Unmasking Vulnerabilities: Cardinality Sketches under Adaptive Inputs

Sara Ahmadian [* 1]   **Edith Cohen** [* 1 2]

Abstract

Cardinality sketches are popular data structures that enhance the efficiency of working with large data sets. The sketches are randomized representations of sets that are only of logarithmic size but can support set merges and approximate cardinality (i.e., distinct count) queries. When queries are not adaptive, that is, they do not depend on preceding query responses, the design provides strong guarantees of correctly answering a number of queries exponential in the sketch size k. In this work, we investigate the performance of cardinality sketches in adaptive settings and unveil inherent vulnerabilities. We design an attack against the "standard" estimators that constructs an adversarial input by post-processing responses to a set of simple non-adaptive queries of size linear in the sketch size k. Empirically, our attack used only $4k$ queries with the widely used HyperLogLog (HLL++) (Flajolet et al., 2007b; Heule et al., 2013) sketch. The simple attack technique suggests it can be effective with post-processed natural workloads. Finally and importantly, we demonstrate that the vulnerability is inherent as *any* estimator applied to known sketch structures can be attacked using a number of queries that is quadratic in k, matching a generic upper bound.

1. Introduction

Composable sketches for cardinality estimation are data structures that are commonly used in practice (Apache Software Foundation, Accessed: 2024; Google Cloud, Accessed: 2024) and had been studied extensively (Flajolet & Martin, 1985; Flajolet et al., 2007a; Cohen, 1997; Bar-Yossef et al., 2002; Kane et al., 2010; Nelson et al., 2014; Cohen, 2015; Pettie & Wang, 2021). The sketch of a set is a compact

representation that supports merge (set union) operations, adding elements, and retrieval of approximate cardinality. The sketch size is only logarithmic or double logarithmic in the cardinality of queries, which allows for a significant efficiency boost over linear size data structures such as Bloom filters.

Formally, for a universe $\mathcal{U}$ of keys, and randomness ρ, a sketch map $U \mapsto S_\rho(U)$ is a mapping from sets of keys $U \in 2^{\mathcal{U}}$ to their sketch $S_\rho(U)$. Sketch maps are designed so that for each U we can recover with high probability (over ρ) an estimate of $|U|$ by applying an estimator $\mathcal{M}$ to $S_\rho(U)$. A common guarantee is a bound on the Normalized Root Mean Squared Error (NRMSE) so that for accuracy parameter ϵ,

$$\forall U, \ \mathsf{E}_\rho \left[\left(\frac{\mathcal{M}(S_\rho(U)) - |U|}{|U|} \right)^2 \right] \le \epsilon^2 \ . \qquad (1)$$

The maps S_ρ are designed to be *composable*: For a set U and key u, the sketch $S_\rho(U \cup \{u\})$ can be computed from $S_\rho(U)$ and u. For two sets U, V, the sketch $S_\rho(U \cup V)$ can be computed from their respective sketches $S_\rho(U)$, $S_\rho(V)$. Composability is a crucial property that makes the sketch representations useful for streaming, distributed, and parallel applications. Importantly, the use of the same internal randomness ρ across all queries is necessary for composability and therefore in typical use cases it is fixed across a system.

The basic technique in the design of cardinality (distinct count) sketches is to randomly prioritize the keys in the universe $\mathcal{U}$ through the use of random hash functions (specified by ρ). The sketch of a set U keeps the lowest priorities of keys that are in the set U. This provides information on the cardinality $|U|$, since a larger cardinality corresponds to the presence of lower priorities keys in U. This technique was introduced by Flajolet & Martin (1985) for counting distinct elements in streaming and as composable sketches of sets by Cohen (1997). The core idea of sampling keys based on a random order emerged in reservoir sampling (Knuth, 1998), and in weighted sampling (Rosén, 1997). Cardinality sketches are also Locality Sensitive Hashing (LSH) maps (Indyk & Motwani, 1998) with respect to set differences.

The specific designs of cardinality sketches vary and in-

---

*Equal contribution   [1]Google Research, United States [2]Department of Computer Science, Tel Aviv University, Israel. Correspondence to: Sara Ahmadian <sahmadian@google.com>, Edith Cohen <edith@cohenwang.com>.

clude *MinHash sketches* (randomly map keys to priorities) or *domain sampling* (randomly map keys to sampling rates). With these methods, the sketch size dependence on the maximum query size $|U| \leq n$ is $\log n$ or $\log \log n$. The sketch size (number of registers) needed for the NRMSE guarantee of Equation (1) is $k = O(\epsilon^{-2})$. The sketch size needed for the following (ϵ, δ) guarantee (confidence $1 - \delta$ of relative error of ϵ)

$$\forall U, \ \Pr_\rho \left[\left| \frac{\mathcal{M}(S_\rho(U)) - |U|}{|U|} \right| > \epsilon \right] \leq \delta \qquad (2)$$

is $k = O(\epsilon^{-2} \log(1/\delta))$. This guarantee means that for any U, almost any sampled ρ works well.

We model the use of the sketching map S_ρ as an interaction between a *source*, that issues queries $U_i \subset \mathcal{U}$, and a *query response (QR)* algorithm, that receives the sketch $S_\rho(U_i)$ (but not U_i) applies an *estimator* $\mathcal{M}$ and returns the estimate $\mathcal{M}(S_\rho(U_i))$ on the cardinality $|U_i|$.

When randomized data structures or algorithms are invoked interactively, it is important to make a distinction between non-adaptive queries, that do not depend on ρ, and adaptive queries. In non-adaptive settings we can treat queries as fixed in advance. In this case, we can apply a union bound with the guarantee of Equation 2 and obtain that the probability that the responses for all r queries are within a relative error of ϵ, is at least $1 - r\delta$. Therefore, the sketch size needed to provide an (ϵ, δ)-guarantee on this ℓ_∞ error is $k = O(\epsilon^{-2} \log(r/\delta))$. In particular, the query response algorithm can be correct on a number of non-adaptive queries that is exponential in the sketch size until a set is encountered for which the estimate is off.

Many settings, however, such as control loops, optimization processes, or malicious behavior, give rise to adaptive inputs. This can happen inadvertently when a platform such as Apache Software Foundation (Accessed: 2024) or SQL (Google Cloud, Accessed: 2024) is used. In such cases, information on the randomness ρ may leak from query responses, and the union bound argument does not hold. An important question that arises is thus to understand the actual vulnerability of our specific algorithms in such settings. Are they practically robust? How efficiently can they be attacked? What can be the consequences of finding such an adversarial input?

Randomized data structures designed for non-adaptive queries can be applied in generic ways with adaptive queries. However, the guarantees provided by the resulting (robust) algorithms tend to be significantly weaker than those of their non-robust counterparts. The straightforward approach is to maintain multiple copies of the sketch maps (with independent randomness) and discard a copy after it is used once to respond to a query. This results in a linear dependence of the number of queries r in the size of the data

structure. Hassidim et al. (2020) proposed the robustness wrapper method that allows for r^2 adaptive queries using $\tilde{O}(r)$ sketch maps. The method uses differential privacy to protect the randomness and the analysis uses generalization (Dwork et al., 2015b; Bassily et al., 2021) and advanced composition (Dwork et al., 2006). The quadratic relation is known to be tight in the worst-case for adaptive statistical queries with an attack (Hardt & Ullman, 2014; Steinke & Ullman, 2015) designed using Fingerprinting Codes (Boneh & Shaw, 1998). But these attacks do not preclude a tailored design with a better utility guarantee for cardinality sketches and also do not apply with "natural" workloads.

Contributions and Overview

We consider the known sublinear composable sketch structures for cardinality estimation, which we review in Section 3. Our primary contribution is designing attacks that construct a set U that is *adversarial* for the randomness ρ. We make this precise in the sequel, but for now, an adversarial set U results in cardinality estimates that are off.

- We consider query response algorithms that use the "standard" cardinality estimators. These estimators optimally use the information in the sketch and report a value that is a function of a sufficient statistic of the cardinality. In Section 4 we present an attack on these estimators. The product of the attack is an adversarial set, one for which the sketch $S_\rho(U)$ is grossly out of distribution. The attack uses linearly many queries $O(k)$ in the sketch size and importantly, issues all queries in a single batch. The only adaptive component is the post processing of the query responses. This single-batch attack suggests that it is possible to construct an adversarial input by simply observing and post-processing a normal and non-adaptive workload of the system. The linear size of the attack matches the straightforward upper bound of using disjoint components of the sketch for different queries.

- We conduct an empirical evaluation of our proposed attack on the HyperLogLog (HLL) sketch (Durand & Flajolet, 2003; Flajolet et al., 2007a) with the HLL++ estimator (Heule et al., 2013). This is the most widely utilized sketch for cardinality estimation in practice. The results reported in Section 5 show that even with a single-batch attack using $4k$ queries, we can consistently construct adversarial inputs on which the estimator substantially overestimates or underestimates the cardinality by 40%.

- In Section 6 and Section 7, we present an attack that broadly applies against *any* correct query response algorithm. By that, we establish inherent vulnerability of the sketch structures themselves. Our attack uses

$\tilde{O}(k^2)$ *adaptive* queries. We show that multiple batches are necessary against strategic query response algorithms. This quadratic attack size matches the generic quadratic upper bound construction of Hassidim et al. (2020). The product of our attack is a small *mask* set M that can poison larger sets U in the sense that $S(M \cup U) \approx S(M)$, making any estimator ineffective. The attack applies even when the query response is for the more specialized *soft threshold* problem: Determine if the cardinality is below or above a range of the form $[A, 2A]$. Moreover, it applies even when the response is tailored to the attack algorithm and its internal state including the distribution from which the query sets are selected at each step. Note that this strengthening of the query response and simplification of the task only makes the query response algorithm harder to attack.

Our attacks have the following structure: We fix a ground set N of keys and issue queries that are subsets $U_i \subset N$. We maintain scores to keys in N that are adjusted for the keys in U_i based on the response. The design has the property that scores are correlated with the priorities of keys and the score is higher when the cardinality is underestimated. The adversarial set is then identified as a prefix or a suffix of keys ordered by their score.

The vulnerabilities we exposed may have practical significance in multiple scenarios: In a non-malicious setting, an adaptive algorithm or an optimization process that is applied in sketch space can select keys that tend to be in overestimated (or underestimates) sets, essentially emulating an attack and inadvertently selecting a biased set on which the estimate is off. In malicious settings, the construction of an adversarial input set U can be an end goal. For example, a system that collects statistics on network traffic can be tricked to report that traffic is much larger or much smaller than it actually is. A malicious player can poison the dataset by injecting a small adversarial set M to the data U, for example, by issuing respective search queries to a system that sketches sets of search queries. The sketch $S_\rho(M \cup U)$ then masks $S_\rho(U)$, making it impossible to recover an estimate of the true cardinality of U. Finally, cardinality sketches have weighted extensions (max-distinct statistics) and are building blocks of sketches designed for a large class of concave sublinear frequency statistics, that include cap statistics and frequency moments with $p \leq 1$ (Cohen, 2018; Cohen & Geri, 2019; Jayaram & Woodruff, 2023), and thus these vulnerabilities apply to these extensions.

2. Related Work

There are prolific lines of research on the effect of adaptive inputs that span multiple areas including dynamic graph algorithms (Shiloach & Even, 1981; Ahn et al., 2012; Gawrychowski et al., 2020; Gutenberg & Wulff-Nilsen, 2020; Wajc, 2020; Beimel et al., 2021), sketching and streaming algorithms (Mironov et al., 2008; Hardt & Woodruff, 2013; Ben-Eliezer et al., 2021b; Hassidim et al., 2020; Woodruff & Zhou, 2021; Attias et al., 2021; Ben-Eliezer et al., 2021a; Cohen et al., 2022a;b), adaptive data analysis (Freedman, 1983; Ioannidis, 2005; Lukacs et al., 2009; Hardt & Ullman, 2014; Dwork et al., 2015a) and machine learning (Szegedy et al., 2013; Goodfellow et al., 2014; Athalye et al., 2018; Papernot et al., 2017).

Reviriego & Ting (2020) and Paterson & Raynal (2021) proposed attacks on the HLL sketch with standards estimators. The proposed attacks were in a streaming setting and utilized many dependent queries in order to detect keys whose insertion results in updates to the cardinality estimate. Our attacks are more general: We construct single-batch attacks with standard estimators and also construct attacks on these cardinality sketches that apply with any estimator. The question of robustness of cardinality sketches to adaptive inputs is related but different than the well studied question of whether they are differentialy private. Cardinality sketches were shown to be not privacy preserving when the sketch randomness or content are public (Desfontaines et al., 2019). Other works (Smith et al., 2020; Pagh & Stausholm, 2021; Knop & Steinke, 2023) showed that cardinality sketches are privacy preserving, but this is under the assumption that the randomness is used once. Our contribution here is designing attacks and quantifying their efficiency. The common grounds with privacy is the high sensitivity of the sketch maps to removal or insertion of low priority key.

Several works constructed attacks on linear sketches, including the Johnson Lindenstrauss Transform (Cherapanamjeri & Nelson, 2020), the AMS sketch (Ben-Eliezer et al., 2021b; Cohen et al., 2022b), and CountSketch (Cohen et al., 2022a;b). The latter showed that the standard estimators for CountSketch and the AMS sketch can be compromised with a linear number of queries and the sketches with arbitrary estimators can be compromised with a quadratic number of queries. The method was to combine low bias inputs with disjoint supports and have the bias amplified, since the bias increases linearly whereas the ℓ_2 norms increases proportionally to $\sqrt{r}$. This approach does not work with cardinality sketches, which required a different attack structure. Combining disjoint sets on which the estimate is slightly biased up will not amplify the bias. The common ground, perhaps surprisingly, is that these fundamental and popular sketches are all vulnerable with adaptive inputs and in a similar manner: Estimators that optimally use the sketch require linear size attacks. Arbitrary correct estimators require quadratic size attacks.

