} = \{D_p\}$ such that $D_p(i) = D(i) + c_p \cdot (-1)^i \binom{R}{i} \cdot Q(i)$, for some distribution D and constants c_p. The difference between the probabilities $D_p(i)$ and $D_q(i)$ is precisely $c_p - c_q$ times $(-1)^i \binom{R}{i} \cdot Q(i)$. Then we can ensure that our moment matching condition is satisfied, since

$$\sum_{i=0}^{R} i^k (D_p(X) - D_q(X)) = (c_p - c_q) \sum_{i=0}^{R} i^k (-1)^i \binom{R}{i} Q(i) = 0$$

for $k \le K \le \mathcal{O}\left(\sqrt{R}\right)$. Furthermore, the bounds on $\sum_{i=0}^{R}\left|\binom{R}{i}\cdot Q(i)\right|$ and $|Q(0)|$ ensure that the range of the distribution family $\beta-\alpha$ can be made $\Omega(1)$ by carefully picking the base distribution D (recall that α,β are the smallest and the largest probabilities at 0 over all distributions in the family).

f) Bounding the total variation distance.: Let $P = D_p$ and $Q = D_q$ be distributions from family $\mathcal{D}$ that match the first K moments for some $p, q \in [\alpha, \beta]$. Suppose P^n and Q^n are probability distributions of n-dimensional vectors, where each entry is drawn independently from P and Q, respectively. As before, let $\mathbf{D}$ denote the dense matrix such that no column satisfies (1) with parameter s. For $\mathbf{x} \sim P^n$ and $\mathbf{x}' \sim Q^n$, let $P_{\mathbf{D}}$ and $Q_{\mathbf{D}}$ be the probability distributions of $\mathbf{Dx}$ and $\mathbf{Dx}'$. Now, we will argue that $d_{\mathrm{tv}}(P_{\mathbf{D}}, Q_{\mathbf{D}}) \le \frac{1}{\mathrm{poly}(n)}$. To see this, we use the following observation from Fourier analysis:

$$\begin{aligned}|P_{\mathbf{D}}(x) - Q_{\mathbf{D}}(x)| &= \left|\int_{[-\pi,\pi)^r} \frac{e^{i\langle \mathbf{u},\mathbf{x}\rangle}}{(2\pi)^r}\left(\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})\right) d\mathbf{u}\right| \\ &\le \frac{1}{(2\pi)^r}\int_{[-\pi,\pi)^r}\left|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})\right| d\mathbf{u}\end{aligned}$$

where the last inequality follows by triangle inequality. So, to bound the difference of $P_{\mathbf{D}}(\mathbf{x})$ and $Q_{\mathbf{D}}(\mathbf{x})$ for a particular value $\mathbf{x}$, we just need to upper bound the quantity $\left|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})\right|$. Let $P_i = \mathbf{Pr}[X = i]$ and $\textsc{Frac}_{2\pi}(x) = 2\pi \cdot \textsc{Frac}\left(\frac{x}{2\pi}\right) \in [-\pi, \pi)$. Then, we can then express $\widehat{P_{\mathbf{D}}}(\mathbf{u})$ (and similarly for $\widehat{Q_{\mathbf{D}}}(\mathbf{u})$) as follows:

$$\begin{aligned}\widehat{P_{\mathbf{D}}}(\mathbf{u}) &= \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{D}}}\left[e^{-i\langle \mathbf{u},\mathbf{z}\rangle}\right] = \mathbb{E}_{\mathbf{x}\sim P^n}\left[e^{-i\langle \mathbf{u},\mathbf{Dx}\rangle}\right] \\ &= \prod_{j\in[n]}\sum_{k\ge 0} P_k \cdot \cos\left(k\cdot\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle\right) \\ &= \prod_{j\in[n]}\sum_{k\ge 0} P_k \cdot \cos\left(k\cdot \textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right).\end{aligned}$$

where the second equality follows since our chosen distribution D_p is symmetric and we draw each coordinate $\mathbf{x}_j \sim D_p$ independently. Now, by the Taylor expansion $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$, we can write

$$\begin{aligned}\widehat{P_{\mathbf{D}}}(\mathbf{u}) &= \prod_{j\in[n]}\sum_{k\ge 0}\left(\sum_{m\ge 0} P_m m^{2k}\right) \\ &\quad\times \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}(-1)^k}{(2k)!}, \\ \widehat{Q_{\mathbf{D}}}(\mathbf{u}) &= \prod_{j\in[n]}\sum_{k\ge 0}\left(\sum_{m\ge 0} Q_m m^{2k}\right) \\ &\quad\times \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}(-1)^k}{(2k)!}\end{aligned}$$

Let $M_P(2k) = \left(\sum_{m\ge 0} P_m \cdot m^{2k}\right)$ and $M_Q(2k) = \left(\sum_{m\ge 0} Q_m \cdot m^{2k}\right)$ denote the $2k$-th moment of P and Q, respectively. At this point, our proof makes use of two key properties to upper bound $\left|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})\right|$:

(1) **Bounded fractional parts.** First, we recall that $\mathbf{D}$ satisfies $|\textsc{Frac}(\mathbf{y}^\top\mathbf{D})_j|^2 \le \frac{1}{s}\cdot\|\textsc{Frac}(\mathbf{y}^\top\mathbf{D})\|_2^2$ for all $y \in \mathbb{R}^r$ and $j \in [n]$. Then, if there exists some index $j \in [n]$ such that $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| \ge \frac{1}{K}$ (for some chosen threshold t), we can use the above property of $\left\|\textsc{Frac}\left(\left\langle \frac{\mathbf{u}}{2\pi}, \mathbf{D}\right\rangle\right)\right\|_2^2$ to upper bound $\widehat{P_{\mathbf{D}}}(\mathbf{u})$ and $\widehat{Q_{\mathbf{D}}}(\mathbf{u})$.

(2) **Moment matching.** Alternatively, suppose there is no such index j; then we can use the fact that $M_P(2k) = M_Q(2k)$ for $k \le K/2$, so we have that the first $K/2$ terms of

$$\sum_{k\ge 0} M_P(2k)\cdot\frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}}{(2k)!}\cdot(-1)^k$$

$$\sum_{k\ge 0} M_Q(2k)\cdot\frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}}{(2k)!}\cdot(-1)^k$$

are exactly the same. By combining this fact with our assumption that $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| < \frac{1}{K}$ for every $j \in [n]$, we obtain the desired upper bound for this case as well.

For the full argument, we refer the readers to Section V. Finally, since $\mathbf{D}$ is a matrix in $\mathbb{Z}^{r\times n}$ with entries bounded in $\mathrm{poly}(n)$, we know that the total support size of $P_{\mathbf{D}}$ and $Q_{\mathbf{D}}$ is $n^{\mathcal{O}(r)}$. So, after we compute an upper bound for $\left|\widehat{P}_{\mathbf{D}}(\mathbf{u}) - \widehat{Q}_{\mathbf{D}}(\mathbf{u})\right|$, we can finish the argument by simply union-bounding over the total size of the support of $\mathbf{Dx}$ to obtain the upper bound of $d_{\mathrm{tv}}(P_{\mathbf{D}}(\mathbf{x}), Q_{\mathbf{D}}(\mathbf{x})) \le \frac{1}{\mathrm{poly}(n)}$ for some choice of parameters K and s.

C. Overview of Attack over Finite Fields

When the sketching matrix $\mathbf{A} \in \mathbb{F}_p^{r\times n}$ for a fixed prime p, our attack is based on the following crucial observation: suppose that U and R are the two subsets of indices of columns of $\mathbf{A}$ such that $\mathbf{A}^U$ and $\mathbf{A}^R$ have the same column span. Then, if $\mathbf{x} \sim \mathbb{F}_p^{|U|}$ and $\mathbf{x}' \sim \mathbb{F}_p^{|R|}$ are sampled uniformly at random, we can show that $\mathbf{A}^U\mathbf{x}$ and $\mathbf{A}^R\mathbf{x}'$ are identically distributed. With this in mind, note that if we can find an independent set of columns T with $|T| = r$, then the streaming algorithm $\mathcal{A}$ will not be able to distinguish $\mathbf{A}^T\mathbf{x}'$ where $\mathbf{x}' \sim \mathbb{F}_p^r$ and $\mathbf{Ax}$ where $\mathbf{x} \sim \mathbb{F}_p^n$. Hence, $\mathcal{A}$ must fail on one of the input distributions (we assume $n \ge 2r$). Therefore, our goal now is to find such a column-independent set.

The way we search for this column independent set is as follows: suppose that the set T is what we have maintained up to now. Then let R be a random sample of $2r$ columns outside T and R^i is the first i column of R. Let μ_i denote the distribution of $f(\mathbf{Ax}^{(i)})$, where $\mathbf{x}^{(i)} \in \mathbb{F}_p^n$ is the random vector that on the support of $T \cup R^i$. From the correctness guarantee we must have the total

variation distance $d_{\mathrm{tv}}(\mu_0, \mu_{2r-1}) = \Omega(1)$ (otherwise we find a distribution that $\mathcal{A}$ fails with constant probability immediately). Then from triangle inequality we have

$$\sum_i d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \geq d_{\mathrm{tv}}(\mu_0, \mu_r) = \Omega(1).$$

From this we get there must exist one j such that $d_{\mathrm{tv}}(\mu_{j-1}, \mu_j) \geq \Omega\left(\frac{1}{r}\right)$, which means that the j-th column in R must be linear independent in T. Note that from the result in statistical testing we can distinguish this case using $\widetilde{O}(r^2)$ samples with error probability at most $1/\operatorname{poly}(r)$. Hence, we can enumerate the index i and do the testing between μ_i and μ_{i+1} to find such column index j.

The above procedure requires $\tilde{\mathcal{O}}\left(r^4\right)$ total number of queries, as we need to find r columns and in each step, we make $2r \cdot \tilde{\mathcal{O}}\left(r^2\right) = \tilde{\mathcal{O}}\left(r^3\right)$ queries. However, the dependence of r can be further improved. Note that in the worst case $\max_i\{d_{\mathrm{tv}}(\mu_{i-1}, \mu_i)\} = \Theta\left(\frac{1}{r}\right)$, we can randomly sample $\mathcal{O}(1)$ indices to find such index j, which suggests a better dependence of r. Indeed, we show that there must exist ℓ for which there exist at least $2^{\ell-1}$ indices i such that $d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \in \left[\frac{1}{2^{\ell+3}\log r}, \frac{1}{2^{\ell+2}\log r}\right)$. Hence, we can make a guess of such ℓ and note that for each different guess, since the range of the total variation distance changes, we can use a different number of the samples in the testing procedure, which results in an overall $\tilde{\mathcal{O}}\left(r^3\right)$ number of queries.

II. Preliminaries

For a positive integer $n > 0$, we write $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use the notation $\operatorname{poly}(n)$ to denote a fixed polynomial in n and $\operatorname{polylog}(n)$ to represent $\operatorname{poly}(\log n)$. We say an event $\mathcal{E}$ occurs with high probability if $\mathbf{Pr}[\mathcal{E}] \geq 1 - \frac{1}{\operatorname{poly}(n)}$, when the dependent variable n is clear from context.

A. Interactive Fingerprinting Codes

An interactive fingerprinting code $\mathcal{F}$ is an efficient adaptive algorithm that defeats any adversary $\mathcal{P}$ in the following two-player game. The adversary $\mathcal{P}$ first selects a secret subset of indices $\mathcal{S} \subset [N]$, where $|\mathcal{S}| = n$. Then, the goal of $\mathcal{F}$ is to construct an adaptive sequence of queries $\{c^t\}_{t\in[\ell]}$ to learn (or "accuse") all of the indices $i \in \mathcal{S}$, while making few false accusations (i.e., incorrectly accusing some $i \notin \mathcal{S}$) in the process. Specifically, in each round $t \in [\ell]$, the interactive fingerprinting code $\mathcal{F}$ selects a query vector $c^t \in \{\pm 1\}^N$, and the adversary $\mathcal{P}$ observes only the coordinates c_i^t for those $i \in \mathcal{S}$, and has no knowledge of c_i^t for $i \notin \mathcal{S}$. Then, the adversary must respond with an answer a^t that is *consistent* with some coordinate of c^t such that $a^t = c_i^t$ for some $i \in \mathcal{S}$. More concretely, if all of the coordinates of $c^t = 1^N$ then a^t must be 1, or if $c^t = (-1)^N$, then a^t must return -1.

Informally, the interactive fingerprinting attack of [SU15] proceeds by assigning a score s_i^t to each index $i \in [N]$ at every round $t \in [\ell]$, which corresponds to a measure of the *correlation* between values of the i^{th} index $(c_i^1, ..., c_i^t)$ and the responses $(a^1, ..., a^t)$ given by the adversary during the first t rounds. The interactive fingerprinting code $\mathcal{F}$ accuses coordinates $i \in [N]$ whose score s_i^t exceeds a threshold σ at some point during the sequence of queries. Using this approach, combined with an appropriate hard distribution for inputs $c^t \in \{\pm 1\}^N$, [SU15] shows that for every $N \in \mathbb{N}$, there exists an interactive fingerprinting code that makes $\ell = \mathcal{O}\left(n^2 \log \frac{1}{\delta}\right)$ queries and, except with negligible probability, identifies all of $\mathcal{S}$ and makes at most $\frac{N\delta}{1000}$ false accusations. Moreover, their attack satisfies a *robustness* property: the result above holds even when the fingerprinting adversary $\mathcal{P}$ only provides a response a^t which is consistent with some coordinate c_i^t in at least $(1-\beta)\ell$ of the rounds, for any $\beta < 1/2$.

We provide a brief overview of the game, as well as the attack of [SU15] here, as a reference.

Definition II.1 (Interactive Fingerprinting Code Game)**.** *The Interactive Fingerprinting Code problem is defined via the following game.*

(1) *First, the adversary $\mathcal{P}$ selects a subset of users $S^1 \subseteq [N]$, $|S^1| = n$, which is unknown to the fingerprinting code $\mathcal{F}$.*
(2) *In each round $j = 1, ..., \ell$:*
   - *$\mathcal{F}$ outputs a column vector $c^j \in \{\pm 1\}^N$.*
   - *Let $c^j_{S^j} \in \{\pm 1\}^{|S^j|}$ be the restriction of c^j to coordinates in S^j: only this restricted copy $c^j_{S^j}$ is given to the adversary $\mathcal{P}$ in each round.*
   - *Then, $\mathcal{P}$ outputs $a^j \in \{\pm 1\}$, which is observed by $\mathcal{F}$.*
   - *Finally, $\mathcal{F}$ accuses a set of users $I^j \subseteq [N]$, and sets $S^{j+1} = S^j \setminus I^j$ as the current "undiscovered" set of coordinates/users.*

a) Construction of attack, c.f., [SU15]: For $0 \leq a < b \leq 1$, let $P_{a,b}$ be the distribution with support (a, b) and probability density function $\mu(p) = \frac{C_{a,b}}{\sqrt{p(1-p)}}$. For $\alpha, \zeta \in (0, \frac{1}{2})$, let $\overline{P_{\alpha,\zeta}}$ be the distribution on $[0, 1]$ such that it returns a sample from $D_{\alpha,1-\alpha}$ with probability $1 - 2\zeta$, and returns 0/1 each with probability ζ. Furthermore, let $\phi : \{-1, 1\} \to \mathbb{R}$ be defined by $\phi^0(c) = \phi^1(c) = 0$ and for $p \in (0, 1)$, we have $\phi^p(1) = \sqrt{\frac{1-p}{p}}$ and $\phi^p(0) = -\sqrt{\frac{p}{1-p}}$. We consider the following parameter regime:

$$\alpha = \frac{(\frac{1}{2} - \beta)}{n}$$

$$\zeta = \frac{3}{8} + \frac{\beta}{4}$$

$$\sigma = \mathcal{O}\left(\frac{n}{(\frac{1}{2} - \beta)^2} \log\left(\frac{1}{\delta}\right)\right)$$

$$\ell = \mathcal{O}\left(\frac{n^2}{(\frac{1}{2} - \beta)^4} \log\left(\frac{1}{\delta}\right)\right)$$

(1) Let $s_i^0 = 0$ for every $i \in [N]$.
(2) For $j = 1, ..., \ell$:
- Draw $p^j \sim \overline{P_{\alpha,\zeta}}$ and $c_{1\cdots N}^j \sim p^j$.
- Issue $c^j \in \{\pm 1\}^N$ as a challenge and get response $a^j \in \{\pm 1\}$.
- For every $i \in [N]$, update score $s_i^j = s_i^{j-1} + a^j \cdot \phi^{p^j}(c_i^j)$

Importantly, the attack enforces the following completeness and soundness properties:

- **Completeness:** If $i \in S^1$, the score of user i will exceed some chosen threshold at *some* step $j \in [\ell]$, i.e. with high probability, there exists j such that $s_i^j > \sigma$.
- **Soundness:** Alternatively, if $i \notin S^1$, the score s_i^j will not exceed σ with high probability. The argument uses the fact that the responses of $\mathcal{P}$ cannot have high correlation with $(c_i^1, ..., c_i^\ell)$ if $\mathcal{P}$ never sees this information.

B. Preliminaries from Information Theory

We recall the following preliminaries from information theory.

Definition II.2 (Entropy and conditional entropy)**.** *The* entropy *of a random variable X taking on possible values in a finite space Ω is defined as*

$$H(X) := \sum_{x \in \Omega} p(x) \log \frac{1}{p(x)},$$

where $p(x) = \mathbf{Pr}[X = x]$ is the probability mass function of X. The conditional entropy *of X with respect to a random variable Y is defined as*

$$H(X|Y) = \mathbb{E}_y H(X|Y = y),$$

where $H(X|Y = y) := \sum_{x \in \Omega} p(x|y) \log \frac{1}{p(x|y)}$, for the conditional probability mass function $p(x|y)$.

