Tricking the Hashing Trick: A Tight Lower Bound on the Robustness of CountSketch to Adaptive Inputs

Edith Cohen[1,3*], **Jelani Nelson**[2,1], **Tamás Sarlós**[1], **Uri Stemmer**[3,1†]

[1]Google Research
[2]UC Berkeley
[3]Tel Aviv University
edith@cohenwang.com, minilek@alum.mit.edu, stamas@google.com, u@uri.co.il

Abstract

CountSketch and Feature Hashing (the "hashing trick") are popular randomized dimensionality reduction methods that support recovery of ℓ_2-heavy hitters (keys i where $v_i^2 > \epsilon\|\boldsymbol{v}\|_2^2$) and approximate inner products. When the inputs are *not adaptive* (do not depend on prior outputs), classic estimators applied to a sketch of size $O(\ell/\epsilon)$ are accurate for a number of queries that is exponential in ℓ. When inputs are adaptive, however, an adversarial input can be constructed after $O(\ell)$ queries with the classic estimator and the best known robust estimator only supports $\tilde{O}(\ell^2)$ queries. In this work we show that this quadratic dependence is in a sense inherent: We design an attack that after $O(\ell^2)$ queries produces an adversarial input vector whose sketch is highly biased. Our attack uses "natural" non-adaptive inputs (only the final adversarial input is chosen adaptively) and universally applies with any correct estimator, including one that is unknown to the attacker. In that, we expose inherent vulnerability of this fundamental method.

1 Introduction

CountSketch (Charikar, Chen, and Farach-Colton 2002) and its variant *feature hashing* (Moody and Darken 1989; Weinberger et al. 2009) are immensely popular dimensionality reduction methods that map input vectors in $\mathbb{R}^n$ to their sketches in $\mathbb{R}^d$ (where $d \ll n$). The methods have many applications in machine learning and data analysis and often are used as components in large models or pipelines (Weinberger et al. 2009; Shi et al. 2009; Pham and Pagh 2013; Chen et al. 2015, 2016; Aghazadeh et al. 2018; Spring et al. 2019; Ahle et al. 2020; Cohen, Pagh, and Woodruff 2020).

The mapping is specified by *internal randomness* $\rho \sim \mathcal{D}$ that determines a set of $d = \ell \cdot b$ linear measurements vectors $(\boldsymbol{\mu}^{(j,k)})_{j\in[\ell],k\in[b]}$ in $\mathbb{R}^n$. The sketch $\texttt{Sketch}_\rho(\boldsymbol{v})$ of a vector $\boldsymbol{v} \in \mathbb{R}^n$ is the matrix of d linear measurements

$$\texttt{Sketch}_{\boldsymbol{\rho}}(\boldsymbol{v}) := \left(\langle \boldsymbol{\mu}^{(j,k)}, \boldsymbol{v}\rangle\right)_{j\in[\ell],k\in[b]} . \qquad (1)$$

The salient properties of CountSketch are that (when setting $b = O(1/\epsilon)$ and $\ell = O(\log n)$) the ℓ_2-heavy hitters of an input $\boldsymbol{v}$, that is, keys i with $v_i^2 > \epsilon\|\boldsymbol{v}\|_2^2$, can be recovered from $\texttt{Sketch}_\rho(\boldsymbol{v})$ and that the inner product of two vectors $\boldsymbol{v}$, $\boldsymbol{u}$ can be approximated from their respective sketches $\texttt{Sketch}_\rho(\boldsymbol{v})$, $\texttt{Sketch}_\rho(\boldsymbol{u})$. This recovery is performed by applying an appropriate *estimator* to the sketch, for example, the median estimator (Charikar, Chen, and Farach-Colton 2002) provides estimates on values of keys and supports heavy hitters recovery. But recovery can also be implicit, for example, when the sketch is used as a compression module in a Neural Network (Chen et al. 2015), the recovery of features is learned.

Randomized data structures and algorithms are typically analysed under an assumption that the input sequence is generated in a way that does not depend on prior outputs and on the sketch randomness ρ. This assumption, however, does not always hold, for example, when there is an intention to construct an adversarial input or when the system has a feedback between inputs and outputs (Spring et al. 2019; Rothchild et al. 2020).

An interactive setting, where inputs are adaptive in that they may depend on prior outputs, is more challenging to analyse and there is growing interest in quantifying performance and in designing methods that are robust to adaptive inputs. Works in this vein span machine learning (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Athalye et al. 2018; Papernot et al. 2017), adaptive data analysis (Freedman 1983; Ioannidis 2005; Lukacs, Burnham, and Anderson 2009; Hardt and Ullman 2014; Dwork et al. 2015), dynamic graph algorithms (Shiloach and Even 1981; Ahn, Guha, and McGregor; Gawrychowski, Mozes, and Weimann 2020; Gutenberg and Wulff-Nilsen 2020; Wajc 2020; Beimel et al. 2021), and sketching and streaming algorithms (Mironov, Naor, and Segev 2008; Ahn, Guha, and McGregor; Hardt and Woodruff 2013; Ben-Eliezer et al. 2021; Hassidim et al. 2020; Woodruff and Zhou 2021; Attias et al. 2021; Ben-Eliezer, Eden, and Onak 2021; Gupta et al. 2021; Cohen et al. 2022a). Robustness to adaptive inputs can trivially be achieved by using a fresh data structure for each query, or more finely, for each time the output changes. Hence, ℓ independent replicas of a non-robust data structure suffice for supporting ℓ adaptive queries. A powerful connection between adaptive robustness

[*] Partially supported by Israel Science Foundation (grant no. 1595/19).

[†] Partially supported by Israel Science Foundation (grant 1871/19) and by Len Blavatnik and the Blavatnik Family foundation.

and differential privacy (Dwork et al. 2015) and utilizing the workhorse of advanced composition, yielded essentially a wrapper around ℓ independent replicas of a non-robust data structure that supports a quadratic number $\tilde{O}(\ell^2)$ of adaptive queries (or changes to the output) (Hassidim et al. 2020; Gupta et al. 2021; Beimel et al. 2021). For the problem of recovering heavy-hitters from CountSketch, the "wrapper" method supports $\approx \epsilon\ell^2$ adaptive queries. The current state of the art (Cohen et al. 2022a) is a robust estimator that works with a variant of CountSketch and supports $\tilde{O}(\ell^2)$ adaptive queries.

Lower bounds on the performance of algorithms to adaptive inputs are obtained by designing an *attack*, a sequence of input vectors, that yields a constructed input that is adversarial to the internal randomness ρ. Tight lower bounds on the robustness of statistical queries were established by (Hardt and Ullman 2014; Steinke and Ullman 2015), who designed an attack with a number of queries that is quadratic in the sample size, which matches the known upper bounds (Dwork et al. 2015). Their construction was based on fingerprinting codes (Boneh and Shaw 1995). A downside of these constructions is that the inputs used in the attack are not "natural" and hence unlikely to shed some understanding on practical vulnerability in the presence of feedback. Hardt and Woodruff (Hardt and Woodruff 2013) provided an impossibility result for the task of estimating the norm of the input within a constant factor from (general) linear sketches. Their construction works with arbitrary correct estimators and produces an adversarial distribution over inputs where the sketch measurements are "far" from their expectations. The attack size, however, has a large polynomial dependence on the sketch size and is far from the respective upper bound. Ben-Eliezer et al (Ben-Eliezer et al. 2021) present an attack on the AMS sketch (Alon, Matias, and Szegedy 1999) for the task of approximating the ℓ_2-norm of the input vector. The attack is tailored to a simplified estimator that is linear in the set of linear measurements (whereas the "classic" estimator uses a median of measurements and is not linear). Their attack is efficient in that the number of queries is of the order of the sketch size, rendering the estimator non-robust. It also has an advantage of using "natural" inputs. More recently, (Cohen et al. 2022a) presented attacks that are tailored to specific estimators for CountSketch, including an attack of size $O(\ell)$ on the classic median estimator and an attack of size $O(\ell^2)$ on their proposed robust estimator.

