(Nearly) All Cardinality Estimators Are Differentially Private

Charlie Dickens[1], Justin Thaler[2], and Daniel Ting[3]

[1]Yahoo
[2]Georgetown University
[3]Meta*

Abstract

We consider privacy in the context of streaming algorithms for cardinality estimation. We show that a large class of algorithms all satisfy ϵ-differential privacy, so long as (a) the algorithm is combined with a simple down-sampling procedure, and (b) the cardinality of the input stream is $\Omega(k/\epsilon)$. Here, k is a certain parameter of the sketch that is always at most the sketch size in bits, but is typically much smaller. We also show that, even with no modification, algorithms in our class satisfy (ϵ, δ)-differential privacy, where δ falls exponentially with the stream cardinality.

Our analysis applies to essentially all popular cardinality estimation algorithms, and substantially generalizes and tightens privacy bounds from earlier works.

1 Introduction

Cardinality estimation, or the distinct counting problem, is a fundamental data analysis task. Typical applications are found in network traffic monitoring [EVF03], query optimization [SAC$^+$79], and counting unique search engine queries [HNH13]. A key challenge is to perform this estimation in small space, say at most logarithmic in the number of distinct items in the input, while processing each data item quickly (ideally in constant time per item).

Typical approaches for solving this problem at scale are the Flajolet-Martin (FM85) sketch [FM85], sometimes referred to as PCSA (short for Probabilistic Counting with Stochastic Averaging), and its more practical variants such as HyperLogLog (HLL) [FFGM07] (we describe details of these algorithms later, in Section 2). These algorithms fall into a class that we call *hash-based, order-invariant* cardinality estimators. This class consists of all algorithms that satisfy the following two properties. The first is that the algorithm utilizes no information about each input identifier x_i other than a uniform random hash of x_i. The second is that the algorithm depends only on the *set* of observed hash values. This means the algorithm satisfies both permutation- and duplication-invariance. That is, the produced sketch does not depend on the ordering of the input stream, nor on the number of times any item is duplicated in the stream.

In addition to the FM and HLL sketches, other popular algorithms in this class include Bottom-k [BYJK$^+$02, CK07, BGH$^+$09] (also called MinCount or k-minimum values (KMV for short)), and Adaptive Sampling [Fla90].[1] For example, the Bottom-k sketch hashes each input item to a value in the interval $[0, 1]$, and stores the smallest k hash values observed.[2]

While research has historically focused on the accuracy, speed, and space usage of these sketches, recent work has examined their privacy guarantees. These privacy-preserving properties have grown in importance as companies have built tools that can grant an appropriate level of privacy to different people and scenarios.

*This work was performed while the author was at Salesforce.

[1]Flajolet [Fla90] attributes the Adaptive Sampling algorithm to an unpublished 1984 communication of Wegman.

[2]In practice, hash values in a Bottom-k sketch consist of a finite number of bits (say, 32 or 64), which are then interpreted as the binary representation of a number in the interval $[0, 1]$.

The tools may help satisfy users' demand for better data stewardship, while also ensuring compliance with regulatory requirements. For example, a database may offer a baseline guarantee that individual aggregates are ϵ-differentially private to a set of trustworthy individuals (ϵ-DP for short), but a public release of data may require all the statistics in the data to be jointly ϵ'-differentially private with a smaller privacy parameter $\epsilon' < \epsilon$.

One view, presented by Desfontaines, Lochbihler, and Basin [DLB19], is that cardinality estimators cannot simultaneously have good utility and preserve privacy. More precisely, any accurate sketch would allow an adversary to identify a user's presence in the original data set. Hence, accurate sketches should be considered as sensitive as raw data when guarding against privacy violations.

However, the impossibility result of [DLB19] applies only to a setting wherein an adversary knows the hash functions used to generate the sketch and no additional noise unknown to the adversary is applied. In fact, in the setting considered in [DLB19], differential privacy is trivially impossible to achieve, because from the perspective of the adversary, the produced sketch is a deterministic function of the input. The main result of [DLB19] is that even weaker notions of privacy cannot be achieved in this context.

Other works have studied more realistic models where either the hashes are public, but private noise is added to the sketch [TS13, CDSKY20, MMNW11, vVT19], or the hashes are secret (i.e., not known to the adversary who is trying to "break" privacy), a setting that turns out to permit less noisy cardinality estimates. For example, Smith et al. [SSGT20] show that an HLL-type sketch is differentially private[3] while [vVT19] modifies the Flajolet-Martin sketch using coordinated sampling, which is based on a private hash. Variants of both models are analyzed by Choi et al. [CDSKY20], and they show (amongst other contributions) a similar result to [SSGT20], establishing that an HLL-type sketch is differentially private. As with these prior works, our focus is the setting when *the hash functions are kept secret* from the adversary.

A related problem, of differentially private estimation of cardinalities under set operations is studied by [PS21], but they assume the inputs to each sketch are already de-duplicated.

Our contributions. We show that *all* hash-based, order-invariant cardinality estimators are ϵ-differentially private so long as the algorithm is combined with a simple down-sampling procedure. A detailed overview of specific sketches to which our results apply can be found in Section 4. As with earlier results, our analysis holds provided that a mild lower bound on the number of distinct items in the stream is satisfied, roughly $\Omega(k/\epsilon)$, where k is a sketch parameter that can be thought of as the number of "buckets" used by the sketch.[4] For any hash-based, order-invariant algorithm, k is always upper bounded by the number of bits in the sketch, but is typically significantly smaller. For example, the number of buckets used by a Bottom-k sketch is k, while the number of bits in the sketch is larger by a factor equal to the number of bits in each hash value.

The $\Omega(k/\epsilon)$ stream cardinality lower bound needed to ensure privacy can be guaranteed to hold by inserting sufficiently many "phantom elements" into the stream when initializing the sketch. One can then subtract off the number of phantom elements from the estimate returned by the sketch, without violating privacy. This padding technique guarantees that the sketch *always* satisfies ϵ-differential privacy, even when run on inputs with very small cardinality. Of course, the insertion of "phantom elements" may increase the error of the sketch on such small-cardinality inputs. But such an increase in error is inherent, as these sketches are clearly non-private (e.g., by virtue of returning the *exact* answer) on small enough input streams.

We also show that, even with no modification, algorithms in our class satisfy (ϵ, δ)-differential privacy, where δ falls exponentially with the stream cardinality.

Our novel analysis has significant benefits. First, prior works on differentially private cardinality estimation have analyzed only specific sketches [TS13, vVT19, CDSKY20, SSGT20]. Moreover, many of the sketches analyzed (e.g., [TS13, SSGT20]), while reminiscent of sketches used in practice, in fact differ from practical sketches in important ways. For example, as we explain in detail later (Section 4.3), Smith et al. [SSGT20] analyze a *variant* of HLL that has much slower update time (by a factor of k) than HLL itself.

[3]Despite the title of [SSGT20] referencing Flajolet and Martin, they actually study an algorithm more closely related to LogLog and HyperLogLog (HLL) sketches, rather than the Flajolet-Martin (FM85) sketch. The former sketches can be thought of as lossy compressions of the latter.