Definition II.3 (Mutual information and conditional mutual information)**.** *We define the* mutual information *between random variables X and Y by*

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X).$$

We define the conditional mutual information *between X and Y conditioned on a random variable Z by*

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z).$$

Theorem II.4 (Data-processing inequality)**.** *Let X, Y, Z be random variables such that $X \to Y \to Z$ forms a Markov Chain, i.e., X and Z are conditionally independent given Y. Then, we have $I(X;Y) \geq I(X;Z)$.*

Theorem II.5 (Chain rule for mutual information)**.** *For random variables $X_1, \ldots, X_n, Z$, we have*

$$I(X_1, \ldots, X_n; Z) = \sum_{i=1}^{n} I(X_i; Z | X_1, ..., X_{i-1}).$$

III. Pre-processing the Sketching Matrix

Our attack will rely on pre-processing and decomposing the sketching matrix $\mathbf{A}$ into *sparse* part and a *dense* part, which will consist of disjoint sets of non-zero indices. This pre-processing procedure will have the property that it can only make the streaming algorithm stronger, by potentially allowing the algorithm to observe more entries of the input vector $\mathbf{x}^{(t)}$. More formally, our new matrix $\mathbf{A}'$ will satisfy several key properties, as stated in the next lemma.

Lemma III.1. *For any algorithm $\mathcal{A}$ with sketching matrix $\mathbf{A} \in \mathbb{Z}^{r \times n}$, there is a pre-processing procedure that produces a new matrix $\mathbf{A}' \in \mathbb{Z}^{r' \times n}$ for $r' = \mathcal{O}(rs \log n)$ satisfying the following properties:*

(1) *The $\mathbf{A}'$ has the form $\begin{bmatrix} \mathbf{D} \\ \mathbf{S} \end{bmatrix}$ where the $\mathbf{D}$ and $\mathbf{S}$ are column-disjoint.*
(2) *We have $|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})_j|^2 \leq \frac{1}{s} \cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})\|_2^2$ for all $\mathbf{y} \in \mathbb{R}^r$ and $j \in [n]$*
(3) *Each row and column of $\mathbf{S}$ has at most one non-zero entry.*

Moreover, without loss of generality, we can assume the algorithm $\mathcal{A}$ uses sketching matrix $\mathbf{A}'$ instead of $\mathbf{A}$.

Proof. We consider the following procedure: we start with the original sketching matrix $\mathbf{A}$, and for each time t, we identify a columns $j_t \in [n]$ such that $|\textsc{Frac}(\mathbf{y}^\top \mathbf{D}^{(t-1)})_{j_t}|^2 > \frac{1}{s} \cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{D}^{(t-1)})\|_2^2$ where $\mathbf{D}^{t-1}$ is the first r rows of $\mathbf{A}^{(t-1)}$, we zero the j_t-th column of $\mathbf{A}^{t-1}$ and add a new row $\mathbf{e}_{j_t}$ to the matrix $\mathbf{A}^{(t-1)}$. We then denote the new resulting matrix as $\mathbf{A}^{(t)}$. Suppose that the above procedure ends in the iteration T. From Lemma III.3 we have $T = \mathcal{O}(rs \log n)$.

Let $\mathbf{D} = \mathbf{D}^{(T)}$ and $\mathbf{S}$ be the remaining rows of $\mathbf{A}^{(T)}$. Then we have $\mathbf{A}' = \begin{bmatrix} \mathbf{D} \\ \mathbf{S} \end{bmatrix}$ has at most $r + T = \mathcal{O}(rs \log n)$ columns. Then from the procedure it is easy to see that the $\mathbf{D}$ and $\mathbf{S}$ are column-disjoint and each row and column of $\mathbf{S}$ has at most one non-zero entry.

At this point, it remains for us to show why we can assume that the sketching matrix used by $\mathcal{A}$ is $\mathbf{A}'$ instead of $\mathbf{A}$. Suppose that the algorithm $\mathcal{A}$ uses the sketching matrix $\mathbf{A}$ and estimator f on $\mathbf{Ax}$, then we consider the following equivalent form of the algorithm $\mathcal{A}$, where it uses the sketching matrix $\mathbf{A}'$ and another estimator g. Given an $\mathbf{A}'\mathbf{x}$, estimator g will first invert the row operations that transfer $\mathbf{A}'\mathbf{x}$ to $\mathbf{Ax}$ (recall that all of the operations we apply on $\mathbf{A}$ are invertible) and output the value $f(\mathbf{Ax})$. From the definition of g, we immediately get that $g(\mathbf{A}'\mathbf{x}) = f(\mathbf{Ax})$ for every input $\mathbf{x}$, which means we can assume $\mathcal{A}$ has the form $g(\mathbf{A}'\mathbf{x})$ without loss of generality. □

A. Bounding the Number of Added Rows

Let $\textsc{Frac}(x) = x - \mathsf{int}(x) \in (-\frac{1}{2}, \frac{1}{2}]$ where $\mathsf{int}(x)$ is the closest integer number to x and for a vector $\mathbf{x} \in \mathbb{R}^n$, let

$\textsc{Frac}(\mathbf{x}) \in \mathbb{R}^n$ be the coordinate-wise fractional parts of $\mathbf{x}$, i.e., $\textsc{Frac}(\mathbf{x})_j = \textsc{Frac}(x_j)$.

Lemma III.2. *Let $\mathbf{A} \in \mathbb{Z}^{r\times n}$ be a fixed matrix and let $\mathbf{x} \in \{-1,0,1\}^n$ such that each coordinate is chosen independently and with probability $1-\frac{2c}{s}$, $x_i = 0$, and with probability $\frac{2c}{s}$, $x_i = 1$ or -1 with equal probability, where c is a sufficiently small constant. Suppose there exists $\mathbf{y} \in \mathbb{R}^r$ and $j \in [n]$ such that for $|\textit{Frac}((\mathbf{y}^\top \mathbf{A})_j)|^2 \geq \frac{1}{s}\cdot \|\textit{Frac}(\mathbf{y}^\top\mathbf{A})\|_2^2$. Then we have $I(\mathbf{Ax}; x_j) = \Omega\left(\frac{1}{s}\right)$.*

Proof. Observe that since by the data-processing inequality,

$$I(\mathbf{Ax}; x_j) \geq I(\mathbf{y}^\top\mathbf{Ax}; x_j) \geq I(\textsc{Frac}(\mathbf{y}^\top\mathbf{Ax}); x_j),$$

then it suffices to show $I(\textsc{Frac}(\mathbf{y}^\top\mathbf{Ax}); x_j) = \Omega\left(\frac{1}{s}\right)$. Let $\mathbf{a} = \mathbf{y}^\top\mathbf{A} \in \mathbb{R}^n$. We sample column vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$, such that all coordinates are selected randomly from $\{-1,0,1\}$ from the distribution in the lemma statement but the j-th coordinate is the same in all t vectors. Since the marginal distributions are the same for each $\mathbf{x}^{(k)}$, we have that $I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle); x_j) = I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k)}\rangle); x_j)$.

We next claim that, by looking at $t = \mathcal{O}(\log s)$ samples $\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k)}\rangle)$ from $k = 1, 2, \cdots t$, we can determine the value of x_j with probability at least $1 - 1/\operatorname{poly}(s)$, which means that

$$I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle, \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(t)}\rangle); x_j) = \Omega(\log s/s).$$

a) Mutual information from independent instances.: Firstly, note that if $\|\textsc{Frac}(\mathbf{a})\|_2^2 > s$, then $\frac{1}{s}\cdot \|\textsc{Frac}(\mathbf{a})\|_2^2 > 1$ and so there cannot exist y such that $|\textsc{Frac}(\mathbf{a}_j)|^2 \geq \frac{1}{s}\cdot\|\textsc{Frac}(\mathbf{a})\|_2^2$. Thus it suffices to consider $\|\textsc{Frac}(\mathbf{a})\|_2^2 \leq s$.

We next consider one of the instances $\mathbf{x}$, let S be the set of indices such that $x_i \neq 0$ and $i \neq j$. Then we have $\mathbb{E}\left[\|\textsc{Frac}(\mathbf{a}_S)\|_2^2\right] \leq \frac{2c}{s}\|\textsc{Frac}(\mathbf{a})\|_2^2$, by Markov's inequality we have with probability at least 0.99, $\|\textsc{Frac}(\mathbf{a}_S)\|_2^2 \leq \frac{1}{100s}\|\textsc{Frac}(\mathbf{a})\|_2^2$. Condition on this event, by Markov's inequality again we can have with probability at least 0.9,

$$\sum_{i\in S}\left(\textsc{Frac}(a_i)\cdot x_i\right)^2 \leq \frac{1}{10s}\|\textsc{Frac}(\mathbf{a})\|_2^2 \leq \frac{1}{10}.$$

which means that $\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle) = \textsc{Frac}(\textsc{Frac}(a_j)\cdot x_j + \alpha)$ where $\alpha = \sum_{i\in S}\textsc{Frac}(a_i)\cdot x_i \leq \frac{1}{3\sqrt{s}}\|\textsc{Frac}(\mathbf{a})\|_2 \leq \frac{1}{3}|\textsc{Frac}(a_j)|$.

Condition on the above events happen, consider the case where $x_j = 0$. In this case, we have $|\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle)| \leq \frac{1}{3}|\textsc{Frac}(a_j)|$. On the other hand, if $x_j \neq 0$, we will have $|\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle)| \geq \frac{2}{3}|\textsc{Frac}(a_j)|$ and the sign of $|\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle)|$ is the same as x_j, which means that we can determine the value of x_j by looking at the value of $\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle)$.

The above procedure succeeds with high constant probability, to boost the success probability, we can instead look at the majority of the outputs by $\mathcal{O}(\log s)$ independent instances

$$\left(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle, \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(t)}\rangle)\right),$$

which makes the error probability $\frac{1}{\operatorname{poly}(s)}$ at most. This means that we have

$$I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle, \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(t)}\rangle); x_j) = \Omega\left(\frac{\log s}{s}\right)$$

b) Mutual information from a single instance.: On the other hand, by the chain rule for mutual information, i.e., Theorem II.5, we have

$$\begin{aligned}&I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle), \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(t)}\rangle); x_j)\\ =&\sum_{k=1}^{t} I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k)}\rangle); x_j \mid \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle), \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k-1)}\rangle)).\end{aligned}$$

Since $\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k)}\rangle)$ is independent of $\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle, \ldots \langle \mathbf{a}, \mathbf{x}^{(k-1)}\rangle)$ conditioned on x_j, we have

$$\begin{aligned}\Omega(1) = I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(1)}\rangle), \ldots, \textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(t)}\rangle); x_j)&\\ \leq \sum_{k=1}^{t} I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}^{(k)}\rangle); x_j)&\\ = \sum_{k=1}^{t} I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle); x_j).&\end{aligned}$$

Thus we have

$$\begin{aligned}I(\textsc{Frac}(\mathbf{y}^\top\mathbf{Ax}; x_j)) = I(\textsc{Frac}(\langle \mathbf{a}, \mathbf{x}\rangle); x_j) &= \Omega\left(\frac{\log s}{st}\right)\\ &= \Omega\left(\frac{1}{s}\right).\end{aligned}$$

□

Lemma III.3. *Let $\mathbf{A} \in \mathbb{Z}^{r\times n}$ be a fixed matrix. There exists a pre-processing procedure to $\mathbf{A}$ and produces a matrix $\mathbf{A}' \in \mathbb{Z}^{r\times n}$ such that $\mathbf{A}'$ zero out at most $\mathcal{O}(rs\log n\log s)$ columns of $\mathbf{A}$. Moreover, we have $|\textit{Frac}(\mathbf{y}^\top\mathbf{A}')_j|^2 \leq \frac{1}{s}\cdot\|\textit{Frac}(\mathbf{y}^\top\mathbf{A}')\|_2^2$ for all $\mathbf{y} \in \mathbb{R}^r$ and $j \in [n]$.*

Proof. Let $\mathbf{A}^{(0)} = \mathbf{A}$ and let $\mathbf{x} \in \{-1,0,1\}^n$ drawn from the same distribution in Lemma III.2, so that $\mathbf{Ax}$ has $\mathcal{O}(r\log n)$ bits. Suppose that the above procedure ends in T rounds where $T \leq n$. Specifically, for each $t \in [T]$, we identify a columns $j_t \in [n]$ such that $|\textsc{Frac}(\mathbf{y}^\top\mathbf{A}^{(t-1)})_j|^2 > \frac{1}{s}\cdot\|\textsc{Frac}(\mathbf{y}^\top\mathbf{A}^{(t-1)})\|_2^2$ and let $\mathbf{A}^{(t)}$ be the matrix $\mathbf{A}^{(t-1)}$ after zeroing out the identified column. We next apply the chain rule for mutual information, i.e., Theorem II.5. we have

$$I(\mathbf{A}^{(T)}\mathbf{x}; x_{j_1}, \cdots, x_{j_T}) = \sum_{t=1}^{T} I(\mathbf{A}^{(T)}\mathbf{x}; x_{j_t} \mid x_{j_{t+1}}, \cdots x_{j_T}).$$

Note that given the matrix $\mathbf{A}$ and $x_{j_{t+1}}, x_{j_{t+2}}, \cdots x_{j_T}$, we can recover $\mathbf{A}^{(t)}$. Hence, by a similar approach to that in Lemma III.2, we have

$I(\mathbf{A}^{(T)}\mathbf{x}; x_{j_t} \mid x_{j_{t+1}}, x_{j_{t+2}}, \cdots x_{j_T}) \geq \mathcal{O}\left(\frac{1}{s \log s}\right)$ from $|\textsc{Frac}(\mathbf{y}^\top \mathbf{A}^{(t-1)})_j|^2 > \frac{1}{s} \cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{A}^{(t-1)})\|_2^2$. Since $\mathbf{Ax}$ can be represented using $\mathcal{O}(r \log n)$ bits, then it follows that $I(\mathbf{A}^{(T)}\mathbf{x}; x_{j_1}, x_{j_2}, \cdots, x_{j_T}) \leq \mathcal{O}(r \log n)$. Putting these two things together, we have that $Cr \log n \geq \frac{T}{s \log s}$ for some constant C, which means that we have $T = \mathcal{O}(rs \log n \log s)$.

□

IV. Attack Against Linear Sketches

In this section, we give a full description of our attack against linear sketches for ℓ_0-estimation. In particular, we prove the following theorem.

Theorem IV.1. *Suppose that $\mathcal{A}$ is a linear streaming algorithm that solves the $(\alpha+c, \beta-c)$-ℓ_0 gap norm problem with some constant α, β and c, where $\mathbf{A} \in \mathbb{Z}^{r \times n}$ is the sketching matrix with $r << n$, $f : \mathbb{Z}^{r \times n} \to \{-1, +1\}$ is any estimator used by $\mathcal{A}$, and $\mathcal{A}$ returns $f(\mathbf{A}, \mathbf{Ax})$ for each query $\mathbf{x}$.*

Then, there exists a randomized algorithm, which after making an adaptive sequence of queries to $\mathcal{A}$, with high constant probability can generate a distribution D on $\mathbb{Z}^n$ such that $\mathcal{A}$ fails on D with constant probability. Moreover, this adaptive attack algorithm makes at most $\tilde{\mathcal{O}}(r^8)$ queries and runs in $\text{poly}(r)$ time.

A. Construction and Analysis Overview

The full description of the attack is given in Figure 1. We first define the probability distribution $P_{a,b}$ with support $[a, b]$ to have probability density function $\mu(p) = \frac{C_{a,b}}{\sqrt{p(1-p)}}$, where $C_{a,b}$ is a normalizing constant.

For $p \in [0, 1]$, we define $\phi^p : \{\pm 1\} \to \mathbb{R}$ by $\phi^0(c) = \phi^1(c) = 0$, and for $p \in (0, 1)$, $\phi^p(1) = -\sqrt{\frac{p}{1-p}}$ and $\phi^p(-1) = \sqrt{\frac{1-p}{p}}$ so that by construction, $\phi^p(c)$ has mean 0 and variance 1 when $\mathbf{Pr}[c = -1] = p$ and $\mathbf{Pr}[c = 1] = 1 - p$.

In this section, we give a high-level description of our algorithm. First, recall that by Lemma III.1 with parameter $s = \mathcal{O}\left(r^3 \log^3 n\right)$, we can assume the sketching matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{D} \\ \mathbf{S} \end{bmatrix}$$

without loss of generality in our attack. Importantly, we recall that $\mathbf{A}$ has the following properties:

(1) The $\mathbf{D}$ and $\mathbf{S}$ are column-disjoint.
(2) We have that $|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})_j|^2 \leq \frac{1}{s} \cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})\|_2^2$ for all $\mathbf{y} \in \mathbb{R}^r$ and $j \in [n]$
(3) $\mathbf{S}$ contains at most $h = \mathcal{O}(rs \log n) = \mathcal{O}\left(r^4 \log^3 n\right)$ non-zero columns.

Suppose that $\mathcal{D}$ is the distribution family in Lemma V.2 with $K = \mathcal{O}(r \log n)$. Then, for $\mathbf{x} \sim D_\alpha$ and $\mathbf{x}' \sim D_\beta$, by Lemma V.4, we know that $d_{\text{tv}}(\mathbf{Dx}, \mathbf{Dx}') \leq \frac{1}{\text{poly}(n)}$. Let $\mathcal{S}$ denote the set of non-zero column indices of $\mathbf{S}$. Then, if we set $\mathbf{x}_i = 0$ and $\mathbf{x}'_i = 0$ for all $i \in \mathcal{S}$, the streaming algorithm $\mathcal{A}$ must fail to solve the ℓ_0 gap norm problem in one of these two cases, with constant probability. With this motivation in mind, the main task of the adaptive adversary is to design an adaptive sequence of queries to learn the set of indices in $\mathcal{S}$.