Contribution

Existing works proposed attacks of size that is far from the corresponding known upper bounds or are tailored to a particular estimator. Specifically for CountSketch, there is an upper bound of $O(\ell^2)$ but it is not even known whether there exist estimators that support a super-quadratic number of adaptive inputs. This question is of particular importance because CountSketch and its variants are the only known efficient sketching method that allow recovery of ℓ_2-heavy hitters and approximating ℓ_2 norms and inner products. Moreover, their form as linear measurements is particularly suitable for efficient implementations and integration as components in larger pipelines. Finally, a recent lower bound precludes hope for an efficient deterministic (and hence fully robust) sketch (Kamath, Price, and Woodruff 2021), so it is likely that the vulnerabilities of CountSketch are inherent to ℓ_2-heavy hitter (and approximate ℓ_2-norm and inner product) recovery from a small sketch.

We construct a *universal* attack on CountSketch, that applies against any unknown, potentially non-linear, possibly state maintaining, estimator. We only require that the estimator is correct. Our attack uses $O(\ell^2)$ queries, matching the $\tilde{O}(\ell^2)$ robust estimator upper bound (Cohen et al. 2022a). Moreover, it suffices for the purpose of the attack that the estimator only reports a set of candidate heavy keys without their approximate values (we only require that heavy hitters are reported with very high probability and 0 value keys are reported with very small probability). Our attack also applies against a correct inner-product estimator (that distinguishes between $\langle \boldsymbol{v}, \boldsymbol{u}\rangle = 0$ (reported with very small probability) and $\langle \boldsymbol{v}, \boldsymbol{u}\rangle^2 \geq \epsilon\|\boldsymbol{v}\|_2^2\|\boldsymbol{u}\|_2^2$ (reported with high probability.)) Additionally, we apply our attack method to ℓ_2-norm estimators applied to an AMS sketch (Alon, Matias, and Szegedy 1999) and obtain that an attack of size $O(\ell^2)$ suffices to construct an adversarial input. The AMS sketch can be viewed as a CountSketch with $b = 1$ and is similar to the Johnson Lindenstrauss transform (Johnson and Lindenstrauss 1984).

The product of our attack (with high probability) is an *adversarial input* $\boldsymbol{v}$ on which the measurement values of $\texttt{Sketch}_\rho(\boldsymbol{v})$ are very biased with respect to their distribution when $\rho \sim \mathcal{D}$. Specifically, the design of CountSketch results in linear measurements that are unbiased for any input $\boldsymbol{v}$ under the sketch distribution $\rho \sim \mathcal{D}$: For each key i and measurement vectors $\boldsymbol{\mu}$ with $\mu_i \neq 0$ it holds that $\mathbb{E}_\rho[\langle \boldsymbol{v}, \boldsymbol{\mu}\rangle/\mu_i - v_i] = 0$ but the corresponding expected values for our adversarial $\boldsymbol{v}$ in $\texttt{Sketch}_\rho(\boldsymbol{v})$ are large ($\geq B\epsilon\|\boldsymbol{v}\|_2^2$ for a desired $B > 1$). This "bias" means that the known standard (and robust) estimators for heavy hitters and inner products would fail on this adversarial input. And generally the usual design goal (for "learned" estimators) of being correct on any input with high probability over the distribution of $\texttt{Sketch}_\rho(\boldsymbol{v})$ is insufficient for an estimator to be correct on adversarial inputs. We note however that our result does not preclude the existence of specialized estimators that are correct on our adversarial inputs. This because some estimators (when attacked) can force the sketch of our adversarial input to be recognizably "out of distribution" (with basic statistics that still falsely match those of an input with a heavy key) and we do not preclude specialized estimators on these "out of distribution" sketches.

Finally, our attacks use "natural" inputs that have the form of a heavy key and random noise. The final adversarial input is a linear combination of the noise components according to the heavy hitter reports and is the only one that depends on prior outputs. The simplicity of this attack suggests "practical" vulnerability of this fundamental sketching technique.

Technique Our attacks construct an adversarial input with respect to key h. The high level structure is to generate "random tails," $(\boldsymbol{z}^{(t)})_{t\in[r]}$, which are vectors with small random entries. Ideally, we would like to determine for each $\boldsymbol{z}^{(t)}$ whether it is biased up or down with respect to key h. Roughly, considering the set T_h of ℓ measurement vectors with $\mu_h \in \{-1, 1\}$, determine the sign $s^{*(t)}$ of $\frac{1}{\ell}\sum_{\boldsymbol{\mu}\in T_h}\langle \boldsymbol{\mu}, \boldsymbol{z}^{(t)}\rangle \cdot \mu_h$. If we had that, the linear combination $\boldsymbol{z}^{*(A)} = \sum_{t\in[r]} s^{*(t)}\boldsymbol{z}^{(t)}$ (with large enough r) is an adversarial input. The intuition why this helps is that the bias accumulates linearly with the number of tails r whereas the standard deviation (essentially the ℓ_2 norm), considering randomness of the selection of tails, increases proportionally to $\sqrt{r}$. The attack strategy is then to design query vectors of the form $v_h^{(t)}\boldsymbol{e}_h + \boldsymbol{z}^{(t)}$ so that from whether or not h is reported as a heavy hitter candidate we obtain $s^{(t)}$ that correlates with $s^{*(t)}$. A higher correlation $\mathbb{E}[s^{*(t)}s^{(t)}]$ yields more effective attacks: With $\mathbb{E}[s^{*(t)}s^{(t)}] = \Omega(1)$ we get attacks of size $r = O(\ell)$ and with $\mathbb{E}[s^{*(t)}s^{(t)}] = \Omega(1/\sqrt{\ell})$ we get attack of size $r = O(\ell^2)$. We show that we can obtain $s^{(t)}$ with $\mathbb{E}[s^{*(t)}s^{(t)}] = \Omega(1/\sqrt{\ell})$ (thus matching the upper bound) against any arbitrary and adaptable estimator as long as it is correct. The difficulty is that such an estimator can be non monotone in the value v_h of the heavy key and change between queries. Our approach is to select the value v_h of the heavy key in the input vectors uniformly at random from an interval that covers the "uncertainty region" between values that must be reported as heavy to values that are hard to distinguish from 0 and hence should not be reported. We then observe that *in expectation* over the random choice, any correct estimator must have a slight reporting advantage with larger v_h. Finally, we show that any particular sketch content is equally likely with symmetrically larger v_h with a tail that is biased down or a smaller v_h with a tail that is biased up. This translates to a slight reporting advantage for h as a candidate heavy hitter when the tail is "biased up." Therefore by taking $s^{(t)} = 1$ when h is reported and $s^{(t)} = -1$ otherwise we have the desired correlation.

Related work Attack vectors of similar random noise form were used against specific estimators in (Cherapanamjeri and Nelson 2020) to attack the Johnson Lindenstrauss transform and as heuristic blackbox attacks on deep neural networks in (Guo et al. 2019). Our work is most related to (Cohen et al. 2022a) in that the structure of our attack vectors is similar to those used in (Cohen et al. 2022a) to construct a tailored attack on the classic CountSketchestimator. The generalization however to a "universal" attack that is effective against arbitrary and unknown estimators was delicate and required multiple new ideas. Our contribution is also related and in a sense complementary to (Hardt and Woodruff 2013) that designed attack on linear sketches that applies with any correct estimator for (approximate) norms. Their attack is much less efficient in that its size is a higher degree polynomial and it uses dependent (adaptive) inputs (whereas with our attack only the final adversarial input depends on prior outputs). The product of their attack are constructed vectors that are in the (approximate) null space of the sketching matrix. These "noise" vectors can have large norms but are "invisible" in the sketch. When such "noise" is added to an input with a signal (say a heavy hitter), the "signal" is suppressed (entry no longer heavy) but can still be recovered from the sketch. Our attack fails the sketch matrix in a complementary way: we construct "noise" vectors that do not involve a signal (a heavy entry) but the sketch mimics a presence of that particular signal.