[4]More precisely, the cardinality lower bound we require is $(k-1)/(1-e^{-\epsilon})$, which is $\Theta(k/\epsilon)$ if $\epsilon \leq 1$ and $\Theta(k)$ if $\epsilon > 1$.

In contrast to the literature, we analyze an entire class of sketches at once. Even when specialized to specific sketches, our error analysis improves upon prior work in many cases. For example, our analysis yields tighter privacy bounds for HLL than the one given in [CDSKY20], yielding an ϵ-DP guarantee rather than an (ϵ, δ)-DP guarantee—see Section 4.3 for details. Crucially, the class of sketches we analyze captures many (in fact, almost all) of the sketches that, to our knowledge, are actually used in practice. This means that existing systems can be used in contexts requiring privacy either without modification (if streams are guaranteed to satisfy the mild cardinality lower bound we require), or via the simple pre-processing step described above if such cardinality lower bounds may not hold. Thus, existing data infrastructure can be easily modified to provide DP guarantees, and in fact existing data (and, as we show, sketches thereof) can be easily migrated to DP summaries.

There is one standard caveat: following prior works [SSGT20, CDSKY20] our privacy analysis assumes a perfectly random hash function. One can remove this assumption both in theory and practice by using a cryptographic hash function. This will yield a sketch that satisfies either a computational variant of differential privacy called SIM-CDP, or standard information-theoretic notions of differential privacy under the assumption that the hash function fools space-bounded computations. See [SSGT20, Section 2.3] for details.

2 Problem Definition

Let $\mathcal{D} = \{x_1, \ldots, x_n\}$ denote a stream of samples with each identifier x_i coming from a large universe U, e.g., of size 2^{64} (or larger). The objective is to estimate the cardinality of $\mathcal{D}$ using an algorithm S which is given privacy parameters $\epsilon, \delta \geq 0$ and a space bound b.

Definition 1 (Differential Privacy [DMNS06])**.** *A randomized algorithm S is (ϵ, δ)-differentially private (or (ϵ, δ)-DP for short) if for any pair of data sets $\mathcal{D}, \mathcal{D}'$ that differ in one record and for all S in the range of S,*

$$\Pr(S(\mathcal{D}') \in S) \leq e^{\epsilon} \Pr(S(\mathcal{D}) \in S) + \delta \tag{1}$$

In Equation (1), the probability is over the internal randomness of the algorithm S. For brevity, $(\epsilon, 0)$-DP is referred to simply as ϵ-DP, or sometimes as *pure* ϵ-differential privacy.

Rather than analyzing any specific sketching algorithm, we analyze a natural class of randomized distinct counting sketches. Algorithms in this class operate in the following manner: each time a new stream item i arrives, i is hashed using some uniform random hash function h, and then $h(i)$ is used to update the sketch, i.e., the update procedure depends only on $h(i)$, and is otherwise independent of i. Our analysis applies to any such algorithm that depends only on the *set* of observed hash values. Equivalently, the sketch state is invariant both to the order in which stream items arrive, and to item duplication.[5] We call this class of algorithms *hash-based, order-invariant* cardinality estimators.

All distinct counting sketches of which we are aware that are invariant to permutations of the input data are included in this class. As we explain in Section 4, this includes MinCount, HLL, LPCA, and probabilistic counting. Note that for any hash-based, order-invariant cardinality estimator, the distribution of the sketch depends only on the cardinality of the stream.

Definition 2 (Hash-Based, Order-invariant Cardinality Estimators)**.** *Any sketching algorithm that depends only on the* set *of hash values of stream items using a uniform random hash function is a* hash-based order-invariant cardinality estimator*. We denote this class of algorithms by $\mathcal{C}$.*

We denote a sketching algorithm over data $\mathcal{D}$ with internal randomness r by $S_r(\mathcal{D})$ (for hash-based algorithms, r will specify the hash function used by the algorithm). The *state of the sketch*, referred to as the *sketch* for short, is denoted by s and is the representation of the data structure defined by algorithm $S_r(\mathcal{D})$. The sample space Ω from which the sketch is drawn varies upon the specific sketching algorithm

[5] By *duplication-invariance*, we mean that the state of the sketch when run on any stream σ is identical to its state when run on the "de-duplicated" stream σ' in which each item appearing one or more times in σ appears exactly once in σ'.

being used. Sketches are first initialized and then items are inserted into the sketch through an `add` operation which may or may not change the state of the sketch.

The size of the sketch is a crucial constraint, and we denote the space consumption in bits by b. For example, FM85 is a sketching algorithm whose state consists of k bitmaps, each of length ℓ. Thus, its state $s \in \Omega = \{0,1\}^{k \times \ell}$. A common value is $\ell = 32$, so that $b = 32k$. Further such examples are given in Section 4.

3 Hash-Based Order-Invariant Estimators are Private

3.1 Conditions Guaranteeing Privacy

Given a collection of n distinct identifiers $\mathcal{D}$ and sketching algorithm S with internal randomness r, denote the resulting sketch by $S_r(\mathcal{D})$. For $i \in \mathcal{D}$ denote the set $\mathcal{D} \backslash \{i\}$ by $\mathcal{D}_{-i}$. For all hash-based, order-invariant cardinality estimators, the distribution of the sketch depends only on the number of distinct elements in the input stream, and so without loss of generality we assume henceforth that $\mathcal{D} = \{1, \ldots, n\}$.

By definition, for an ϵ-differential privacy guarantee, we must show

$$e^{-\epsilon} < \frac{\Pr_r(S_r(\mathcal{D}) = s)}{\Pr_r(S_r(\mathcal{D}_{-i}) = s)} < e^{\epsilon} \quad \forall s \in \Omega, i \in \mathcal{D}. \tag{2}$$

Overview of the privacy results. The main result in our analysis bounds the privacy loss of a hash-based, order-invariant sketch in terms of just two sketch-specific quantities. Both quantities intuitively capture how sensitive the sketch is to the removal or insertion of a single item from the data stream.

The first quantity is $k_{max} \coloneqq \sup_r K_r$, where K_r denotes the number of items from $\mathcal{D}$ that would change the sketch if removed from the stream, when the internal randomness used by the sketch is r (see Equation (4) for details). As we show later, k_{max} is always bounded above by the number of bits in the sketch, but for most sketches is much smaller. The second quantity is $\pi^* \coloneqq \sup_{s \in \Omega} \pi(s)$, where $\pi(s)$ denotes the probability over the algorithm's internal randomness r that adding one more item to the stream would change the sketch, conditioned on the sketch's state prior to the addition being s (see Equation (10) for details).

The main sub-result in our analysis (Theorem 7) roughly states that the sketch is ϵ-DP so long as (a) π^* is not too close to 1 (specifically, so long as $\pi^* < 1 - e^{-\epsilon}$), and (b) the stream cardinality n is larger than $\frac{k_{max}-1}{e^{\epsilon}-1}$, which for small values of ϵ is $\Theta(k_{max}/\epsilon)$.

We actually show that Property (a) *must* be satisfied by any ϵ-DP algorithm, if the algorithm works over data universes of unbounded size. Unfortunately, Property (a) does *not* directly hold for natural sketching algorithms. But we show (Section 3.2.2) that Property (a) can be generically satisfied by combining any hash-based, order-invariant sketching algorithm with a simple down-sampling procedure.