In each iteration, we query a random vector $\mathbf{x}^t \sim D_p^n$ where p is sampled in $P_{\alpha,\beta}$. We maintain a score s_i^t for each coordinate $i \in [n]$, which represents some measure of the *correlation* between the i-th coordinate of the inputs and the outputs of the algorithm $\mathcal{A}$ up until step t. In particular, at the t-th iteration and for each coordinate, let $c_i^t = 1$ if $x_i^t \neq 0$ and $c_i^t = -1$ if $x_i^t = 0$. Suppose that the output of the algorithm is a^t, then we update each coordinate's current store $s_i^t = s_i^{t-1} + a^t \cdot \phi^p(c_i^t)$, where $\phi^p(\cdot)$ is some specially-chosen function which depends on the choice of p. If for coordinates i, the score s_i^t exceeds a pre-determined threshold σ, then we accuse this coordinate and treat it as a coordinate in the secret set $\mathcal{S}$. Furthermore, we set this coordinate to 0 in all future queries, so the algorithm cannot get any information about this coordinate in future iterations.

From the guarantee of the algorithm $\mathcal{A}$, we get that when p is close to α, with high probability it should output -1 and when p is close to β, it should output 1. However, from Lemma V.4 we know that the algorithm $\mathcal{A}$ can almost get nothing about the value of p from the part of the sketch $\mathbf{Dx}_D$, which means its output should have a higher correlation with the coordinates in $\mathcal{S}$. With this ind mind, the proof is comprised of the following two parts:

- **Soundness:** For any coordinate $i \notin \mathcal{S}$, the score s_i will never exceed σ with high probability, which means that coordinate i will never be falsely accused.
- **Completeness:** Let $\mathcal{S}^j$ be the remaining (undiscovered) coordinates in $\mathcal{S}$ in the j-th round. We will show that if the algorithm still has the correctness guarantee on D_α^n and D_β^n after we zero out the accused coordinates, then the sum $\sum_{i \in \mathcal{S}} s_i$ will increase faster than the scores of other coordinates. This means that as we accuse more and more coordinates in $\mathcal{S}$, we either find a distribution that $\mathcal{A}$ or find more coordinates in $\mathcal{S}$ (note that if find the whole set $\mathcal{S}$, the algorithm $\mathcal{A}$ must fail in the next iteration).

To simplify our argument in Sections IV-B and IV-C, we will first prove that soundness and completeness hold in the case that the streaming algorithm $\mathcal{A}$ only uses $\mathbf{x}_S$ to estimate the ℓ_0 norm at each step. Then, in Section IV-D, we show that the soundness and completeness guarantees hold for an arbitrary algorithm $\mathcal{A}$ which uses both the sparse part $\mathbf{x}_S$ and the dense part $\mathbf{D}x_D$ to compute its responses to each query.

B. Soundness

Lemma IV.2. *For $p \in [\alpha, \beta]$, let $\tau = \min(\alpha, 1 - \beta)$ and $t \in \left[-\frac{\sqrt{\tau}}{2}, \frac{\sqrt{\tau}}{2}\right]$. Then*

$$\mathop{\mathbb{E}}_{v \sim D_p}\left[e^{t\phi^p(c)}\right] \leq e^{t^2},$$

Let α and β be defined as in Lemma V.2
Let $\mathcal{D}$ be the distribution family in Lemma V.2 with $K = \mathcal{O}(r \log n)$
$h \leftarrow \mathcal{O}(rs \log n) = \mathcal{O}\left(r^4 \log^3 n\right)$, $\sigma \leftarrow \mathcal{O}(h \log(n))$, $\ell \leftarrow \mathcal{O}(h) \cdot \sigma$, $c \leftarrow \mathcal{O}(1)$
Let $z_J(v)$ denote the vector where we make v_i to 0 for all $i \in J$.
$\mathcal{A} \leftarrow$ An instantiation of the ℓ_0 gap-norm algorithm.
Initialize $s_i^0 = 0$ for all $i \in [n]$.
For $j \in [\ell]$:
Sample $u^1, \cdots, u^c \sim D_\alpha^n$ and $v^1, \cdots, v^c \sim D_\beta^n$.
If $\mathcal{A}$ fails with constant probability on one of $z_{I^{j-1}}(u^i)$ or $z_{I^{j-1}}(v^i)$: Output this distribution as the attack.
Sample $p^j \sim P_{\alpha,\beta}$ and $v^j \sim D_{p^j}^n$.
For each $i \in [n]$, set $c_i^j = 1$ if $v_i^j \neq 0$ and $c_i^j = -1$ otherwise if $v_i^j = 0$.
Query $z_{I^{j-1}}(v^j) \in \mathbb{Z}^n$ and receive $a^j = \mathcal{A}(z_{I^{j-1}}(v^j)) \in \{\pm 1\}$ as the output.
For $i \in [n]$, update $s_i^j \leftarrow s_i^{j-1} + a^j \cdot \phi^{p^j}(c_i^j)$.
Set $I^j = I^{j-1} \cup \{i \in [n] \mid s_i^j > \sigma\}$ and $\mathcal{S}^{j+1} = \mathcal{S} \setminus I^j$.

Fig. 1. Construction of Our Attack

where $c = 1$ if v is nonzero and $c = -1$ if v is zero.

Proof. Although the statement is slightly different from Lemma 2.4 in [SU15] due to drawing $v \sim D_p$, the proof is almost verbatim; we include it for completeness.

Observe that $\mathop{\mathbb{E}}_{v \sim D_p}[\phi^p(c)] = p \cdot \phi^p(-1) + (1-p) \cdot \phi^p(1)$, since $\mathop{\mathbf{Pr}}_{v \sim D_p}[v = 0] = p$. Then we have $\mathop{\mathbb{E}}_{v \sim D_p}[\phi^p(c)] = 0$ and $\mathop{\mathbb{E}}_{v \sim D_p}\left[(\phi^p(c))^2\right] = 1$. Moreover, for $c \in \{\pm 1\}$, we have $|\phi^p(c)| \leq \frac{1}{\sqrt{\tau}}$. Thus, we have $|\phi^p(c) \cdot t| \leq \frac{1}{2}$.

Since $e^x \leq 1 + x + x^2$ for $x \in \left[-\frac{1}{2}, \frac{1}{2}\right]$, then

$$\begin{aligned}\mathop{\mathbb{E}}_{v \sim D_p}\left[e^{t\phi^p(c)}\right] &\leq 1 + t \cdot \mathop{\mathbb{E}}_{v \sim D_p}[\phi^p(c)] + t^2 \cdot \mathop{\mathbb{E}}_{v \sim D_p}\left[(\phi^p(c))^2\right] \\ &= 1 + t^2 \leq e^{t^2}.\end{aligned}$$

□

Lemma IV.3. *Let $p^1, \ldots, p^m \in [\alpha, \beta]$ and $v_i \sim D_{p^j}$. Let $a^1, \ldots, a^m \in [-1, 1]$ be fixed and $\tau = \min(\alpha, 1-\beta)$. Then for all $\lambda \geq 0$,*

$$\mathbf{Pr}\left[\sum_{j \in [m]} a^j \phi^{p^j}(c_i^j) \geq \lambda\right] \leq e^{-\lambda^2/4m} + e^{-\sqrt{\tau}\lambda/4},$$

where for all $i \in [n]$, we have $c_i = 1$ if $v_i^j \neq 0$ and $c_i^j = -1$ otherwise if $v_i^j = 0$.

Proof. The proof follows exactly along the lines of Lemma 2.5 in [SU15], using $\tau = \min(\alpha, 1-\beta)$ instead due to the range of $p \in [\alpha, \beta]$ as the probability of D_p drawing a zero. We include the proof for completeness. By Lemma IV.2, for all $t \in \left[-\frac{\sqrt{\tau}}{2}, \frac{\sqrt{\tau}}{2}\right]$,

$$\mathop{\mathbb{E}}_{v}\left[e^{t\sum_{i \in [m]} a^j \phi^{p^j}(c_i^j)}\right] \leq \prod_{j \in [m]} \mathop{\mathbb{E}}_{v_i^j \sim D_{p^j}}\left[e^{ta^j \phi^{p^j}(c_i^j)}\right] \leq e^{t^2 m}.$$

By Markov's inequality, we have

$$\begin{aligned}\mathbf{Pr}\left[\sum_{i \in [m]} a^j \phi^{p^j}(c_i^j) \geq \lambda\right] &\leq \frac{\mathbb{E}\left[e^{t\sum_{i \in [m]} a^j \phi^{p^j}(c_i^j)}\right]}{e^{t\lambda}} \\ &\leq e^{t^2 m - t\lambda}.\end{aligned}$$

We set $t = \min\left(\frac{\sqrt{\tau}}{2}, \frac{\lambda}{2m}\right)$. Then for $\lambda \in [0, m\sqrt{\tau}]$, we have $t = \frac{\lambda}{2m}$ and so

$$\mathbf{Pr}\left[\sum_{i \in [m]} a^j \phi^{p^j}(c_i^j) \geq \lambda\right] \leq e^{-\lambda^2/4m},$$

and for $\lambda \geq m\sqrt{\tau}$,

$$\mathbf{Pr}\left[\sum_{i \in [m]} a^j \phi^{p^j}(c_i^j) \geq \lambda\right] \leq e^{\tau m/4 - \sqrt{\tau}\lambda/2} \leq e^{-\frac{\sqrt{\tau}\lambda}{4}}.$$

□

Theorem IV.4 (Etemadi's inequality)**.** *[Ete85] Let $X_1, \ldots, X_n$ be independent random variables and for all $k \in [n]$, let $S_k = \sum_{i=1}^k X_i$ be the k-th partial sum of the sequence $X_1, \ldots, X_n$. Then for all $\lambda > 0$,*

$$\mathbf{Pr}\left[\max_{k \in [n]} |S_k| > 4\lambda\right] \leq 4 \cdot \max_{k \in [n]} \mathbf{Pr}\left[|S_k| > \lambda\right].$$

Lemma IV.5 (Individual soundness)**.** *For all $i \in [n] \setminus \mathcal{S}$, we have*

$$\mathbf{Pr}\left[i \in I^\ell\right] \leq \frac{1}{n^2}.$$

Proof. The proof follows similarly from Proposition 2.7 in [SU15]. Consider a fixed $i \in [n] \setminus \mathcal{S}$. Since the adversary does not see c_i^j, without the loss of generality we can assume that the outputs a^j are fixed and v_i^j and then c_i^j

are subsequently drawn since $i \notin \mathcal{S}$. By Lemma IV.3, we have for every $j \in [\ell]$,

$$\mathbf{Pr}\left[s_i^j > \frac{\sigma}{4}\right] = \mathbf{Pr}\left[\sum_{k\in[j]} a^k \phi^{p_k}(c_i^k) > \frac{\sigma}{4}\right] \leq e^{-\frac{\sigma^2}{64\ell}} + e^{-\sigma\sqrt{\tau}/16}$$

Similarly, by Lemma IV.3, for every $j \in [\ell]$,

$$\mathbf{Pr}\left[s_i^j < -\frac{\sigma}{4}\right] = \mathbf{Pr}\left[\sum_{k\in[j]} a^k \phi^{p_k}(c_i^k) < -\frac{\sigma}{4}\right] \leq e^{-\frac{\sigma^2}{64\ell}} + e^{-\sigma\sqrt{\tau}/16}$$

Thus by Theorem IV.4,

$$\begin{aligned}\mathbf{Pr}\left[i \in I^j\right] &\leq \mathbf{Pr}\left[\max_{t\in[j]} |s_i^t| > \sigma\right] \\ &\leq 4 \max_{t\in[j]} \mathbf{Pr}\left[|s_i^t| > \frac{\sigma}{4}\right] \\ &\leq 8(e^{-\frac{\sigma^2}{64\ell}} + e^{-\sigma\sqrt{\tau}/16}) \leq \frac{1}{n^2}.\end{aligned}$$

□

Lemma IV.6 (Soundness)**.**

$$\mathbf{Pr}\left[|I^\ell \setminus \mathcal{S}| \geq 1\right] \leq \frac{1}{n}.$$

Proof. The proof follows similarly from Theorem 2.8 in [SU15], as follows. For $i \in [n] \setminus \mathcal{S}$, we use Y_i to denote the indicator random variable for the event $i \in I^\ell \setminus \mathcal{S}$. By Lemma IV.5, we have that $\mathbb{E}[Y_i] \leq \frac{1}{n^2}$ for all $i \in [n] \setminus \mathcal{S}$. By Markov's inequality,

$$\mathbf{Pr}\left[|I^\ell \setminus \mathcal{S}| \geq 1\right] \leq \mathbb{E}\left[\sum_{i\in[n]\setminus\mathcal{S}} Y_i\right] \leq \frac{1}{n^2}(n-r) \leq \frac{1}{n}. \quad \square$$

Lemma IV.7. *For each $i \in [n]$, let $j_i \in [\ell+1]$ be the first j such that $i \in I^j$, where we set $I^{\ell+1} = [n]$. Then for $\tau = \min(\alpha, \beta)$ and for any $J \subset [n]$,*

$$\mathbf{Pr}\left[\sum_{i\in J}(s_i^\ell - s_i^{j_i-1}) > \lambda\right] \leq e^{-\frac{\lambda^2}{4|J|\ell}} + e^{-\sqrt{\tau}\lambda/4}.$$

Proof. The proof is nearly identical to that of Lemma 2.10 in [SU15], as follows. Observe that

$$\sum_{i\in J}(s_i^\ell - s_i^{j_i-1}) = \sum_{i\in J}\sum_{j\in[\ell]} \mathbb{I}(j \geq j_i) a^j \phi^{p_j}(c_i^j),$$

where $\mathbb{I}$ denotes the standard indicator function. Again, since we zero out the i-th coordinate after time $j_t - 1$, we can take the view that the outputs a^j are fixed and then the terms $\phi^{p_j}(c_i^j)$ are subsequently drawn for $j \geq j_t$. Then by Lemma IV.3, we have

$$\mathbf{Pr}\left[\sum_{i\in J}(s_i^\ell - s_i^{j_i-1}) > \lambda\right] \leq e^{-\frac{\lambda^2}{4|J|\ell}} + e^{-\sqrt{\tau}\lambda/4},$$

as desired. □

C. Completeness

Recall that we use h to denote the size of $\mathcal{S}$, where $\mathcal{S}$ corresponds to the non-zero indices of columns in $\mathbf{S}$.

1) Fourier Analysis:

Lemma IV.8. *Let $f : \mathbb{R}^h \to \mathbb{R}$ and let $g : [0,1] \to \mathbb{R}$ be defined so that $g(p) = \mathbb{E}_{v_1,\ldots,v_h\sim D_p}[f(v)]$, where for all $i \in [h]$, $c_i = 1$ if v_i is nonzero and $c_i = -1$ if v_i is zero. Then for any $p \in [\alpha, \beta]$,*

$$\mathbb{E}_{v_1,\ldots,v_h\sim D_p}\left[f(v) \cdot \sum_{i\in[h]} \phi^p(c_i)\right] = g'(p)\sqrt{p(1-p)}$$

Proof. The analysis is similar to Lemma 2.11 of [SU15], but with differing probability distributions and thus correspondingly slightly differing score functions ϕ. We include the full proof for completeness.

For $p \in (0,1)$ and $T \subset [h]$, we define $\phi_T^p : \{\pm 1\}^h \to \mathbb{R}$ by $\phi_T^p(c) = \prod_{i\in T} \phi^p(c_i)$, so that the functions ϕ_T^p form an orthonormal basis with respect to the product distribution with bias p. Specifically, we have that for all $T, U \subset [h]$,

$$\mathbb{E}_{v_1,\ldots,v_h\sim D_p}[\phi_T^p(c) \cdot \phi_U^p(c)] = \begin{cases} 1 & T = U \\ 0 & T \neq U, \end{cases}$$

where for all $i \in [h]$, $c_i = 1$ if v_i is nonzero and $c_i = -1$ if v_i is zero. We use $c(v)$ to denote this mapping from v to c. Therefore, we can decompose f by

$$f(v) = \sum_{T\subset[h]} \widehat{f^p}(s) \cdot \phi_T^p(c(v)),$$

where we have the Fourier coefficients

$$\widehat{f^p}(T) = \mathbb{E}_{v_1,\ldots,v_h\sim D_p}\left[f(v) \cdot \phi_T^p(c(v))\right],$$

where all $T \subset [h]$. Then for $p, q \in (0,1)$, we can expand $g(q)$ by

$$\begin{aligned} g(q) &= \mathbb{E}_{v_1,\ldots,v_h\sim D_q}[f(v)] \\ &= \sum_{T\subset[h]} \widehat{f^p}(T) \cdot \mathbb{E}_{v_1,\ldots,v_h\sim D_q}[\phi_T^p(c(v))] \\ &= \sum_{T\subset[h]} \widehat{f^p}(T) \cdot \prod_{i\in T} \mathbb{E}_{v_i\sim D_q}[\phi_T^p(c_i(v_i))] \\ &= \sum_{T\subset[h]} \widehat{f^p}(T) \cdot \left(q \cdot \sqrt{\frac{1-p}{p}} + (1-q) \cdot \sqrt{\frac{p}{1-p}}\right)^{|T|} \end{aligned}$$

since for $v_i \sim D_q$, the probability that v_i is zero (and thus c_i is -1) is q. Thus, for $T \neq \emptyset$, we have

$$\begin{aligned} g'(q) = \sum_{T\subset[h]} \widehat{f^p}(T)|T| &\left(q\sqrt{\frac{1-p}{p}} + (1-q)\sqrt{\frac{p}{1-p}}\right)^{|T|-1} \\ &\cdot \left(\sqrt{\frac{1-p}{p}} + \sqrt{\frac{p}{1-p}}\right) \end{aligned}$$

and

$$g'(p) = \sum_{i\in[h]} \widehat{f}^p(\{i\}) \cdot \left(\sqrt{\frac{1-p}{p}} + \sqrt{\frac{p}{1-p}}\right).$$

Since $\widehat{f}^p(\{i\}) = \mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_p}[f(v)\cdot\phi^p(c_i(v_i))]$, then

$$\mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_p}\left[f(v)\cdot\sum_{i\in[h]}\phi^p(c_i(v_i))\right] = \sum_{i\in[h]}\widehat{f}^p(\{i\})$$
$$= \frac{g'(p)}{\sqrt{\frac{1-p}{p}}+\sqrt{\frac{p}{1-p}}} = g'(p)\cdot\sqrt{p(1-p)}.$$