3. Preliminaries

An *attack* is an interaction designed to construct a set that is *adversarial* to the randomness ρ. An adversarial set can be identified by trying out a large number of inputs. We measure the efficiency of the attack by its *size* (number of issued queries) and *concurrency* (number of batches of concurrent queries).

A set U is *adversarial* for the randomness ρ if a *sufficient statistics* for the cardinality that is computed from $S_\rho(U)$ is very skewed with respect to its distribution under sampling of ρ. That is, it has proportionally too few or two many low priority keys.

Definition 3.1 (Sufficient Statistics). A statistic T on the sketch domain $S \mapsto \mathbb{R}$ is *sufficient* for the cardinality $|U|$ if it includes all information the sketch provides on the cardinality $|U|$. That is, for each t, the conditional distribution of the random variable $S_\rho(U)$ given $T(S_\rho(U)) = t$, does not depend on $|U|$.

3.1. Composable Cardinality Sketches

The underlying technique in all small space cardinality sketches is to use random hashmaps h that assign "priorities" to keys in $\mathcal{U}$.[1] The sketch of a set is specified by the priorities of a small set of keys with the lowest priorities. This information is related to the cardinality as smaller lowest priorities in a subset U are indicative of larger cardinality. Therefore cardinality estimates can be recovered from the sketch. We describe several common designs. MinHash sketches (see surveys (Cohen, 2008; 2023)) are suitable for insertions only (set unions and insertions of new elements) and are also suitable for sketch-based sampling. Domain sampling has priorities that are discretized sampling rates and has the advantage that the sketch can be represented as random linear maps (specified by ρ) of the data vector and therefore have support for deletions (negative entries in sketched vectors) (Ganguly, 2007).

MinHash sketches   Types of MinHash sketches

- *k-mins* (Flajolet & Martin, 1985; Cohen, 1997) k random hash functions $h_1, \ldots, h_k$ that map each key $x \in \mathcal{U}$ to i.i.d samples from the domain of the hash function. The sketch $S_\rho(U)$ of a set U is the list $(\min_{x \in U} h_i(x))_{i \in [k]}$ of minimum values of each hash function over the keys in U. The sketch distribution for a subset U is $\mathsf{Exp}[|U|]^k$, a set of k i.i.d. exponentially distributed random variables with parameter $|U|$. The sum $T(S) := \|S\|_1$ is a *sufficient statistics* for estimating the parameter $|U|$. An unbiased cardinality

estimator is $(k-1)/T(S)$.

- *Bottom-k* (Rosén, 1997; Cohen, 1997; Bar-Yossef et al., 2002) One random hash function h that maps $x \in \mathcal{U}$ to i.i.d samples from a distribution. The sketch $\{h(x) \mid x \in U\}_{(1:k)}$ stores the k smallest hash values of keys $x \in U$. The kth smallest value $T(S) := \{h(x) \mid x \in U\}_{(k)}$ is a sufficient statistics for estimating $|U|$. When the distribution is $U[0, 1]$, the unbiased cardinality estimate is $(k-1)/T(S)$.

- *k-partition* (Flajolet et al., 2007a) One hash $P : \mathcal{U} \to [k]$ randomly partition keys to k parts. One hash function $h : \mathcal{U}$ maps keys to i.i.d $\mathsf{Exp}[1]$. The sketch includes the minimum in each part $(\min_{x \in U \mid P(x)=i} h(x))_{i \in [k]}$.

Note that the choice of (continuous) distribution does not affect the information content in the sketch. Variations of these sketches store rounded/truncated numbers (HLL (Flajolet et al., 2007a) stores a maximum negated exponent). When studying vulnerabilities of query response algorithms, the result is stronger when the full precision representation is available to them.

The cardinality estimates obtained with these sketches have NRMSE error (1) of $1/\sqrt{k}$.

Definition 3.2 (bias of the sketch). We say that the sketch $S_\rho(U)$ of a set U is *biased up* by a factor of $1/\alpha$ when $T(S_\rho(U)) \leq \alpha k/|U|$ and we say it is *biased down* by a factor of α when $T(S_\rho(U)) \geq (1/\alpha)k/|U|$.

For our purposes, $\alpha \leq 1/2$ would places the sketch at the $\delta = e^{-\Omega(k)}$ tail of the distribution under sampling of ρ and we say that U is adversarial for ρ.

Domain sampling   These cardinality sketches can be expressed as discretized bottom-k sketches. Therefore vulnerabilities of bottom-k sketches also apply with domain sampling sketches. The input is viewed as a vector of dimension $|\mathcal{U}|$ where the set U corresponds to its nonzero entries. The cardinality $|U|$ is thus the sparsity (number of nonzero entries). The sketch map S_ρ is a dimensionality reduction via a random linear map (specified by ρ).

We sample the domain $\mathcal{U} = [n]$ with different rates $p = 2^{-j}$. For each rate, we collect a count c_j (capped by k) of the number of sampled keys from our set X. This can be done by storing the first k distinct keys we see or (approximately) by random hashing into a domain of size k and considering how many cells were hit. A continuous version known as *liquid legions* (Kreuter et al., 2020)) is equivalent to a bottom-k sketch: Each key is assigned a random i.i.d. priority (lowest sampling rate in which it is counted with domain sampling) and we seek the sampling rate with which we have k keys.

---

[1](Chakraborty et al., 2022) proposed a method for streaming cardinality estimates that does not require hashmaps but the sketch is not composable.

Algorithm 1: `Attack ''standard'' estimators`

Input: ρ, n, r, T
Fix a set N of n keys `// selected randomly independently of ρ)`
foreach *key* $x \in N$ **do**   `// initialize`
  $t[x] \leftarrow 0$
  $c[x] \leftarrow 0$
foreach $i = 1, \ldots, r$ **do**
  $U \leftarrow$ include each $x \in N$ independently with prob $\frac{1}{2}$
  foreach *key* $x \in U$ **do**   `// score keys`
    $t[x] \leftarrow t[x] + 1$
    $c[x] \leftarrow c[x] + T(S_\rho(U))$

---

return *The keys in N ordered by average score $A[x] = \frac{c[x]}{t[x]}$.*

Specifying keys for the sketch   Note that with all these sketch maps, the sketch of a set U is specified by a small subset $U_0 \subset U$ of the "lowest priority" keys in U. With k-mins and k-partition sketches it is the keys $\arg\min_{x \in U}\{h_i(x)\}$ for $i \in [k]$. With bottom-k sketches, it is the keys with k smallest values in $\{h_i(x)\}_{x \in U}$. With domain sampling, it is the keys with the highest sampling rate. Note that $|U_0| = O(k) \ll |U|$ but $S_\rho(U_0) = S_\rho(U)$.

4. Attack on the "standard" estimators

The "standard" cardinality estimators optimally use the content in the sketch. They can be equivalently viewed as reporting a sufficient statistics. We design a single-batch attack described in Algorithm 1. The algorithm fixes a ground set N of keys. For r queries, it samples a random subset $U \subset N$ where each $u \in N$ is included independently with probability $1/2$. It receives from the estimator the value of the sufficient statistics $T(S_\rho(U)) := 1/M(S_\rho(U))$ (we use the inverse of the cardinality estimate). For each key $x \in N$ it computes a score $A[x]$ that is the average value of $T(S_\rho(U))$ over all subsets where $x \in U$.

We show that for $\alpha > 0$, an attack of size $O(r/\alpha^2)$ produces an adversarial set with sketch that is biased up by a factor α (see Definition 3.2).

Theorem 4.1 (Utility of Algorithm 1). *Consider Algorithm 1 with k-mins or bottom-k sketches and $T(S)$ being the inverse of the cardinality estimate as specified in Section 3.1. For $\alpha > 0$, set the parameters $n = \Omega(\frac{1}{\alpha}k\log(kr))$ and $r = O\left(\frac{k}{\alpha^2}\right)$. Then with probability at least 0.99, the sketch $S_\rho(U_\alpha)$, where $U_\alpha \subset N$ is the of the αn lowest $A[u]$ scores, is biased up by a factor of $\Omega(1/\alpha)$:*

$$E[M(U_\alpha)] = \Theta(n) .$$

Our analysis extends to the case when the estimator reports $T(S_\rho(U))$ with relative error $O(1/\sqrt{k})$. That is, as long as the estimates are sufficiently accurate (within the order

of the accuracy guarantees of a size k sketch), then $O(k)$ attack queries suffice.

Analysis Highlights   The proof of Theorem 4.1 is presented in Appendix A. The high level idea is as follows. We establish that scores are correlated with the priorities of keys – the keys with lowest priorities have in expectation lower scores. Therefore a prefix of the order will contain disproportionately more of them and overestimate the cardinality and a suffix will contain disproportionately fewer of them and underestimate the cardinality.

We consider, for each key $x \in N$, the distributions of $T(S_\rho(U))$ conditioned on $x \in U$. We bound from above the variance of these distributions and bound from below the gap in the means of the distributions between the keys that have the "lowest priority" in N and the bulk of other keys in N. We then apply Chebyshev's Inequality to bound the number of rounds that is needed so that enough of the low priority keys have lower average scores $A[u]$ than "most" other keys. A nuance we overcame in the analysis was to handle the dependence of the sketches of the different queries that are selected from the same ground set.

5. Experimental Evaluation

In this section, we empirically demonstrate the efficacy of our proposed attack (Algorithm 1) against the Hyper-LogLog (HLL) sketch (Durand & Flajolet, 2003; Flajolet et al., 2007a) with the HLL++ estimator (Heule et al., 2013). This is the most widely utilized sketch for cardinality estimation in practice. Given an accuracy parameter ϵ, the HLL sketch stores $k = 1.04\epsilon^{-2}$ values that are the negated exponents of a k-partition MinHash sketch (described in Section 3.1).

The HLL++ estimator is a hybrid that was introduced in order to improve accuracy on low cardinality queries. When the sketch representation is sparse (fewer parts are populated), which is the case with cardinality lower than the sketch size, HLL++ uses the sketch as a hash table and estimates cardinality based on the number of populated parts. This yields essentially precise values. When all parts are populated, HLL++ uses an estimator based on the MinHash property. We set the size of our ground set $n \gg k$ to be in this relevant regime.

We conduct two primary experiments: (i) For a fixed sketch size, we analyze the efficacy of the attack with a varying number of queries. (ii) For different sketch sizes, we evaluate the effectiveness of the attack with the number of queries linearly proportional to the sketch size. In the following section, we will first provide a detailed explanation of the ingredients required for our experimental setup. Subsequently, we will present the results of each experiment.

Experiment setup. To generate the data, we ensure that a ground set with a size of at least $10 \cdot k$ is produced for a given sketch size k. The size of the ground set must be at least linearly larger than the sketch size to prevent the sketch from memorizing the entire dataset. Given the desired size of the ground set, we generate random strings using the English alphabet of a fixed length, where the length is appropriately chosen so that we can generate the desired size set with different strings.

We utilize the open-source implementation of HLL++ algorithm in github. In this implementation, the sketch is fixed by giving the error rate $\epsilon \in (0, 1)$ and the sketch size k for error rate ϵ is $\lceil 1.04/\epsilon^2 \rceil$ (consistent with Flajolet et al. (2007a)).

5.1. Efficacy with a varying number of queries

In this experiment we examine the impact of introducing a variable quantity of queries. The attack is executed with the same ground set for eight distinct query counts, where each count is a power of 4. At the conclusion, the algorithm generates scores and returns keys sorted in ascending order according to their scores. Keys with high score correspond to low-priority keys which are expected to appear when the estimate is biased up. By including these keys in the adversarial set, we basically can trick the estimator to think that they are seeing a sketch of a large set. Similarly we can construct adversarial input sets by including keys with low scores and trick the estimator to think they are seeing a sketch of a small set.

We present two sets of plots corresponding to how the estimator overestimates or underestimates as keys are incrementally added to the adversarial input in the increasing or decreasing order of their scores. We consider two different error rates, $\epsilon = 0.1$ with corresponding sketch size $k = 104$ and $\epsilon = 0.05$, with corresponding sketch size $k = 416$. We use the same ground set comprising of 5000 keys for both sets of experiments. It is worth noting that the plot with one query, which oscillates around the line $y = x$ (denoted by a dashed line), is close to a non-adversarial setting and we can see that the estimates are within the desired specified error of ϵ.

Figure 1 reports cardinality estimates when keys are added incrementally in increasing order of their scores. We can see that as we increase the number of queries, the gap between estimated value and the $y = x$ line (actual value) widens. This gap indicate the overestimation error. Our algorithm is able to construct more effective adversarial input with a larger number of queries. However the gain in effectiveness becomes marginal at some point. For example, for $k = 104$, we already see good degree of error in estimation with 4096 queries. Figure 2 reports results when keys are added incrementally in *decreasing* order of their scores. The gap
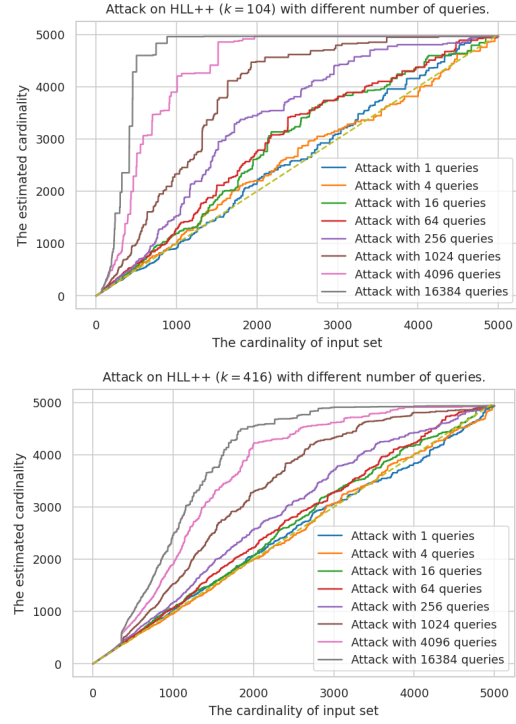

Figure 1. Attack on the HLL++ sketch and estimator, for varying number of queries. Cardinality estimates for the prefix of keys with lowest average score after $r = 4^i$ queries.

here corresponds to an *underestimation* error. We can see that the attacks are more effective with more queries.