Overview of the analysis. To establish the main result outlined above (Theorem 7) our basic strategy proceeds in two steps. First, as indicated above, we bound the quantity K_r, i.e., we show that for any fixed value of the sketching procedure's internal randomness r, very few items $i \in \mathcal{D}$ actually affect in the sketch. That is, there are very few items $i \in \mathcal{D}$ whose removal from $\mathcal{D}$ would change the resulting sketch, in the sense that $S_r(\mathcal{D}) \neq S_r(\mathcal{D}_{-i})$. In particular, we show that the number K_r of such items is always at most the size b of the sketch in bits (and typically is significantly smaller). Second, the order-invariance of the sketch generates a great deal of symmetry that we can exploit. Effectively, we show that for any $i \in \mathcal{D}$, the probability that $S_r(\mathcal{D}) \neq S_r(\mathcal{D}_{-i})$ is at most K_r/n, or more precisely, at most the probability that a random subset of $\mathcal{D}$ of size K_r contains i. This is enough to establish Equation (2), provided that n is sufficiently larger than K_r (roughly by a factor of $1/\epsilon$).

The above two-step description is, however, a significant simplification of the full argument. A more detailed overview follows.

Intuitively, the first step of our analysis works as follows. Since a distinct counting sketch must perform a deduplication operation to ensure only *distinct* items are counted, it also allows for approximate set

membership queries, albeit with *very* high error. In order to remain small (at most b bits), only information about a few items can be stored in the sketch at any given time (certainly at most b items, but for many sketches of interest, information about even fewer than b items is stored). Other than these at most b items, all remaining items from $\mathcal{D}$ do not affect the final state of the sketch. Thus, for any sketch outcome $s \in \Omega$,

$$\Pr_r(S_r(\mathcal{D}) = s | S(\mathcal{D}_{-i}) = s) \approx 1. \tag{3}$$

While the definition of the conditional probability on the left hand side of Equation (3) is tantalizingly close to the Bayes ratio we wish to bound, namely $\frac{\Pr_r(S_r(\mathcal{D})=s)}{\Pr_r(S_r(\mathcal{D}_{-i})=s)}$, the above reasoning only provides a lower bound on the ratio. It does not rule out the possibility that adding item i changes the sketch *to* s with high probability. That is, $\Pr_r(S_r(\mathcal{D}) = s) \geq \Pr_r(S_r(\mathcal{D}) = s \wedge S_r(\mathcal{D}_{-i}) \neq s)$, and the right hand side may still be large.

In order to convert the Bayes ratio into a form where we can explicitly compute the relevant probabilities, we find some other item $j \neq i$ whose inclusion does *not* change the sketch. Using symmetry we can change the item i in the denominator to j and obtain a probability conditional on $S_r(\mathcal{D}_{-j}) = s$. A combinatorial argument further exploiting symmetry gives our final bound.

Details of the Analysis. Let

$$\mathcal{K}_r := \{i \in \mathcal{D} : S_r(\mathcal{D}_{-i}) \neq S_r(\mathcal{D})\} \tag{4}$$

denote the set of items which, when removed from the data set, change the sketch and denote its cardinality by $K_r := |\mathcal{K}_r|$. Also, let

$$J_r := \min\{i : S_r(\mathcal{D}_{-i}) = S_r(\mathcal{D})\}$$

denote the smallest index amongst the remaining $n - K_r$ items in $\mathcal{D}$ that do not change the sketch. If removing *any* item changes the sketch, i.e., if $S_r(\mathcal{D}_{-i}) \neq S_r(\mathcal{D})$ for all $i \in \mathcal{D}$, then $K_r = n$. For this case, we define J_r to be a special symbol $\perp$.

The following lemmas relate the state of a sketch over data $\mathcal{D}$, $S_r(\mathcal{D})$, to the states of the sketch when an item is omitted, $S_r(\mathcal{D}_{-i})$.

Lemma 3. *Suppose* $n > \sup_r K_r$. *Then* $\Pr_r(K_r = n) = 0$, *and*

$$\frac{\Pr_r(S_r(\mathcal{D}) = s)}{\Pr_r(S_r(\mathcal{D}_{-i}) = s)} = \sum_{j \in \mathcal{D}} \Pr_r(J_r = j \,|\, S_r(\mathcal{D}_{-j}) = s). \tag{5}$$

Proof. First, let us rewrite $\Pr_r(S_r(\mathcal{D}) = s)$ as a sum over all possible values of J_r:

$$\Pr_r(S_r(\mathcal{D}) = s) = \sum_{j \in \mathcal{D} \cup \{\perp\}} \Pr_r(J_r = j \wedge S_r(\mathcal{D}) = s). \tag{6}$$

Next, we split the right hand side of Equation (6) into the distinct cases wherein $J_r = \perp$ and $j \in \mathcal{D}$, as we will ultimately deal with each case separately:

$$\sum_{j \in \mathcal{D} \cup \{\perp\}} \Pr_r(J_r = j \wedge S_r(\mathcal{D}) = s) =$$
$$\Pr_r(J_r = \perp \wedge S_r(\mathcal{D}) = s) + \sum_{j \in \mathcal{D}} \Pr_r(J_r = j \wedge S_r(\mathcal{D}_{-j}) = s). \tag{7}$$

Next, the summands over $j \in \mathcal{D}$ are decomposed via conditional probabilities. Specifically, the right hand side of Equation (7) equals:

$$\Pr_r(J_r = \perp \wedge S_r(\mathcal{D}) = s) + \sum_{j \in \mathcal{D}} \Pr_r(J_r = j \,|\, S_r(\mathcal{D}_{-j}) = s) \Pr_r(S_r(\mathcal{D}_{-j}) = s). \tag{8}$$

For any hash-based, order-invariant sketch, the distribution of $S(\mathcal{D})$ depends only on the number of distinct elements in $\mathcal{D}$, and hence the factor $\Pr_r(S_r(\mathcal{D}_{-j}) = s)$ appearing in the jth summand of Equation (8) equals $\Pr_r(S_r(\mathcal{D}_{-i}) = s)$, where i is the element of $\mathcal{D}$ referred to in the statement of the lemma. Accordingly, we can rewrite Expression (8) as:

$$\Pr_r(J_r = \perp \wedge S_r(\mathcal{D}) = s) + \sum_{j \in \mathcal{D}} \Pr_r(J_r = j \mid S_r(\mathcal{D}_{-j}) = s) \Pr_r(S_r(\mathcal{D}_{-i}) = s). \tag{9}$$

Clearly, $J_r \neq \perp$ whenever the number of items in the data set, n, exceeds K_r. Hence, if $n > \sup_r K_r$, $\Pr_r(J_r = \perp \wedge S_r(\mathcal{D}) = s) = 0$. We obtain Equation (5) as desired. □

Lemma 4 states that $\sup_r K_r$ is always at most b, the size of the sketch $S_r(\mathcal{D})$ in bits, though as explained in Section 4, $\sup_r K_r$ is much smaller than b for popular sketching algorithms.