Overview Our attack is described in Section 3 with detailed proofs provided in the full version (Cohen et al. 2022b). The analysis is based on that of a corresponding interaction with a mean estimator, that is described in Section 5. In Section 6 we describe another application of our attack technique, to ℓ_2 norm estimators for the AMS sketch (Alon, Matias, and Szegedy 1999).

2 Preliminaries

We use boldface notation for vectors $\boldsymbol{v}$, non boldface for scalars v, $\langle \boldsymbol{v}, \boldsymbol{u}\rangle = \sum_i v_i u_i$ for inner product, and $v \cdot u$ for scalar product. For a vector $\boldsymbol{v} \in \mathbb{R}^n$ we refer to $i \in [n]$ as a *key* and v_i as the value of the ith key (entry) and denote by $\overline{v} = \frac{1}{n}\sum_{i=1}^n v_i$ the mean value. For exposition clarity, we use $\approx$ to mean "within a small relative error." We denote by $\mathcal{N}(v, \sigma^2)$ the normal distribution with mean v and variance σ^2 and by $\boldsymbol{u} \sim \mathcal{N}_\ell(v, \sigma^2)$ a vector in $\mathbb{R}^\ell$ with entries that are i.i.d. $\mathcal{N}(v, \sigma^2)$. The probability density function of $\mathcal{N}_\ell(v, \sigma^2)$ is

$$f_v(\boldsymbol{u}) = \prod_{i\in[\ell]} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{u_i - v}{\sigma}\right)^2} . \qquad (2)$$

Definition 2.1. *(heavy hitter) For $\epsilon > 0$, and a vector $\boldsymbol{v} \in \mathbb{R}^n$, key $i \in [n]$ is an ℓ_2-ϵ-*heavy hitter *if $v_i^2 \geq \epsilon\|\boldsymbol{v}\|_2^2$.*

Clearly, there can be at most $1/\epsilon$ ℓ_2-ϵ heavy hitters.

Definition 2.2. *(heavy hitters estimator) A ℓ_2-ϵ-heavy hitters estimator is applied to a sketch of an input vector $\boldsymbol{v} \in \mathbb{R}^n$ and returns a set of entries $K \subset [n]$. The output is* correct *if K includes all the ℓ_2-ϵ-heavy hitters keys and does not include keys with $v_i = 0$.*

Remark 2.3. *Our correctness definition is a weaker requirement than what the classic* CountSketch *estimator provides (Charikar, Chen, and Farach-Colton 2002). Since we design attacks, the design is stronger against weaker requirements as less information on the randomness is revealed.*

Definition 2.4. *(inner product estimator) An inner-product estimator is applied to sketches of two input vectors $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^n$ and returns $s \in \{-1, 1\}$. The output is* correct *if $s = -1$ when $\langle \boldsymbol{v}, \boldsymbol{u}\rangle = 0$ and is $s = 1$ when $\langle \boldsymbol{v}, \boldsymbol{u}\rangle^2 \geq \epsilon\|\boldsymbol{v}\|_2^2\|\boldsymbol{u}\|_2^2$.*

2.1 CountSketch

The sketch (Charikar, Chen, and Farach-Colton 2002) is specified by parameters (n, ℓ, b), where n is the dimension of input vectors, and $d = \ell \cdot b$. The internal randomness ρ specifies a set of random hash functions $h_r : [n] \to [b]$ $(r \in [\ell])$ with the marginals that $\forall k \in [b]$, $i \in [n]$, $\Pr[h_j(i) = k] = 1/b$, and $s_j : [n] \to \{-1, 1\}$ $(j \in [\ell])$ so that $\Pr[s_j(i) = 1] = 1/2$. These hash functions define $d = \ell \cdot b$ measurement vectors, $\boldsymbol{\mu}^{(j,k)}$ $(j \in [\ell], k \in [b])$ where

$$\mu_i^{(j,k)} := \mathbb{1}_{\{h_j(i)=k\}} s_j(i),$$

organized as ℓ sets of b vectors each.

For an input vector $\boldsymbol{v} \in \mathbb{R}^n$, $\texttt{Sketch}_\rho(\boldsymbol{v}) := (\langle \boldsymbol{\mu}^{(j,k)}, \boldsymbol{v}\rangle)_{j,k}$ is the set of the respective measurement values. Note that for each key $i \in [n]$ there are exactly ℓ measurement vectors with a nonzero ith entry: $(\boldsymbol{\mu}^{(j,h_j(i))})_{j\in[\ell]}$ and these measurement vectors are independent (as the only dependency is between measurement in the same set of b, and there is exactly one from each set). The respective set of ℓ adjusted measurements:

$$(\langle \boldsymbol{\mu}^{(j,h_j(i))}, \boldsymbol{v}\rangle \mu_i^{(j,h_j(i))})_{j\in[\ell]} \tag{3}$$

are unbiased estimates of v_i: $\mathbb{E}_\rho[\langle \boldsymbol{\mu}^{(j,h_j(i))}, \boldsymbol{v}\rangle \mu_i^{(j,h_j(i))}] = v_i$.

The median estimator (Charikar, Chen, and Farach-Colton 2002) uses the median adjusted measurement to estimate the value v_i of each key i. The $O(1/\epsilon)$ keys with highest magnitude estimates are then reported as heavy hitters. When the same randomness ρ is used for r non-adaptive inputs (inputs selected independently of ρ and prior outputs), sketch parameter settings of $\ell = \log(r \cdot n/\delta)$ and $b = O(\epsilon^{-1})$ guarantee that with probability $1-\delta$, all outputs are correct (in the sense of Remark 2.3).

CountSketch also supports estimation of inner products. For two vectors $\boldsymbol{v}, \boldsymbol{u}$, we obtain an unbiased estimate of their inner product from the respective inner product of the $j \in [\ell]$th row of measurements:

$$\sum_{k\in[b]} \langle \boldsymbol{\mu}^{(j,k)}, \boldsymbol{v}\rangle \cdot \langle \boldsymbol{\mu}^{(j,k)}, \boldsymbol{u}\rangle \ . \tag{4}$$

The median of these ℓ estimates is within relative error $\sqrt{\epsilon}$ with probability $1 - \exp(\Omega(-\ell))$.

We note that pairwise independent hash functions h_j and s_j suffice for obtaining the guarantees of Remark 2.3 (Charikar, Chen, and Farach-Colton 2002) whereas 4-wise independence is needed for approximate inner products. The analysis of the attack we present here, however, holds even under full randomness.