5.2. Efficacy of the attack with a varying sketch sizes

In this section, our focus is on examining HLL++ with different sketch sizes, namely we consider six different error rates corresponding to sketch sizes $k = 2^i$ for i ranging from 6 to 11. For each sketch size k, we generate a ground set of size $n = 10 * 10^{\lceil log_{10}(k) \rceil}$ to ensure that the ground set is larger than sketch size and the MinHash component of the HLL++ estimator is used. In Figure 3, we report the ratio of estimated size to actual size of the set for all subsets constructed as a prefix of the order on keys, sorted by increasing average scores $A[x]$ for a fixed number of queries set to $4k$. In this cardinality regime, HLL++ is nearly unbiased and we expect a ratio that is close to 1 when the queries are not adversarial. However by running attacks with enough number of queries (linear in the size of sketch), we are able to identify keys with low-priority and then trick the estimator to give an estimate for a set much higher than the actual size.
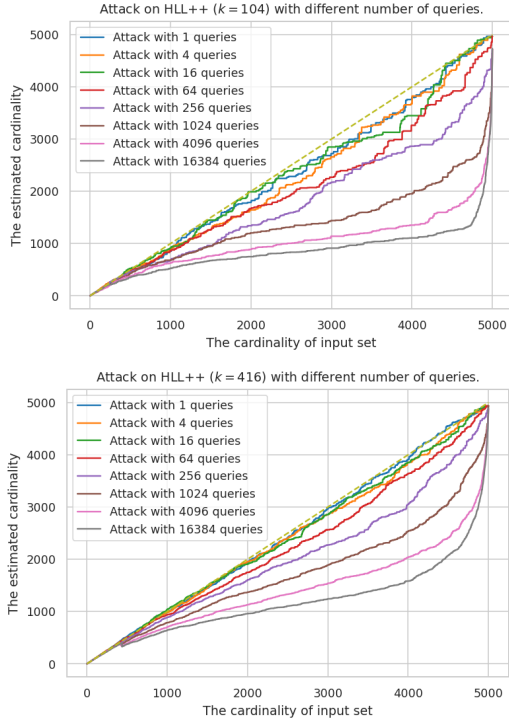
Figure 2. Attack on the HLL++ sketch and estimator, for varying number of queries. Cardinality estimates for the prefix of keys with largest average score after $r = 4^i$ queries.

6. Attack Setup Against Strategic Estimators

We design attacks that apply generally against any query response (QR) algorithm. The attacks are effective even when the specifics of the attack and the full internal state of the attack algorithm are shared with the QR algorithm, including the per-step distribution from which the attacker selects each query. Moreover, we can even assume that the QR algorithm is provided with an enhanced sketch that includes the identities of the low priority keys that determined the sketch and that after the QR algorithm responds to a query, the full query set U is shared with it. The only requirement from the QR algorithm is that it selects *correct* response maps (with respect to the query distribution). Note that such a powerful QR algorithm precludes attacks that use queries of fixed cardinality, since the QR algorithm can simply return that cardinality value without even considering the actual input sketch.[2]

Moreover, the task of the QR algorithm is the following problem that is more specialized than cardinality estimation:

Problem 6.1 (Soft Threshold A). Return 0 when $|U| \leq A$ and 1 when $|U| \geq 2A$.

---

[2]Our attack in Algorithm 1 is also precluded, since a fixed response of $n/2$ satisfies the requirements (the cardinality is $\mathsf{Binom}(n,p)$ and for $n \gg k$, all queries have size close to $n/2$).
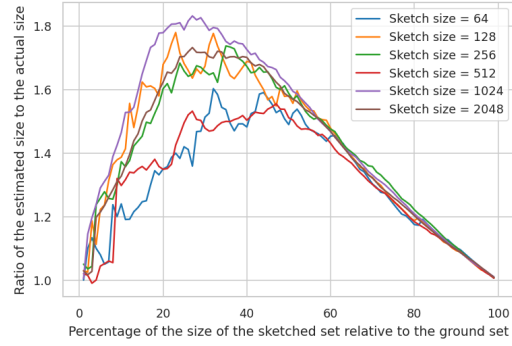

Figure 3. Attack on HLL++ for varying sketch sizes while utilizing queries of size 4 times the sketch size.

Remark 6.2. Soft Threshold can be solved via a cardinality estimate with a multiplicative error of at most $\sqrt{2}$ by reporting 1 when the estimate is larger than $\sqrt{2}A$ and 0 otherwise. When estimates are computed from cardinality sketches with randomness ρ that does not depend on the queries, a sketch of size $k = \Theta(\log(1/\delta))$ is necessary and suffices for providing correct responses with probability $\geq 1 - \delta$.

Attack Framework We describe the attack framework. We specify attacks in Sections 7 and 8. We model the interaction as a process between three parties: the *Attacker*, the *QR algorithm*, and *System*. The attacker fixes a ground set N from which it samples query sets. The product of the attack is a subset $M \subset N$ which we refer to as a *mask*. The aim is for the mask to have size that is much smaller than our query subset sizes and the property that for uniformly sampled $U \subset N$ ($|U| \gg |M|$) the information in the sketch $S_\rho(M \cup U)$ is insufficient to estimate $|U|$. The attack proceeds in steps described in Algorithm 2:

Algorithm 2: Attack Interaction Step

• Attacker specifies a distribution $\mathcal{D}$ over the support 2^N and sends it to QR. It then selects a query $U \sim \mathcal{D}$ and sends it to *System*.
• QR selects a *correct* map with $\delta = O(1/\sqrt{k})$ (as in Definition 6.3) of sketches to probabilities $S \mapsto \pi(S) \in [0, 1]$. The selection may depend on the prior interaction transcript and on $\mathcal{D}$. If there is no correct map QR reports **failure** and halts.
• *System* computes the sketch $S_\rho(U)$ and sends it to QR.
• QR sends $Z \sim \mathsf{Bern}[\pi(S_\rho(U))]$ to Attacker. Attacker shares U and its internal state with QR.

Definition 6.3 (Correct Map). We say that the map π is *correct* for A and δ and query distribution $\mathcal{D}$ if for any cardinality value, over the query distribution for this value, it returns a correct response to a soft threshold problem with

A with probability at least $1 - \delta$. That is,

$$\text{for } c < A, \ \mathsf{E}_{U \sim \mathcal{D}||U|=c}(\pi(S_\rho(U))) \leq \delta$$
$$\text{for } c > 2A, \ \mathsf{E}_{U \sim \mathcal{D}||U|=c}(\pi(S_\rho(U))) \geq 1 - \delta \ .$$

Remark 6.4 (Many correct maps). There can be multiple correct maps and QR may choose any one at any step. Since the output when $|U| \in [A, 2A]$ is not specified, the probability $\mathsf{E}_{U \sim \mathcal{D}} \pi(S_\rho(U))$ of reporting $Z = 1$ may vary by $\approx \Pr_{U \sim \mathcal{D}}[|U| \in [A, 2A]] + \delta$ between correct maps.

Recall that our attack on the standard estimators (Algorithm 1) issued a single batch of queries (all drawn from the same pre-specified distribution $\mathcal{D}_0$). We show that multiple batches are necessary to attack general QR algorithms:

Lemma 6.5 (Multiple batches are necessary). *Any attack of polynomial size in k on a soft threshold estimator must use multiple batches.*

Proof. When there is a single batch of r queries, we can apply the standard estimator while accessing only a "component" of the sketch that is of size $k' = O(\log(r/\delta))$ and obtain correct responses on all queries. This component is the same k' hash functions with k-mins sketches, only k' parts in a k-partition sketch, or the bottom-k' values in a bottom-k sketch. Since the query response only leaks information on this component, the attacker is only able to compromise that component in this single-batch attack. Therefore, an exponential number of queries in k is needed in order to construct an adversarial input in a single batch. $\square$

7. Single-batch attack on symmetric QR

Algorithm 3 specifies a single-batch attack. We establish that the attack succeeds when we set the size $r = \tilde{O}(k^2)$ and QR is constrained to be *symmetric* (see Definition 7.2). In essence symmetry means that the QR algorithm does not make a significant distinction between components of the sketch. Symmetry excludes strategies that distinguish between components of the sketch as in the proof of Lemma 6.5 but still allows for flexibility including randomly selecting components of the sketch. In Section 8 we extend this attack to an adaptive attack that works against *any* QR algorithm.

The attacker initializes the scores $C[x] \leftarrow 0$ of all keys $x \in N$ in the ground set. Each query is formed by sampling a rate q (as described in Algorithm 5) and selecting a random subset $U \subset N$ so that each key in N is included independently with probability q. The attacker receives Z and increments by Z the score $C[x]$ of all keys $x \in U$. The final product is the set M of keys with scores that are higher by $\tilde{\Omega}(r/k)$ than the median score.

Theorem 7.1 (Utility of Algorithm 3 with symmetric maps).

Algorithm 3: `Single Batch Attacker`

Input: ρ, n, r
Select a set N of n keys   `// Randomly from` $\mathcal{U}$
$A \leftarrow n/16$
foreach *key* $x \in N$ **do** $C[x] \leftarrow 0$ `// initialize`
for $i = 1, \dots, r$ **do**
     Sample q as specified in Algorithm 5   `// using` A, n
     $U \leftarrow$ includes each $u \in N$ independently with prob q
     send U `//` $\rightarrow$ `System`
     receive Z `//` $\leftarrow$ `Symmetric Query Response`
     foreach *key* $x \in U$ **do** $C[x] \leftarrow C[x] + Z$ `// score`
$\overline{C} \leftarrow \text{median}\{C[N]\}$ `// Compute median score`
return $M \leftarrow \{x \in N \mid C[x] > \overline{C} + \tilde{\Omega}(\frac{r}{k})\}$ `// Mask`

For $\alpha > 0$, set $n = \Omega(\frac{1}{\alpha} k \log(kr))$ and $r = \tilde{\Omega}\left(\frac{k^2}{\alpha^2}\right)$. Then $\Pr[(S_\rho(M) = S_\rho(N)) \wedge (|M| < \alpha n)] \geq 0.99$.

7.1. Proof Overview

See Appendix B for details. We work with the *rank-domain* representations of the sketches with respect to the ground set N. This representation simplifies our analysis as it only depends on the rank order of keys by their hash values and by that factors out the hash values. The rank-domain sketches $S^R(U)$ have the form $(Y_1, \dots, Y_k)$ where Y_i are positive integers in $[N]$. The sketch distribution over the sampling of U for fixed q is that of k independent $\mathsf{Geom}[q]$ random variables Y_i. The sum $T = \sum_{i=1}^{k} Y_i$ is a sufficient statistics for q from the sketch.

Definition 7.2 (symmetric map). A map π is *symmetric* if it uses the rank-domain sketch as an *unordered* set and (ii) is monotone in that if a sketch $S_1 \leq S_2$ coordinate-wise then then $\pi(S_1) \geq \pi(S_2)$.

We denote by N_0^* the set of k lowest-rank (lowest priority) keys. It includes the bottom-k keys with bottom-k sketches, and the minimum hash key with respect to each of the k hashmaps with k-mins and k-partition sketches. Note that $S_\rho(N) = S_\rho(N_0^*)$. We denote by $N' \subset N$ a set of keys that are *transparent* – very unlikely to influence the sketch if included in the attack subsets of Algorithm 3. We show that a key in N_0^* obtains in expectation a higher score than a key in N' and use that to establish the utility claim:

Lemma 7.3 (separation with symmetric maps). *Let π be correct and symmetric (Definition 7.2). Then for any $m \in N_0^*$ and $u \in N'$,*

$$\mathsf{E}_U[\pi(S_\rho(U) \cdot \mathbf{1}(m \in U)] - \mathsf{E}_U[\pi(S_\rho(U) \cdot \mathbf{1}(u \in U)] = \tilde{\Omega}(1/k)$$

Proof. See Appendix B.5. The gap only holds on average for general correct maps (Lemma B.6) but holds per-key when specialized to symmetric maps. $\square$

Proof of Theorem 7.1. The distribution of the score $C[x]$ of

---

Algorithm 4: `Adaptive Attacker`

Input: ρ, n, r

Select a set N of n keys   `// Randomly from` $\mathcal{U}$

$A \leftarrow n/16; M \leftarrow \emptyset$

foreach *key* $x \in N$ **do** $C[x] \leftarrow 0$ `// initialize`

for $i = 1, \ldots, r$ **do**

    Sample $U \sim \mathcal{D}_0$ as in Algorithm 3

    send $M \cup U$ to system

    receive Z from QR

    if *failure* **then exit**

    foreach *key* $x \in U$ **do**   `// score keys`

        $C[x] \leftarrow C[x] + Z$

        if $C[x] \geq median(C[N \setminus M]) + \sqrt{i \log(200nr)/2}$

        then   `// test if score is high`

            $M \leftarrow M \cup \{x\}$

    send M, C, U to QR `// share internal state`

return M `// Mask`

---

all transparent keys $u \in N'$ is identical and is the sum of r independent Poisson random variables $\sum_{i=1}^{r} Z_i \cdot \mathsf{Bern}[q_i]$.