□

Lemma IV.9. *Let $f:\mathbb{R}^h\to\mathbb{R}$ and let $g:[0,1]\to\mathbb{R}$ be defined so that $g(p) = \mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_p}[f(v)]$, where for all $i\in[h]$, $c_i = 1$ if v_i is nonzero and $c_i=-1$ if v_i is zero. Then there exists a constant $\zeta>0$ such that*

$$\mathop{\mathbb{E}}_{p\sim P_{\alpha,\beta}}\left[\mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_p}\left[f(v)\cdot\sum_{i\in[h]}\phi^p(c_i)\right]\right] \geq \zeta\cdot(g(\beta)-g(\alpha)).$$

Proof. The proof follows similarly from Proposition 2.12 in [SU15], as follows. Recall that $P_{\alpha,\beta}$ has probability density function $\mu(p) = \frac{C_{\alpha,\beta}}{\sqrt{p(1-p)}}$ entirely on the interval $[\alpha,\beta]$. By Lemma IV.8,

$$\mathop{\mathbb{E}}_{p\sim P_{\alpha,\beta}}\left[\mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_p}\left[f(v)\cdot\sum_{i\in[h]}\phi^p(c_i)\right]\right]$$
$$= \mathop{\mathbb{E}}_{p\sim P_{\alpha,\beta}}\left[g'(p)\cdot\sqrt{p(1-p)}\right]$$
$$= \int_\alpha^\beta g'(p)\sqrt{p(1-p)}\cdot\mu(p)\,dp$$
$$= C_{\alpha,\beta}\cdot\int_\alpha^\beta g'(p)\,dp$$
$$= C_{\alpha,\beta}\cdot(g(\beta)-g(\alpha)).$$

The proof then follows from setting $C_{\alpha,\beta}=\zeta$. □

2) Concentration:

Lemma IV.10. *Suppose that at the j-th round, the algorithm $\mathcal{A}$ has error probability $\delta_\alpha^j,\delta_\beta^j\leq c$ over the input distribution $z_{I^{j-1}}(u)$ and $z_{I^{j-1}}(v)$ where $u\in D_\alpha^n, v\in D_\beta^n$ for some small constant c, then we have there exists a function $f^j:\mathbb{R}^h\to\mathbb{R}$ that only depends on the interaction up to round $j-1$ and satisfies the value of $f^j(v)$ is decided by the coordinates of v in $\mathcal{S}^j$ and $f^j(v^j_{\mathcal{S}^j}) = a^j$ where $\mathcal{S}^j = \mathcal{S}\setminus I^{j-1}$. Moreover, we have and $g^j(\beta)-g^j(\alpha)\geq 2-\eta$ for some $\eta=\mathcal{O}(1)$.*

Proof. From the assumption we have $\mathop{\mathbb{E}}_{v_1,\ldots,v_n\sim D_\alpha}[\mathcal{A}(z_{I^{j-1}}(v))] \leq -(1-2c)$ and $\mathop{\mathbb{E}}_{v_1,\ldots,v_n\sim D_\beta}[\mathcal{A}(z_{I^{j-1}}(v))] \geq 1-2c$. From the fact that we have zeroed out the coordinates of v in I^{j-1}, we can without loss of generality get that the output a^j can be represented by the value of a function f^j that is only decided by the coordinates on $\mathcal{S}^j$. From this and the assumption that $\delta_\alpha^j,\delta_\beta^j\leq c$ we have that

$$g^j(\beta)-g^j(\alpha) = \mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_\beta}[f(v)] - \mathop{\mathbb{E}}_{v_1,\ldots,v_h\sim D_\alpha}[f(v)] \geq 2-\eta$$

for some $\eta=\mathcal{O}(1)$. □

For $p\sim P_{\alpha,\beta}$ and $v_1,\ldots,v_h\sim D_p$, we set $\xi_{\alpha,\beta}(f) = f(v)\cdot\sum_{i\in[h]}\phi^p(c_i)$, where for all $i\in[h]$, $c_i=1$ if v_i is nonzero and $c_i=-1$ if v_i is zero.

Lemma IV.11. *Let $f:\mathbb{R}^h\to\{\pm1\}$, $\tau=\min(\alpha,1-\beta)$ and $t\in\left[-\frac{\sqrt{\tau}}{8},\frac{\sqrt{\tau}}{8}\right]$. Then for $C=\frac{32e^{h\tau/64}}{\tau}$, we have*

$$\mathbb{E}\left[e^{t\xi_{\alpha,\beta}(f)-\mathbb{E}[t\xi_{\alpha,\beta}(f)]}\right]\leq e^{Ct^2}.$$

Proof. Let $Y=\sum_{i\in[h]}\phi^p(c_i)$. By Lemma IV.2 and independence, we have that

$$\mathbb{E}\left[e^{tY}\right] = \mathbb{E}\left[e^{t\sum_{i\in[n]}\phi^p(c_i)}\right] = \left(\mathop{\mathbb{E}}_{v\sim D_p}\left[e^{t\phi^p(c)}\right]\right)^h \leq e^{t^2h}$$

for $t\in\left[-\frac{\sqrt{\tau}}{8},\frac{\sqrt{\tau}}{8}\right]$. Pick $t\in\{\pm\frac{\sqrt{\tau}}{8}\}$ such that

$$\sum_{k=0}^\infty\frac{t^{2k+1}}{(2k+1)!}\mathbb{E}\left[Y^{2k+1}\right]\geq 0.$$

Then by dropping the positive terms, for all $j\geq1$, we have

$$0\leq\mathbb{E}\left[Y^{2j}\right]\leq\frac{(2j)!}{t^{2j}}\sum_{k=0}^\infty\frac{t^k}{k!}\mathbb{E}\left[Y^k\right] = \frac{(2j)!}{t^{2j}}\mathbb{E}\left[e^{tY}\right]$$
$$\leq\frac{(2j)!}{t^{2j}}e^{t^2h} = \frac{8^j(2j)!}{\tau^j}e^{\tau h/64}.$$

This means that we have bounded the even moment of Y. For $k=2j+1\geq3$, by Cauchy-Schwartz,

$$\mathbb{E}\left[|Y|^k\right]\leq\sqrt{\mathbb{E}\left[Y^{2j}\right]\cdot\mathbb{E}\left[Y^{2j+2}\right]}$$
$$\leq\sqrt{\frac{8^j(2j)!}{\tau^j}e^{h\tau/64}\frac{8^{j+1}(2j+2)!}{\tau^{j+1}}e^{h\tau/64}}$$
$$=\frac{8^{k/2}k!}{\tau^{k/2}}e^{h\tau/64}\sqrt{\frac{k+1}{k}}.$$

Since $|f(c)|\leq1$, we have $\mathbb{E}\left[|f(c)\cdot Y|^k\right]\leq\mathbb{E}\left[|Y|^k\right]\leq 2\cdot8^{k/2}k!e^{h\tau/64}/\tau^{k/2}$.

For $t \in \left[-\frac{\sqrt{\tau}}{8}, \frac{\sqrt{\tau}}{8}\right]$, we have

$$\begin{aligned}
\mathbb{E}\left[e^{t\xi_{\alpha,\beta}(f)}\right] &\le 1 + t\mathbb{E}\left[\xi_{\alpha,\beta}(f)\right] + \sum_{k=2}^{\infty} \frac{|t|^k}{k!}\mathbb{E}\left[|\xi_{\alpha,\beta}(f)|^k\right] \\
&\le 1 + t\mathbb{E}\left[\xi_{\alpha,\beta}(f)\right] + \sum_{k=2}^{\infty} \frac{|t|^k}{k!}\frac{2\cdot 8^{k/2}k!e^{h\tau/64}}{\tau^{k/2}} \\
&= 1 + t\mathbb{E}\left[\xi_{\alpha,\beta}(f)\right] + 2e^{h\tau/64}\sum_{k=2}^{\infty}\left(\frac{\sqrt{8}t}{\sqrt{\tau}}\right)^k \\
&\le 1 + t\mathbb{E}\left[\xi_{\alpha,\beta}(f)\right] + 2e^{h\tau/64}\sum_{k=2}^{\infty}\left(\frac{\sqrt{8}t}{\sqrt{\tau}}\right)^2 (\sqrt{8})^{-(k-2)} \\
&\le 1 + t\mathbb{E}\left[\xi_{\alpha,\beta}(f)\right] + 32e^{h\tau/64}\frac{t^2}{\tau} \\
&\le e^{t\mathbb{E}[\xi_{\alpha,\beta}(f)]+Ct^2} .
\end{aligned}$$

□

Theorem IV.12 (Azuma-Doob Inequality, Theorem 2.16 in [SU15])**.** *Let $X_1, \ldots, X_m \in \mathbb{R}$, $\mu_1, \ldots, \mu_m \in \mathbb{R}$, and $\mathcal{U}_0, \ldots, \mathcal{U}_m \in \Omega$ be random variables such that for all $i \in [m]$:*

- *X_i and $\mathcal{U}_{i-1}$ are fixed by $\mathcal{U}_i$*
- *μ_i is fixed by $\mathcal{U}_{i-1}$.*

Suppose that for all $i \in [m]$, $u \in \Omega$, and $t \in [-c, c]$, we have

$$\mathbb{E}\left[e^{t(X_i-\mu_i)} \mid \mathcal{U}_{i-1} = u\right] \le e^{Ct^2}.$$

Then for $\lambda \in [0, 2Cmc]$, we have

$$\mathbf{Pr}\left[\left|\sum_{i\in[m]}(X_i - \mu_i)\right| \ge \lambda\right] \le 2e^{-\frac{\lambda^2}{4Cm}},$$

and for $\lambda \ge 2Cmc$, we have

$$\mathbf{Pr}\left[\left|\sum_{i\in[m]}(X_i - \mu_i)\right| \ge \lambda\right] \le 2e^{-\frac{-c\lambda}{2}}.$$

3) Lower Bounding the Correlation:

Lemma IV.13. *Let $\tau = \min(\alpha, 1-\beta)$ and let ζ be the constant from Lemma IV.9. Suppose that for every $j \in [\ell]$-th round, the algorithm $\mathcal{A}$ has error probability $\delta_\alpha^j, \delta_\beta^j \le c$ over the distribution $z_{I^{j-1}}(D_\alpha^n), z_{I^{j-1}}(D_\beta^n)$ for some small constant c, where $z_{I^{j-1}}$ means we zero out the coordinates in I^{j-1}. Then for any $\lambda \in \left[0, \frac{15\ell}{\sqrt{\tau}}\right]$,*

$$\mathbf{Pr}\left[\sum_{i\in\mathcal{S}} s_i^\ell < 2\ell\zeta(1-\eta) - \lambda\right] < 2e^{-\frac{\lambda^2\tau}{200\ell}}.$$

Proof. For each $j \in [\ell]$, from the discussion of Lemma IV.10 we can have a function $f^j : \mathbb{R}^h \to \{\pm 1\}$ that only depends on the interaction up to round $j-1$ and satisfies $f^j(v_{S^j}^j) = a^j$. Define

$$X_j = f^j(v_{S^j}^j)\sum_{i\in[h]}\phi^p(c_i^j) \sim \xi_{\alpha,\beta}(f^j) ,$$

where $\sim$ denotes that has the same distribution. We than have

$$\sum_{i\in\mathcal{S}} s_i^\ell = \sum_{j\in[\ell]} X_j .$$

From Lemma IV.9 and Lemma IV.10. We have that

$$\mu_j = \mathbb{E}\left[X_j\right] \ge 2\zeta(1-\eta)$$

for all f^j. Then, from Lemma IV.11 we have,

$$\mathbb{E}\left[e^{t(X^j-\mu_j)}\right] = \mathbb{E}\left[e^{t\xi_{\alpha,\beta}(f^j)-\mathbb{E}\left[t\xi_{\alpha,\beta}(f^j)\right]}\right] \le e^{Ct^2} .$$

Define $\mathcal{U}_j = (f^1, p^1, v^1, \cdots, f^j, p^j, v^j, f^{j+1})$. Now $X_1, ..., X_\ell$, $\mu_1, ..., \mu_\ell$, and $\mathcal{U}_1, ..., \mathcal{U}_\ell$ satisfies the condition in Lemma IV.11. For $\lambda \in [0, 2Cmc] = [0, 15\ell/\sqrt{\tau}]$, we have that

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{i\in\mathcal{S}} s_i^\ell < 2\ell\zeta(1-\eta) - \lambda\right] &\le \mathbf{Pr}\left[\left|\sum_i X_i - \mu_i\right|\right] \\
&\le 2e^{-\lambda^2/4Cm} < 2e^{-\frac{\lambda^2\tau}{200\ell}}.
\end{aligned}$$

□

Lemma IV.14. *Let $\tau = \min(\alpha, 1-\beta)$. Then for all $\lambda > 0$,*

$$\mathbf{Pr}\left[\sum_{i\in\mathcal{S}} s_i^\ell > \lambda + h\sigma + \frac{h}{\sqrt{\tau}}\right] \le e^{-\frac{\lambda^2}{4h\ell}} + e^{-\sqrt{\tau}\lambda/4}.$$

Proof. For each $i \in [n]$, let j_i be as in Lemma IV.7. That is, $i \notin \mathcal{S}^{j_i}$ and $i \in \mathcal{S}^{j_i-1}$, where we define $\mathcal{S}^{\ell+1} = \emptyset$ and $\mathcal{S}^0 = [n]$. By the definition of j_i, we have that $s_i^{j_i-2} \le \sigma$ for all $i \in \mathcal{S}$. Hence we have

$$\begin{aligned}
\sum_{i\in\mathcal{S}} s_i^{j_i-1} &= \sum_{i\in\mathcal{S}} s_i^{j_i-2} + a^{j_i-1}\phi^{j_i-1}(c_i^{j_i-1}) \\
&\le \sum_{i\in\mathcal{S}}(\sigma + \frac{1}{\sqrt{\tau}}) \le h\sigma + \frac{h}{\sqrt{\tau}}.
\end{aligned}$$

By Lemma IV.7 we have

$$\mathbf{Pr}\left[\sum_{i\in S}(s_i^\ell - s_i^{j_i-1}) > \lambda\right] \le e^{-\frac{\lambda^2}{4h\ell}} + e^{-\sqrt{\tau}\lambda/4}$$

which completes the proof. □

Lemma IV.15 (Completeness)**.** *With high constant probability, at the end of ℓ rounds of the attack, we can find a distribution on $\mathbb{Z}^n$ such that the algorithm $\mathcal{A}$ fails with constant probability when the input is sampled from this distribution.*

Proof. Suppose that at some round $j \in [\ell]$ we have $\max\{\delta_\alpha^j, \delta_\beta^j\} = \Omega(1)$, then from Chernoff's bound we have with probability at least 0.99, we can find this distribution $z_{I^{j-1}}(D_\alpha)$ or $z_{I^{j-1}}(D_\beta)$ that the algorithm $\mathcal{A}$ fails from a constant number of samples.

We next consider the other case where $\delta_\alpha^j, \delta_\beta^j \le c$ over the distribution D_α^n, D_β^n for all $j \in [\ell]$. Then from Lemma IV.13 we have

$$\sum_{i\in\mathcal{S}} s_i^\ell \ge 2\ell\zeta(1-\eta) - \lambda = \Omega(\ell)$$

with probability at least $1 - 2\exp(-\Omega(\ell))$ by setting $\lambda = \mathcal{O}(\ell)$. On the other hand, from Lemma IV.14 we have

$$\sum_{i\in\mathcal{S}} s_i^\ell < \lambda + h\sigma + \frac{h}{\sqrt{\tau}} \le 3h\sigma$$

with probability at least $1 - 2\exp(-\Omega(\sigma))$ by setting $\lambda = h\sigma$. This is a contradiction when $\ell \ge C \cdot h\sigma$ for a sufficient large constant C. □

D. Proof of Our Main Theorem

We are now ready to prove Theorem IV.1.

Proof of Theorem IV.1. First, without loss of generality, we can assume that $n = \text{poly}(r)$. This follows since we can always query on the first $\text{poly}(r)$ coordinates and make the remaining $\text{poly}(r)$ coordinates of the query vector $\mathbf{x}$ to 0 (in this case, we are attacking the first $\text{poly}(r)$ columns of the sketching matrix $\mathbf{A}$).

We now prove the correctness of our attack. Suppose that the algorithm $\mathcal{A}$ which we attack uses the estimator f, and suppose we sample $\mathbf{x} \sim D_{p^t}$ at time t. Next, we consider $\mathcal{A}'$ which uses the same estimator f, but instead takes the input $\begin{bmatrix}\mathbf{D}\mathbf{x}' \\ \mathbf{S}\mathbf{x}_S\end{bmatrix}$ where $\mathbf{x}' \sim D_\gamma^{|D|}$ for a fixed $\gamma \in [\alpha, \beta]$, which is independent of input $\mathbf{x}$. Since $\mathbf{D}\mathbf{x}$ and $\mathbf{S}\mathbf{x}$ are independent conditioned on p^t, by Lemma V.4, we know that for each iteration t, the total variation distance between $\begin{bmatrix}\mathbf{D}\mathbf{x}_D^{(t)} \\ \mathbf{S}\mathbf{x}_{\mathcal{S}}^{(t)}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{D}\mathbf{x}' \\ \mathbf{S}\mathbf{x}_{\mathcal{S}}^{(t)}\end{bmatrix}$ is at most $1/\text{poly}(n)$. Therefore, we have that

$$d_{\text{tv}}\left(\{\mathcal{A}(\mathbf{x}^{(t)})\}_{t=1,2,\cdots,\ell}, \{\mathcal{A}'(\mathbf{x}^{(t)})\}_{t=1,2,\cdots,\ell}\right) \le \ell \cdot \frac{1}{\text{poly}(n)} = \frac{1}{\text{poly}(n)}$$

Hence, it suffices for us to show that by interacting with $\mathcal{A}'$, we can find the attack distribution on which $\mathcal{A}'$ fails with high constant probability. Note that $\mathcal{A}'$ has the property that it only uses $x_{\mathcal{S}}$ in the computation. From Lemma IV.6, we see that with probability at least $1 - 1/n$, we never falsely accuse any index $i \notin \mathcal{S}$; Additionally, by Lemma IV.15, we know that with high constant probability, our attack correctly identifies (some, or all) coordinates $i \in S$ and outputs a distribution on which $\mathcal{A}'$ fails. From the above discussion we can see that the algorithm $\mathcal{A}$ must also fail on this distribution with constant probability. By conditioning on these two events and taking a union bound, it follows that our attack finds some hard query distribution $\mathbf{q}$ on which $\mathcal{A}$ fails with constant probability.