2.2 Adversarial Input for CountSketch

Definition 2.5 (adversarial input). *We say that an attack $\mathcal{A}$ that is applied to a sketch with randomness ρ and outputs $i \in [n]$ and $\boldsymbol{z}^{(A)} \in \{-1,0,1\}^n$ (with $z_i^{(A)} = 0$) is* (B,β)-adversarial *(for $B > 1$) if, with probability at least $1-\beta$ over the randomness of $\rho, \mathcal{A}$, the adjusted measurements* (3) *satisfy:*

$$\Pr_{\rho\sim\mathcal{D},\mathcal{A}}\left[\frac{1}{\ell}\sum_{j\in[\ell]} \langle \boldsymbol{\mu}^{(j,h_j(i))}, \boldsymbol{z}^{(A)}\rangle \mu_i^{(j,h_j(i))} \geq \sqrt{\frac{B}{b}}\|\boldsymbol{z}^{(A)}\|_2\right] \geq 1-\beta. \tag{5}$$

The adversarial input $\boldsymbol{z}^{(A)}$ is a noise vector (with no heavy hitters) but $\texttt{Sketch}_\rho(\boldsymbol{z}^{(A)})$ "looks like" (in terms of the average adjusted measurement of key i) a sketch of a vector with a heavy key i. It follows from the standard analysis of CountSketch that the event

$$\frac{1}{\ell}\sum_{j\in[\ell]} \langle \boldsymbol{\mu}^{(j,h_j(i))}, \boldsymbol{v}\rangle \mu_i^{(j,h_j(i))} \geq \sqrt{\frac{B}{b}}\|\boldsymbol{v}\|_2 \ . \tag{6}$$

(that corresponds to (5)) is very likely for vectors $\boldsymbol{v}$ such that h is a heavy hitter (Definition 2.1) and extremely unlikely when h is not a heavy hitter, and in particular, when $v_h = 0$. Similarly for inner products, considering the inner product of $\boldsymbol{v}$ with the standard basis vector $\boldsymbol{e}_i$, the sketch-based estimates (4) of each $j \in \ell$ computed from $\texttt{Sketch}_\rho(\boldsymbol{e}_i)$ and $\texttt{Sketch}_\rho(\boldsymbol{v})$ are equal to the respective adjusted measurement (3) of i from $\texttt{Sketch}_\rho(\boldsymbol{v})$. The event (6) is very likely when $\langle \boldsymbol{e}_i, \boldsymbol{v}\rangle \geq \epsilon\|\boldsymbol{v}\|_2^2$ and very unlikely when $\langle \boldsymbol{e}_i, \boldsymbol{v}\rangle = 0$, noting that for our adversarial input $\boldsymbol{z}^{(A)}$ it holds that $\langle \boldsymbol{e}_i, \boldsymbol{z}^{(A)}\rangle = 0$.

While this follows from the standard analysis, the simple structure of our noise vectors (described in Section 3.1) allows for a particularly simple argument for bounding the probability of (6) for $\boldsymbol{v}$ with the structure of $\boldsymbol{z}^{(A)}$: The distribution of an adjusted measurement of a $\boldsymbol{v} \in \{-1,0,1\}^n$ and $v_i = 0$ with support size $m = |\operatorname{supp}(\boldsymbol{v})| = \|\boldsymbol{v}\|_2^2$ approaches $\mathcal{N}(0, \frac{m}{b})$ (for large m/b), and thus the average approaches $\mathcal{N}(0, \frac{m}{\ell \cdot b})$. Therefore, the probability of (5) on a random sketch of $\boldsymbol{z}^{(A)}$ is $\leq \exp(-\ell B/2)$ (applying tail bounds on the probability of value exceeding $\sqrt{\ell B}$ standard deviations).

3 Attack Description

We describe our attack against heavy hitters estimators. The modifications needed for it to apply with inner product estimator are described in Section 3.3. The attack is an interaction between the following components:

- Internal randomness $\rho \sim \mathcal{D}$ that specifies linear measurement vectors $(\boldsymbol{\mu}^{(j,k)})_{j\in[\ell],k\in[b]}$. The sketch $\texttt{Sketch}_\rho(\boldsymbol{v})$ of a vector $v \in \mathbb{R}^n$ is the set of measurements $(\langle \boldsymbol{\mu}^{(j,k)}, \boldsymbol{v}\rangle)_{j\in[\ell],k\in[b]}$.
- A *query-response algorithm* that at each step t chooses a heavy hitters estimator (see Definition 2.2). The choice may depend on the randomness ρ and prior queries and responses $(\texttt{Sketch}_\rho(\boldsymbol{v}^{(t')}), K^{(t')})_{t'<t}$. The algorithm receives $\texttt{Sketch}_\rho(\boldsymbol{v}^{(t)})$, applies the estimator to the sketch, and outputs $K^{(t)}$.
- An *adversary* that issues a sequence of input queries $(\boldsymbol{v}^{(t)})_t$ and collects the responses $(K^{(t)})_t$. The randomness ρ and the sketches of the query vectors $(\texttt{Sketch}_\rho(\boldsymbol{v}^{(t)}))_t$ are not known to the adversary. The goal is to construct an *adversarial input vector* $\boldsymbol{z}^{(A)}$ (see Definition 2.5).

Our adversary generates the *query vectors* $(\boldsymbol{v}^{(t)})_{t\in[r]}$ non-adaptively as described in Section 3.1. The attack interaction and its properties are stated in Section 3.2.

3.1 Query Vectors

Our attack query vectors $(\boldsymbol{v}^{(t)})_{t\in[r]}$ have the form:

$$\boldsymbol{v}^{(t)} := v_h^{(t)}\boldsymbol{e}_h + \boldsymbol{z}^{(t)} \in \mathbb{R}^n, \tag{7}$$

where h is a special *heavy* key, that is selected uniformly $h \sim \mathcal{U}[n]$ and remains fixed, $\boldsymbol{e}_h$ is the standard basis vector (axis-aligned unit vector along h), and the vectors $\boldsymbol{z}^{(t)}$ are *tails*. The (randomized) construction of tails is described in Algorithm 1. The tail vectors $(\boldsymbol{z}^{(t)})_{t\in[r]}$ have support of size $|\operatorname{supp}(\boldsymbol{z}^{(t)})| = m$ that does not include key h ($h \notin \operatorname{supp}(\boldsymbol{z}^{(t)})$) and so that the supports of different tails are disjoint:

$$t_1 \neq t_2 \implies \operatorname{supp}(\boldsymbol{z}^{(t_1)}) \cap \operatorname{supp}(\boldsymbol{z}^{(t_2)}) = \emptyset\ .$$

For query t and key $i \in \operatorname{supp}(\boldsymbol{z}^{(t)})$, the values are selected i.i.d. Rademacher $z_i^{(t)} \sim \mathcal{U}[\{-1,1\}]$. Note that $\|\boldsymbol{v}^{(t)}\|_2^2 = (v_h^{(t)})^2 + \|\boldsymbol{z}^{(t)}\|_2^2 = (v_h^{(t)})^2 + m$. Note that the tails, and (as we shall see) the selection of $v_h^{(t)}$, and hence the input vectors are constructed non-adaptively. Only the final adversarial input vector depends on the output of the estimator on prior queries. The parameter m is set to a value that is polynomial in the sketch size and large enough so that certain approximations hold (see Section 4).