From Lemma 7.3 $\mathsf{E}[C[m]] - \mathsf{E}[C[u]] = \tilde{\Omega}(r/k)$. For $x \in N_0^*$, the gap random variables of different steps may be dependent, but the lower bound on the expected gap in Lemma 7.3 holds even conditioned on transcript. Additionally, the expected gap in each step is bounded in $[-1, 1]$. We can apply Chernoff bounds (Chernoff, 1952) to bound the probability that a sum deviates by more than λ from its expectation

$$\Pr[|C[x] - \mathsf{E}[C[x]]| \geq \lambda] \leq 2e^{-2\lambda^2/r} . \qquad (3)$$

Setting $\lambda = cr/k$ separates a key in N_0^* from a key in N' with probability $1 - 2e^{-2c^2r/k^2}$. Choosing $r = O(k^2 \log |N|)$ we get that the order separates out with high probability all the keys N_0^* from all the keys N'. Note that there are only $\tilde{\Omega}(k)$ non transparent keys $N_0 := N \setminus N'$. Therefore with high probability $N_0^* \subset M \subset N_0$ and (we can fix constants so that) $|M| = \tilde{O}(k) \leq \alpha n$. Since $N_0^* \subset M, S_\rho(M) = S_\rho(N)$. $\qquad \square$

8. Adaptive Attack on General QR

An attack on general QR algorithms is given in Algorithm 4. The attacker maintains an initially empty set $M \subset N$ of keys which we refer to as *mask*. The query sets have the form $M \cup U$, where $U \sim \mathcal{D}_0$ is sampled and scored as in Algorithm 3. A key is added to M when its score separates out from the median score. We establish the following:

Theorem 8.1 (Utility of Algorithm 4). *For* $\alpha > 0$, *set* $n = \Omega(\frac{1}{\alpha} k \log(kr))$ *and* $r = \tilde{\Omega}\left(\frac{k^2}{\alpha^2}\right)$. *Then with probability at least* 0.99, $|M| < \alpha n$ *and there is no correct map for the query distribution* $M \cup U$ *where* $U \sim \mathcal{D}_0$ *is as in Algorithm 3.*

We overview the proof with details deferred to Appendix B. The condition for adding a key to M is such that with probability at least 0.99, only N_0 keys are placed in M, so $|M| \leq \alpha n$ (Claim B.9). If the QR algorithm fails, there is no correct map for the distribution $M \cup \mathcal{D}_0$.[3] It remains to consider the case where the attack is not halted.

Since the mask M is shared with QR "for free," QR only needs to estimate $|U|$ (or q). But the sketch of $M \cup U$ partially masks the sketch of U. The set of non-transparent keys $N_0' \subset N_0$ decreases as M increases. Additionally, the *effective* sketch size $k' \leq k$ is lower (that is, QR only obtains k' i.i.d $\mathsf{Geom}[q]$ random variables). Recall that when $k' < \log(k)/2$, there is no correct map.

With general correct maps, we can only establish a weaker *average* version of the score gap over N_0' keys. This allows some N_0' keys to remain indistinguishable by score from transparent keys. But what works in Attacker's favour is that in this case the score of other N_0 keys must increase faster. Let $p(\pi, M, x)$ be the probability that key x is scored with map π and mask M. The probability is the same for all transparent keys $x \notin N_0'$ and we denote it by $p'(\pi, M)$. We establish (see Lemma B.8) that for a correct map π for $M \cup \mathcal{D}_0$ it holds that

$$\sum_{x \in N_0'} (p(\pi, M, x) - p'(\pi, M)) = \tilde{\Omega}(1) .$$

Therefore, in $r = \tilde{O}(k^2)$ steps, the combined score advantage of N_0 keys is (concentrated well around) $\tilde{O}(k^2)$. But crucially, any one key can not get too much advantage: once $C(x) - \overline{C} = \tilde{\Omega}(k)$ (where $\overline{C}$ is the median score), then key x is placed in the mask M, exits N_0', and stops getting scored. Therefore if QR does not fail, $\tilde{\Omega}(r/k) > |N_0|$ keys are eventually placed in M, which must include all N_0 keys.

9. Conclusion

We demonstrated the inherent vulnerability of the known composable cardinality sketches to adaptive inputs. We designed attacks that use a number of queries that asymptotically match the upper bounds: A linear number of queries with the "standard" estimator and a quadratic number of queries with *any* estimator applied to the sketch. Empirically, our attacks are simple and effective with small constants. An interesting direction for further study is to show that this vulnerability applies with any composable sketch structure. On the positive side, we suspect that restricting the maximum number of queries that any one key can participate in to sublinear (with standard estimators) or subquadratic (with general estimators) would enhance robustness.

---

[3]A situation of no correct maps can be identified by Attacker, by tracking the error rate of QR, even if not declared by QR.

Acknowledgements

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

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)*, pp. 459–467, 2012. doi: 10.1137/1.9781611973099.40. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973099.40.

Apache Software Foundation. DataSketches, Accessed: 2024. URL https://datasketches.apache.org. Apache Software Foundation Documentation.

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

Attias, I., Cohen, E., Shechner, M., and Stemmer, U. A framework for adversarial streaming via differential privacy and difference estimators. *CoRR*, abs/2107.14527, 2021.

Bar-Yossef, Z., Jayram, T. S., Kumar, R., Sivakumar, D., and Trevisan, L. Counting distinct elements in a data stream. In *RANDOM*. ACM, 2002.

Bassily, R., Nissim, K., Smith, A. D., Steinke, T., Stemmer, U., and Ullman, J. R. Algorithmic stability for adaptive data analysis. *SIAM J. Comput.*, 50(3), 2021. doi: 10.1137/16M1103646. URL https://doi.org/10.1137/16M1103646.

Beimel, A., Kaplan, H., Mansour, Y., Nissim, K., Saranurak, T., and Stemmer, U. Dynamic algorithms against an adaptive adversary: Generic constructions and lower bounds. *CoRR*, abs/2111.03980, 2021.

Ben-Eliezer, O., Eden, T., and Onak, K. Adversarially robust streaming via dense-sparse trade-offs. *CoRR*, abs/2109.03785, 2021a.

Ben-Eliezer, O., Jayaram, R., Woodruff, D. P., and Yogev, E. A framework for adversarially robust streaming algorithms. *SIGMOD Rec.*, 50(1):6–13, 2021b.

Boneh, D. and Shaw, J. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inf. Theory*, 44(5):1897–1905, 1998. doi: 10.1109/18.705568. URL https://doi.org/10.1109/18.705568.

Chakraborty, S., Vinodchandran[1], N. V., and Meel, K. S. Distinct elements in streams: An algorithm for the (text) book. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPICS.ESA.2022.34. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2022.34.

Cherapanamjeri, Y. and Nelson, J. 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*, 2020.

Chernoff, H. A measure of the asymptotic efficiency for test of a hypothesis based on the sum of observations. *Annals of Math. Statistics*, 23:493–509, 1952.

Cohen, E. Size-estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences*, 55:441–453, 1997.

Cohen, E. *Min-Hash Sketches*, pp. 1–7. Springer US, Boston, MA, 2008. ISBN 978-3-642-27848-8. doi: 10.1007/978-3-642-27848-8_573-1. URL https://doi.org/10.1007/978-3-642-27848-8_573-1.

Cohen, E. All-distances sketches, revisited: HIP estimators for massive graphs analysis. *TKDE*, 2015. URL http://arxiv.org/abs/1306.3284.

Cohen, E. Stream sampling framework and application for frequency cap statistics. *ACM Trans. Algorithms*, 14(4):52:1–52:40, 2018. ISSN 1549-6325. doi: 10.1145/3234338.

Cohen, E. Sampling big ideas in query optimization. In Geerts, F., Ngo, H. Q., and Sintos, S. (eds.), *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2023, Seattle, WA, USA, June 18-23, 2023*, pp. 361–371. ACM, 2023. doi: 10.1145/3584372.3589935. URL https://doi.org/10.1145/3584372.3589935.

Cohen, E. and Geri, O. Sampling sketches for concave sublinear functions of frequencies. In *NeurIPS*, 2019.

Cohen, E., Lyu, X., Nelson, J., Sarlós, T., Shechner, M., and Stemmer, U. On the robustness of countsketch to adaptive inputs. In *ICML*, volume 162 of *Proceedings*

of Machine Learning Research, pp. 4112–4140. PMLR, 2022a.

Cohen, E., Nelson, J., Sarlós, T., and Stemmer, U. Tricking the hashing trick: A tight lower bound on the robustness of CountSketch to adaptive inputs. *arXiv:2207.00956*, 2022b. doi: 10.48550/ARXIV.2207.00956. URL https://arxiv.org/abs/2207.00956.

Desfontaines, D., Lochbihler, A., and Basin, D. A. Cardinality estimators do not preserve privacy. In *Privacy Enhancing Technologies Symposium*, volume 19, 2019. doi: https://doi.org/10.2478/popets-2019-0018. URL http://arxiv.org/abs/1808.05879.

Durand, M. and Flajolet, P. Loglog counting of large cardinalities (extended abstract). In *ESA*, 2003.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. L. Preserving statistical validity in adaptive data analysis. In *STOC*, pp. 117–126. ACM, 2015a.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. L. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pp. 117–126, New York, NY, USA, 2015b. Association for Computing Machinery. ISBN 9781450335362. doi: 10.1145/2746539.2746580. URL https://doi.org/10.1145/2746539.2746580.

Flajolet, P. and Martin, G. N. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31:182–209, 1985.

Flajolet, P., Fusy, É., Gandouet, O., and Meunier, F. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. *Discrete mathematics & theoretical computer science*, (Proceedings), 2007a.

Flajolet, P., Fusy, E., Gandouet, O., and Meunier, F. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *Analysis of Algorithms (AofA)*. DMTCS, 2007b.

Freedman, D. A. A note on screening regression equations. *The American Statistician*, 37(2):152–155, 1983. doi: 10.1080/00031305.1983.10482729. URL https://www.tandfonline.com/doi/abs/10.1080/00031305.1983.10482729.

Ganguly, S. Counting distinct items over update streams. *Theoretical Computer Science*, 378 (3):211–222, 2007. ISSN 0304-3975. doi:

https://doi.org/10.1016/j.tcs.2007.02.031. URL https://www.sciencedirect.com/science/article/pii/S0304397507001223. Algorithms and Computation.

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

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

Google Cloud. BigQuery Documentation: Approximate Aggregate Functions, Accessed: 2024. URL https://cloud.google.com/bigquery/docs/reference/standard-sql/approximate_aggregate_functions. Google Cloud Documentation.

Gutenberg, M. P. and Wulff-Nilsen, C. 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, pp. 2542–2561, USA, 2020. Society for Industrial and Applied Mathematics.

Hardt, M. and Ullman, J. Preventing false discovery in interactive data analysis is hard. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 454–463. IEEE Computer Society, 2014. doi: 10.1109/FOCS.2014.55. URL https://doi.ieeecomputersociety.org/10.1109/FOCS.2014.55.

Hardt, M. and Woodruff, D. P. How robust are linear sketches to adaptive inputs? In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pp. 121–130, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320290. doi: 10.1145/2488608.2488624. URL https://doi.org/10.1145/2488608.2488624.

Hassidim, A., Kaplan, H., Mansour, Y., Matias, Y., and Stemmer, U. Adversarially robust streaming algorithms via differential privacy. In *Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Heule, S., Nunkesser, M., and Hall, A. HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *EDBT*, 2013.

Indyk, P. and Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc.*

30th Annual ACM Symposium on Theory of Computing, pp. 604–613. ACM, 1998.

Ioannidis, J. P. A. Why most published research findings are false. *PLoS Med*, (2):8, 2005.

Janson, S. Tail bounds for sums of geometric and exponential variables, 2017a. URL https://arxiv.org/abs/1709.08157.

Janson, S. Tail bounds for sums of geometric and exponential variables, 2017b. URL https://arxiv.org/abs/1709.08157.

Jayaram, R. and Woodruff, D. P. Towards optimal moment estimation in streaming and distributed models. *ACM Trans. Algorithms*, 19(3), jun 2023. ISSN 1549-6325. doi: 10.1145/3596494. URL https://doi.org/10.1145/3596494.

Kane, D. M., Nelson, J., and Woodruff, D. P. An optimal algorithm for the distinct elements problem. In *PODS*, 2010.

Knop, A. and Steinke, T. Counting distinct elements under person-level differential privacy. *CoRR*, abs/2308.12947, 2023. doi: 10.48550/ARXIV.2308.12947. URL https://doi.org/10.48550/arXiv.2308.12947.

Knuth, D. E. *The Art of Computer Programming, Vol 2, Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1998.

Kreuter, B., Wright, C. W., Skvortsov, E. S., Mirisola, R., and Wang, Y. Privacy-preserving secure cardinality and frequency estimation. Technical report, Google, LLC, 2020.

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

Mironov, I., Naor, M., and Segev, G. Sketching in adversarial environments. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 651–660, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470. doi: 10.1145/1374376.1374471. URL https://doi.org/10.1145/1374376.1374471.

Nelson, J., Nguyễn, H. L., and Woodruff, D. P. On deterministic sketching and streaming for sparse recovery and norm estimation. *Lin. Alg. Appl.*, 441:152–167, January 2014. Preliminary version in RANDOM 2012.

Pagh, R. and Stausholm, N. M. Efficient Differentially Private F Linear Sketching. In Yi, K. and Wei, Z. (eds.), *24th International Conference on Database Theory (ICDT 2021)*, volume 186 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 18:1–18:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-179-5. doi: 10.4230/LIPIcs.ICDT.2021.18. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICDT.2021.18.

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

Paterson, K. G. and Raynal, M. Hyperloglog: Exponentially bad in adversarial settings. Cryptology ePrint Archive, Paper 2021/1139, 2021. URL https://eprint.iacr.org/2021/1139. https://eprint.iacr.org/2021/1139.

Pettie, S. and Wang, D. Information theoretic limits of cardinality estimation: Fisher meets shannon. In Khuller, S. and Williams, V. V. (eds.), *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pp. 556–569. ACM, 2021. doi: 10.1145/3406325.3451032. URL https://doi.org/10.1145/3406325.3451032.

Reviriego, P. and Ting, D. Security of hyperloglog (HLL) cardinality estimation: Vulnerabilities and protection. *IEEE Commun. Lett.*, 24(5):976–980, 2020. doi: 10.1109/LCOMM.2020.2972895. URL https://doi.org/10.1109/LCOMM.2020.2972895.