Lemma 4. *For any distinct counting sketch with size in bits bounded by b, $\sup_r K_r \leq b$.*

Proof. Consider some data set $\mathcal{D}$ and sketch $s = S_r(\mathcal{D})$. Recall that we denote the set of items whose removal would change the sketch by $\mathcal{K}_r(\mathcal{D}) := \{i \in \mathcal{D} : S_r(\mathcal{D}_{-i}) \neq S_r(\mathcal{D})\}$. Consider any subset $\Lambda \subset \mathcal{K}_r(\mathcal{D})$. Then we claim that, for any $x \in \mathcal{K}_r$, adding x to the sketch $S_r(\Lambda)$ will change it if and only if $x \in \mathcal{K}_r(\mathcal{D}) \setminus \Lambda$. That is, if $\Lambda \circ x$ denotes the stream consisting of one occurrence of each item in Λ, followed by x, then $S_r(\Lambda) \neq S_r(\Lambda \circ x)$ if and only if $x \in \mathcal{K}_r(\mathcal{D}) \setminus \Lambda$.

To see this, first observe that duplication-invariance of the sketching algorithm implies that if $x \in \Lambda$ then $S_r(\Lambda) = S_r(\Lambda \circ x)$. Second if $x \notin \Lambda$, suppose by way of contradiction that $S_r(\Lambda) = S_r(\Lambda \circ x)$, and let $T = \mathcal{D} \setminus (\Lambda \cup \{x\})$. Since $S_r(\Lambda) = S_r(\Lambda \circ x)$, it holds that $S_r(\Lambda \circ x \circ T) = S_r(\Lambda \circ T) = S_r(\mathcal{D}_{-x})$. Yet by order-invariance of the sketching algorithm, $S_r(\Lambda \circ x \circ T) = S_r(\mathcal{D})$. We conclude that $S_r(\mathcal{D}_{-x}) = S_r(\mathcal{D})$, contradicting the assumption that $x \in \mathcal{K}_r(\mathcal{D})$.

The above means that for any fixed r, the sketch $S_r(\Lambda)$ losslessly encodes the arbitrary subset Λ of $\mathcal{K}_r$. Hence, the sketch requires at least $\log_2(2^{|\mathcal{K}_r|}) = |\mathcal{K}_r|$ bits to represent. Thus, any sketch with size bounded by m bits can have at most m items that affect the sketch. □

Comparing Equations (2) and (5), we see that to establish $(\epsilon, 0)$-DP we must show the right hand side of Equation (5) lies in the interval $[e^{-\epsilon}, e^{\epsilon}]$. To do so, we define

$$\pi(s) \coloneqq \Pr_r(S_r(\mathcal{D}) \neq S_r(\mathcal{D}_{-i}) | S_r(\mathcal{D}_{-i}) = s) \tag{10}$$

to be the probability that adding item i to a sketch in state s will change the sketch. Conceptually, it can be helpful to think of $\pi(s)$ as the probability that, when processing an as-yet-unseen item i when the sketch is in state s, i gets "sampled" by the sketch. For a sketch such as Bottom-k, which computes a sample of (hashes of) stream items, $\pi(s)$ is literally the probability that i is sampled by the sketch. Accordingly, we refer to $\pi(s)$ later in this manuscript as a "sampling probability".

Lemma 5. *Under the same assumptions as Lemmas 3 and 4,*

$$\sum_{j \in \mathcal{D}} \Pr_r(J_r = j \mid S_r(\mathcal{D}_{-j}) = s) = (1 - \pi(s)) \mathbb{E}_r \left(1 + \frac{K_r}{n - K_r + 1} \middle| S_r(\mathcal{D}_{-1}) = s\right). \tag{11}$$

Proof. We begin by writing

$$\Pr_r(J_r = j \mid S_r(\mathcal{D}_{-j}) = s) = \sum_{k=1}^{n} \Pr_r(J_r = j \mid K_r = k, S_r(\mathcal{D}_{-j}) = s) \Pr_r(K_r = k | S_r(\mathcal{D}_{-j}) = s). \tag{12}$$

To analyze Expression (12), we first focus on the $\Pr_r(J_r = j \,|\, K_r = k, S_r(\mathcal{D}_{-j}) = s)$ term. Given that $K_r = k$ and $S_r(\mathcal{D}_{-j}) = s$, we know that $J_r = j$ if and only if the items $\{1, \ldots, j-1\}$ are *all* in $\mathcal{K}_r$ *and* item j is not in $\mathcal{K}_r$. The first condition occurs with probability $\frac{\binom{n-(j-1)}{k-(j-1)}}{\binom{n}{k}} = \frac{\binom{k}{j-1}}{\binom{n}{j-1}}$. This is because there are $\binom{n-(j-1)}{k-(j-1)}$ subsets K_r of $\{1, \ldots, n\}$ of size k that contain items $1, \ldots, j-1$, out of $\binom{n}{k}$ subsets of $\mathcal{K}_r$ of size k. Meanwhile, $j \notin \mathcal{K}_r$ means that $S_r(\mathcal{D}_{-j}) = S_r(\mathcal{D})$, which occurs with probability that is exactly the complement of the sampling probability $\pi(s)$ (see Equation (10)).

By the above reasoning, the left hand side of Expression (11) equals:

$$\begin{aligned}
&\sum_{j \in \mathcal{D}} \sum_{k=1}^{n} \Pr_r(J_r = j \,|\, K_r = k, S_r(\mathcal{D}_{-j}) = s) \Pr_r(K_r = k | S_r(\mathcal{D}_{-j}) = s) \\
&= \sum_{k=1}^{n} \sum_{j \in \mathcal{D}} \frac{\binom{k}{j-1}}{\binom{n}{j-1}} (1 - \pi(s)) \Pr_r(K_r = k | S_r(\mathcal{D}_{-1}) = s) \\
&= (1 - \pi(s)) \sum_{k=1}^{n} \left(1 + \frac{k}{n-k+1}\right) \Pr_r(K_r = k | S_r(\mathcal{D}_{-1}) = s) \\
&= (1 - \pi(s)) \mathbb{E}_r \left(1 + \frac{K_r}{n - K_r + 1} \middle| S_r(\mathcal{D}_{-1}) = s\right). \qquad (13)
\end{aligned}$$

□

Having established Lemma 5, we are finally in a position to derive a result showing that a hash-based, order-invariant sketch is ϵ-DP so long as the stream cardinality is large enough and $\sup_{s \in \Omega} \pi(s)$ is not too close to 1.

Corollary 6. *Let*

$$\pi_0 := 1 - e^{-\epsilon} \qquad (14)$$

and let Ω denote the set of all possible states of a hash-based order-invariant distinct counting sketching algorithm. When run on a stream of cardinality $n > \sup_r K_r$, the sketch output by the algorithm satisfies ϵ-DP if

$$\pi_0 = 1 - e^{-\epsilon} > \sup_{s \in \Omega} \pi(s) \qquad (15)$$

and, for all sketch states $s \in \Omega$,

$$e^{\epsilon} > \mathbb{E}_r \left(\frac{n}{n - K_r + 1} \middle| S_r(\mathcal{D}_{-1}) = s \right). \qquad (16)$$

Furthermore, if the data stream $\mathcal{D}$ consists of items from a universe U of unbounded size, Condition 15 is necessarily satisfied by any *sketching algorithm satisfying ϵ-DP.*

Proof. Since $1 - \pi(s) \leq 1$ for all s and $\frac{n}{n-K_r+1} = 1 + \frac{K_r - 1}{n-K_r+1} \geq 1$, it follows from Lemma 5, (15) and (16) that the right hand side of Equation (5) lies in the interval $[e^{-\epsilon}, e^{\epsilon}]$, as required for an ϵ-DP guarantee.