Next, we analyze the query complexity and time complexity of our attack. In each of the ℓ iterations, we make $\mathcal{O}(1)$ queries. Thus, the total number of queries is $\mathcal{O}(\ell) = \mathcal{O}\left(r^8 \log^7 n\right) = \tilde{\mathcal{O}}\left(r^8\right)$. Since we only maintain the accumulated score s_i^t in each iteration $t \in [\ell]$, the total runtime of the attack is $\mathcal{O}(\ell n) = \text{poly}(r)$, since it suffices to consider $n = \text{poly}(r)$. □

V. Constructing the Hard Input Distribution

In this section, we give the construction of the hard distribution family that is used in Section IV. We will make use of the following lemma.

Lemma V.1 (Claim 1 of [LWY20])**.** *For every $\varepsilon > 2^{-\mathcal{O}(R)}$, there exists a univariate polynomial Q of degree at most $R - \Omega\left(\sqrt{R\log\frac{1}{\varepsilon}}\right)$ such that*

$$|Q(0)| > \varepsilon \cdot \sum_{i=0}^{R} \left|\binom{R}{i} \cdot Q(i)\right| = \varepsilon.$$

Furthermore, this polynomial Q has the property that

$$\sum_{i=0}^{R} (-1)^i \binom{R}{i} \cdot Q(i) \cdot i^t = 0.$$

for all non-negative integers $t \le \mathcal{O}\left(\sqrt{R\log\frac{1}{\varepsilon}}\right)$.

Lemma V.2. *For any K, there exist constants $0 \le \alpha < \beta \le 1$ such that there exists a family $\mathcal{D} = \{D_p\}$ of probability distributions parameterized by $p \in [\alpha, \beta]$ with support on $\{-R, \ldots, R\}$ where $R = \mathcal{O}\left(K^2\right)$ such that:*

1. *For $D_p \in \mathcal{D}$, we have $D_p(0) = p$ and $D_p(1) = \Omega(1)$.*
2. *For all $p \in [\alpha, \beta]$ and for all $X \in [R]$, we have $D_p(X) = D_p(-X)$, so that D_p is a symmetric distribution.*
3. *For all $p, q \in [\alpha, \beta]$, we have $\mathop{\mathbb{E}}_{X\sim D_p}\left[X^k\right] = \mathop{\mathbb{E}}_{X\sim D_q}\left[X^k\right]$ for all $k \in [K]$.*

Proof. By Lemma V.1 with $R = \Theta(K^2)$ and $\varepsilon = 1/4$, there exists a univariate polynomial Q of degree at most $R - \Omega\left(\sqrt{R}\right)$ such that

$$|Q(0)| > \frac{1}{4} \cdot \sum_{i=0}^{R} \left|(-1)^i \binom{R}{i} \cdot Q(i)\right|.$$

Moreover, for every non-negative integer $t \le K$, we have

$$\sum_{i=0}^{R} (-1)^i \binom{R}{i} \cdot Q(i) \cdot i^t = 0.$$

Let $u(i) = (-1)^i\binom{R}{i} \cdot Q(i)$ for all $i \in [R]$ and let $U = \sum_{i\in[R]} |u(i)|$. Without loss of generality, suppose $Q(0) > 0$, so that $u(0) > 0$ and $u(0) > \frac{1}{4} \cdot U$. Moreover, since $\sum_{i\in[R]} u(i) = 0$, then $u(0) \le \frac{1}{2} \cdot U$.

We set $\alpha = \left|\frac{u(0)}{2U}\right|$ and $\beta = 2\left|\frac{u(0)}{2U}\right|$. We first define:

$$B(i) = \begin{cases} \left|\frac{u(0)}{2U}\right|, & i = 0 \\ \frac{1}{2}\left(\frac{1}{2} + \left|\frac{u(1)}{2U}\right|\right) & i = \pm 1 \\ \frac{1}{2}\left(\left|\frac{u(i)}{2U}\right|\right), & |i| \in \{2, \ldots, R\} \end{cases}$$

Then for a fixed $p \in [\alpha, \beta]$, we define

$$D_p(0) = B(0) + \left(\frac{p}{\alpha} - 1\right) \cdot \frac{u(0)}{2U},$$

and

$$D_p(i) = B(i) + \left(\frac{p}{\alpha} - 1\right) \cdot \frac{u(i)}{4U},$$

for all i with $|i| \in \{1, \ldots, R\}$.

We first prove that $D_p(i)$ is a probability distribution. Since $\sum_{i=0}^{R} |u(i)| = U$, then $\sum_{i=0}^{R} \frac{|u(i)|}{2U} = \frac{1}{2}$, and thus

$$\sum_{i:|i|\in\{0,1,\ldots,R\}} B(i) = \frac{1}{2} + \sum_{j=0}^{R} \frac{|u(j)|}{2U} = 1.$$

Moreover, since $|u(i)| \leq \frac{U}{2}$, then $B(i) \in [0,1]$ for all i and thus B is a probability distribution. We also have $\sum_{i=0}^{R} \frac{u(i)}{U} = 0$. Thus we have

$$\sum_{i:|i|\in\{0,1,\ldots,R\}} D_p(i) =$$

$$= \left(\sum_{i:|i|\in\{0,1,\ldots,R\}} B(i)\right) + \left(\sum_{i:i\in\{0,1,\ldots,R\}} \left(\frac{p}{\alpha} - 1\right) \frac{u(i)}{2U}\right)$$

$$= \sum_{i:|i|\in\{0,1,\ldots,R\}} B(i) = 1.$$

We also have $\sum_i \frac{|u(i)|}{2U} = \frac{1}{2}$ and thus $\frac{|u(i)|}{2U} \leq \frac{1}{2}$. Moreover, note that for $p \in [\alpha, \beta]$ with $\alpha = \left|\frac{u(0)}{2U}\right|$ and $\beta = 2\left|\frac{u(0)}{2U}\right|$, then $\left(\frac{p}{\alpha} - 1\right) \in [0,1]$. Thus $D_p(i) \in [0,1]$ for all i and so D_p is a valid probability distribution.

By construction, we have

$$\begin{aligned} D_p(0) &= \left|\frac{u(0)}{2U}\right| + \left(\frac{p}{\alpha} - 1\right) \cdot \frac{u(0)}{2U} \\ &= \left|\frac{u(0)}{2U}\right| + \left(\frac{2Up}{|u(0)|} - 1\right) \cdot \frac{u(0)}{2U} \\ &= p, \end{aligned}$$

since $u(0) > 0$ by assumption. Hence, the first part of the claim follows.

By construction, we have D_p is symmetric distribution for all $p \in [\alpha, \beta]$, which gives the second part of the claim.

It thus remains to prove the third part of the claim. Let $p < q$ be fixed, for $p, q \in [\alpha, \beta]$. To that end, observe that $\mathop{\mathbb{E}}_{X\sim D_p}\left[X^j\right] = \mathop{\mathbb{E}}_{X\sim D_q}\left[X^j\right]$ if and only if $\sum_{X\in[R]} X^j \cdot (D_p(X) - D_q(X)) = 0$. Now, for each $X \in [R]$, we have $D_p(X) - D_q(X) = \frac{q-p}{\alpha} \cdot \frac{u(X)}{2U}$. Since $u(X) = (-1)^X \binom{R}{X} \cdot Q(X)$, then it suffices to show that $\sum_{X\in[R]} X^j \cdot (-1)^X \binom{R}{X} \cdot Q(X) = 0$, which is true by Lemma V.1. Thus, the third part of the claim follows.

As an alternative view, we can first observe that since D_p and D_q are symmetric distributions, then their odd moments are all 0. To match their even moments, we can define $\mathbf{M} \in \mathbb{R}^{K\times R}$ be the following transposition of a Vandermonde matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 4 & 9 & \ldots & R^2 \\ 1 & 16 & 81 & \ldots & R^4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{2K} & 3^{2K} & \ldots & R^{2K} \end{bmatrix},$$

then $\mathop{\mathbb{E}}_{X\sim D_p}\left[X^{2j}\right]$ is the j-th row of the matrix-vector product $\mathbf{Mv}$, where $v_i = 2 \cdot D_p(i)$. Similarly, $\mathop{\mathbb{E}}_{X\sim D_q}\left[X^{2j}\right]$ is the j-th row of the matrix-vector product $\mathbf{Mv'}$, where $v'_i = 2 \cdot D_q(i)$ and thus $\mathop{\mathbb{E}}_{X\sim D_p}\left[X^{2j}\right] = \mathop{\mathbb{E}}_{X\sim D_q}\left[X^{2j}\right]$ if and only if $\mathbf{Mv} - \mathbf{Mv'} = 0^K$, i.e., the all zeros vector of length K, so that $\mathbf{v} - \mathbf{v'}$ is in the kernel of $\mathbf{M}$. Now, the j-th entry of $\mathbf{Mv} - \mathbf{Mv'}$ is precisely $2\sum_{X\in[R]} X^j \cdot (D_p(X) - D_q(X)) = 0$ and we proceed as before. □

A. Bounding the Total Variation Distance

Let $\mathbf{D}$ denote the dense part of sketching matrix $\mathbf{A}$, and let $\mathbf{x} \sim D_p^n$ and $\mathbf{x'} \sim D_q^n$, respectively. Before we proceed to prove that $d_{\mathrm{tv}}(\mathbf{Dx}, \mathbf{Dx'}) \leq \frac{1}{\mathrm{poly}(n)}$, we state the following useful lemma.

Lemma V.3. $|\prod_{i\in[n]}(a_i + \delta_i) - \prod_{i\in[n]} a_i| \leq \sum_{i\in[n]} |\delta_i| \cdot e^{\sum_{j\in[n]} |\delta_j|}$ *if* $|a_i + \delta_i| \leq 1$ *for all* $i \in [n]$.

Proof. We have

$$\begin{aligned} &\left|\prod_{j<i}(a_j + \delta_j) \prod_{j\geq i} a_j - \prod_{j<i+1}(a_j + \delta_j) \prod_{j\geq i+1} a_j\right| \\ &= |\delta_i| \cdot \prod_{j>i} |a_j + \delta_j| \prod_{j\geq i+1} |a_j|. \end{aligned}$$

Since $|a_j + \delta_j| \leq 1$, then we have $|a_j| \leq 1 + |\delta_j|$ by triangle inequality. Thus,

$$\begin{aligned} &\left|\prod_{j<i}(a_j + \delta_j) \prod_{j\geq i} a_j - \prod_{j<i+1}(a_j + \delta_j) \prod_{j\geq i+1} a_j\right| \\ &\leq |\delta_i| \cdot \prod_{j\geq i+1} (1 + |\delta_j|) \\ &\leq |\delta_i| \prod_{j\in[n]} e^{|\delta_i|} \leq |\delta_i| \cdot e^{\sum_j |\delta_j|}. \end{aligned}$$

Now, note that we can write

$$\begin{aligned} &\left|\prod_{i\in[n]}(a_i + \delta_i) - \prod_{i\in[n]} a_i\right| \\ &= \sum_{i=1}^{n} \left|\prod_{j<i}(a_j + \delta_j) \prod_{j\geq i} a_j - \prod_{j<i+1}(a_j + \delta_j) \prod_{j\geq i+1} a_j\right|. \end{aligned}$$

Therefore, we have that

$$\left|\prod_{i\in[n]}(a_i + \delta_i) - \prod_{i\in[n]} a_i\right| \leq \sum_{i\in[n]} |\delta_i| \cdot e^{-\sum_{j\in[n]} |\delta_j|}.$$

□

Lemma V.4. *For fixed* p *and* $p' \in [\alpha, \beta]$, *let* $P = D_p$ *and* $Q = D_{p'}$ *be the pair of probability distributions defined in Lemma V.2. Let* P^n *and* Q^n *be the probability distributions of vectors of dimension* n, *with each entry drawn*

independently from P and Q, respectively. Let $\mathbf{D} \in \mathbb{Z}^{r\times n}$ with entries bounded in $[-\operatorname{poly}(n), \operatorname{poly}(n)]$ and

$$|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})_j|^2 \le \frac{1}{s}\cdot \|\textsc{Frac}(\mathbf{y}^\top \mathbf{D})\|_2^2 \,.$$

for all $\mathbf{y} \in \mathbb{R}^r$ and $j \in [n]$. Let $P_{\mathbf{D}}$ and $Q_{\mathbf{D}}$ be the probability distributions of $\mathbf{Dx}$ and $\mathbf{Dx}'$ for $\mathbf{x} \sim P^n$ and $\mathbf{x}' \sim Q^n$ respectively. Let K and R be the parameter from Lemma V.2 and s be the parameter from Lemma III.3 with $s = \Omega(R^{5/2})$. Then the total variation distance between $P_{\mathbf{D}}$ and $Q_{\mathbf{D}}$ is at most $n^{\mathcal{O}(r)}\left(n\cdot e^{-\Omega(K)} + e^{-\Omega(K)}\right)$.

Proof. For $\mathbf{u} \in [-\pi,\pi]^r$ and $\mathbf{z} = \mathbf{Dx}$, we have

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \mathop{\mathbb{E}}_{\mathbf{z}\sim P_{\mathbf{D}}}\left[e^{-\langle \mathbf{u},\mathbf{z}\rangle i}\right] = \mathop{\mathbb{E}}_{\mathbf{z}\sim P_{\mathbf{D}}}\left[e^{-\langle \mathbf{u}^\top \mathbf{Dx}\rangle i}\right].$$

We have $\mathbf{Dx} = \sum_{j\in[n]} \mathbf{D}^{(j)} x_j$, where $\mathbf{D}^{(j)}$ is the j-th column of $\mathbf{D}$. For all $i \in [R]$, let P_i be the probability that $\mathop{\mathbf{Pr}}_{X\sim P}[X = i]$. Since each coordinate of $\mathbf{x}$ is drawn independently from P, then we have

$$\begin{aligned}\widehat{P_{\mathbf{D}}}(\mathbf{u}) &= \prod_{j\in[n]} \mathbb{E}\left[e^{-\mathbf{u}^\top \mathbf{D}^{(j)} x_j i}\right] \\ &= \prod_{j\in[n]}\sum_{m\ge 0} P_m \cdot \left(\cos(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle m) + i\cdot \sin(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle m)\right).\end{aligned}$$

Since $P_i = P_{-i}$, then we have

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{m\ge 0} P_m \cdot \cos(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle m).$$

As before, we define $\textsc{Frac}(x) = x - \mathsf{int}(x) \in \left[-\frac{1}{2}, \frac{1}{2}\right)$ and $\textsc{Frac}_{2\pi}(x) = 2\pi\cdot \textsc{Frac}\left(\frac{x}{2\pi}\right) \in [-\pi,\pi)$, so that $\cos(m\theta) = \cos\left(m\cdot \textsc{Frac}_{2\pi}(\theta)\right)$. Then

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{m\ge 0} P_m \cdot \cos\left(m\cdot \textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right).$$

Rewriting $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ in its Taylor expansion, we have

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{m\ge 0} P_m \sum_{k\ge 0} \frac{\left(m\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}(-1)^k}{(2k)!}$$

Since $\cos(x)$ is well-defined, the summation is absolutely convergent, and so

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{k\ge 0}\left(\sum_{m\ge 0} P_m m^{2k}\right) \cdot \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}(-1)^k}{(2k)!}$$

Let $M_P(2k) = \left(\sum_{m\ge 0} P_m\cdot m^{2k}\right)$ be the $2k$-th moment of P and $M_Q(2k) = \left(\sum_{m\ge 0} Q_m\cdot m^{2k}\right)$, so that

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{k\ge 0} M_P(2k)\cdot \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}}{(2k)!}\cdot(-1)^k$$

and similarly

$$\widehat{Q_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{k\ge 0} M_Q(2k)\cdot \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}}{(2k)!}\cdot(-1)^k.$$

We claim $|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})| \le n\cdot e^{-\Omega(K)} + e^{-\Omega(K)}$ for all $\mathbf{u} \in [-\pi,\pi]^n$. Now, for a fixed $\mathbf{u}$, either there exists $j \in [n]$ such that $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| > \frac{1}{4K}$ or for all $j \in [n]$, we have $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| \le \frac{1}{4K}$. We analyze these cases separately.