Rosén, B. Asymptotic theory for order sampling. *J. Statistical Planning and Inference*, 62(2):135–158, 1997.

Shiloach, Y. and Even, S. An on-line edge-deletion problem. *J. ACM*, 28(1):1–4, jan 1981. ISSN 0004-5411. doi: 10.1145/322234.322235. URL https://doi.org/10.1145/322234.322235.

Smith, A., Song, S., and Guha Thakurta, A. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19561–19572. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e3019767b1b23f82883c9850356b71d6-Paper.pdf.

Steinke, T. and Ullman, J. 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*,

pp. 1588–1628, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Steinke15.html.

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

Wajc, D. *Rounding Dynamic Matchings against an Adaptive Adversary*. Association for Computing Machinery, New York, NY, USA, 2020. URL https://doi.org/10.1145/3357713.3384258.

Woodruff, D. P. and Zhou, S. 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)*, 2021.

A. Analysis of the Attack on the Standard Estimators

This section includes the proof of Theorem 4.1. We first consider k-mins sketches and $T(S) = \|S\|_1$. The modification needed for bottom-k sketches are in Section A.4.

A.1. Preliminaries

The following are order statistics properties useful for analysing MinHash sketches. Let $X_i \sim \mathsf{Exp}[1]$ for $i \in [n]$ be i.i.d. random variables. Then the distribution of the minimum value and of the differences between the $i + 1$ and the ith order statistics (smallest values) are independent random variables with distributions

$$\Delta_1 := \min_{i \in [n]} X_i \sim \mathsf{Exp}[n]$$

$$\Delta_i := \{X_i\}_{(i+1)} - \{X_i\}_{(i)} \sim \mathsf{Exp}[n - i] \qquad\qquad i > 1$$

Lemma A.1 (Chebyshev's Inequality).
$$Pr\left[|Z - E[Z]| \le c\sigma^2\right] \le 1/c^2 \ .$$

We set some notation: For a fixed ground set N and randomness ρ, for each hash function $i \in [k]$, let $m_j^i \in N$ be the key with the jth rank in the ith hashmap, that is, $h_i(m_j^i)$ is the jth smallest in $\{h_i(u)\}_{u \in N}$. Let

$$L := \log_2(rk) + 10$$
$$N_0^i := \{m_j^i\}_{j \le L}$$
$$N_0 := \bigcup_{i \in [k]} N_0^i$$
$$N' := N \setminus N_0$$

be a rank threshold L, for $i \in [k]$ the set N_0^i of keys with rank up to L in the ith hashmap, the set N_0 that is the union of these keys across hashmaps, and the set N' of the remaining keys in N.

We show that a choice of $n = O(kL/\alpha)$ ensures that certain properties that simplify our analysis hold. Our analysis applies to the event that these properties are satisfied:

Lemma A.2 (Good draws). *For $n = \Omega(\frac{1}{\alpha} k \log(rk))$, the following hold with probability at least 0.99:*

p1 *(property of ρ and N) The keys m_j^i for $i \in [k]$ and $j \le L$ are distinct.*

p2 *In a run of Algorithm 1, all r steps, for all $i \in [k]$, U includes a key from N_0^i.*

p3 *$n \ge 3kL/\alpha$*

Proof. p3 is immediate. For p1, note that if we set $n \ge \frac{2}{p} kL$ then the claim follows with probability $1 - p$ using the birthday paradox. For p2, the probability that a random U does not include one of the L smallest values of a particular hash function is 2^{-L}. The probability that this happens for any of the k hash functions in any of the r rounds is at most $rk2^{-L}$. Substituting $L = \log(rk) + 10$ we get the claim. $\qquad\square$

For fixed N and ρ, consider the random variable $Z := T(S_\rho(U))$ over sampling of U and the contributions Z_i of hash function $i \in [k]$ to Z.

$$Z_i := \min_{x \in U} h_i(x)$$
$$Z := \sum_{i \in [k]} Z_i$$

For a key $u \in N$, we consider the random variables $Z_i \mid u \in U$ and $Z \mid u \in U$ that are conditioned on $u \in U$. From property p1 in Lemma A.2, the Z_i are independent and are also independent when conditioned on $u \in U$.

A.2. Proof outline

We will need the following two Lemmas (the proofs are deferred to Section A.3). Intuitively we expect $\mathsf{E}_U[Z]$ to be lower when conditioned on $m_1^i \in U$. We bound this gap from below. For fixed ρ and N, let

$$G(u, v) := \mathsf{E}_{U|u \in U}[Z] - \mathsf{E}_{U|v \in U}[Z] .$$

Lemma A.3 (Expectations gap bound). *For each $i \in [k]$ and $\delta > 0$,*

$$\mathsf{Pr}_{\rho,N} \left[\min_{u \in N'} G(u, m_1^i) \geq \frac{\delta}{3n} \right] \geq 1 - \delta .$$

We bound from above the maximum over $u \in N$ of $\mathsf{Var}_{U|u \in U}[Z]$:

Lemma A.4 (Variance bound). *For $\delta > 0$ there is a constant c,*

$$\mathsf{Pr}_{\rho,N} \left[\max_{u \in N} \mathsf{Var}_{U|u \in U}[Z] \leq c \left(1 + \frac{1}{\sqrt{k\delta}} \right) \frac{k}{n^2} \right] \geq 1 - \delta .$$

We use the following to bound the number r of attack queries needed so that the sorted order by average score separates the minimum hash keys from the bulk of the keys in N':

Lemma A.5 (Separation). *Let $\alpha > 0$. Assume that*

- $\min_{u \in N'} G(u, m_1^i) \geq M > 0$

- $\max_{u \in N} \mathsf{Var}_{U|u \in U}[Z] \leq V^2$

- *During Algorithm 1, the keys $u \in N'$ and m_1^i are selected in U in at least $r' \geq \frac{2V^2}{M^2} \frac{1}{\alpha}$ rounds each.*

Then

$$\mathsf{Pr}[A[u] > A[m_1^i]] \leq \alpha$$

Proof. Consider the random variable

$$Y = A[u] - A[m_1^i] .$$

From our assumptions:

$$\mathsf{E}[Y] \geq M$$

$$\mathsf{Var}[Y] \leq \frac{2V^2}{r'}$$

We get

$$\begin{aligned}
\mathsf{Pr}[Y < 0] &\leq \mathsf{Pr}[|Y - \mathsf{E}[Y]| \geq \mathsf{E}[Y]] \\
&\leq \mathsf{Pr}[|Y - \mathsf{E}[Y]| \geq M] \\
&= \mathsf{Pr}[|Y - \mathsf{E}[Y]| \geq \frac{1}{\sqrt{\alpha}} \cdot \frac{V}{\sqrt{2r'}}] \leq \alpha
\end{aligned}$$

Using Chebyshev's inequality. $\qquad\square$

We are now ready to conclude the utility proof of Algorithm 1:

Lemma A.6 (Utility of Algorithm 1). *For $\alpha, \delta > 0$. Consider Algorithm 1 with $r = O(\frac{k}{\delta\alpha})$. Then for each $i \in [r]$, with probability $1 - \delta$, for any $u \in N'$ we have $\mathsf{Pr}[A[u] < A[m_1^i]] \leq \alpha$.*

Proof. Consider a key m_1^i and a key $u \notin N'$. We bound the probability that $A[u] < A[m_1^i]$.

From Lemma A.4, with probability $1 - 1/k$ we have $V^2 = O\left(\frac{k}{n^2}\right)$. From Lemma A.3, for each i, with probability $1 - \delta$, we have $M \geq \delta/(3n)$. If we choose $r = 4r'$ in Algorithm 1, then with high probability a key $x \in N$ is selected to U at least r' times. The claim follows from Lemma A.5 by setting $r' = O(\frac{k}{\delta\alpha})$. $\qquad\square$

We are now ready to conclude the proof of Theorem 4.1. Recall that a subset U of keys of size $M = 2\alpha n$ has $T(S(U))$ over random ρ concentrated around k/M with standard error $\sqrt{k}/M$). We now consider the prefix U of $M = 2\alpha n$ keys in the sorted order by average scores. This selection has $\mathsf{E}[T(S(U))] \leq (3\alpha) \cdot k/(2\alpha n)$. To see this, note that $(1 - \delta)$ of the MinHash values are the same as in the sketch of N. Therefore they have expected value $1/n$. The remaining δ fraction have expected value $1/(2\alpha n)$. Therefore $\mathsf{E}[T_\rho(S(U))] = k((1 - \delta)/n + \delta/(2\alpha n)) = (k/(2\alpha n))(\delta + (1 - \delta)2\alpha)$. Therefore T is a factor of $1/(\delta + 2\alpha)$ too small, which for small α and $\delta < \alpha$ is a large constant multiplicative error.

A.3. Proofs of Lemma A.3 and Lemma A.4

For fixed ρ (and N), for $i \in [k]$ and $j \in [N - 1]$, denote by

$$\Delta_1^i := \min_{x \in N} h_i(x) \equiv h_i(m_1^i)$$

$$\Delta_j^i := \{h_i(x) \mid x \in N\}_{(j+1)} - \{h_i(x) \mid x \in N\}_{(j)} \equiv h_i(m_{j+1}^i) - h_i(m_j^i) \qquad \text{for } j > 1$$

the gap between the j and $j + 1$ smallest values in $\{h_i(x)\}_{x \in N}$.

Lemma A.7 (Properties of Δ_j^i). *The random variables Δ_j^i $i \in [k]$, $j \in [L]$ over the sampling of N are independent with distributions $\Delta_j^i \sim \mathsf{Exp}[N - j]$. This holds also when conditioning on properties p1 and p2 of Lemma A.2.*

Proof. It follows from properties of the exponential distribution that over the sampling of ρ, $\Delta_j^i \sim \mathsf{Exp}[N - j]$ are independent random variables for i, j. Note that p1 and p2 are independent of the actual values of $h_i(m_j^i)$ and only depend on the rank in the order. $\square$

We can now express the distribution of the random variable Z_i in terms of Δ_j^i: We have

For $j \geq 1$, the probability that $Z_i = \sum_{\ell=1}^j \Delta_\ell^i$ is $2^{-j}/(1 - 2^{-L})$. This corresponds to the event that U does not include the keys m_ℓ^i for $\ell < j$ and includes the key m_j^i. The normalizing factor $(1 - 2^{-L})$ arises from property p2 in Lemma A.2. In the sequel we omit this normalizing factor for brevity.

$$Z_i = \begin{cases} \Delta_1^i & \text{with probability } 2^{-1}/(1 - 2^{-L}) \\ \Delta_1^i + \Delta_2^i & \text{with probability } 2^{-2}/(1 - 2^{-L}) \\ \Delta_1^i + \Delta_2^i + \Delta_3^i & \text{with probability } 2^{-3}/(1 - 2^{-L}) \\ \quad \vdots \\ \Delta_1^i + \cdots + \Delta_j^i & \text{with probability } 2^{-j}/(1 - 2^{-L}) \\ \quad \vdots \end{cases}$$

We now consider Z_i conditioned on the event $m_j^i \in U$. Clearly for $j = 1$ (conditioning on the event $m_1^i \in U$) we have $Z_i \equiv \Delta_1^i$. For $j \geq 1$, we have $Z_i = \sum_{\ell=1}^h \Delta_\ell^i$ with probability 2^{-h} for $h < j$ and $Z_i = \sum_{\ell=1}^j \Delta_\ell^i$ with probability 2^{-j+1}.

We bound the expected value of Z_i conditioned on presence of a key $u \in U$.

Lemma A.8. *(i) (Anti-concentration) For $u \neq m_1^i$, the random variable over sampling of ρ, N*

$$G = \max_{u \in N \setminus \{m_1^i\}} \mathsf{E}_{U|u \in U}[Z_i] - \mathsf{E}_{U|m_1^i \in U}[Z_i]$$

is such that for all $c > 0$, $Pr_{\rho,N}\left[G \geq \frac{c}{n-1}\right] \geq e^{-2c}$.

(ii) (Concentration) For $u \in N$, the random variable

$$G = \mathsf{E}_{U|u \in U}[Z_i] - \mathsf{E}_{U|m_j^i \in U}[Z_i]$$

is such that for $c \geq 1$

$$Pr_{\rho,N}\left[G \geq c \cdot \frac{3}{n2^j}\right] \leq ce^{-c} .$$

Proof. Per Lemma A.2 we are assuming the event (that happens with probability $1 - \delta$ that U includes a key $u \in \{m_j^i\}_{j \in [L]}$ for all $i \in [k]$. Therefore, for a key $u \in N'$ it holds that

$$\mathsf{E}_{U|u \in U}[Z_i] = \mathsf{E}_U[Z_i] \,.$$

Recall that $Z_i \mid m_1^i \in U = \Delta_1^i$. Otherwise (when not conditioned on $m_1^i \in U$ or conditioned on presence of $u \neq m_1^i$) $Z_i = \Delta_1^i$ with probability $1/2$ (when the random U includes m_1^i) and $Z_i \geq \Delta_1^i + \Delta_2^i$ otherwise (when U does not include m_1^i). Thus,

$$\mathsf{E}_U[Z_i] - \mathsf{E}_{U|m_1^i \in U}[Z_i] \geq \Delta_2^i/2$$
$$\mathsf{E}_{U|m_j^i \in U}[Z_i] - \mathsf{E}_{U|m_1^i \in U}[Z_i] \geq \Delta_2^i/2 \text{ for } 1 < j \leq L$$

Therefore for $u \neq m_1^i$,

$$G := \mathsf{E}_{U|u \in U}[Z_i] - \mathsf{E}_{U|m_1^i \in U}[Z_i] \geq \Delta_2^i/2 \,.$$

Since $\Delta_2^i \sim \mathsf{Exp}[N-1]$, we have for all $t > 0$, $\mathsf{Pr}_\rho[G \geq t] \geq e^{-2t/(n-1)}$. This establishes claim (i).