For the necessity of Condition 15, note that if the universe of possible items is infinite, then for any possible sketch state s, there exists an arbitrarily long sequence of distinct items that results in state s if $\pi(s) < 1$. One simply needs to search for a sequence of items which do not change the sketch. Combining Lemma 3 with Equation (11) in Lemma 5 therefore implies that

$$e^{-\epsilon} < \inf_s (1 - \pi(s)) \qquad (17)$$

and hence $\sup_s \pi(s) < 1 - e^{-\epsilon}$ as claimed. □

The above corollary may be difficult to apply directly since the expectation in Condition (16) is often difficult to compute and depends on the unknown cardinality n. Our main result provides sufficient criteria to ensure that Condition (16) holds. The criteria is expressed in terms of a minimum cardinality n_0 and sketch-dependent constant k_{max}. This constant k_{max} is a bound on the maximum number of items which change the sketch when removed. That is, for all input streams $\mathcal{D}$ and all r, $k_{max} \geq |\mathcal{K}_r|$, where recall from Equation (4) that for a given input stream $\mathcal{D}$, $\mathcal{K}_r = \{i \in \mathcal{D} \colon S_r(\mathcal{D}_{-i}) \neq S_r(\mathcal{D})\}$ denotes the set of items whose absence changes the sketch. We derive k_{max} for a number of popular sketch algorithms in Section 4.

Theorem 7. *For a hash-based, order-invariant distinct counting sketch, let $k_{max} = \sup_r K_r$ (recall that K_r is defined in Equation (4)). The sketch output by the algorithm satisfies an ϵ-DP guarantee if*

$$\sup_{s \in \Omega} \pi(s) < \pi_0 := 1 - e^{-\epsilon} \tag{18}$$

and the number of unique items in the input stream is strictly greater than

$$n_0 \coloneqq \frac{k_{max} - 1}{1 - e^{-\epsilon}}. \tag{19}$$

Proof. We can upper bound the expectation on the right hand side of Condition (16) using k_{max} and n_0. Corollary 6 and solving for n_0 then gives the desired result. Specifically, by Corollary 6, the sketch satisfies ϵ-DP if:

$$e^{\epsilon} > \sup_{n \geq n_0} \mathbb{E}_r \left(1 + \frac{K_r - 1}{n - K_r + 1} \middle| S_r(\mathcal{D}_{-1}) = s \right) \geq \sup_{n \geq n_0} \left(1 + \frac{k_{max} - 1}{n - k_{max} + 1} \right) = 1 + \frac{k_{max} - 1}{n_0 - k_{max} + 1}. \tag{20}$$

Hence, the sketch satisfies ϵ-DP when run on a stream of cardinality n so long as:

$$n > n_0 = k_{max} - 1 + \frac{k_{max} - 1}{e^{\epsilon} - 1} = \frac{k_{max} - 1}{1 - e^{-\epsilon}}. \tag{21}$$

□

Later, we explain how to modify existing sketching algorithms in a black-box way to ensure that Conditions (18) and (19) are satisfied. But for most sketching algorithms used in practice, if left unmodified there will be some sketch values $s \in \Omega$ for which Condition 18 is not satisfied, i.e, $\pi(s) > 1 - e^{-\epsilon}$. Let us call such sketch values s "privacy-violating". Fortunately, for practical sketching algorithms, privacy-violating sketch values s only arise with tiny probability. The next theorem states that, so long as this probability is smaller than δ, the sketch satisfies (ϵ, δ)-DP without modification.

Theorem 8. *Let n_0 be as in Theorem 7. Given a hash-based, order-invariant distinct counting sketch with bounded size, let Ω' be the set of sketch states such that $\pi(s) \geq \pi_0$. If the cardinality n of the input stream $\mathcal{D}$ is greater than n_0, then the sketch is (ϵ, δ) differentially private where $\delta = \Pr_r(S_r(\mathcal{D}) \in \Omega')$.*

Proof. This trivially follows from Theorem 7. □

3.2 Constructing Private Sketches

3.2.1 Approximate Differential Privacy

Theorem 8 states that, when run on a stream with $n > n_0$ distinct items, any hash-based order-invariant algorithm directly (see Algorithm 1a) satisfies (ϵ, δ)-differential privacy where δ denotes the probability that the final sketch state s is "privacy-violating", i.e., $\pi(s) > 1 - e^{-\epsilon}$. Concrete bounds on δ as a function of the input cardinality n for specific practical sketching algorithms are given in Section 4; in all cases considered, δ falls exponentially quickly with n. In the remainder of this subsection, we provide intuition for why this is true and how we later establish the concrete bounds for specific sketches.

If s_t denotes the state of the sketch after processing the first t input items, we show that $\pi(s_t)$ is monotonically decreasing with t (see, for example, the proof of Corollary 11). We can prove the desired bound on δ by analyzing sketches in a manner similar to the coupon collector problem. Assuming a perfect, random hash function, the hash values of a universe of items defines a probability space. For each sketch considered in Section 4, we identify $v \leq k_{max}$ events or coupons, $C_1, \ldots, C_v$, such that $\pi(s)$ is guaranteed to be less than π_0 after all events have occurred. A simple union bound then guarantees that the probability δ that a sketch fails to satisfy an ϵ-DP guarantee decreases exponentially as the cardinality grows.

As additional intuition for why unmodified sketches satisfy an (ϵ, δ)-DP guarantee when the cardinality is large, we note that the inclusion probability $\pi(s)$ is closely tied to the cardinality estimate in most sketching algorithms. For example, the cardinality estimators used in HLL and KMV are inversely proportional to the sampling probability $\pi(s)$, i.e., $\hat{N}(s) \propto 1/\pi(s)$, while for LPCA and Adaptive Sampling, the cardinality estimators are monotonically decreasing with respect to $\pi(s)$. Thus, for most sketching algorithms, when run on a stream of sufficiently large cardinality, the resulting sketch is privacy-violating only when the cardinality estimate is also inaccurate. The following theorem is useful when analyzing the privacy of such sketching algorithms, as it characterizes the probability δ of a "privacy violation" in terms of the probability the sketch returns an estimate $\hat{N}(S_r(\mathcal{D}))$ lower than some threshold $\tilde{N}(\pi_0)$.

Theorem 9. *Recall from Equation* (19) *that* $n_0 = \frac{k_{max}-1}{1-e^{-\epsilon}}$. *Let* S_r *be a sketching algorithm with estimator* $\hat{N}(S_r)$. *Suppose the estimate returned on sketch* s *is a strictly decreasing function of* $\pi(s)$, *so that* $\hat{N}(s) = \tilde{N}(\pi(s))$ *for some function* $\tilde{N}$. *Then, if* $n > n_0$, *the sketching algorithm* S_r *is* (ϵ, δ)*-DP where* $\delta = P(\hat{N}(S_r(\mathcal{D})) < \tilde{N}(\pi_0))$.