Suppose there exists $j \in [n]$ such that $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| > \frac{1}{4K}$. We write $\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j := \textsc{Frac}_{2\pi}\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle$. Then $|\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j| > \frac{1}{4K}$ and the definition $\textsc{Frac}_{2\pi}(x) = 2\pi\cdot \textsc{Frac}\left(\frac{x}{2\pi}\right) \in [-\pi,\pi)$ implies

$$\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j^2 = \left|\textsc{Frac}\left(\left\langle \frac{\mathbf{u}}{2\pi}, \mathbf{D}^{(j)}\right\rangle\right)^2\right| > \frac{1}{(16K^2)\cdot 2\pi}.$$

Since we have $|\textsc{Frac}(\mathbf{y}^\top\mathbf{D})_j|^2 < \frac{1}{s}\cdot\|\textsc{Frac}(\mathbf{y}^\top\mathbf{D})\|_2^2$ for all vectors $\mathbf{y} \in \mathbb{R}^r$, then it follows that

$$\left\|\iota\left(\frac{\mathbf{u}}{2\pi}\right)\right\|_2^2 \ge \frac{s}{(16K^2)2\pi} = \frac{K}{32\pi}.$$

by setting $s = \mathcal{O}\left(K^3\right)$. From before, we have

$$\begin{aligned}|\widehat{P_{\mathbf{D}}}(\mathbf{u})| &= \left|\prod_{j\in[n]}\sum_{m\ge 0} P_m\cdot\cos\left(m\cdot \textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)\right| \\ &= \prod_{j\in[n]}\left|\sum_{m\ge 0} P_m\cdot\cos\left(m\cdot\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j\cdot 2\pi\right)\right|.\end{aligned}$$

Since we have $P_1 = \Omega(1)$, then

$$\begin{aligned}|\widehat{P_{\mathbf{D}}}(\mathbf{u})| &\le \prod_{j\in[n]}\left|1 - P_1\left(1-\cos\left(m\cdot\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j\cdot 2\pi\right)\right)\right| \\ &\le \prod_{j\in[n]} e^{-\Omega\left(\left(\iota\left(\frac{\mathbf{u}}{2\pi}\right)_j\right)^2\right)},\end{aligned}$$

where the last inequality holds by the Taylor expansion $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ and the inequality $1 - x \le e^{-x}$. We thus have

$$\begin{aligned}|\widehat{P_{\mathbf{D}}}(\mathbf{u})| &\le e^{-\Omega\left(\left\|\iota\left(\frac{\mathbf{u}}{2\pi}\right)\right\|_2^2\right)} \\ &\le e^{-\Omega(K)},\end{aligned}$$

and similarly $|\widehat{Q_{\mathbf{D}}}(\mathbf{u})| \le e^{-\Omega(K)}$. Thus in this case, $|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})| \le e^{-\Omega(K)}$, by triangle inequality.

In the other case, we have that for all $j \in [n]$, $|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)| \le \frac{1}{4K}$. From before, we have

$$\widehat{P_{\mathbf{D}}}(\mathbf{u}) = \prod_{j\in[n]}\sum_{k\ge 0} M_P(2k)\cdot \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)}\rangle)\right)^{2k}}{(2k)!}\cdot(-1)^k.$$

At this point, we recall that $R = \mathcal{O}\left(K^2\right)$ by Lemma V.2. So, using $R = \mathcal{O}\left(K^2\right)$ and the fact that for all $j \in [n]$,

$|\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)} \rangle)| \leq \frac{1}{4K}$ (as well as Stirling's approximation), we can upper bound the higher moments as follows:

$$\left| \sum_{k \geq K/2} M_P(2k) \cdot \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)} \rangle)\right)^{2k}}{(2k)!} \cdot (-1)^k \right|$$
$$\leq \sum_{k > K/2} R^{2k} \cdot \frac{1}{(2k)!} \cdot \left(\frac{1}{16K^2}\right)^k$$
$$\leq \frac{K^{4K}}{(2K)^{2K}/e^{2K} \cdot \sqrt{4\pi K} \cdot (16)^K} \cdot \frac{1}{K^{2K}} \leq e^{-\Omega(K)}$$

We now apply Lemma V.3 with $a_j = \sum_{k \leq K/2} M_P(2k)$ and $\delta_j = \sum_{k > K/2} M_P(2k)$ so that

$$\left| \widehat{P_{\mathbf{D}}}(\mathbf{u}) - \prod_{j \in [n]} \sum_{k \leq K/2} M_P(2k) \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)} \rangle)\right)^{2k} (-1)^k}{(2k)!} \right|$$
$$\leq n \cdot e^{-\Omega(K)}.$$

Similarly, we have

$$\left| \widehat{Q_{\mathbf{D}}}(\mathbf{u}) - \prod_{j \in [n]} \sum_{k \leq K/2} M_Q(2k) \frac{\left(\textsc{Frac}_{2\pi}(\langle \mathbf{u}, \mathbf{D}^{(j)} \rangle)\right)^{2k} (-1)^k}{(2k)!} \right|$$
$$\leq n \cdot e^{-\Omega(K)}.$$

Moreover, we have $M_Q(2k) = M_Q(2k)$ for $k \leq K/2$ and thus by triangle inequality, we have $|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})| \leq n \cdot e^{-\Omega(K)}$.

Thus, combining both cases, we have $|\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u})| \leq n \cdot e^{-\Omega(K)} + e^{-\Omega(K)}$ for all $\mathbf{u} \in [-\pi, \pi]^n$, as desired. Now, we have

$$|P_{\mathbf{D}}(\mathbf{x}) - Q_{\mathbf{D}}(\mathbf{x})|$$
$$= \left| \frac{1}{(2\pi)^r} \int_{[-\pi,\pi)^r} e^{i\langle \mathbf{u}, \mathbf{x} \rangle} \left(\widehat{P_{\mathbf{D}}}(\mathbf{u}) - \widehat{Q_{\mathbf{D}}}(\mathbf{u}) \right) d\mathbf{u} \right|$$
$$\leq n \cdot e^{-\Omega(K)} + e^{-\Omega(K)}.$$

Finally, we observe that since $\mathbf{D} \in \mathbb{Z}^{r \times n}$ with entries bounded in $[-\operatorname{poly}(n), \operatorname{poly}(n)]$, then $P_{\mathbf{D}}(\mathbf{x})$ and $Q_{\mathbf{D}}(\mathbf{x})$ only have support on a set of size $n^{\mathcal{O}(r)}$. Thus, we have that

$$d_{\mathrm{tv}}(P_{\mathbf{D}}(\mathbf{x}), Q_{\mathbf{D}}(\mathbf{x})) \leq n^{\mathcal{O}(r)} \left(n \cdot e^{-\Omega(K)} + e^{-\Omega(K)} \right)$$

□

At this point, we note that $s = \mathcal{O}(K^3)$ in the proof of Lemma V.4. So, by setting $K = r \log n$, we see that $s = \mathcal{O}((r \log n)^3)$. For this choice of parameters s, K, we get that $d_{\mathrm{tv}}(P_{\mathbf{D}}(\mathbf{x}), Q_{\mathbf{D}}(\mathbf{x})) \leq \frac{1}{\operatorname{poly}(n)}$, as desired.

VI. Attack against Linear Sketches over Finite Fields

In this section, we present our attack against linear sketches for ℓ_0-estimation in the case that the sketching matrix $\mathbf{A} \in \mathbb{F}_p^{r \times n}$ and inputs $\mathbf{x} \in \mathbb{F}_p^n$ come from a finite field for some prime p. Formally, we have the following theorem.

Theorem VI.1. *There exists an adaptive attack that makes $\tilde{\mathcal{O}}(r^3)$ queries and with high constant probability outputs a distribution D over $\mathbb{Z}^n$ such that when $\mathbf{x} \sim D$, $\mathcal{A}$ fails to distinguish between $\|x\|_0 \leq 1.1n$ and $\|x\|_0 \geq 1.9n$ with constant probability.*

Algorithm 1 Attack on L_0 algorithms that use a sketching matrix over $\mathbb{F}_p^{r \times n}$

Input: Algorithm $\mathcal{A}$ that decides whether input vector $\mathbf{x}$ satisfies $\|\mathbf{x}\|_0 \leq 1.1r$ or $\|\mathbf{x}\|_0 \geq 1.9r$, using a sketching matrix $\mathbf{A} \in \mathbb{F}_p^{r \times n}$
Output: A query distribution on which $\mathcal{A}$ does not succeed with constant probability.

```
T ← ∅
while |T| < r do
    Randomly choose R ⊆ ([n] \ T) of size 2r
    Let x^(1) ∈ F_p^n be a random vector with support only on T.
    Let x^(2) ∈ F_p^n be a random vector with support only on T ∪ R.
    if A fails on x^(1) or x^(2) then
        Return this distribution x^(i).
    for ℓ = 1 to ℓ = 5 + log log r + log r do   ▷ 2^ℓ indices of TVD O(1/2^ℓ)
        if FindColumn(T, R, ℓ) outputs a column j then
            T ← T ∪ {j}
Let x^(1) ∈ F_p^n be a random vector with support only on T.
Let x^(2) be a random vector from F_p^n.
Return one of x^(1) and x^(2)
```

Algorithm 2 FindColumn(T, R, ℓ)

Input: Set T, Set $R, \ell \in [5 + \log\log r + \log r]$
Output: A column j that is linear independent to T

```
1: Let R^i denote the first columns of R
2: for m_3 = O(r/2^ℓ log r) times do
3:     Randomly choose i ∈ [2r]
4:     for m_4 = O(2^{2ℓ} log r) times do
5:         Randomly generate v^(3) ∈ F_p^n with support only on R^i ∪ T.
6:         Randomly generate v^(4) ∈ F_p^n with support only on R^{i+1} ∪ T.
7:         Query A on v^(3) and v^(4)
8:     Let D_3 and D_4 be the output distributions of {v^(3)} and {v^(4)}.
9:     if d_tv(D_3, D_4) ≥ 1/(2^{ℓ+3} log r) then
10:        return j
11: return FAIL
```

The full description of our algorithm is given in Algo-

rithm 1. The basic idea of our attack is due to the following observation: let T and R be two subsets of columns in $\mathbf{A}$ such that T and R have the same column span. Then $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)}, \mathbf{A}\mathbf{x}^{(2)}) = 0$, where $\mathbf{x}^{(1)} \in \mathbb{F}_p^n$ and $\mathbf{x}^{(2)} \in \mathbb{F}_p^n$ are uniformly random vectors with support on T and R, respectively (Corollary VI.3). Thus, if we can find a column-independent set T with r columns, the algorithm $\mathcal{A}$ must fail on one of the following two cases where $\mathbf{x}$ is a random vector that is on the support T or a random vector over $\mathbb{F}_p^n$, as they correspond to the different outputs of $\mathcal{A}$. Therefore, the remaining task is to devise a strategy to find the column independent set T.

Lemma VI.2. *Let T be a subset of columns in $\mathbf{A}$ and suppose column j is linearly dependent with the columns in T. Then $d_{\mathrm{tv}}(\mathbf{A}\mathbf{v}^{(1)}, \mathbf{A}\mathbf{v}^{(2)}) = 0$, i.e., the distributions of the sketch on $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are identical. Here $\mathbf{x}^{(1)} \in \mathbb{F}_p^n$ is random vector with support on T and $\mathbf{x}^{(2)} \in \mathbb{F}_p^n$ be a random vector with support on $T \cup \{j\}$.*

Proof. From the condition, we have that there exist $\alpha_1, \ldots, \alpha_{|T|} \in \mathbb{F}_p$ such that

$$\alpha_1 \mathbf{A}^{(T_1)} + \ldots + \alpha_{|T|} \mathbf{A}^{(T_{|T|})} = \mathbf{A}^{(j)}.$$

Thus there exists a one-to-one correspondence for the setting where the coordinate of $\mathbf{v}^{(1)}$ corresponding to the i-th index of T is $\beta_i \in \mathbb{F}_p$ and the setting where the coordinate of $\mathbf{v}^{(2)}$ corresponding to the i-th index of T is β_i, i.e., the coordinate of $\mathbf{v}^{(1)}$ corresponding to the i-th index of T is $\beta_i - \alpha_i$. Thus, the output distributions of the sketch on $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are identical, i.e., $d_{\mathrm{tv}}(\mathbf{A}\mathbf{v}^{(1)}, \mathbf{A}\mathbf{v}^{(2)}) = 0$. □

Corollary VI.3. *Let T and R be two subsets of columns in $\mathbf{A}$ and suppose that they have the same column span. Then $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)}, \mathbf{A}\mathbf{x}^{(2)}) = 0$, where $\mathbf{x}^{(1)} \in \mathbb{F}_p^n$ is random vector with support on T and $\mathbf{x}^{(2)} \in \mathbb{F}_p^n$ be a random vector with support on R.*

We next give some high-level intuition of our procedure that searches for this column-independent set: suppose T is the current set of linear columns found, then we randomly sample $2r$ columns in $[n] \setminus T$, and let R denote the set of these new columns. Then from the correctness guarantee of the algorithm $\mathcal{A}$ we have that $d_{\mathrm{tv}}(\mathcal{A}(\mathbf{x}^{(1)}), \mathcal{A}(\mathbf{x}^{(2)})) \geq 1/3$ (as otherwise we find the distribution on which $\mathcal{A}$ fails immediately), where $\mathbf{x}^{(1)}$ is a random vector with support on T and $\mathbf{x}^{(2)}$ is a random vector with support on $T + R$. Next let R^i denote the first i columns in R and μ_i denote the distribution of $\mathcal{A}(\mathbf{x}^{(i)})$ where $\mathbf{x}^{(i)}$ is the random vector in the support of $T \cup R^i$. From the triangle inequality we have

$$\sum_i d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \geq d_{\mathrm{tv}}(\mu_0, \mu_{2r}) \geq \frac{1}{3}. \tag{2}$$

One natural way at this point is from the above, we have there must exist j such that $d_{\mathrm{tv}}(\mu_{j-1}, \mu_j) \geq \Omega(1/r)$, and then such j should be a column that is linearly independent to the columns in T, as otherwise the total variation distance should be 0. Hence, we can enumerate all $i \in [2r]$ to find such column j (from the results in statistical testing, we can distinguish whether two binary distributions have 0 distance or have total variation distance larger than $1/r$ using $\widetilde{O}(r^2)$ samples with error probability at most $1/\operatorname{poly}(r)$ (Lemma VI.5)). However, such a way might not be optimal, as in the worst case we need to search every $i \in [2r]$. To get a better r dependence, we consider the following level-set argument: define the level set $I_0 = [\frac{1}{\log r}, 1)$ and $I_\ell = \left[\frac{1}{2^{\ell+3}\log r}, \frac{1}{2^{\ell+2}\log r}\right)$, then since $\sum_i d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \geq \frac{1}{3}$, there exists $\ell \in [5 + \log\log r + \log r]$ for which there exist at least $2^{\ell-1}$ indices i such that $d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \in I_{\ell-1}$ (Lemma VI.4). Hence, we can guess the value of ℓ, and for each value of ℓ, we use a proper sampling rate to sample the indices in $[2r]$. Note that since the range of the total variation distance is different for each ℓ, we can use different number of samples (which depends on ℓ) to do the distribution testing. This results in a better r^3 dependence.

Lemma VI.4. *Suppose that*

$$\sum_{i=0}^{2r-1} d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \geq \frac{1}{3}.$$

Define the level set $I_0 = [\frac{1}{\log r}, 1)$ and $I_\ell = \left[\frac{1}{2^{\ell+3}\log r}, \frac{1}{2^{\ell+2}\log r}\right)$. There exists $\ell \in [5 + \log\log r + \log r]$ for which there exist at least $2^{\ell-1}$ indices i such that $d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \in I_{\ell-1}$.

Proof. Suppose by way of contradiction that for all $\ell \in [5 + \log\log r + \log r]$, there exists fewer than $2^{\ell-1}$ indices i such that $d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \in I_{\ell-1}$. Let N_ℓ be the number of indices i such that $d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \in I_{\ell-1}$. Then we have

$$\sum_{i=0}^{2r-1} d_{\mathrm{tv}}(\mu_i, \mu_{i+1}) \leq \sum_{\ell=1}^{5+\log\log r+\log r} \frac{N_\ell}{2^{\ell+1}\log r} + 2r \cdot \frac{1}{32r} < \frac{1}{4}$$

□

Before proving our main theorem. We need the following result in the discrete distribution testing.

Lemma VI.5 ([CDVV14])**.** *Suppose that p and q are two distributions on $[n]$ There is an algorithm that uses $\mathcal{O}\left(\max\{n^{2/3}/\varepsilon^{4/3}, n^{1/2}/\varepsilon^2\}\right)$ samples to distinguish whether $p = q$ or $d_{\mathrm{tv}}(p, q) \geq \varepsilon$ with probability at least $2/3$.*

Note that the distributions we test is binary as the algorithm $\mathcal{A}$ only output 0 or 1. And to boost the error probability to δ, we can run $\log(1/\delta)$ independent copies and then take the majority.

We are now ready to prove our Theorem VI.1.

Proof of Theorem VI.1: Consider Algorithm 1. With probability at least $1 - 1/\operatorname{poly}(r)$, all of the distribution testing subroutines succeeded, this is because we make an extra of $\mathcal{O}(\log r)$ factor in the number of samples for each testing procedure and take a union bound. Condition on

this event, we only need to show in each iteration, with probability at least $1-1/\operatorname{poly}(r)$ we can find a new column j that is linearly independent to T.

Consider a fixed iteration and let $\mathbf{x}^{(1)}$ is a random vector with support on T and $\mathbf{x}^{(2)}$ is a random vector with support on $T+R$. We first consider the case where $d_{\mathrm{tv}}(\mathcal{A}(\mathbf{x}^{(1)}),\mathcal{A}(\mathbf{x}^{(2)})) \leq 1/3$, then from the guarantee of the algorithm $\mathcal{A}$, $\mathcal{A}$ must fail on one of the distributions.

We next consider the other case $d_{\mathrm{tv}}(\mathcal{A}(\mathbf{x}^{(1)}),\mathcal{A}(\mathbf{x}^{(2)})) \geq 1/3$. First, if during the process, FindColumn successfully finds a column j, since we assume the correctness of the property testing subroutines, this means column j must be linearly independent to T (as otherwise the total variation distance is 0). One the other hand, from Lemma VI.4, we know that there exists there exists $\ell \in [5+\log\log r+\log r]$ for which there exist at least $2^{\ell-1}$ indices i such that $d_{\mathrm{tv}}(\mu_i,\mu_{i+1}) \in I_{\ell-1}$. Since we sample $\mathcal{O}\left(\frac{r}{2^\ell}\log r\right)$ index i in this range, with probability at least $1-1/\operatorname{poly}(r)$, we can find such a j that $d_{\mathrm{tv}}(\mu_j,\mu_{j+1}) \in I_{\ell-1}$.

Now, assume that we have found such a column-independent set T with r columns. Let $\mathbf{x}^{(1)} \in \mathbb{F}_p^n$ is random vector with support on T and $\mathbf{x}^{(2)} \in \mathbb{F}_p^n$ be a random vector on $\mathbb{F}_p^n$. Recall that $\mathbf{A} \in \mathbb{F}_p^{r\times n}$, this means that we have $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)}) = 0$, which means that the algorithm $\mathcal{A}$ must fail on one of the distributions.