For claim (ii), note that

$$\mathsf{E}_{U|u \in U}[Z_i] \leq \mathsf{E}[Z_i] \leq \sum_{\ell=1}^{L} \frac{1}{2^{\ell-1}} \Delta_\ell^i$$

$$\mathsf{E}_{U|m_j^i \in U}[Z_i] \geq \sum_{\ell=1}^{j-1} \frac{1}{2^{j-1}} \Delta_j^i$$

Therefore

$$G := \mathsf{E}_{U|u \in U}[Z_i] - \mathsf{E}_{U|m_j^i \in U}[Z_i] \leq \sum_{\ell=j}^{L} \frac{1}{2^{\ell-1}} \Delta_\ell^i \,.$$

This is a sum of independent exponential random variables and recall that we assumed $L \leq n/3$. Therefore, this is stochastically smaller that the respective geometrically decreasing weighted sum of independent $\mathsf{Exp}[3/n]$ random variables. It follows that

$$\mathsf{E}[G] \leq \frac{3}{n} \sum_{\ell=j}^{L} \frac{1}{2^{\ell-1}} = \frac{3}{n2^j}$$

We apply an upper bound on the tail (Janson, 2017a) that shows that this concentrates almost as well as a single exponential random variable: $\mathsf{Pr}[G \geq t\mu] \leq te^{-t}$ and obtain claim (ii)

$$\mathsf{Pr}\left[G \geq t\frac{3}{n2^j}\right] \leq te^{-t} \,.$$

$\square$

Lemma A.3 is a corollary of the first claim of Lemma A.8.

We now express a bound on the variance of Z_i, also when conditioned on the presence of any key $u \in U$, for fixed ρ, N.

Lemma A.9. *For fixed N, ρ, and any $u \in N$*

$$\textit{Var}_{U|u \in U}[Z_i], \textit{Var}[Z_i] = \Theta(\sum_j (3/2)^{-j}(\Delta_j^i)^2) \,.$$

Proof.

$$\mathsf{Var}_{U|u\in U}[Z_i], \mathsf{Var}[Z_i] \le \mathsf{E}[Z_i^2] \le \sum_{j\ge 1} 2^{-j} \left(\sum_{\ell=1}^{j} \Delta_j^i \right)^2 \le \sum_{j\ge 1} 2^{-j} j \sum_{\ell=1}^{j} (\Delta_j^i)^2$$

$$= \sum_{j\ge 1} \left(\sum_{\ell\ge j} \frac{\ell}{2^\ell} \right) (\Delta_j^i)^2 = \Theta(\sum_j (3/2)^{-j} (\Delta_j^i)^2)$$

$\square$

Proof of Lemma A.4. Since the Z_i, also when conditioned on $u \in U$, are independent (modulu our simplifying assumption in Lemma A.2), it follows from Lemma A.9 that

$$\mathsf{Var}_{U|u\in U}[Z] = \Theta \left(\sum_{i\in[k]} \sum_{j\ge 1} (3/2)^{-j} (\Delta_j^i)^2 \right) .$$

Therefore,

$$\max_{u\in N} \mathsf{Var}_{U|u\in U}[Z] = \Theta \left(\sum_{i\in[k]} \sum_{j\ge 1} (3/2)^{-j} (\Delta_j^i)^2 \right) .$$

The right hand side

$$Y := \max_{u\in N} \mathsf{Var}_{U|u\in U}[Z]$$

is a random variable over ρ, N that is a a a weighted sum of the squares of independent exponential random variables Δ_j^i. The PDF of a squared exponential random variable $\mathsf{Exp}[w]^2$ is $\frac{w}{2\sqrt{t}} e^{-w\sqrt{t}}$. The mean is $\frac{2}{w^2}$ and the variance is at most $\mathsf{E}[t^2] = 24/w^4$. Applying this, we obtain that $\mathsf{E}[Y] = \Theta(k/n^2)$ and $\mathsf{Var}[Y] = O(k/n^4)$.

From Chebyshev's inequality, $\Pr[Y - \mathsf{E}[Y] \ge c \cdot \sqrt{k}/n^2] = O(1/c^2)$ and we obtain for any $\delta > 0$ and a fixed constant c

$$\Pr_{\rho,N} \left[\max_{u\in N} \mathsf{Var}_{U|u\in U}[Z] \ge c \left(1 + \frac{1}{\sqrt{k\delta}} \right) \frac{k}{n^2} \right] \le \delta .$$

$\square$

A.4. Attack on the Bottom-k standard estimator

The argument is similar to that of k-mins sketches. We highlight the differences. Recall that a bottom-k sketch uses a single hash function h with the sketch storing the k smallest values $S(U) := \{h(x) \mid x \in U\}_{(1:k)}$. We use the kth order statistics (kth smallest value) $T(S) := \{h(x) \mid x \in U\}_{(k)}$.

For fixed ρ and N, let $m_j \in N$ ($j \in [n]$) be the key with the jth smallest hashmap $h(m_j) = \{h(x) \mid x \in U\}_{(k)}$. Define $\Delta_1 := h(m_1)$ and for $j > 1$, $\Delta_j := h(m_j) - h(m_{j-1})$.

Let R be the random variable that is the rank in N of the key with the kth smallest hashmap in U. The distribution of R is the sum of k i.i.d. Geometric random variables $\mathsf{Geom}[q = 1/2]$. We have $\mathsf{E}[R] = k/q$ and the concentration bound (Janson, 2017a) that for any $c \ge 1$, $\Pr[R > c\mathsf{E}[R]] \le ce^{-c}$.

Let $L = 10 + 2\log r$, $N_0 = \{m_j\}_{j\le kL/q}$ be the keys with the $L(k/q)$ smallest hashmaps. Let $N' = N \setminus N_0$ be the remaining keys. We show that the attack separates with probability α a key with one of the bottom-k ranks and a key in N'.

Assume $n > 3|N_0|$. Assume that we declare failure when a set U selected by the algorithm does not contain k keys from N_0. The probability of such selection is at most $Le^{-L} < 0.01/r$ and at most 0.01 in all r steps.

For fixed ρ and N, consider the random variable $Z := T(S(U))$. The following parallels Lemma A.3 and Lemma A.9:

Lemma A.10. *Fixing ρ, N,*

(i) let

$$G(u, v) := \mathsf{E}_{U|u\in U}[Z] - \mathsf{E}_{U|v\in U}[Z]$$

For each $i \leq [k]$ and $\delta > 0$,

$$\mathsf{Pr}\left[\min_{u\in N'} G(u, m_i) \geq \frac{\delta}{3n}\right] \geq 1 - \delta .$$

(ii) for $\delta > 0$ and some constant c,

$$\mathsf{Pr}_{\rho, N}\left[\max_{u\in N} \mathsf{Var}_{U|u\in U}[Z] \leq c\left(1 + \frac{1}{\sqrt{k\delta}}\right)\frac{k}{n^2}\right] \geq 1 - \delta .$$

Proof. (i) Note that $Z = m_R$. When $i \leq [k]$, $Z = m_{R+1}$. The gap is a weighted average of Δ_j for $j \in [R, |N_0|]$. These are independent $\mathsf{Exp}[n - i]$ random variables with $i \leq n/3$. The expected value is $\Theta(1/n)$ and the tail bounds are at least as tight as for a single $\mathsf{Exp}[n/3]$ random variable.

(ii) We use the concentration bound on R to express the variance for fixed ρ, N as a weighted sum with total weight $\Theta(k)$ and each of weight $O(1)$ of independent squared exponential random variables. The argument is as in the proof of Lemma A.9. $\qquad\square$

Using the same analysis, a subset $U \subset N$ of size αn has $T(S(U))$ that in expectation has the k/α smallest rank in N with standard error $\sqrt{k}/\alpha$ and normalized standard error $1/\sqrt{k}$. The subset U selected as a prefix of the order generating in the attack includes the $(1 - \delta)k$ of the bottom-k in N and δk of the bottom in U. This means that in expectation $T(S(U))$ has the $k\delta/\alpha$ rank in N. That is, error that is $(1/\delta)$ factor off.

B. Analysis of Attack on General Query Response Algorithms

We include details for Sections 7 and 8.

B.1. Rank-domain representation of sketches

We use the *rank domain* representation $S_\rho^R(U)$ of the input sketch $S_\rho(U)$. This representation is defined for subsets of a fixed ground set N. Instead of hash values, it includes the ranks in N of the keys that are represented in the sketch $S_\rho(U)$ with respect to the relevant hashmaps.

Definition B.1. (Rank domain representation) For a fixed ground set N, and a subset $U \subset N$, the *rank domain* representation $S_\rho^R(U)$ of a respective MinHash sketch has the form $(Y_1, \dots, Y_k)$, where $Y_i \in \mathbb{N}$.

- k-mins sketch: For $i \in [k]$ and $j \geq 1$, let m_j^i be the key $x \in N$ with the jth smallest $h_i(x)$. For $i \in [k]$, let $Y_i := \arg\min_j m_j^i \in U$. That is, Y_i is the smallest j such that $m_j^i \in U$.

- k-partition sketch: For $i \in [k]$ and $j \geq 1$, let m_j^i be the key $x \in N$ that is in part i with the jth smallest $h(x)$. For $i \in [k]$, let $Y_i := \arg\min_j m_j^i \in U$ be the smallest rank in the ith part.

- Bottom-k sketch: For $j \geq 1$, let m_j be the key $x \in N$ with the jth smallest $h(x)$ value. Let the bottom-k keys in U be $m_{i_1}, m_{i_2}, \dots, m_{i_k}$ where $i_1 < i_2 < \cdots < i_k$. We then define $Y_1 := i_i$ and $Y_j := i_j - i_{j-1}$ for $1 < j \leq k$.

Note that when the ground set N is available to the query response algorithm the rank domain and MinHash representations are equivalent (we can compute one from the other).

The following properties of the rank domain facilitate a simpler analysis: (i) It only depends on the order induced by the hashmaps and not on actual values and thus allows us to factor out dependence on ρ, (ii) It subsumes the information on q (and $|U|$) available from $S_\rho(U)$ and (iii) It has a unified form and facilitates a unified treatment across the MinHash sketch types.

The subsets $U \sim \mathcal{D}_0$ generated by our attack algorithm selects a rate q and then sample U by including each $x \in N$ independently with probability q. We consider the distribution, which we denote by $S^R[q]$ of the rank domain sketch under this sampling of U with rate q. We show that for a sufficiently large $|N| = n$, the rank domain representation is as follows:

Lemma B.2 (distribution of rank-domain sketches). *For $\delta > 0$, $q \in (0,1)$, and an upper bound r on the attack size, let $L = \log_2(rk/\delta)/q + 10$, and assume $n > 3kL/\delta$. Then for all the three MinHash sketch types, the distribution $S^R[q]$ is within total variation distance δ from $(Y_1, \ldots, Y_k)$ that are k independent geometric random variables with parameter q: $Y_i \sim \textsf{Geom}[q]$.*

Proof. As in Lemma A.2. Applying the birthday paradox with $n > 3kL/\delta$, with probability at least $1 - \delta$: For k-mins sketches, the keys m_j^i for $i \in [k]$ and $j \in [L]$ are distinct. For k partition sketches, there are at least L keys assigned to each part so the keys m_j^i for $i \in [k]$ and $j \in [L]$ are well specified.

A sketch from $S^R[q]$ can be equivalently sampled using the following process:

- k-mins and k-partition sketch: For each $i \in [k]$, process keys m_j^i by increasing $j \geq 1$ until $\textsf{Bern}[q]$ and then set $Y_i = j$.

- Bottom-k sketch: Process keys m_j in increasing $j \geq 1$ until we get $\textsf{Bern}[q]$ k times.

We next establish that with our choice of n, with probability at least $1 - \delta$, in all of r sampling of U, the sketch $S^R(U)$ is determined by the Lk smallest rank keys. Therefore there are sufficiently many keys for the sketch to agree with the sampling k i.i.d. $\textsf{Geom}[k]$ random variables.

For k-mins and k-partition sketches, the probability that for a single hashmap $i \in [k]$ none of the L smallest rank is included is at most $(1-q)^L$. Taking a union bounds over k maps and r sampling and using that $\log(1/(1-q)) \approx q$ gives the claim. With bottom-k sketches the requirement is that in all r selections, the kth smallest rank is $O(kL)$. $\qquad\square$

Remark B.3. Estimating q from a sketch from $S^R[q]$ is a standard parameter estimation problem. A sufficient statistic T for estimating q is $T(S^R) := \sum_{i=1}^k Y_i$. Note the following properties:

- The distribution $S^R[q]$ does not provide additional information on the cardinality $|U|$ beyond an estimate of q.

- The distribution of $S^R[q]$ conditioned on $T(S) = \tau$ is the same for all q (this follows from the definition of sufficient statistic).

- The statistic T has expected value k/q, variance $k(1-q)/q^2$, and single-exponential concentration (Janson, 2017b).

B.1.1. CONTINUOUS RANK DOMAIN REPRESENTATION

We now cast the distribution of $S^R[q]$ using a continuous representation S^C. This is simply a tool we use in the analysis.

We can sample a sketch from $S^R[q]$ as follows

- Set the rate $q' := -\ln(1-q)$

- Sample a sketch $S^C(U) = (Y_1', \ldots, Y_k')$ where $Y_i' \sim \textsf{Exp}[q']$ are i.i.d

- Compute $S^R(U)$ from $S^C(U)$ using $Y_i \leftarrow 1 + \lfloor Y_i' \rfloor + 1$ for $i \in [k]$.