Proof. Since $\tilde{N}$ is invertible and decreasing, $P(\hat{N}(S_r(\mathcal{D})) < \tilde{N}(\pi_0)) = P(\pi(S_r(\mathcal{D})) > \pi_0) = \delta$. □

3.2.2 Pure Differential Privacy

Theorem 7 guarantees that a sketch satisfies ϵ-DP if two conditions hold (Conditions (18) and (19)). The first condition requires that factor $\sup_{s \in \Omega} \pi(s)$ be smaller than $1 - e^{-\epsilon}$, i.e., requiring that the "sampling probability" of the sketching algorithm be sufficiently small regardless of the sketch's state s (smaller than $1 - e^{-\epsilon}$). Meanwhile, Condition (19) requires that the number of distinct items in the input stream must be sufficiently large.

We observe that one can take *any* hash-based, order-invariant distinct counting sketching algorithm and turn it into one that satisfies these two conditions by adding a simple pre-processing step, which does two things. First, it "downsamples" the input stream by hashing each input (using a different hash function than the one used by the algorithm being pre-processed), interpreting the hash values as numbers in $[0, 1]$, and simply ignoring numbers whose hashes are larger than π_0. This ensures that Condition (18) is satisfied, by simply discarding each input item with probability π_0. Second, it artificially adds n_0 items to the input stream to ensure that Condition (19) is satisfied (these n_0 items should also be downsampled as per the first modification). An unbiased estimate of the cardinality of the unmodified stream can then be easily recovered from the sketch via a post-processing correction. Note that for this estimate to be unbiased, these n_0 artificial items must be distinct from any items that appear in the "real" stream. Pseudocode for the modified algorithm, which is guaranteed to satisfy ϵ-DP, is given in Algorithm 1c.

In settings where there is an a priori guarantee that the number of distinct stream items n is greater than $n_0 = \frac{k_{max}-1}{1-e^{-\epsilon}}$, the addition of artificial items is not necessary to ensure ϵ-DP. Pseudocode for the resulting ϵ-DP algorithm is given in Algorithm 1b.

Corollary 10. *The functions* `DPSketchLargeSet` *(Algorithm 1b) and* `DPSketchAnySet` *(Algorithm 1c) yield* ϵ*-DP distinct counting sketches provided that* $n \geq n_0$ *and* $n \geq 1$, *respectively.*

Proof. Under their respective assumptions, `DPSketchLargeSet` and `DPSketchAnySet` satisfy Conditions (18) and (19) of Theorem 7. □

function BASE(items, ϵ)

$S \leftarrow InitSketch()$

for $x \in items$ **do**

$S.add(x)$

return $\hat{N}(S)$

(a) Provides (ϵ, δ)-DP guarantee for sufficiently large n.

function DPSKETCHLARGESET(items, ϵ)

$S \leftarrow InitSketch()$

$\pi_0 \leftarrow 1 - e^{-\epsilon}$

for $x \in items$ **do**

if $hash(x) < \pi_0$ **then**

$S.add(x)$

return $\hat{N}(S)/\pi_0$

(b) Provides ϵ-DP guarantee for $n > n_0$.

function DPSKETCHANYSET(items, ϵ)

$S, n_0 \leftarrow DPInitSketch(\epsilon)$

$\pi_0 \leftarrow 1 - e^{-\epsilon}$

for $x \in items$ **do**

if $hash(x) < \pi_0$ **then**

$S.add(x)$

return $\hat{N}(S)/\pi_0 - n_0$

(c) Provides ϵ-DP guarantee for any n.

Algorithms 1: Differentially private cardinality estimation algorithms from black box sketches. The function $InitSketch()$ initializes a sketch in a black-box fashion. The output of the uniform random hash function is interpreted as a number in $[0, 1]$. Note that this hash function is chosen *independently* of the internal randomness of the black-box sketching procedure S (in particular, the hash function used in pre-processing is independent of any hash function used by S). The cardinality estimate returned by sketch S is denoted $\hat{N}(S)$. `DPInitSketch` is given in Algorithm 2a.

function DPINITSKETCH(ϵ)

$S \leftarrow InitSketch()$

$\pi_0 \leftarrow 1 - e^{-\epsilon}$

$n_0 \leftarrow \left\lceil \frac{k_{max}-1}{\pi_0} \right\rceil$

$M \sim Binomial(n_0, \pi_0)$

for $i = 1 \rightarrow M$ **do**

$x \leftarrow NewItem()$

if $hash(x) < \pi_0$ **then**

$S.add(x)$

return S, n_0

(a)

function DPINITSKETCHFORMERGE(ϵ)

$S \leftarrow InitSketch()$

$\pi_0 \leftarrow 1 - e^{-\epsilon}$

$n_0 \leftarrow \left\lceil \frac{k_{max}-1}{\pi_0} \right\rceil$

$v \leftarrow 0$

repeat

$x \leftarrow NewItem()$

$S.add(x)$

$v \leftarrow v + 1$

until $\pi(S) \leq \pi_0$ and $v \geq n_0$

return S, v

(b)

Algorithms 2: Initialization routines for generating ϵ-DP sketches. The function $NewItem()$ returns an item that is guaranteed to come from a data universe disjoint from the universe over which stream items are drawn. In `DPInitSketch`, the binomial draw M simulates inserting n_0 unique items into the sketch, with downsampling probability π_0.

3.2.3 Constructing ϵ-DP Sketches from Existing Sketches: Algorithm 3

As regulations change and new ones are added, existing data may need to be appropriately anonymized. However, if the data has already been sketched, the underlying data may no longer be available, and even if it is retained, it may be too costly to reprocess it all. Our theory allows these sketches to be directly converted into differentially private sketches when the sketch has a merge procedure.

That is, the algorithm assumes that it is possible to take a sketch $S_r(\mathcal{D}_1)$ of a stream $\mathcal{D}_1$ and a sketch $S_r(\mathcal{D}_2)$ of a stream $\mathcal{D}_2$, and "merge" them to get a sketch of the concatenation of the two streams $\mathcal{D}_1 \circ \mathcal{D}_2$. This is the case for most practical hash-based order-invariant discount count sketches. Denote a merge operation between sketches $S_r(\mathcal{D}_1)$ and $S_r(\mathcal{D}_2)$ by $S_r(\mathcal{D}_1) \cup S_r(\mathcal{D}_2)$. Since merging *requires* the same randomness r to be used, we will often suppress the dependency on r in the notation. The merge step is a property of the specific sketching algorithm used and operates on the sketch states s and s' so we also overload the notation to denote the merge over states by $s \cup s'$.

In this setting, we think of the existing non-private sketch s being converted to a sketch that satisfies ϵ-DP by Algorithm 3. Since sketch s is already constructed, items cannot be first downsampled in the building phase the way they are in Algorithms 1b and 1c. To achieve the stated privacy, Algorithm 3 constructs a noisily initialized sketch, t, which satisfies both the downsampling condition (Condition (18)) and the minimum stream cardinality requirement (Condition (19)) and returns the merged sketch $s \cup t$. As formalized in Corollary 11 below, the merged sketch is guaranteed to satisfy both requirements needed for a privacy guarantee.