Algorithm 1: `AttackTails`

```
Input: Input dimension $n$, support size $m$, number of tails $r$
$h \leftarrow \mathcal{U}[n]$                    // Special heavy hitter key
$S \leftarrow \{h\}$                              // Keys used in support
for $t \in [r]$ do
    $S' \leftarrow$ random subset of size $m$ from $[n] \setminus S$
    $\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{0}$
    foreach $i \in S'$ do
        $\boldsymbol{z}_i^{(t)} \sim \mathcal{U}[\{-1,1\}]$
    $S \leftarrow S \cup S'$
return $h, (\boldsymbol{z}^{(t)})_{t\in[r]}$
```

Remark 3.1. *The only piece of information needed from the output of the estimator is whether the particular key h is reported as a candidate heavy hitter of $\boldsymbol{v}^{(t)}$, that is, whether $h \in K^{(t)}$. Note that disclosing additional information can only make the estimator* more *vulnerable to attacks.*

3.2 Universal Attack

Our attack interaction is described in Algorithm 2. We generate r attack tails using Algorithm 1. We then construct r queries of the form (7) with i.i.d. $v_h^{(t)} \sim \mathcal{U}[a\cdot\sigma, (c+2a)\cdot\sigma]$. At each step $t \in [r]$, we feed the sketch of $\boldsymbol{v}^{(t)}$ to the HH estimator selected by the query response algorithm and collect the output $K^{(t)}$ of the estimator. We then set $s^{(t)} \leftarrow 1$ if $h \in K^{(t)}$ and $s^{(t)} \leftarrow -1$ if $h \notin K^{(t)}$. The final step computes the *adversarial input*:

$$\boldsymbol{z}^{(A)} := \sum_{t\in[r]} s^{(t)}\boldsymbol{z}^{(t)}\ . \tag{8}$$

The statements below apply only to measurement vectors with nonzero value for key h. To simplify the notation, we use $\boldsymbol{\mu}^{(j)} := \boldsymbol{\mu}^{(j,h_j(h))}$ for $j \in [\ell]$. For randomness ρ, we use $\boldsymbol{\mu}^{(j)}(\rho)$ for the respective measurement vectors and $\mathcal{A}(\rho)$ for the output distribution of Algorithm 2 applied with randomness ρ.

Algorithm 2: Attack on `CountSketch` Heavy Hitters Estimators

```
Set $a = \Theta(\sqrt{\frac{\ln(1/\delta_2)}{\ell}})$ and $c = \Theta(1)$    // With universal
  constants as in Lemma 4.1
Input: Initialized $\texttt{CountSketch}_\rho$ with parameters
       $(n, \ell, b)$, Query-response algorithm, number of
       queries $r$, tail support size $m$
$(h, (\boldsymbol{z}^{(t)})_{t\in[r]}) \leftarrow \texttt{AttackTails}(n, m, r)$ // Algorithm 1
for $t \in [r]$ do // Compute Query Vectors
    $v_h^{(t)} \sim \mathcal{U}[a\cdot\sigma, (c+2a)\cdot\sigma]$            // $\sigma := \sqrt{m/b}$
    $\boldsymbol{v}^{(t)} \leftarrow v_h^{(t)}\boldsymbol{e}_h + \boldsymbol{z}^{(t)}$                // Query vectors
for $t \in [r]$ do // Apply Query Response
    Choose a correct HH estimator $M^{(t')}$    // With correct
      reporting function (Definition 5.1), may depend on
      $(v_h^{(t')}, K^{(t')}, M^{(t')})_{t'<t}$ and $\rho$
    $K^{(t)} \leftarrow M^{(t)}(\texttt{CountSketch}_\rho(\boldsymbol{v}^{(t)}))$        // Apply
      estimator to sketch
    if $h \in K^{(t)}$ then $s^{(t)} \leftarrow 1$ else $s^{(t)} \leftarrow -1$
return $\boldsymbol{z}^{(A)} \leftarrow \sum_{t\in[r]} s^{(t)}\boldsymbol{z}^{(t)}$        // Adversarial input
```

The adversarial input has $z_h^{(A)} = 0$ and norm $\|\boldsymbol{z}^{(A)}\|_2^2 = r\cdot m$ (it has support of size $r\cdot m$ with values in the support i.i.d Rademacher $U[\{-1,1\}]$).

Let the random variable $M(\rho)$ be the average adjusted measurement of an adversarial vector $\boldsymbol{z}^{(A)}$ constructed for randomness ρ_0 ($\rho_0 \sim \mathcal{D}, \boldsymbol{z}^{(A)} \sim \mathcal{A}(\rho_0)$) and sketched with randomness ρ:

$$M(\rho) := \frac{1}{\ell}\sum_{j\in[\ell]} \langle \boldsymbol{z}^{(A)}, \boldsymbol{\mu}^{(j)}(\rho)\rangle \cdot \mu_h^{(j)}(\rho)\ .$$

When an adversarial input $\boldsymbol{z}^{(A)} \sim \mathcal{A}(\rho_0)$ is sketched using a *random* $\rho \sim \mathcal{D}$ it holds that for all $j \in [\ell]$:

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0),\rho\sim\mathcal{D}}{\mathbb{E}}\left[\langle \boldsymbol{z}^{(A)}, \boldsymbol{\mu}^{(j)}(\rho)\rangle\right] = 0$$

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0),\rho\sim\mathcal{D}}{\mathrm{Var}}\left[\langle \boldsymbol{z}^{(A)}, \boldsymbol{\mu}^{(j)}(\rho)\rangle\right] \approx \frac{r\cdot m}{b} = r\sigma^2\ .$$

and since the ℓ measurements are independent we get:

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0),\rho\sim\mathcal{D}}{\mathbb{E}}[M(\rho)] = 0$$

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0),\rho\sim\mathcal{D}}{\mathrm{Var}}[M(\rho)] \approx \frac{r}{\ell}\cdot\sigma^2.$$

The adversarial input $\boldsymbol{z}^{(A)} \sim \mathcal{A}(\rho_0)$ behaves differently with respect to the particular randomness ρ_0 it was constructed for. We will establish the following:

Lemma 3.2 (Properties of the adversarial input)**.**

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0)}{\mathbb{E}}[M(\rho_0)] \approx \frac{r}{\ell}\cdot\frac{2\sigma}{c+a}$$

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0)}{\mathrm{Var}}[M(\rho_0)] \approx \frac{r}{\ell}\cdot\sigma^2$$

The proof is provided in the full version (Cohen et al. 2022b). The high level idea, as hinted in the introduction, is that we establish that the event $h \in K^{(t)}$ and thus $s^{(t)} = 1$ is correlated with "positive bias", that is, with the event $\frac{1}{\ell}\sum_{j\in[\ell]}\langle s^{(t)}\boldsymbol{z}^{(t)}, \boldsymbol{\mu}^{(j)}(\rho_0)\rangle \cdot \mu_h^{(j)}(\rho_0) > 0$. In the sum $\sum_{t\in[r]} s^{(t)}\boldsymbol{z}^{(t)}$ the bias (which is "forced" error on the estimates) increases linearly with r while the ℓ_2 norm, which corresponds to the standard deviation of the error, increases proportionally to $\sqrt{r}$. The main technique is abstracted through an interaction described in Section 5.

As a corollary of Lemma 3.2, it follows that $\boldsymbol{z}^{(A)}$ is an adversarial input (see Definition 2.5).

Theorem 3.1 (Adversarial input)**.** *If for $B > 1$ we use attack of size $r = B \cdot \ell^2$ then*

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0)}{\mathbb{E}}[M(\rho_0)] \approx \frac{2}{c+a}\sqrt{\frac{B}{b}}\|\boldsymbol{z}^{(A)}\|_2$$

$$\underset{\rho_0\sim\mathcal{D},\boldsymbol{z}^{(A)}\sim\mathcal{A}(\rho_0)}{\mathrm{Var}}[M(\rho_0)] \approx \frac{1}{\ell\cdot b}\|\boldsymbol{z}^{(A)}\|_2^2 .$$

Proof. Using Lemma 3.2, the expected value with attack size $r = B\ell^2$ is

$$\frac{r}{\ell}\cdot\frac{2\sigma}{c+a} = \frac{r}{\ell}\sqrt{\frac{m}{b}}\cdot\frac{2}{c+a} = \frac{2}{c+a}\frac{\sqrt{r}}{\ell}\frac{1}{\sqrt{b}}\sqrt{r\cdot m}$$

$$= \frac{2}{c+a}\sqrt{\frac{B}{b}}\|\boldsymbol{z}^{(A)}\|_2 \text{ since } \|\boldsymbol{z}^{(A)}\|_2 = \sqrt{r\cdot m}$$

The variance of the average is

$$\frac{r}{\ell}\sigma^2 = \frac{r\cdot m}{\ell\cdot b} = \frac{1}{\ell\cdot b}\|\boldsymbol{z}^{(A)}\|_2^2$$

□

3.3 Attack of an Inner-Product Estimator

We describe the modifications to Algorithm 2 needed for the attack to apply with an inner-product estimator. We compute the same query vectors $\boldsymbol{v}^{(t)}_{t\in[r]}$. At each step t, the query response algorithm chooses a correct inner-product estimator $M^{(t)}$ (see Definition 2.4). The query is issued for the inner product of $\boldsymbol{v}^{(t)}$ with the standard basis vector $\boldsymbol{e}_h$. Note that the value of the inner product is exactly $v_h^{(t)}$ and the requirement of correct reporting of the inner product (Definition 2.4) on these query vectors matches the requirement of a correct heavy hitters reporting of the key h (Definition 2.2).