The correctness of this transformation is from the relation between a geometric $\textsf{Geom}[q]$ and exponential $\textsf{Exp}[q']$ distributions:
$$\Pr[Y_i = t] = \Pr[t - 1 \leq Y_i' < t] = e^{-q'(t-1)} - e^{-q't} = (1 - e^{-q'}) \cdot e^{-q't} = q \cdot (1-q)^t .$$

Note that we can always recover S^R from S^C but we need to know q in order to compute S^C from S^R:
$$Y_i' \sim \textsf{Exp}[-\ln(1-q)] \mid Y_i' \in [Y_i - 1, Y_i) .$$

Therefore being provided with the continuous representation only makes the query response algorithm more informed and potentially more powerful. Also note that $|q - q'| < q^2/2$.

A sufficient statistic for estimating q' from $S^C[q']$ is $T' := \sum_{i=1}^k Y_i'$. In the sequel we will work with S^C and omit the prime from q and T.

Now note that the distribution of $T := \|S^C[q]\|_1$ for a given k and q is the sum of k i.i.d. $\mathsf{Exp}[q]$ random variables. This is the Erlang distribution that has density function for $x \in [0, \infty]$:

$$f_T(k, q; x) = \frac{q^k}{(k-1)!} x^{k-1} e^{-qx} \tag{4}$$

The distribution has mean $\mathsf{E}[T] = k/q$, variance $\mathsf{Var}[T] = k/q^2$ and exponential tail bounds (Janson, 2017a):

$$\text{For } c > 1 \colon \mathsf{Pr}[T \geq c \cdot k/q] \leq \frac{1}{c} e^{-k(c-1-\ln c)} \tag{5}$$

$$\text{For } c < 1 \colon \mathsf{Pr}[T \leq c \cdot k/q] \leq e^{-k(c-1-\ln c)} \tag{6}$$

Consider the random variable

$$Z = (T - \mathsf{E}[T])/\sqrt{\mathsf{Var}[T]} \tag{7}$$

that is the number of standard deviations of T from its mean. We have $T = \frac{k}{q} + Z \cdot \frac{\sqrt{k}}{q}$ and $Z = \frac{qT}{\sqrt{k}} - \sqrt{k}$.

The domain of Z is $[-\sqrt{k}, \infty)$ and the density function of Z is

$$\begin{aligned} f_Z(k; z) &= \frac{\sqrt{k}}{q} \frac{q^k}{(k-1)!} \left(\frac{k}{q} + z\frac{\sqrt{k}}{q}\right)^{k-1} e^{-q(\frac{k}{q} + z\frac{\sqrt{k}}{q})} \\ &= \frac{\sqrt{k}}{(k-1)!} (k + z\sqrt{k})^{k-1} e^{-(k+z\sqrt{k})} \end{aligned} \tag{8}$$

This density satisfies

$$\int_{-\sqrt{k}}^{\infty} f_Z(k; x) x \, dx = 0 \tag{9}$$

$$\int_{-\sqrt{k}}^{\infty} f_Z(k; x) x^2 \, dx = 1 \tag{10}$$

$$\int_{-\sqrt{k}/4}^{\sqrt{k}/4} f_Z(k; x) x^2 \, dx = \Theta(1) \tag{11}$$

$$\text{for } c \in (0, 1], \int_{-c}^{0} f_Z(k; x) dx, \int_{0}^{c} f_Z(k; x) dx = \Omega(c) \tag{12}$$

$$\text{for } c \geq 0, \mathsf{Pr}[T \geq c \cdot \sqrt{k}] \leq \frac{1}{c+1} e^{-k \cdot (c - \ln(c+1))} \tag{13}$$

$$\text{for } c \in (0, 1), \mathsf{Pr}[T \leq -c \cdot \sqrt{k}] \leq e^{-k \cdot (1 - c - \ln(1-c))} \tag{14}$$

Note that T is available to the query response algorithm but q, and thus the value of Z are not available.

B.2. Correct maps

A map $S \mapsto \pi(S) \in [0, 1]$ maps sketches to the probability of returning 1. We require that the maps selected by QR are correct as in Definition 6.3 with $\delta = O(1/\sqrt{k})$.

For a map π and τ we denote by $\overline{\pi}(\tau)$ the mean value of $\pi(S)$ over sketches with statistic value $T(S) = \tau$. This is well defined since for the query distribution in our attacks $\mathcal{D}_0 \mid q = q^*$, even when conditioned on a fixed rate q^*, the distribution of the sketch conditioned on $T(S) = \tau$ does not depend on q^* (See Remark B.3).

We now specify conditions on the map $\overline{\pi}(\tau)$ that must be satisfied by a correct π. A correct map may return an incorrect output, when conditioned on cardinality, with probability δ. This means that there are correct maps with large error on certain τ (since each cardinality has a distribution on τ). We therefore can not make a sharp claim on $\overline{\pi}(\tau)$ that must hold for any τ in an applicable range. Instead, we make an average claim: For any interval of τ values that is wide enough to include $\Omega(1)$ of the values for some cardinality value $c \notin [A, 2A]$, the average error of the mapping must be $O(\delta)$.

Claim B.4. For any $\xi > 0$, there is $c_0 > 0$ such that for any correct map π for A and $\delta \leq c_0/\sqrt{k}$ and $\tau_b > (1 + 0.1/\sqrt{k})\tau_a$ it holds that

$$
\begin{cases}
\frac{1}{\tau_b - \tau_a} \int_{\tau_a}^{\tau_b} \overline{\pi}(x)dx < \xi & \text{if } \tau_a > \frac{kn}{A}(1 - 1/\sqrt{k}) \\
\frac{1}{\tau_b - \tau_a} \int_{\tau(1-a/\sqrt{k})}^{\tau} \overline{\pi}(x)dx > 1 - \xi & \text{if } \tau_b < \frac{kn}{2A}(1 + 1/\sqrt{k})
\end{cases}
\tag{15}
$$

Proof. For a cardinality value c, the distribution of the statistic T conditioned on a cardinality value c is $f_T(k, c/n; x)$ (4). With cardinality value c, it holds that $\Pr[T < kn/c] \geq 1/e$ and $\Pr[T > kn/c] \geq 1/e$. Moreover, the density in the interval $\frac{kn}{c}[1 - 0.1/\sqrt{k}, 1 + 0.1/\sqrt{k}]$ is $\Theta(1)$. It follows from the correctness requirement for cardinality value $c = k/\tau$ that there exists $c_1 > 0$ such that:

$$
\begin{cases}
\int_{\tau}^{\tau(1+0.1/\sqrt{k})} \overline{\pi}(x)dx < c_1\delta & \text{if } \tau > \frac{kn}{A}(1 - 1/\sqrt{k}) \\
\int_{\tau(1-0.1/\sqrt{k})}^{\tau} \overline{\pi}(x)dx > 1 - c_1\delta & \text{if } \tau < \frac{kn}{2A}(1 + 1/\sqrt{k})
\end{cases}
\tag{16}
$$

Therefore for $\tau_a > \frac{kn}{A}(1 - 1/\sqrt{k})$

$$
\int_{\tau_a}^{\tau_b} \overline{\pi}(x)dx \leq 10\sqrt{k}(\tau_b - \tau_a)c_1\delta
$$

and for $\tau_b < \frac{kn}{2A}(1 + 1/\sqrt{k})$

$$
\int_{\tau_a}^{\tau_b} \overline{\pi}(x)dx \geq (\tau_b - \tau_a) \cdot (1 - 10\sqrt{k}c_1\delta) .
$$

Choosing $c_0 \leq 10c_1$ establishes the claim. $\qquad\square$

For fixed q, the cardinality $|U|$ of the selected U has distribution $\mathsf{Binom}(q, n)$. The n chosen for the attack is large enough so that for all our r queries $||U| - qn|/(qn) \ll 1/\sqrt{k}$. That is, the variation in $|U|$ for fixed q is small compared with the error of the sketch and $|U| \approx qn$.

B.3. Relating Z and sampling probability of low rank keys

The sketch is determined by k keys that are lowest rank in U. We can view the sampling of U to the point that the sketch is determined in terms of a process, as in the proof of Lemma B.2, that examines keys from the ground set N in a certain order until the k that determine the sketch are selected. The process selects each examined key with probability q. For a bottom-k sketch, keys are examined in order of increasing rank until k are selected. With k-mins and k-partition, keys in each part (or hash map) are examined sequentially by increasing rank until there is selection for the part. For all sketch types, the statistic value T corresponds to the number of keys from N that are examined until k are selected. This applies also with the continuous representation S^C.

We denote by N_0 the set of keys that are examined with probability at least $\delta_c \leq 1/(rk)$ when the rate is at least q_a. We refer to these keys as *low rank* keys. It holds that $|N_0| \leq k\ln(1/\delta_c)/q_a$. For $\delta_c = 1/O(rk)$ we have $|N_0| = O(k\log(rk))$. The remaining keys $N' := N \setminus N_0$ are unlikely to impact the sketch content and we refer to them as *transparent*.

With rate q, the probability that a certain key is included in U is q. We now consider a rate q and the inclusion probability *conditioned on the normalized deviation from the mean Z*. Transparent keys have inclusion probability q. The low rank keys N_0 however have *average* inclusion probability that depends on Z. Qualitatively, we expect that when $Z < 0$, the inclusion probability is larger than q and this increases with magnitude $|Z|$. When $Z > 0$, the inclusion is lower and decreases with the magnitude. This is quantified in the following claim:

Claim B.5. Fix a rate q and Δ. Consider the distribution of U conditioned on $Z = \Delta$. The average probability over N_0 keys to be in U is $q - \Delta\frac{\sqrt{k}}{|N_0|}$.

Proof. Equivalently, consider the distribution conditioned on $T = \frac{1}{q}(k + \Delta\sqrt{k})$. The sampling process selects k keys after examining T keys. The average effective sampling rate for the examined keys is $q_e = k/T$. There are T keys out of N_0 that are processed with effective rate q_e and the remaining keys in N_0 have effective rate q.

Averaging the effective rate over the $T = \frac{1}{q}(k + Z\sqrt{k})$ processed keys and the remaining N_0 keys we obtain

$$\frac{T \cdot q_e + (|N_0| - T) \cdot q}{|N_0|} = \frac{T \cdot \frac{k}{T} + (|N_0| - T) \cdot q}{|N_0|} = \frac{k + (|N_0| - (\frac{k+\Delta\sqrt{k}}{q})) \cdot q}{|N_0|}$$

$$= q - \Delta\frac{\sqrt{k}}{|N_0|}$$

$\square$

B.4. Scoring probability gap

For a map π, let $p'(\pi)$ be the score probability, over the distribution of q and Z, of a key in N'. Let $p_0(\pi)$ be the *average* over N_0 of the score probability of keys in N_0.

Let $f_\lambda(x)$ be the density function of the selected rate, described by Algorithm 5. Note that the selected rate is in the interval

---

Algorithm 5: `Sample rate`

Input: A, n
$\omega \leftarrow \frac{n}{2A}$
$\omega_a \leftarrow \frac{1}{2}\omega; \omega_b \leftarrow \frac{5}{2}\omega$ // range of inverse rates
$D \sim U[0, \omega/4]$
$\omega_a^* \leftarrow \omega_a + D; \omega_b^* \leftarrow \omega_a^* + \frac{7}{4}\omega$ // range of sampled inverse rate
return $q \sim \frac{1}{U[\omega_a^*, \omega_b^*]}$

$q \in [\frac{1}{\omega_b}, \frac{1}{\omega_a}] = \frac{1}{\omega} \cdot [\frac{2}{5}, 2] = \frac{A}{n} \cdot [\frac{4}{5}, 4]$.

For each transparent key, the score probability is:

$$p'(\pi) = \int_{q_a}^{q_b} \int_{-\sqrt{k}}^{\infty} \overline{\pi}(\frac{k}{q}(1 + z/\sqrt{k})) f_Z(k; z) \cdot dz \cdot q \cdot f_\lambda(q) dq \tag{17}$$

On average over the low-rank keys N_0 using Claim B.5 it is

$$p_0(\pi) = \int_{q_a}^{q_b} \int_{-\sqrt{k}}^{\infty} \overline{\pi}(\frac{k}{q}(1 + z/\sqrt{k})) f_Z(k; z) \cdot \left(q - z\frac{\sqrt{k}}{|N_0|}\right) \cdot dz \cdot f_\lambda(q) dq \tag{18}$$

For a correct map π (as in Definition 6.3), we express the gap between $p'(\pi)$ and $p_0(\pi)$. Note that we bound the gap without assuming much on the actual values, as they can highly vary for different correct π.

Lemma B.6 (Score probability gap). *Consider a step of the algorithm and a correct map π (see Remark 6.4). Then*

$$p_0(\pi) - p'(\pi) = \Omega\left(\frac{1}{|N_0|}\right) = \Omega\left(\frac{1}{k\log(kr)}\right)$$

In the remaining part of this section we present the proof of Lemma B.6. We will need the following claim, that relates Δ and scoring probability.