Algorithm 3 Turn an existing sketch into one with an ϵ-DP guarantee.

function MAKEDP(S, ϵ)
  $T, v \leftarrow DPInitSketchForMerge(\epsilon)$ ▷ Algorithm 2b
  return $S \cup T$, $\hat{N}(S \cup T) - v$
    ▷ return private sketch and associated cardinality estimate for stream S is a sketch of.

Corollary 11. *Regardless of the sketch s provided as input to the function `MakeDP` (Algorithm 3), `MakeDP` yields an ϵ-DP distinct counting sketch.*

Proof. Given sketches S, T with states s and t, respectively, we claim that any item that does not modify T also cannot modify the merged sketch $S \cup T$ by the order-invariance of S, T. To see this, let $\mathcal{D}_S$ and $\mathcal{D}_T$ respectively denote the streams that were processed by sketches S and T, and consider an item i that does not appear in $\mathcal{D}_S$ or $\mathcal{D}_T$ and whose insertion into $\mathcal{D}_T$ would not change the sketch T. Since the state of the sketch T is the same after processing $\mathcal{D}_T \circ i$ as it was after processing $\mathcal{D}_T$, $S \cup T$ is also the sketch of $\mathcal{D}_T \circ i \circ \mathcal{D}_S$, where $\circ$ denotes stream concatenation. By order-invariance, $S \cup T$ is also a sketch for $\mathcal{D}_T \circ \mathcal{D}_S \circ i$. Also by order-invariance, $S \cup T$ is a sketch for $\mathcal{D}_T \circ \mathcal{D}_S$. Hence, we have shown that the insertion of i into $\mathcal{D}_T \circ \mathcal{D}_S$ does not change the resulting sketch.

It follows that $\pi(s \cup t) \leq \pi(t) \leq \pi_0$, where the last inequality holds by the stopping condition of the loop in `DPInitSketchForMerge` (Algorithm 2b). Hence, `MakeDP` also satisfies Condition (18). The requirement that $v \geq n_0$ in `DPInitSketchForMerge` also ensures that $S \cup T$ is a sketch of a stream satisfying Condition (19). Hence, Theorem 7 implies that the sketch $S \cup T$ returned by `MakeDP` satisfies ϵ-DP. Since the additional value v that affects the estimate returned by `MakeDP` does not depend on the data, there is no additional privacy loss incurred by returning it. □

3.3 Utility

When processing a data set with n unique items, denote the expectation and variance of a sketch and its estimator by $\mathbb{E}_n(\hat{N})$ and $\mathsf{Var}_n(\hat{N})$ respectively. We show that our algorithms all yield unbiased estimates. Furthermore, we show that for Algorithms 1a-1c, if the base sketch satisfies a *relative error guarantee* (defined below), the DP sketches add no additional error asymptotically.

Establishing unbiasedness. To analyze the expectation and variance of each algorithm's estimator, $\hat{N}(S(\mathcal{D}))$, note that each estimator uses a 'base estimate' $\hat{N}_{base}$ from the base sketch S and has the form

$$\hat{N}(S(\mathcal{D})) = \frac{\hat{N}_{base}}{p} - V, \tag{22}$$

where V is the number of artificial items added and p is the downsampling probability. This allows us to express expectations and variance in terms of the variance of the base estimator.

Theorem 12. *Consider a base sketching algorithm $S \in \mathcal{C}$ with an unbiased estimator $\hat{N}_{base}$ for the cardinality of items added to the base sketch. All three algorithms 1 (a)-(c) and Algorithm 3 yield unbiased estimators.*

Proof. Trivially, Algorithm 1a is unbiased by assumption, as it does not modify the base sketch. Given V, there are $Z \sim Binomial(n+V, p)$ items added to the base sketch. Since the base sketch's estimator is unbiased, $\mathbb{E}(\hat{N}_{base}|Z) = Z$. Algorithms 1b, 1c, and Algorithm 3 all have expectation:

$$\begin{aligned}\mathbb{E}\left(\hat{N}(S_r(\mathcal{D}))|V\right) &= \mathbb{E}\left(\mathbb{E}\left(\frac{\hat{N}_{base}}{p} - V \middle| V, Z\right)\right) \\ &= \mathbb{E}\left(\frac{Z}{p} - V \middle| V\right) = n + V - V = n.\end{aligned}$$

□

Bounding the variance. First, observe that Theorem 12 yields a clean expression for the variance of our private algorithms.

Corollary 13. *The variance of the estimates produced by Algorithms 1a-1c and Algorithm 3 is given by*

$$\mathsf{Var}\left(\hat{N}(S_r(\mathcal{D}))\right) = \mathbb{E}\left(\mathsf{Var}\left(\frac{\hat{N}_{base}}{p} \middle| V\right)\right). \tag{23}$$

Proof. This follows from the law of total variance and the fact that the estimators are unbiased. □

Let us say that the base sketch *satisfies a relative-error guarantee* if with high probability, the estimate returned by the sketching algorithm when run on a stream of cardinality n is $(1 \pm 1/\sqrt{c})n$ for some constant $c > 0$. Let $\hat{N}_{base,n}$ denote the cardinality estimate when the base algorithm is run on a stream of cardinality n, as opposed to $\hat{N}_{base}$ denoting the cardinality estimate produced by the base sketch on the sub-sampled stream used in our private sketches `DPSketchLargeSet` (Algorithm 1b) and `DPSketchAnySet` (Algorithm 1c). The relative error guarantee is satisfied when $\mathsf{Var}_n(\hat{N}_{base,n}) < n^2/c$; this is an immediate consequence of Chebyshev's inequality.

When the number of artificially added items V is constant as in Algorithms 1b and 1c, Corollary 13 provides a precise expression for the variance of the differentially private sketch. In Theorem 14 below, we use this expression to establish that the modification of the base algorithm to an ϵ-DP sketch as per Algorithms 1b and 1c satisfy the exact same relative error guarantee asymptotically. In other words, the additional error due to any pre-processing (down-sampling and possibly adding artificial items) is insignificant for large cardinalities n.

Theorem 14. *Recall from Equation (19) that $n_0 = \frac{k_{max}-1}{1-e^{-\epsilon}}$. Suppose $\hat{N}_{base,n}$ satisfies a relative error guarantee $\mathsf{Var}_n(\hat{N}_{base,n}) < n^2/c$ for all n and for some constant c. Then Algorithms 1b and 1c satisfy*

$$\mathsf{Var}_n(\hat{N}) \leq \frac{(n+v)^2}{c} + \frac{(n+v)(v+\pi_0^{-1})}{k_{max}} = \frac{(n+v)^2}{c} + O(n) \tag{24}$$

$$\frac{\mathsf{Var}_n(\hat{N})}{\mathsf{Var}_n(\hat{N}_{base,n})} \to 1 \quad \textit{as } n \to \infty. \tag{25}$$

where $v = 0$ for Algorithm 1b and $v = n_0$ for Algorithm 1c.