The input to the estimator are the sketches $\texttt{Sketch}_\rho(\boldsymbol{e}_h)$ and $\texttt{Sketch}_\rho(\boldsymbol{v}^{(t)})$. Note that the information available to the estimator from the provided sketches on $v_h^{(t)}$ is the same as with heavy-hitter queries: $\texttt{Sketch}_\rho(\boldsymbol{e}_h)$ is simply the vector with entries $\mu_h^{(j)}$, which does not add information as ρ and h are assumed to be known to the estimator. The same analysis therefore applies.

4 Sketch Distribution and Estimators

We show (see full version (Cohen et al. 2022b)) that with our particular query inputs (7), for large enough m, the sketch content that is relevant to determining whether h is a candidate heavy hitter is approximately $\boldsymbol{u}^{(t)} \sim \mathcal{N}_\ell(v_h^{(t)}, \sigma^2)$ where $\sigma = \sqrt{\frac{m}{b}}$. The random variables $\boldsymbol{u}^{*(t)} = \boldsymbol{u}^{(t)} - v_h^{(t)}\boldsymbol{1}_\ell$ for $t \in [r]$ are approximately i.i.d. from $\mathcal{N}_\ell(0, \sigma^2)$.

We establish properties of any correct ℓ_2 ϵ-heavy hitters estimator that is applied to our query vectors. In its most general form, a query response algorithm fixes before each query t an estimator $M^{(t)}$. The estimator is applied to the content of the sketch, which on our inputs are i.i.d. vectors $(\boldsymbol{u}^{(t)} \sim \mathcal{N}_\ell(v_h^{(t)}, \sigma^2))_{t\in[r]}$. The estimator is specified by a *reporting function* $p^{(t)} : \mathbb{R}^\ell \to [0,1]$ so that $p^{(t)}(\boldsymbol{u}^{(t)}) := \Pr[h \in M^{(t)}(\boldsymbol{u}^{(t)})]$ specifies the probability that the returned $K^{(t)}$ includes key h when the sketch content is $\boldsymbol{u}^{(t)}$. We allow the query response algorithm to modify the estimator arbitrarily between queries and in a way that depends on sketches of prior inputs, prior outputs, and on a maintained state from past queries $(\boldsymbol{u}^{(t')}, K^{(t')})_{t'<t}$. The only constraint that we impose is that (at each step t) the output is correct with high probability: ℓ_2-ϵ-heavy hitters are reported and 0 value keys are not reported (see Definition 2.2). We show that a correct estimator on our query inputs must satisfy the following:

Lemma 4.1 (Correct HH estimator basic property)**.** *For $\delta_1, \delta_2 \ll 1$, there are $a = \Theta(\sqrt{\frac{\ln(1/\delta_2)}{\ell}})$ and $c = \Theta(1)$ so that the following holds. If the estimator satisfies (i) if h is a heavy hitter then* $\Pr[h \in K] \geq 1 - \delta_1$ *and (ii) if $v_h = 0$ then* $\Pr[h \notin K] \geq 1 - \delta_2$. *Then*

- $|v_h| \geq c \cdot \sigma \implies \Pr[h \in K] \geq 1 - \delta_1$
- $|v_h| \leq a \cdot \sigma \implies \Pr[h \notin K] \geq 1 - \frac{1}{\delta_2^{\Omega(1)}}$
- *Otherwise, unrestricted*

5 Mean Estimation Interaction

In this section we describe and state properties of an interaction, stated in Algorithm 3, with a mean estimator for i.i.d Normal random variables. Properties of Algorithm 2 can be established through correspondence to Algorithm 3 (see full version (Cohen et al. 2022b)). In Section 6 we describe an attack on ℓ_2-norm estimators applied to the AMS sketch that can be analyzed also through Algorithm 3.

Algorithm 3: `MeanEstAttack`

```
Input: Parameters $(a, c, \sigma, \delta)$, number of queries $r$,
       $b \in (0, a]$, A query response algorithm $\mathcal{A}$ that
       chooses $(a, c, \sigma, \delta)$-correct reporting functions
for $t \in [r]$ do // Generate queries and reporting functions
    $\mathcal{A}$ chooses a $(a, c, \sigma, \delta)$-correct reporting function
      $p^{(t)} : \mathbb{R}^\ell \to [0, 1]$        // Definition 5.1. Choice may
      depend on $(\boldsymbol{u}^{(t')}, s^{(t')})_{t'<t}$
    $v^{(t)} \sim \mathcal{U}[a \cdot \sigma, (c + 2b) \cdot \sigma]$
    $\boldsymbol{u}^{*(t)} \sim \mathcal{N}_\ell(0, \sigma^2)$
    for $j \in [\ell]$ do $u_j^{(t)} \leftarrow v^{(t)} + u_j^{*(t)}$ // Compute
      $\boldsymbol{u}^{(t)} \in \mathbb{R}^\ell$, note that $\boldsymbol{u}^{(t)} \sim \mathcal{N}_\ell(v^{(t)}, \sigma^2)$
    $s^{(t)} \leftarrow 1$ w.p. $p^{(t)}(\boldsymbol{u}^{(t)})$ and $s^{(t)} \leftarrow -1$ otherwise
return $\boldsymbol{u}^{(A)} \leftarrow \sum_{t\in[r]} s^{(t)}\boldsymbol{u}^{*(t)}$
```

We use the following definition:

Definition 5.1 (correct reporting function)**.** *A reporting function* $p : \mathbb{R}^\ell \to [0,1]$ *is* correct *with respect to parameters:* $(\delta, c > a > 0, \ell, \sigma)$ *if*

$$\forall |v| \geq c \cdot \sigma, \quad \underset{\boldsymbol{u} \sim \mathcal{N}_\ell(v,\sigma^2)}{\mathbb{E}}[p^{(t)}(\boldsymbol{u})] \geq 1 - \delta$$

$$\forall |v| \leq a \cdot \sigma, \quad \underset{\boldsymbol{u} \sim \mathcal{N}_\ell(v,\sigma^2)}{\mathbb{E}}[p^{(t)}(\boldsymbol{u})] \leq \delta \; .$$

A correct reporting function can be viewed as a simple mean estimator applied to ℓ i.i.d. samples from $\mathcal{N}(v, \sigma^2)$: With probability $1 - \delta$, the output is 1 when $|v| > c \cdot \sigma$ and -1 when $|v| < a \cdot \sigma$.