Finally, we analyze the query complexity. in each step of the finding of the r columns in T, we make $\log r+\log\log r+5$ guess about the value of ℓ and in each guess we sample $\mathcal{O}\left(\frac{r}{2^\ell}\log r\right)$ column j and in each sample we make $\mathcal{O}\left(2^{2\ell}\log r\right)$ samples of the two distributions, then it follows that the overall query complexity is

$$r\cdot\left(\sum_{\ell=1}^{(\log r+\log\log r+5)}\frac{r}{2^\ell}\log r\cdot 2^{2\ell}\log r\right) = r^3\cdot\operatorname{polylog}(r)\ .$$

□

VII. Attack against Real-Valued Linear Sketches

In this section, we consider the case where the sketching matrix $\mathbf{A}\in\mathbb{R}^{r\times n}$ has all subdeterminants at least $\frac{1}{\operatorname{poly}(r)}$ (note that the known sketches have this property). Formally, we prove the following theorem.

Theorem VII.1. *Suppose that $\mathbf{A}$ with the estimator f solves the $(\alpha+c,\beta-c)$- ℓ_0 gap norm problem with some constants α,β, and c, where $\mathbf{A}\in\mathbb{R}^{r\times n}$ is the sketching matrix and has all nonzero subdeterminants at least $\frac{1}{\operatorname{poly}(r)}$, and $f:\mathbb{R}^{r\times n}\to\{-1,+1\}$ is any estimator used by $\mathcal{A}$, and $\mathcal{A}$ returns $f(\mathbf{A},\mathbf{A}\mathbf{x})$ for each query $\mathbf{x}$.*

Then, there exists a randomized algorithm, which after making an adaptive sequence of queries to $\mathcal{A}$, with high constant probability can generate a distribution D on $\mathbb{R}^n$ such that $\mathcal{A}$ fails on D with constant probability. Moreover, this adaptive attack algorithm makes at most $\operatorname{poly}(r)$ queries and runs in $\operatorname{poly}(r)$ time.

We follow a similar procedure as we did for $\mathbf{x}\in\mathbb{Z}^n$, where the strategy is to design queries to learn the significant columns of the sketching matrix $\mathbf{A}$. However, since the sketching matrix $\mathbf{A}$ is real-valued, we may need to redefine the significance of columns and re-design the hard input distribution family for the insignificant coordinates. Specifically, we consider the following condition for the significance of column i:

$$\exists\mathbf{y}^\top\in\mathbb{R}^r,(\mathbf{y}^\top\mathbf{A})_i^2\geq\frac{1}{s}\cdot\|\mathbf{y}^\top\mathbf{A}\|_2^2\ .$$

Next, we argue that we can iteratively remove a (small) number of columns of a matrix $\mathbf{A}\in\mathbb{R}^{r\times n}$ such that the resulting matrix $\mathbf{A}'$ has leverage scores at most $\frac{1}{s}$ (note that since $\mathbf{A}$ and $\mathbf{x}$ are real-valued matrix and vector now, the previous information theoretic argument in Section III no longer works). Since the sum of the leverage scores is at most r, we would like to argue that we can just remove rs columns. However, this may not be true, since the leverage scores of some columns may increase when we zero-out other columns during the pre-processing. Thus, we require a more involved volume argument to bound the total number of added rows e_i, which has previously been used to bound the sum of online leverage scores [CMP20], [BDM+20].

Lemma VII.2 (Matrix determinant lemma)**.** *For any vector $\mathbf{u}\in\mathbb{R}^d$ and matrix $\mathbf{M}\in\mathbb{R}^{d\times d}$, we have*

$$\det(\mathbf{M}+\mathbf{u}\mathbf{u}^\top)=\det(\mathbf{M})\cdot(1+\mathbf{u}\mathbf{M}^{-1}\mathbf{u}).$$

Now, we show that for matrices with *bounded entries* and *bounded subdeterminants*, we can only zero-out columns with high leverage scores for a fixed number of times before the remaining columns have bounded leverage score. Among this class of matrices is the class of integer matrices with bounded entries. We remark that if each entry in a general matrix is represented using b bits, then by rescaling, this translates to an integer matrix whose entries are bounded by at most 2^b in magnitude.

We further remark that although the following statement for matrices with subdeterminant at least $\frac{1}{\operatorname{poly}(r)}$, the statement easily extends to matrices with subdeterminants at least κ by removing $\mathcal{O}\left(rs\log(\kappa nr)\right)$ columns, e.g., matrices with subdeterminants at least $\frac{1}{n^{\operatorname{poly}(r)}}$ would require $\operatorname{poly}(r)\cdot\log n$ columns to be removed.

Lemma VII.3. *Let $\mathbf{A}\in\mathbb{R}^{r\times n}$ be a matrix with nonzero entries bounded by $\operatorname{poly}(r)$ and all subdeterminants either zero or at least $\frac{1}{\operatorname{poly}(r)}$. Let $s\geq 1$ be a given parameter. Then there exists a pre-processing procedure to $\mathbf{A}$ that produces a matrix $\mathbf{A}'\in\mathbb{Z}^{r\times n}$ that zeros out at most $\mathcal{O}\left(r^2s\log(nr)\right)$ columns of $\mathbf{A}$ such that the leverage score of all columns of $\mathbf{A}'$ is at most $\frac{1}{s}$.*

Proof. Let $\mathbf{S}=\mathbf{A}\mathbf{A}^\top\in\mathbb{R}^{r\times r}$. By the matrix determinant lemma, c.f., Lemma VII.2, we have for any vector $\mathbf{u}\in\mathbb{R}^r$, $\det(\mathbf{S}+\mathbf{u}\mathbf{u}^\top)=\det(\mathbf{S})\cdot(1+\mathbf{u}^\top\mathbf{S}^{-1}\mathbf{u})$. Suppose a column $\mathbf{A}_i$ is removed from the sketching matrix $\mathbf{A}$, so that $\mathbf{S}$

decreases by $\mathbf{A}_i\mathbf{A}_i^\top$. By the matrix determinant lemma, c.f., Lemma VII.2, we have $\det(\mathbf{S}-\mathbf{A}_i\mathbf{A}_i^\top) = \det(\mathbf{S})(1-(\mathbf{A}_i^\top\mathbf{S}^{-1}\mathbf{A}_i))$. Note by the definition of leverage score, $\mathbf{A}_i^\top\mathbf{S}^{-1}\mathbf{A}_i$ is the i-th leverage score ℓ_i of the current $\mathbf{S}$. Hence, we have $\det(\mathbf{S}-\mathbf{A}_i\mathbf{A}_i^\top) = \det(\mathbf{S})(1-\ell_i)$. Observe that if $\ell_i = 1$, then the rank of $\mathbf{S}$ decreases, and the analysis can be restarted with a new linearly independent subset of columns of the matrix $\mathbf{A}$ at that time. Thus we can have $\ell_i = 1$ at most r times and for the remainder of the analysis, we shall consider the number of columns that must be removed while not decreasing the rank of $\mathbf{A}$.

Now in the case that no columns have leverage score 1, we seek to remove columns with leverage score $\ell_i > \frac{1}{s}$. In this case, we have $|\det(\mathbf{S}-\mathbf{A}_i\mathbf{A}_i^\top)| \le |\det(\mathbf{S})|\cdot\left(1-\frac{1}{s}\right)$. On the other hand, we have that $|\det(\mathbf{S})| \le \|\mathbf{S}\|_F^r \le (n\cdot\mathrm{poly}(r))^r$, since $\mathbf{S} = \mathbf{A}^\top\mathbf{A}$ so that $\|\mathbf{S}\|_F \le \|\mathbf{A}\|_F^2 \le n\cdot\mathrm{poly}(r)$ since each of the entries of $\mathbf{A}$ have magnitude at most $\mathrm{poly}(r)$. Hence after $\mathcal{O}(rs\log(nr))$ iterations of removing columns with leverage score at least $\frac{1}{s}$, we have $|\det(\mathbf{S}-\mathbf{A}_i\mathbf{A}_i^\top)| < \frac{1}{\mathrm{poly}(r)}$, which contradicts $|\det(\mathbf{S}-\mathbf{A}_i\mathbf{A}_i^\top)| \ge \frac{1}{\mathrm{poly}(r)}$, given the assumption that any subdeterminant of $\mathbf{A}$ has value at least $\frac{1}{\mathrm{poly}(r)}$. Thus, we remove $\mathcal{O}(rs\log(nr))$ columns for a given rank, and have at most r changes to the rank of the matrix. Therefore, at most $\mathcal{O}\left(r^2 s\log(nr)\right)$ columns are removed in total before no remaining columns have leverage score at last $\frac{1}{s}$. □

We next consider the construction of the hard distribution for the insignificant coordinates. Let $D = \mathcal{N}(0,1)$ and let D_p for a constant $p \in (0,1)$ be the distribution such that for $x \sim D_p$ satisfies $\mathbf{Pr}[x=0] = 1-p$. Otherwise, with probability p, $x \sim \mathcal{N}\left(0,\frac{1}{p}\right)$. Note that we can also write $x \sim D_p$ by $x = \frac{1}{\sqrt{p}}\cdot\mathrm{Bern}(p)\cdot\mathcal{N}(0,1)$, where $\mathrm{Bern}(p)$ denotes a Bernoulli random variable with parameter p, i.e., 1 with probability p and 0 with probability $1-p$, Thus, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}_1}[x] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_2}[x] = 0, \qquad \mathop{\mathbb{E}}_{x\sim\mathcal{D}_1}\left[x^2\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_2}\left[x^2\right] = 1.$$

We next turn to bound the total variation distance $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)}, \mathbf{A}\mathbf{x}^{(2)})$ where $\mathbf{x}^{(1)} \sim D_p$ and $\mathbf{x}^{(2)} \sim D_q$ for p and q randomly sampled in $(\alpha,1)$ for some small constant α. We first recall the following statement of Azuma's inequality:

Theorem VII.4 (Azuma's inequality)**.** *Let $Z_1,\ldots,Z_n$ be mean-zero random variables and $\beta_1,\ldots,\beta_n$ be upper bounds such that for all $i\in[n]$, $|Z_i| \le \beta_i$. Then*

$$\mathbf{Pr}\left[\left|\sum_{i=1}^n Z_i\right| > t\right] \le \exp\left(-\frac{t^2}{2\sum_{i\in[n]}\beta_i^2}\right).$$

We next show that a random matrix $\mathbf{B}$ formed by rescaling columns sampled from a matrix $\mathbf{A}$ is a good subspace embedding of $\mathbf{A}$.

Lemma VII.5. *Let $\gamma \ge 1$ be a fixed constant. Let $\mathbf{A} \in \mathbb{R}^{r\times n}$ and $s = \Theta\left(\frac{\gamma^2}{p^2}\cdot r^4\log r\right)$ be fixed so that no column of $\mathbf{A}$ has leverage score more than $\frac{1}{s}$. Let $\mathbf{B}\in\mathbb{R}^{r\times n}$ be a random matrix formed by sampling and scaling by $\frac{1}{p}$ each column of $\mathbf{A}$ with probability p, otherwise zeroing out the column entirely. Then with high probability, we have that simultaneously for all $\mathbf{x}\in\mathbb{R}^r$*

$$\left(1-\frac{1}{\gamma r}\right)\cdot\|\mathbf{x}^\top\mathbf{B}\|_2^2 \le \|\mathbf{x}^\top\mathbf{A}\|_2^2 \le \left(1+\frac{1}{\gamma r}\right)\cdot\|\mathbf{x}^\top\mathbf{B}\|_2^2.$$

Proof. Let $\mathbf{x}\in\mathbb{R}^r$ be any fixed vector and let $\mathbf{y} = \mathbf{A}^\top\mathbf{x} \in \mathbb{R}^n$. If $\mathbf{y}$ is the all zeros vector, then all columns of $\mathbf{A}$ have dot product 0 with $\mathbf{x}$ and so $\mathbf{y} = \mathbf{x}^\top\mathbf{B}$. Thus it suffices to consider the case where $\mathbf{y}$ is nonzero, in which case we can suppose $\mathbf{y}$ has unit L_2 norm without loss of generality. Now for $i\in[n]$, let $Z_i = \frac{1}{p}\cdot y_i^2\cdot X_i - y_i^2$, where $X_i \sim \mathrm{Bern}(p)$. Hence, we have $\mathbb{E}[Z_i] = 0$ and $|Z_i| \le \left(\frac{1}{p}-1\right)y_i^2 \le \left(\frac{1}{p}-1\right)\ell_i$, where ℓ_i is the leverage score of column i, since $\ell_i = \max_{\mathbf{v}\in\mathbb{R}^r}\frac{\langle\mathbf{v},\mathbf{a}_i\rangle^2}{\|\mathbf{A}^\top\mathbf{v}\|_2^2}$. Thus for all $i\in[n]$, we can set $\beta_i = \frac{\ell_i}{p}$, so that $\sum_{i\in[n]}\beta_i \le \sum_{i\in[n]}\frac{\ell_i}{p} \le \frac{r}{p}$. Moreover, we have that for all $i\in[r]$, $\ell_i \le \frac{1}{s}$ by our pre-processing, and thus $\beta_i \le \frac{1}{s}$. We also have $\sum_{i=1}^n y_i^2 = \|\mathbf{y}\|_2^2 = 1$. Hence for $s = \Theta\left(\frac{\gamma^2}{p^2}\cdot r^4\log r\right)$, Azuma's inequality, c.f., Theorem VII.4, implies

$$\mathbf{Pr}\left[1-\sum_{i\in[n]}2\cdot y_i^2\cdot X_i > \frac{1}{\gamma r}\right] \le \exp\left(-\frac{p^2}{\gamma^2 r^2\cdot 2\cdot\frac{r}{s}}\right) = \exp(-\Theta(\gamma^2 r\log r))$$

for $s = \Theta\left(\frac{\gamma^2}{p^2}\cdot r^4\log r\right)$.

Now we take a $\frac{1}{\gamma r}$-net $\mathcal{N}$ of the unit vectors $\mathbf{y}$ in the row-span of $\mathbf{A}$. We have $|\mathcal{N}| \le (\gamma r)^{\mathcal{O}(r)}$ and thus by a union bound over all $\mathcal{N}$, we have that all net points $\mathbf{y}'\in\mathcal{N}$ have their length preserved up to $1\pm\frac{1}{\gamma r}$. Finally to show that correctness over the net $\mathcal{N}$ implies correctness everywhere, we can view our estimation procedure as generating a diagonal sampling matrix $\mathbf{D}$ with $\sqrt{1/p}$ on the diagonal entries corresponding to the columns sampled into $\mathbf{B}$, and 0 otherwise. Thus for an arbitrary unit vector $\mathbf{y}$ in the row-span of $\mathbf{A}$, let $\mathbf{y}'$ be the vector of $\mathcal{N}$ closest to $\mathbf{y}$. Then by triangle inequality, we have

$$\begin{aligned}\|\mathbf{D}\mathbf{y}\|_2 &\le \|\mathbf{D}\mathbf{y}'\|_2 + \|\mathbf{D}(\mathbf{y}-\mathbf{y}')\|_2\\ &\le 1+\frac{1}{\gamma r}+\sqrt{2}\cdot\|\mathbf{y}-\mathbf{y}\|_2 \le 1+\mathcal{O}\left(\frac{1}{\gamma r}\right),\\ \|\mathbf{D}\mathbf{y}\|_2 &\ge \|\mathbf{D}\mathbf{y}'\|_2 - \|\mathbf{D}(\mathbf{y}-\mathbf{y}')\|_2\\ &\ge 1-\frac{1}{\gamma r}-\sqrt{2}\cdot\|\mathbf{y}-\mathbf{y}\|_2 \ge 1-\mathcal{O}\left(\frac{1}{\gamma r}\right).\end{aligned}$$

Since $\mathbf{D}\mathbf{y} = \mathbf{x}^\top\mathbf{B}$, then the desired claim follows. □

Recall the following definition of KL divergence:

Definition VII.6 (Kullback-Leibler Divergence)**.** *For continuous distributions P and Q of a random variable*

with probability densities p and q on support Ω, the KL divergence is

$$d_{\mathrm{KL}}(P||Q) = \int_{x\in\Omega} p(x)\log\frac{p(x)}{q(x)}\,dx.$$

The following statement about the KL divergence of a multivariate Gaussian distribution from another multivariate Gaussian distribution is well-known, e.g., [Duc20].

Lemma VII.7. *For $P = \mathcal{N}(\mu_1,\Sigma_1)$ and $Q = \mathcal{N}(\mu_2,\Sigma_2)$, we have $d_{\mathrm{KL}}(P||Q) = \frac{1}{2}\left(\log\frac{\det(\Sigma_2)}{\det(\Sigma_1)} - r + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^\top\Sigma_2^{-1}(\mu_2-\mu_1)\right)$.*

Recall the following relationship between total variation distance and KL divergence:

Theorem VII.8 (Pinsker's inequality)**.** *For probability distributions P and Q, we have*

$$d_{\mathrm{tv}}(P,Q) \le \sqrt{\frac{1}{2}d_{\mathrm{KL}}(P||Q)}.$$

We now upper bound the total variation distance of the image of $\mathbf{A}$ after right-multiplying with two vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ whose entries are drawn from a normal distribution and a sparse scaled normal distribution, respectively.