Claim B.7. For $|\Delta| < \sqrt{k}/4$

$$\int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}(1 + \frac{\Delta}{\sqrt{k}})) \cdot f_\lambda(q) dq = \int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}) \cdot f_\lambda(q) dq + \Theta(\frac{\Delta}{\sqrt{k}})$$

Proof. Using the distribution specified in Algorithm 5, for any $g()$:

$$\int_{q_a}^{q_b} g(q) f_\lambda(q) dq = -\frac{4}{\omega} \int_0^{\omega/4} dD \frac{4}{7\omega} \int_{\omega_a + D}^{\omega_a + D + \frac{7}{4}\omega} g(1/x) dx \tag{19}$$

We use $w_a^* = \omega_a + D$ and $w_b^* = \omega_a + D + \frac{7}{4}\omega$ and get

$$\int_{\omega_a^*}^{\omega_b^*} \overline{\pi}(kx(1 + \frac{\Delta}{\sqrt{k}})) \cdot dx = \frac{1}{1 + \frac{\Delta}{\sqrt{k}}} \int_{\omega_a^* \cdot (1+\Delta/\sqrt{k})}^{\omega_b^* \cdot (1+\Delta/\sqrt{k})} \overline{\pi}(ky)dy \quad \text{(change variable x to $y = x(1 + \Delta/\sqrt{k})$)}$$

$$= \frac{1}{1 + \frac{\Delta}{\sqrt{k}}} \left(\int_{\omega_a^*}^{\omega_b^*} \overline{\pi}(kx)dx - \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx + \int_{\omega_b^*}^{\omega_b^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \right)$$

Therefore,[4]

$$\int_{\omega_a^*}^{\omega_b^*} \left(\overline{\pi}(kx(1 + \frac{\Delta}{\sqrt{k}})) - \overline{\pi}(kx) \right) \cdot dx = \tag{20}$$

$$= -\Theta(\frac{\Delta}{\sqrt{k}}) \cdot \int_{\omega_a^*}^{\omega_b^*} \overline{\pi}(kx)dx - \Theta(1) \cdot \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx + \Theta(1) \cdot \int_{\omega_b^*}^{\omega_b^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx$$

$$= -O(\frac{\Delta}{\sqrt{k}}\omega) - \Theta(1) \cdot \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx + \Theta(1) \cdot \int_{\omega_b^*}^{\omega_b^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \ .$$

The last equality follows using

$$\int_{\omega_a^*}^{\omega_b^*} \overline{\pi}(kx)dx \in [0, \omega_b^* - \omega_a^*] = [0, \frac{7}{4}\omega]$$

$$\int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}(1 + \frac{\Delta}{\sqrt{k}})) \cdot f_\lambda(q)dq - \int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}) \cdot f_\lambda(q)dq = \tag{21}$$

$$= \int_{q_a}^{q_b} \left(\overline{\pi}(\frac{k}{q}(1 + \frac{\Delta}{\sqrt{k}})) - \overline{\pi}(\frac{k}{q}) \right) \cdot f_\lambda(q)dq =$$

$$= \frac{4}{\omega} \int_0^{\omega/4} dD \frac{4}{7\omega} \int_{\omega_a+D}^{\omega_a+D+\frac{7}{4}\omega} \left(\overline{\pi}(kx(1 + \frac{\Delta}{\sqrt{k}})) - \overline{\pi}(kx) \right) dx \quad \text{(Using (19))}$$

$$= -O(\frac{\Delta}{\sqrt{k}}) - \Theta(\frac{1}{\omega^2}) \cdot \int_0^{\omega/4} dD \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx + \Theta(\frac{1}{\omega^2}) \cdot \int_0^{\omega/4} dD \cdot \int_{\omega_b^*}^{\omega_b^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \quad \text{(Apply (20))}$$

We now separately bound terms[5]:

$$\int_0^{\omega/4} dD \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \geq \int_0^{\omega/4} dD \int_{\omega_a+D}^{\omega_a+D+\frac{\Delta}{\sqrt{k}}\omega_a} \overline{\pi}(kx)dx$$

$$= \int_0^{\frac{\Delta}{\sqrt{k}}(\omega/4)} dW \int_{\omega_a+W}^{\omega_a+\omega/4+W} \overline{\pi}(kx)dx$$

$$\geq \frac{\Delta}{\sqrt{k}}\frac{\omega}{4} \cdot \frac{\omega}{4}(1 - \xi) = \Theta(\omega^2\frac{\Delta}{\sqrt{k}}) \quad \text{(Using Claim B.4)} \tag{22}$$

$$\int_0^{\omega/4} dD \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \leq \int_0^{\omega/4} dD \int_{\omega_a+D}^{\omega_a+D+\frac{\Delta}{\sqrt{k}}(\omega_a+\omega/4)} \overline{\pi}(kx)dx$$

$$= \int_0^{\frac{\Delta}{\sqrt{k}}(\omega_a+\omega/4)} dW \int_{\omega_a+W}^{\omega_a+\omega/4+W} \overline{\pi}(kx)dx \qquad = O(\omega^2\frac{\Delta}{\sqrt{k}}) \tag{23}$$

---

[4]Note that Δ can be negative. In which case in order to streamline expressions we interpret the asymptotic notation $O(c\Delta)$ as $-O(c|\Delta|)$.

[5]Argument for negative Δ is similar

Combining (22) and (23) we obtain that

$$\int_0^{\omega/4} dD \int_{\omega_a^*}^{\omega_a^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx = \Theta(\omega^2 \frac{\Delta}{\sqrt{k}}) \tag{24}$$

We next bound the last term:

$$\begin{aligned}
\int_0^{\omega/4} dD \int_{\omega_b^*}^{\omega_b^*(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx &= \int_0^{\omega/4} dD \int_{\frac{9}{4}\omega+D}^{(\frac{9}{4}\omega+D)\cdot(1+\Delta/\sqrt{k})} \overline{\pi}(kx)dx \\
&\leq \int_0^{\omega/4} dD \int_{\frac{9}{4}\omega+D}^{(\frac{9}{4}\omega+D)+\frac{5}{2}\omega\cdot\frac{\Delta}{\sqrt{k}}} \overline{\pi}(kx)dx \\
&= \int_0^{\frac{5}{2}\omega\cdot\frac{\Delta}{\sqrt{k}}} dW \int_{\frac{9}{4}\omega+W}^{\frac{9}{4}\omega+W+\omega/4} \overline{\pi}(kx)dx \\
&\leq \frac{5}{2}\omega\frac{\Delta}{\sqrt{k}}\cdot\frac{\omega}{4}\xi = \frac{5}{8}\frac{\Delta}{\sqrt{k}}\omega^2\xi \quad \text{(Apply Claim B.4)}
\end{aligned} \tag{25}$$

We substitute (24) and (25) in (21) to conclude the proof, choosing a small enough constant ξ. $\qquad\square$

Proof of Lemma B.6. We express the difference between the average score probability of a key in N_0 (18) and the score probability of a key in N' (17):

$$\begin{aligned}
p_0(\pi) - p'(\pi) &= \frac{\sqrt{k}}{|N_0|} \cdot \int_{q_a}^{q_b} \int_{-\sqrt{k}}^{\infty} \overline{\pi}(\frac{k+\sqrt{k}z}{q}) f_Z(k;z)\cdot z dz \cdot f_\lambda(q)dq \\
&= \frac{\sqrt{k}}{|N_0|} \cdot \int_{-\sqrt{k}}^{\infty} \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k+\sqrt{k}z}{q}) \cdot f_\lambda(q)dq \right) \cdot f_Z(k;z)\cdot z dz
\end{aligned} \tag{26}$$

We separately consider Z in the range $I_{\text{in}} := [-\sqrt{k}/4, \sqrt{k}/4]$ and Z outside this range in $I_{\text{out}} = [-\sqrt{k}, \sqrt{k}/4] \cup [\sqrt{k}/4, \infty]$

For outside the range we use that $\int_{q_a}^{q_b} \overline{\pi}(\frac{k+\sqrt{k}z}{q})\cdot f_\lambda(q)dq \in [0,1]$ and tail bounds on $f_Z(k;z)$ (13) (14) and get:

$$\frac{\sqrt{k}}{|N_0|} \int_{I_{\text{out}}} \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k+\sqrt{k}z}{q}) \cdot f_\lambda(q)dq \right) \cdot f_Z(k;z) \cdot z dz = \frac{1}{N_0}e^{-\Omega(k)} \quad \text{Apply (14) and (13)} \tag{27}$$

For inside the range we apply Claim B.7:

$$\begin{aligned}
&\frac{\sqrt{k}}{|N_0|} \cdot \int_{I_{\text{in}}} \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k+\sqrt{k}z}{q}) \cdot f_\lambda(q)dq \right) \cdot f_Z(k;z) \cdot z dz \\
&= \frac{\sqrt{k}}{|N_0|} \cdot \left(\int_{I_{\text{in}}} \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}) \cdot f_\lambda(q)dq \right) \cdot f_Z(k;z) \cdot z dz + \int_{I_{\text{in}}} \Theta(\frac{z}{\sqrt{k}}) \cdot f_Z(k;z) \cdot z dz \right) && \text{Claim B.7} \\
&= \frac{\sqrt{k}}{|N_0|} \cdot \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}) \cdot f_\lambda(q)dq \cdot \int_{I_{\text{in}}} f_Z(k;z) \cdot z dz + \frac{1}{\sqrt{k}} \cdot \Theta\left(\int_{I_{\text{in}}} f_Z(k;z) \cdot z^2 dz \right) \right) \\
&= \frac{\sqrt{k}}{|N_0|} \cdot \left(\int_{q_a}^{q_b} \overline{\pi}(\frac{k}{q}) \cdot f_\lambda(q)dq \cdot 0 + \frac{1}{\sqrt{k}} \cdot \Theta(1) \right) && \text{Using (9) and (11)} \\
&= \frac{\sqrt{k}}{|N_0|} \cdot \frac{1}{\sqrt{k}} \cdot \Theta(1) = \Theta(\frac{1}{|N_0|}) \tag{28}
\end{aligned}$$

The statement of the Lemma follows by combining (27) and (28) in (26). $\qquad\square$

B.5. The case of symmetric estimators

The proof of Lemma 7.3 (gap for symmetric estimators) follows as a corollary of the proof of Lemma B.6.

Proof of Lemma 7.3. For symmetric maps (Definition 7.2) keys in N_0 that have lower rank can only have higher scoring probabilities. That is, when $j < j'$, the score probability of m_j^i is no lower than that of $m_{j'}^i$. With bottom-k sketches, the score probability of m_j is no lower than that of $m_{j'}$. In particular, the keys in N_0^* have the highest average score among keys in their components. Additionally, there is symmetry between components. Therefore, the average score of each of the k lowest rank keys in N_0^* is no lower than the average over all N_0 keys:

$$\mathsf{E}_U[\pi(S_\rho(U) \cdot \mathbf{1}(m \in U)] \geq p_0(\pi) .$$

Therefore using Lemma B.6:

$$\mathsf{E}_U[\pi(S_\rho(U) \cdot \mathbf{1}(m \in U)] - p'(\pi) \geq p_0(\pi) - p'(\pi) - \mathsf{E}_U[\pi(S_\rho(U) \cdot \mathbf{1}(u \in U)] = \Omega(\frac{1}{k \log(kr)}) .$$

$\square$

B.6. Analysis details of the Adaptive Algorithm

We consider the information available to the query response algorithm. The mask M is shared with the query response and hence it only needs to estimate the cardinality of (the much larger) set U. The mask keys M hide information in $S_\rho(U)$ and make additional keys trasparent. For k-partition and k-mins sketches, keys m_h^i where $h > \arg\min_j m_j^i \in M$ are transparent. For bottom-k sketches, we see only $k' \leq k$ bottom ranks in U if $k - k'$ keys from M have lower ranks.

We describe the sampling of $S(U \cup M)$ (for given mask M) as an equivalent process that examines keys in U in order, selecting each examined key with probability q, until the sketch is determined. This process generalizes the process we described for the case without a mask in the proof of Lemma B.2:

- Bottom-k sketch: Set counter $c \leftarrow k$. $t \sim \mathsf{Geom}[q]$. Process keys m_j in increasing j:
    1. If $m_j \in M$ decrease c and output m_j. If $c = 0$ halt.
    2. If $m_j \notin M$ then decrease t.
        (a) If $t = 0$ output m_j, decrease c, and sample a new $t \sim \mathsf{Geom}[q]$. If $c = 0$ halt.

- k-mins and k-partition sketches: For $i \in [k]$ let $h^i \leftarrow \arg\min_\ell m_\ell^i \in M$. Sample $t \sim \mathsf{Geom}[q]$. Process $i \in [k]$ in order:
    1. If h is defined and $h \leq t$ then $t \leftarrow t - h + 1$ and continue with next i.
    2. If h is undefined or $h > t$ then output m_t^i, sample new $t \sim \mathsf{Geom}[q]$. Continue to next i.

The QR algorithm has the results of the process which yields $k' \leq k$ i.i.d. $\mathsf{Geom}[q]$ random variables. As keys are added to the mask M the information we can glean on q from the sketch, that corresponds to the number k' of $\mathsf{Geom}[q]$ samples we obtain, decreases. As the mask gets augmented, the number of keys, additional keys in N_0 become transparent in the sense that they have probability smaller than δ/r to impact the sketch if included in U. With k-mins and k-partition sketches keys where $m_j^i > h^i$ become transparent. With bottom-k sketches keys m_j where $j > \min_i(k - |M \cap (m_\ell)_{\ell<j}|) \cdot \Omega(\log(r))$ are transparent. These keys are no longer candidates to be examined by the process above. We denote by $N_0' \subset N_0$ the set of keys that remain non-transparent. It holds that $|N_0'| = O(\overline{k'} \log(kr))$, where $\overline{k'}$ is the mean k' with our current mask. When $k' = O(\log(kr))$ becomes too small (see Remark 6.2), there are no correct maps and the algorithm halts and returns M.

Let $p(\pi, M, x)$ be the probability that key x is scored with map π and mask M. The probability is the same for all transparent keys $x \notin N_0'$ and we denote it by $p'(\pi, M)$.

Lemma B.8 (Score probability gap with mask). *Let π be a correct map for $M \cup \mathcal{D}_0$. Then*

$$\sum_{x \in N_0'} (p(\pi, M, x) - p'(\pi, M)) = \Omega\left(\frac{1}{\log(kr)}\right) .$$

Proof. The proof is similar to that of Lemma B.6 applies with respect to k' and using that $|N_0'| = O(\overline{k'}\log(kr))$. □

Claim B.9. With probability at least 0.99, no transparent keys are placed in M.

Proof. First note that all transparent keys have the same score distribution (see proof of Theorem 7.1). Keys get placed in M when their score separates from the median score in $N \setminus M$. Note that since nearly all keys (except α fraction) are transparent, the median score is the score of a transparent key. From Chernoff bounds (3) the probability that a transparent key at a given step is placed in M (and deviates by more than λ from its expectation) is $< 1/(100nr)$. Taking a union bound over all steps and transparent keys we obtain the claim. □