We show the following (See (Cohen et al. 2022b)):

Lemma 5.2 (`MeanEstAttack` Properties)**.** *Consider Algorithm 3 where b is such that*

$$\sqrt{\frac{\ell}{2\pi}} e^{-\ell b^2/2} \ll \frac{1}{c - a + 2b}.$$

Then the output $\boldsymbol{u}^{(A)} \in \mathbb{R}^\ell$ *satisfies*

$$\underset{\mathcal{A}}{\mathbb{E}}\left[\overline{u^{(A)}}\right] \approx \frac{r}{\ell} \cdot \frac{2\sigma}{c - a + 2b} \tag{9}$$

$$\frac{r}{\ell}\sigma^2 \left(1 - \frac{2b}{c - a + 2b}\right)^2 \lessapprox \underset{\mathcal{A}}{\mathrm{Var}}\left[\overline{u^{(A)}}\right] \lessapprox \frac{r}{\ell}\sigma^2 \tag{10}$$

6 Attack on the AMS Sketch

Algorithm 4: Attack on AMS norm estimation

```
Input: $\tau$, $\epsilon$, $\delta$          // Estimator parameters (Definition 6.1)
Initialized AMS $\texttt{Sketch}_\rho$ with parameters $(n, \ell)$, number
        of queries $r$, tail support size $m$
$\sigma \leftarrow \tau\sqrt{1/2}$; $c \leftarrow \sqrt{1+2\epsilon}$; $a \leftarrow 1$; $b \leftarrow \Theta(\frac{1}{\sqrt{\ell}} \ln(\sqrt{\frac{\ell}{\epsilon}})$
$(h, (\boldsymbol{z}^{(t)})_{t\in[r]}) \leftarrow \texttt{AttackTails}(n, m, r)$ // Choose tails
  (Algorithm 1)
$(\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{z}^{(t)} \cdot \frac{\sigma}{\sqrt{m}})_{t\in[r]}$  // Rescale tails to have $\ell_2$ norm $\sigma$
for $t \in [r]$ do // Generate Query Vectors
    $v_h^{(t)} \sim \mathcal{U}[a \cdot \sigma, (c + 2b) \cdot \sigma]$
    $\boldsymbol{v}^{(t)} \leftarrow v_h^{(t)} \boldsymbol{e}_h + \boldsymbol{z}^{(t)}$                 // Query vectors
for $t \in [r]$ do // Apply Query Response
    Choose an $(\epsilon, \delta, \tau)$-correct estimator $M^{(t)}$
      // Definition 6.1, may depend on
      $(v_h^{(t')}, s^{(t')}, M^{(t')})_{t'<t}$ and $\rho$
    $s^{(t)} \leftarrow M^{(t)}(\texttt{Sketch}_\rho(\boldsymbol{v}^{(t)})$      // Apply estimator to
      sketch
return $\boldsymbol{z}^{(A)} \leftarrow \sum_{t\in[r]} s^{(t)} \boldsymbol{z}^{(t)}$            // Adversarial input
```

The AMS sketch (Alon, Matias, and Szegedy 1999) and the related Johnson Lindenstrauss transform (Johnson and Lindenstrauss 1984) are randomized linear maps of input vectors $\boldsymbol{v} \in \mathbb{R}^n$ to their sketches $\texttt{Sketch}_\rho(\boldsymbol{v}) \in \mathbb{R}^\ell$. The sketches support recovery of ℓ_2 norms and distances. In this section we describe an attack on the AMS sketch that applies with any correct norm estimator. It suffices that the estimator returns only one bit, comparing the norm to τ with accuracy ϵ and confidence $1 - \delta$:

Definition 6.1 (correct ℓ_2-norm estimator)**.** *A norm estimator is correct for parameters* (ϵ, δ) *and* τ *if for all* $\boldsymbol{v}$*, when* $\|\boldsymbol{v}\|_2^2 \geq (1+\epsilon)\tau^2$*, the output is* 1 *with probability* $\geq 1 - \delta$ *and if* $\|\boldsymbol{v}\|_2^2 \leq \tau^2$ *then the output is* -1 *with probability* $\geq 1 - \delta$*.*

The sketch is specified by parameters (n, ℓ), where n is the dimension of input vectors and ℓ is the number of measurements. The internal randomness ρ specifies a set of random hash functions $s_j : [n] \to \{-1, 1\}$ $(j \in [\ell])$ so that $\Pr[s_j(i) = 1] = 1/2$. These hash functions define ℓ measurement vectors $\boldsymbol{\mu}^{(j)}$ $(j \in [\ell])$: $\mu_i^{(j)} := s_j(i)$. The sketch has the property that for any $\boldsymbol{v} \in R^n$ and $j \in [\ell]$, $\mathbb{E}_\rho[\langle \boldsymbol{\mu}, \boldsymbol{v}\rangle^2] = \|\boldsymbol{v}\|_2^2$ and $\mathrm{Var}_\rho[\langle \boldsymbol{\mu}, \boldsymbol{v}\rangle^2] = O(\|\boldsymbol{v}\|_2^2)$. The average estimator for the norm $M(\texttt{Sketch}_\rho(\boldsymbol{v})) = \frac{1}{\ell}\sum_{j\in[\ell]} \langle \boldsymbol{\mu}^{(j)}(\rho), \boldsymbol{v}\rangle^2$ when $\ell = O(\epsilon^{-2}\log(1/\delta))$, for all $\boldsymbol{v} \in \mathbb{R}^n$,

$$\Pr_\rho[| \|\boldsymbol{v}\|_2^2 - M(\texttt{Sketch}_\rho(\boldsymbol{v}))| \geq \epsilon \|\boldsymbol{v}\|_2^2] < \delta \; .$$

Definition 6.2 (adversarial input for AMS)**.** *We say that an attack* $\mathcal{A}$ *that is applied to a sketch with randomness* ρ *and outputs* $\boldsymbol{z}^{(A)} \in \{-1, 0, 1\}^n$ *is* (ξ, β)-adversarial *(for* $\xi > 0$*) if with probability at least* $1 - \beta$ *over the randomness of* ρ, $\mathcal{A}$ *it holds that:*

$$\Pr_{\rho, \mathcal{A}}\left[M(\mathit{Sketch}_\rho(\boldsymbol{z}^{(A)})) \geq (1 + \xi) \cdot \|\boldsymbol{z}^{(A)}\|_2^2\right] \geq 1 - \beta \; . \tag{11}$$

Therefore, for (ξ, δ) such that $\xi = \Omega(\sqrt{\ln(1/\delta)/\ell})$, the probability of the event in (11) on a non-adaptively chosen input is smaller than δ but is at least $1 - \beta$ on an adversarial input. Algorithm 4 describes an attack that constructs an adversarial input for a sketch with randomness ρ. Here we consider both the accuracy ϵ of the estimator, which is at least $\epsilon \geq 1/\sqrt{\ell}$ (otherwise correct estimators do not exist) and the bias $\xi = \Omega(1/\sqrt{\ell}$ in the product of the attack. We establish the following (the proof is presented in the full version (Cohen et al. 2022b)):

Lemma 6.3. *For* $\epsilon, \xi = \Omega(1/\sqrt{\ell})$*, any constant* $\beta > 0$*, and attack size* $r = O(\xi \ell^2 \min\{\epsilon^2, \epsilon\})$*, the output of Algorithm 4 is* (ξ, β)*-adversarial.*

Intuitively, more accurate estimators (that is, smaller ϵ) leak more information on the randomness and hence are easier to attack. We therefore expect attack size to increase with ϵ. A larger bias ξ is harder to accrue and would require a larger attack and hence attack size also increases with ξ.

Conclusion

Our results suggest interesting directions for future work. We suspect that our attack technique can be generalized so that it applies with any randomized linear sketch that supports recovery of heavy hitters keys or other properties of the input vectors. Our attack techniques can be viewed as constructing "random noise" that mimics the presence of a signal. Similar blackbox attacks were used heuristically in (Guo et al. 2019) on trained neural networks with adaptive inputs. Our results suggest that attacks can be effective even with non-adaptive inputs (with only the final attack input depending on prior outputs) and provide theoretical grounding to the observed effectiveness.