Lemma VII.9. *Let $\gamma \ge 1$ be any fixed constant and let $s = \Theta\left(\frac{\gamma^2}{p^2}\cdot r^4\log r\right)$. Let $D = \mathcal{N}(0,1)$ and let $D_p = \mathrm{Bern}(p)\cdot\mathcal{N}\left(0,\frac{1}{p}\right)$ for a constant $p\in(0,1)$. Let $\mathbf{x}^{(1)}\sim D$ and $\mathbf{x}^{(2)}\sim D_p$. Let $\mathbf{A}\in\mathbb{R}^{r\times n}$ be a matrix with leverage score at most $\frac{1}{s}$. Then $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)}) \le \mathcal{O}\left(\frac{1}{\gamma}\right)$.*

Proof. Since $\mathbf{x}^{(1)}$ is a multivariate Gaussian with identity covariance matrix and mean 0^n, then $\mathbf{A}\mathbf{x}^{(1)}$ is a multivariate Gaussian with mean 0^r and covariance matrix $\mathbf{A}\mathbf{A}^\top$. Let $\mathbf{x}^{(2)}\sim\mathcal{D}_2$ and let S be the support of $\mathbf{x}^{(2)}$. Note that each coordinate of $\mathbf{x}^{(2)}$ in the support of S is drawn from the Gaussian distribution $\mathcal{N}(0,\frac{1}{p})$. Therefore, $\mathbf{A}\mathbf{x}^{(2)}$ is a multivariate Gaussian with mean 0^r and covariance matrix $\mathbf{B}\mathbf{B}^\top$ for some matrix $\mathbf{B}$. By Lemma VII.7, we have that

$$\begin{aligned}&d_{\mathrm{KL}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)})\\ &= \frac{1}{2}\left(\log\frac{\det(\mathbf{B}^\top\mathbf{B})}{\det(\mathbf{A}^\top\mathbf{A})} - r + \mathrm{Tr}((\mathbf{B}^\top\mathbf{B})^{-1}(\mathbf{A}^\top\mathbf{A})\right)\end{aligned}$$

Let $\mathcal{E}$ be the event that $\left(1-\frac{1}{\gamma r}\right)^2\mathbf{B}^\top\mathbf{B} \preceq \mathbf{A}^\top\mathbf{A} \preceq \left(1+\frac{1}{\gamma r}\right)^2\mathbf{B}^\top\mathbf{B}$, so that by Lemma VII.5, $\mathbf{Pr}[\mathcal{E}] \ge 1 - \frac{1}{\mathrm{poly}(r)}$. Then we have conditioned on $\mathcal{E}$,

$$\begin{aligned}&d_{\mathrm{KL}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)} \mid \mathcal{E})\\ &\le \frac{1}{2}\left(r\cdot\log\left(1+\frac{1}{\gamma r}\right) - r + r\cdot\left(1+\frac{1}{\gamma r}\right)\right)\\ &= \mathcal{O}\left(\frac{1}{\gamma}\right)\end{aligned}$$

Let α and β be two constants such that α is close to 0 and β is close to 1.
Let $\mathcal{D}$ be the distribution family where $D_p = \mathrm{Bern}(p)\cdot\mathcal{N}(0,\frac{1}{p})$
$h \leftarrow \mathcal{O}\left(r^2 s\log r\right) = \mathcal{O}\left(r^{12}\log r\right)$, $\sigma \leftarrow \mathcal{O}(h\log(n))$, $\ell\leftarrow\mathcal{O}(h)\cdot\sigma$, $c\leftarrow\mathcal{O}(1)$
Let $z_J(v)$ denote the vector where we make v_i to 0 for all $i\in J$.
$\mathcal{A}\leftarrow$ An instantiation of the ℓ_0 gap-norm algorithm.
Initialize $s_i^0 = 0$ for all $i\in[n]$
For $j\in[\ell]$:
- Sample $u^1,\cdots,u^c\sim D_\alpha^n$ and $v^1,\cdots,v^c\sim D_\beta^n$.
- If $\mathcal{A}$ fails with constant probability on one of $z_{I^{j-1}}(u^i)$ or $z_{I^{j-1}}(v^i)$: Output this distribution as the attack.
- Sample $p^j\sim P_{\alpha,\beta}$ and $v^j\sim D_{p^j}^n$.
- For all $i\in[n]$, set $c_i^j = 1$ if $v_i^j\neq 0$ and $c_i^j = -1$ otherwise if $v_i^j = 0$.
- Query $z_{I^{j-1}}(v^j)\in\mathbb{R}^n$ and receive $a^j = \mathcal{A}(z_{I^{j-1}}(v^j))\in\{\pm1\}$ as the output.
- For $i\in[n]$, update $s_i^j\leftarrow s_i^{j-1} + a^j\cdot\phi^{p^j}(c_i^j)$.
- Set $I^j = I^{j-1}\cup\{i\in[n] \mid s_i^j > \sigma\}$ and $\mathcal{S}^{j+1} = \mathcal{S}\setminus I^j$.

Fig. 2. Construction of Our Attack over the Reals

since $\log(1+x) = \mathcal{O}(x)$ for $x\in\left(0,\frac{1}{2}\right)$, e.g., by the Taylor series expansion of $\log(1+x)$. By Theorem VII.8, we thus have

$$d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)} \mid \mathcal{E}) \le \mathcal{O}\left(\frac{1}{\gamma}\right).$$

Since $\mathbf{Pr}[\mathcal{E}] \ge 1 - \frac{1}{\mathrm{poly}(r)}$, then it follows that

$$d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)}) \le \mathcal{O}\left(\frac{1}{\gamma}\right).$$

□

Combining Lemma VII.9 and the triangle inequality, we have the following lemma immediately.

Lemma VII.10. *Let $\gamma\ge1$ be any fixed constant and let $s = \Theta(\gamma^2 r^4\log r)$. Let $D_p = \mathrm{Bern}(p)\cdot\mathcal{N}\left(0,\frac{1}{p}\right)$ and $D_q = \mathrm{Bern}(p)\cdot\mathcal{N}\left(0,\frac{1}{p}\right)$ for constant $p,q\in(0,1)$. Let $\mathbf{x}^{(1)}\sim D_p$ and $\mathbf{x}^{(2)}\sim D_q$. Let $\mathbf{A}\in\mathbb{R}^{r\times n}$ be a matrix with leverage score at most $\frac{1}{s}$. Then $d_{\mathrm{tv}}(\mathbf{A}\mathbf{x}^{(1)},\mathbf{A}\mathbf{x}^{(2)}) \le \mathcal{O}\left(\frac{1}{\gamma}\right)$.*

We are now ready to present our attack over real-valued inputs, which is shown in Figure 2. Note that this is analogous to the attack for $\mathbb{Z}^{r\times n}$, and the only difference is that we use a different input distribution and a different setting of parameters h,σ.

We now prove Theorem VII.1.

Theorem VII.1. *Suppose that* $\mathbf{A}$ *with the estimator* f *solves the* $(\alpha + c, \beta - c)$- ℓ_0 *gap norm problem with some constants* α, β, *and* c, *where* $\mathbf{A} \in \mathbb{R}^{r \times n}$ *is the sketching matrix and has all nonzero subdeterminants at least* $\frac{1}{\text{poly}(r)}$, *and* $f : \mathbb{R}^{r \times n} \to \{-1, +1\}$ *is any estimator used by* $\mathcal{A}$, *and* $\mathcal{A}$ *returns* $f(\mathbf{A}, \mathbf{Ax})$ *for each query* $\mathbf{x}$.

Then, there exists a randomized algorithm, which after making an adaptive sequence of queries to $\mathcal{A}$, *with high constant probability can generate a distribution* D *on* $\mathbb{R}^n$ *such that* $\mathcal{A}$ *fails on* D *with constant probability. Moreover, this adaptive attack algorithm makes at most* $\text{poly}(r)$ *queries and runs in* $\text{poly}(r)$ *time.*

Proof. Our argument is similar to that of Section IV. Recall that we set $\gamma = r^3$ in Lemma VII.10, which means that $s = \mathcal{O}\left(r^{10} \log r\right)$ and $r^2 s = \mathcal{O}\left(r^{12} \log r\right)$ and hence, we can assume the sketching matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{D} \\ \mathbf{S} \end{bmatrix} ,$$

where $\mathbf{S}$ has at most $r^2 s$ non-zero columns and $\mathbf{D}$ satisfies

$$\forall \mathbf{y}^\top \in \mathbb{R}^r, (\mathbf{y}^\top \mathbf{D})_i^2 \leq \frac{1}{s} \cdot \|\mathbf{y}^\top \mathbf{D}\|_2^2 . \tag{3}$$

Let $\mathcal{S}$ denote the indices of the at most $r^2 s$ non-zero columns.

a) Soundness.: Consider the coordinates $i \in I \setminus \mathcal{S}$. From the choice of the parameters we have the total variation distance of $\mathbf{D}\mathbf{x}^D$ for different $p \in [\alpha, \beta]$ is $\mathcal{O}\left(\frac{1}{r^3}\right)$ where $\mathbf{x} \sim D_p$. Then, from this we can get that there are at most $\mathcal{O}\left(r^9\right)$ coordinates $i \in I \setminus \mathcal{S}$ such that the expectation of $s_i^t - s_i^{t-1}$ is $\Omega\left(\frac{1}{r^{12}}\right)$ given $\mathbf{D}\mathbf{x}^D$, which means that with high probability there are at most $\tilde{\mathcal{O}}\left(r^9\right) = o(s)$ coordinates in $I \setminus \mathcal{S}$ will be accused in the procedure (as there are total $\tilde{\mathcal{O}}\left(r^2 s^4\right)$ number of queries).

Suppose that $\mathbf{D}$ is the matrix that satisfies (3) and $\mathbf{D}'$ is the matrix where we zero out $o(s)$ columns of $\mathbf{D}$. Then for any $\mathbf{y}^\top \in \mathbb{R}^r$, since for each remaining index i satisfies $(\mathbf{y}^\top \mathbf{D})_i^2 \leq \frac{1}{s} \cdot \|\mathbf{y}^\top \mathbf{D}\|_2^2$, then

$$\forall \mathbf{y}^\top \in \mathbb{R}^r, (\mathbf{y}^\top \mathbf{D}')_i^2 \leq \frac{1.1}{s} \cdot \|\mathbf{y}^\top \mathbf{D}'\|_2^2 .$$

Hence, in the rest of the argument, we can assume the total variation bound for $\mathbf{D}'\mathbf{x}^D$ for different p still holds.

b) Completeness.: Suppose that the algorithm $\mathcal{A}$ we attack uses the estimator f. We consider $\mathcal{A}'$ to be an algorithm that uses the same estimator f, but rather than f takes the sketch $\mathbf{Ax}$ as the input, it takes the input $\begin{bmatrix} \mathbf{D}'\mathbf{x}' \\ \mathbf{S}\mathbf{x}_S \end{bmatrix}$ where $\mathbf{x}' \sim D_\gamma^{|D|}$ for a fixed $\gamma \in [\alpha, \beta]$, which is sampled by the algorithm $\mathcal{A}$ and is independent of the input $\mathbf{x}$. From Lemma V.4, we know that for each iteration t, the total variation distance between $\begin{bmatrix} \mathbf{D}'\mathbf{x}_D^{(t)} \\ \mathbf{S}\mathbf{x}_\mathcal{S}^{(t)} \end{bmatrix}$ and $\begin{bmatrix} \mathbf{D}'\mathbf{x}' \\ \mathbf{S}\mathbf{x}_\mathcal{S}^{(t)} \end{bmatrix}$ is at most $\mathcal{O}\left(\frac{1}{r^3}\right)$ for some small constant γ. If $\mathcal{A}$ succeeds with probability at least $1 - \delta$ over some input distribution D, then over this distribution $\mathcal{A}$ will also succeed with probability at least $1 - \delta - \mathcal{O}\left(\frac{1}{r^3}\right)$.

Let us now first assume the algorithm we attack is $\mathcal{A}'$. Note that $\mathcal{A}'$ has the property that it only uses $x_\mathcal{S}$ in the computation. From Lemma IV.6, we see that with probability at least $1 - \frac{1}{n}$, we never falsely accuse any index $i \notin \mathcal{S}$. Additionally, by Lemma IV.15, we know that with high constant probability, our attack correctly identifies (some, or all) coordinates $i \in S$ and outputs a distribution on which $\mathcal{A}'$ fails (note that the increase of the $\mathcal{O}\left(\frac{1}{r^3}\right)$ in the error probability will make the $g(\beta) - g(\alpha)$ in Lemma IV.10 decrease by at most $\mathcal{O}\left(\frac{1}{r^3}\right)$, which is still $\Omega(1)$). It follows that with high constant probability, our attack finds some hard query distribution $\mathbf{q}$ on which $\mathcal{A}'$ fails with constant probability. Now, let us now consider the original algorithm $\mathcal{A}$. For the property of the total variation distance we know that with probability at least $1 - \frac{1}{r^3}$, the output of $\mathcal{A}$ can be seen as sampled from the same distribution from the output of $\mathcal{A}'$, and the probability is only over the random choice over input vector $\mathbf{x}$. Hence, without loss of generality it can be seen as the algorithm $\mathcal{A}'$ but with a $\mathcal{O}\left(\frac{1}{r^3}\right)$ more inconsistency, and the probability here is only over the choice of the random input vector $\mathbf{x}$ (note that the algorithm is based on a linear sketch and the output of the algorithm is binary). Therefore, the same argument still applies.

□

References

[ABJ+22] Miklós Ajtai, Vladimir Braverman, T. S. Jayram, Sandeep Silwal, Alec Sun, David P. Woodruff, and Samson Zhou. The white-box adversarial data stream model. In *PODS '22: International Conference on Management of Data*, pages 15–27, 2022. 2

[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. 2

[ACSS23] Idan Attias, Eden Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. In *14th Innovations in Theoretical Computer Science*, pages 8:1 – 8:19, 2023. 2

[AHLW16] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC*, pages 20:1–20:22, 2016. 3

[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. ACM, 2019. 2

[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. 21

[BEO22] Omri Ben-Eliezer, Talya Eden, and Krzysztof Onak. Adversarially robust streaming via dense-sparse tradeoffs. In *5th Symposium on Simplicity in Algorithms, SOSA@SODA*, pages 214–227, 2022. 3

[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 Systems, NeurIPS*, pages 3544–3557, 2021. 2

[BJK+02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Randomization and Approximation Techniques, 6th International Workshop, RANDOM, Proceedings*, volume 2483, pages 1–10, 2002. 3

[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. 2, 3

[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. 2

[BLV18] Elette Boyle, Rio Lavigne, and Vinod Vaikuntanathan. Adversarially robust property-preserving hash functions. Cryptology ePrint Archive, Paper 2018/1158, 2018. https://eprint.iacr.org/2018/1158. 4

[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. 2

[CA24] Eden Cohen and Sara Ahmadian. Unmasking vulnerabilities: Cardinality sketches under adaptive inputs. In *Forty-first International Conference on Machine Learning*, 2024. 2

[CDVV14] Siu On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1193–1203. SIAM, 2014. 20

[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. 2, 3

[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. 2

[CMP20] Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *Theory Comput.*, 16:1–25, 2020. 21

[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. 2

[CNSS23] Edith Cohen, Jelani Nelson, Tamás Sarlós, and Uri Stemmer. Tricking the hashing trick: A tight lower bound on the robustness of countsketch to adaptive inputs. In *Thirty-Seventh Conference on Artificial Intelligence, AAAI*, pages 7235–7243, 2023. 2

[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. 2

[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. 2

[Duc20] John Duchi. Derivations for linear algebra and optimization. 2007. *URL: http://web. stanford. edu/~ jduchi/projects/general_notes. pdf*, 2020. 23

[Ete85] Nasrollah Etemadi. On some classical results in probability theory. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 215–221, 1985. 12

[FM85] Philippe Flajolet and G Nigel Martin. Probabitistic counting algorithms for data base applications. *Journal of Computer and Systems Sciences*, 31:182–209, 1985. 3

[GGMW20] Shafi Goldwasser, Ofer Grossman, Sidhanth Mohanty, and David P. Woodruff. Pseudo-deterministic streaming. In *11th Innovations in Theoretical Computer Science Conference, ITCS*, pages 79:1–79:25, 2020. 4

[HKMM20] Avinatan Hassidim, Haim Kaplan, Yishay Mansour, and Yossi Matias. Adversarially robust streaming algorithms via differential privacy. In *Conference on Neural Information Processing Systems*, 2020. 2, 3

[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. 3

[IW05] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, 2005. 3

[JPW22] Shunhua Jiang, Binghui Peng, and Omri Weinstein. Dynamic least-squares regression. *CoRR*, abs/2201.00228, 2022. 2

[KMNS21] Haim Kaplan, Yishay Mansour, Kobbi Nissim, and Uri Stemmer. Separating adaptive streaming from oblivious streaming. In *CRYPTO*, 2021. 3

[KNPW11] Daniel Kane, Jelani Nelson, Ely Porat, and P. David Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 745–754, 2011. 3

[KNW10] Daniel Kane, Jelani Nelson, and David 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*, pages 41–52, 2010. 3

[KP20] John Kallaugher and Eric Price. Separations and equivalences between turnstile streaming and linear sketching. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1223–1236. ACM, 2020. 3

[LNW14] Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC*, pages 174–183, 2014. 3

[LWY20] Kasper Green Larsen, Omri Weinstein, and Huacheng Yu. Crossing the logarithmic barrier for dynamic boolean data structure lower bounds. *SIAM J. Comput.*, 49(5), 2020. 6, 16

[MNS11] Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. *SIAM J. Comput.*, 40(6):1845–1870, 2011. 2

[MRU11] Andrew McGregor, Atri Rudra, and Steve Uurtamo. Polynomial fitting of data streams with applications to codeword testing. In *Symposium on Theoretical Aspects of Computer Science (STACS2011)*, volume 9, pages 428–439, 2011. 4

[NY19] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. *ACM Trans. Algorithms*, 15(3):35:1–35:30, 2019. 2

[SU15] Thomas Steinke and Jonathan R. Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Proceedings of The 28th Conference on*

Learning Theory, COLT, pages 1588–1628, 2015. 5, 8, 12, 13, 14, 15

[Wor21] STOC 2021 Workshop. Robust streaming, sketching, and sampling, June 2021. https://rajeshjayaram.com/stoc-2021-robust-streaming-workshop.html. 3

[Wor23] FOCS 2023 Workshop. Exploring the frontiers of adaptive robustness, November 2023. https://samsonzhou.github.io/focs-2023-workshop-adaptive-robustness. 3

[WZ21] 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. 2

[WZZ23] David P. Woodruff, Fred Zhang, and Samson Zhou. On robust streaming for learning with experts: Algorithms and lower bounds. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2023. 2