References

Aghazadeh, A.; Spring, R.; LeJeune, D.; Dasarathy, G.; Shrivastava, A.; and Baraniuk, R. G. 2018. MISSION: Ultra Large-Scale Feature Selection using Count-Sketches. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 80–88. PMLR.

Ahle, T. D.; Kapralov, M.; Knudsen, J. B. T.; Pagh, R.; Velingker, A.; Woodruff, D. P.; and Zandieh, A. 2020. Oblivious Sketching of High-Degree Polynomial Kernels. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*. SIAM.

Ahn, K. J.; Guha, S.; and McGregor, A. ???? Analyzing Graph Structure via Linear Measurements. In *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 459–467.

Alon, N.; Matias, Y.; and Szegedy, M. 1999. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58: 137–147.

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing robust adversarial examples. In *International conference on machine learning*, 284–293. PMLR.

Attias, I.; Cohen, E.; Shechner, M.; and Stemmer, U. 2021. A Framework for Adversarial Streaming via Differential Privacy and Difference Estimators. *CoRR*, abs/2107.14527.

Beimel, A.; Kaplan, H.; Mansour, Y.; Nissim, K.; Saranurak, T.; and Stemmer, U. 2021. Dynamic Algorithms Against an Adaptive Adversary: Generic Constructions and Lower Bounds. *CoRR*, abs/2111.03980.

Ben-Eliezer, O.; Eden, T.; and Onak, K. 2021. Adversarially Robust Streaming via Dense-Sparse Trade-offs. *CoRR*, abs/2109.03785.

Ben-Eliezer, O.; Jayaram, R.; Woodruff, D. P.; and Yogev, E. 2021. A Framework for Adversarially Robust Streaming Algorithms. *SIGMOD Rec.*, 50(1): 6–13.

Boneh, D.; and Shaw, J. 1995. Collusion-Secure Fingerprinting for Digital Data (Extended Abstract). In Coppersmith, D., ed., *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, volume 963 of *Lecture Notes in Computer Science*, 452–465. Springer.

Charikar, M.; Chen, K.; and Farach-Colton, M. 2002. Finding Frequent Items in Data Streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, 693–703. Springer-Verlag. ISBN 3540438645.

Chen, W.; Wilson, J. T.; Tyree, S.; Weinberger, K. Q.; and Chen, Y. 2015. Compressing Neural Networks with the Hashing Trick. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 2285–2294. JMLR.org.

Chen, W.; Wilson, J. T.; Tyree, S.; Weinberger, K. Q.; and Chen, Y. 2016. Compressing Convolutional Neural Networks in the Frequency Domain. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, 1475–1484. ACM. ISBN 978-1-4503-4232-2.

Cherapanamjeri, Y.; and Nelson, J. 2020. On Adaptive Distance Estimation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cohen, E.; Lyu, X.; Nelson, J.; Sarlós, T.; Shechner, M.; and Stemmer, U. 2022a. On the Robustness of CountSketch to Adaptive Inputs. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Cohen, E.; Nelson, J.; Sarlós, T.; and Stemmer, U. 2022b. Tricking the Hashing Trick: A Tight Lower Bound on the Robustness of CountSketch to Adaptive Inputs. *arXiv:2207.00956*.

Cohen, E.; Pagh, R.; and Woodruff, D. 2020. WOR and p's: Sketches for ℓ_p-Sampling Without Replacement. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 21092–21104. Curran Associates, Inc.

Dwork, C.; Feldman, V.; Hardt, M.; Pitassi, T.; Reingold, O.; and Roth, A. L. 2015. Preserving Statistical Validity in Adaptive Data Analysis. In *STOC*, 117–126. ACM.

Freedman, D. A. 1983. A Note on Screening Regression Equations. *The American Statistician*, 37(2): 152–155.

Gawrychowski, P.; Mozes, S.; and Weimann, O. 2020. Minimum Cut in O(m $\log^2$ n) Time. In *ICALP*, volume 168 of *LIPIcs*, 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guo, C.; Gardner, J. R.; You, Y.; Wilson, A. G.; and Weinberger, K. Q. 2019. Simple Black-box Adversarial Attacks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2484–2493. PMLR.

Gupta, V.; Jung, C.; Neel, S.; Roth, A.; Sharifi-Malvajerdi, S.; and Waites, C. 2021. Adaptive Machine Unlearning. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc.

Gutenberg, M. P.; and Wulff-Nilsen, C. 2020. Decremental SSSP in Weighted Digraphs: Faster and against an Adaptive Adversary. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, 2542–2561. USA: Society for Industrial and Applied Mathematics.

Hardt, M.; and Ullman, J. 2014. Preventing False Discovery in Interactive Data Analysis Is Hard. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 454–463. IEEE Computer Society.

Hardt, M.; and Woodruff, D. P. 2013. How Robust Are Linear Sketches to Adaptive Inputs? In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, 121–130. New York, NY, USA: Association for Computing Machinery. ISBN 9781450320290.

Hassidim, A.; Kaplan, H.; Mansour, Y.; Matias, Y.; and Stemmer, U. 2020. Adversarially Robust Streaming Algorithms via Differential Privacy. In *Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*.

Ioannidis, J. P. A. 2005. Why Most Published Research Findings Are False. *PLoS Med*, (2): 8.

Johnson, W.; and Lindenstrauss, J. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Math.*, 26.

Kamath, A.; Price, E.; and Woodruff, D. P. 2021. *A Simple Proof of a New Set Disjointness with Applications to Data Streams*. Dagstuhl, DEU: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 9783959771931.

Lukacs, P. M.; Burnham, K. P.; and Anderson, D. R. 2009. Model selection bias and Freedman's paradox. *Annals of the Institute of Statistical Mathematics*, 62(1): 117.

Mironov, I.; Naor, M.; and Segev, G. 2008. Sketching in Adversarial Environments. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, 651–660. New York, NY, USA: Association for Computing Machinery. ISBN 9781605580470.

Moody, J. E.; and Darken, C. J. 1989. Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Comput.*, 1(2): 281–294.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Pham, N.; and Pagh, R. 2013. Fast and Scalable Polynomial Kernels via Explicit Feature Maps. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, 239–247. New York, NY, USA: Association for Computing Machinery. ISBN 9781450321747.

Rothchild, D.; Panda, A.; Ullah, E.; Ivkin, N.; Stoica, I.; Braverman, V.; Gonzalez, J.; and Arora, R. 2020. FetchSGD: Communication-Efficient Federated Learning with Sketching. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 8253–8265. PMLR.

Shi, Q.; Petterson, J.; Dror, G.; Langford, J.; Smola, A.; Strehl, A.; and Vishwanathan, S. V. N. 2009. Hash Kernels. In van Dyk, D.; and Welling, M., eds., *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, 496–503. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.

Shiloach, Y.; and Even, S. 1981. An On-Line Edge-Deletion Problem. *J. ACM*, 28(1): 1–4.

Spring, R.; Kyrillidis, A.; Mohan, V.; and Shrivastava, A. 2019. Compressing Gradient Optimizers via Count-Sketches. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5946–5955. PMLR.

Steinke, T.; and Ullman, J. 2015. Interactive Fingerprinting Codes and the Hardness of Preventing False Discovery. In Grünwald, P.; Hazan, E.; and Kale, S., eds., *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, 1588–1628. Paris, France: PMLR.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Wajc, D. 2020. *Rounding Dynamic Matchings against an Adaptive Adversary*. New York, NY, USA: Association for Computing Machinery.

Weinberger, K.; Dasgupta, A.; Langford, J.; Smola, A.; and Attenberg, J. 2009. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 1113–1120. New York, NY, USA: Association for Computing Machinery. ISBN 9781605585161.

Woodruff, D. P.; and Zhou, S. 2021. Tight Bounds for Adversarially Robust Streams and Sliding Windows via Difference Estimators. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*.