Proof. Let $Z \sim Binomial(n+v, \pi_0)$ denote the actual number of items inserted into the base sketch. From Corollary 13 and since V is constant, the variance is

$$\begin{aligned}
\mathsf{Var}\,\hat{N}(S_r(\mathcal{D})) &= \left(\mathsf{Var}\left(\frac{\hat{N}_{base}}{\pi_0}\middle| V = v\right)\right) \\
&= \left(\frac{\mathbb{E}\,\mathsf{Var}(\hat{N}_{base}|Z) + \mathsf{Var}\,\mathbb{E}(\hat{N}_{base}|Z)}{\pi_0^2}\right) \\
&\leq \left(\frac{\mathbb{E}\,Z^2/c + \mathsf{Var}(Z)}{\pi_0^2}\right) \\
&= \frac{(\mathbb{E}\,Z)^2}{c\pi_0^2} + \frac{\mathsf{Var}(Z)(c+1)}{c\,\pi_0^2} \\
&= \frac{(n+v)^2}{c} + \frac{(n+v)(1-\pi_0)}{c\pi_0}.
\end{aligned}$$

Trivially, $\frac{\mathsf{Var}_n(\hat{N})}{\mathsf{Var}_n(\hat{N}_{base,n})} = \frac{(n+v)^2}{n^2} + O(1/n) \to 1$ as $n \to \infty$.

□

In Algorithm 3, the number of artificial items added V is a random variable. We can ensure that the algorithm satisfies a utility guarantee if we can bound V with high probability. This is equivalent to showing that the base sketching algorithm satisfies an (ϵ, δ)-DP guarantee since for any $n^* \geq n_0$ and data set $\mathcal{D}^*$ with $|\mathcal{D}^*| = n^*$, an (ϵ, δ_{n^*})-DP guarantee ensures $\delta_{n^*} > P(\pi(S_r(\mathcal{D}^*)) > \pi_0) = P(V > n^*)$ where the last equality follows from the definition of V for Algorithm 2b.

We provide (ϵ, δ)-DP results for all the specific sketching algorithms considered in Section 4.

Corollary 15. *Assume that the conditions of Theorem 14 hold. Further assume the base sketching algorithm satisfies an (ϵ, δ_n) privacy guarantee where $\delta_n \to 0$ as $n \to \infty$. For any given $n^* > n_0$, we say Algorithm 3 succeeded if $V < n^*$. Then with probability at least $1 - \delta_{n^*}$*

$$\mathsf{Var}_n(\hat{N}|Success) \leq \frac{(n+n^*)^2}{c} + \frac{(n+n^*)(n_0 + \pi_0^{-1})}{k_{max}}$$

and

$$\frac{\mathsf{Var}_n(\hat{N}|V)}{\mathsf{Var}_n(\hat{N}_{base,n})} \xrightarrow{p} 1 \quad as\ n \to \infty,$$

where the notation $X_n \xrightarrow{p} 1$ denotes convergence in probability: $P(|X_n - 1| < \Delta) \to 1$ as $n \to \infty$ for any $\Delta > 0$.

4 Example Hash-based, Order-Invariant Cardinality Estimators

Recall that the quantities of interest are the number of bins used in the sketch k, the size of the sketch in bits b and the number of items whose absence changes the sketch k_{max}. From Section 3, Lemma 4 we know that $k_{max} \leq b$ but for several common sketches delineated in this section, we can give a stronger bound showing that $k_{max} = k$. The relationship between these parameters for various sketching algorithms is summarized in Table 1.

We remind the reader that, per Equation (14), $\pi_0 = 1 - e^{-\epsilon}$, and per Equation (19)

$$n_0 = \frac{k_{max} - 1}{1 - e^{-\epsilon}}.$$

Sketch	b: size (bits)	standard error	k_{max}	$\log \delta$	reference
PCSA / FM85	$32k$	$0.649n/\sqrt{k}$	$32k$	$-\frac{\pi_0}{2k}n + o(1)$	[Lan17]
MinCount/KMV	$64k$	$n/\sqrt{k}$	k	$-\frac{1}{2}\frac{\pi_0}{1-\pi_0}n + o(1)$	[Gir09]
HLL	$5k$	$1.04\,n/\sqrt{k}$	k	$-\frac{\pi_0}{k}n + o(1)$	[FFGM07]
LPCA	k	$n/\sqrt{k}^{\dagger}$	k	$-\frac{\pi_0}{\tilde{N}(\pi_0)}n + O(\log n)$	[WVZT90]

Table 1: Properties of each sketch with k "buckets" (see each sketch's respective section for details of what this parameter means for the sketch). Each sketch provides an (ϵ, δ)-DP guarantee, where the column $\log \delta$ provides an upper bound on the natural logarithm of δ established in the relevant subsection of Section 4. The † symbol denotes an approximation that holds for $n < k$. A better approximation of the standard error is $\sqrt{k(\exp(n/k) - n/k - 1)}$.

Furthermore, recall that once we bound the parameter k_{max} for any given hash-based order-invariant sketching algorithm, Corollary 10 states that the derived algorithms `DPSketchLargeSet` (Algorithm 1b) and `DPSketchAnySet` (Algorithm 1c) satisfy ϵ-DP provided that $n \geq n_0$ and $n \geq 1$, respectively. Moreover, if the original sketch returns an unbiased estimate, so too does the derived private sketch. Additionally, Theorem 14 bounds the variance of the estimate returned by the derived private sketch in terms of the variance of the original sketch's estimate. Accordingly, in the rest of this section, we bound k_{max} for each example sketch of interest, which has the consequences for pure ϵ-differential privacy delineated above.

4.1 Probabilistic Counting / Flajolet-Martin 85

The Probabilistic Counting with Stochastic Averaging (PCSA) or Flajolet-Martin '85 (Flajolet-Martin (FM85)) sketch consists of k bitmaps B_i of length ℓ. Each item is hashed into a bitmap and index (B_i, G_i) and sets the indexed bit in the bitmap to 1. The chosen bitmap is uniform amongst the k bitmaps and the index $G_i \sim \mathit{Geometric}(1/2)$. If ℓ is the length of each bitmap, then the total number of bits used by the sketch is $b = k\ell$ and $k_{max} = k\ell$ for all seeds r. A typical value for ℓ is 32 bits, as used in Table 1.

Past work [vVT19] proposed an ϵ differentially private version of PCSA using a similar subsampling idea but combined it with random bit flips.

Theorem 16. *Let $v = \lceil -\log_2 \pi_0 \rceil$ and $\tilde{\pi}_0 := 2^{-v} \in (\pi_0/2, \pi_0]$. If $n \geq n_0$, then the FM85 sketch is (ϵ, δ)-DP with $\delta \leq kv \exp\left(-\tilde{\pi}_0 \frac{n}{k}\right)$.*

Proof. To obtain an (ϵ, δ) guarantee, note that bit s_{ij} in the sketch has probability $2^{-i}/k$ of being selected by any item. If $v = \lceil -\log_2 \pi_0 \rceil$ and all bits s_{ij} with $j \leq v$ are flipped, then $\pi(s) < \pi_0$. The probability $\Pr_r(s_{ij} = 0) = (1 - 2^{-i}/k)^n \leq \exp(-2^{-i}n/k)$. A union bound gives that $\Pr_r(\pi(S_r(\mathcal{D})) \geq \pi_0) \leq k \sum_{i=1}^{v} \exp(-2^{-i}n/k) \leq kv \exp(-2^{-v}n/k) = kv \exp\left(-\tilde{\pi}_0 \frac{n}{k}\right)$ where $\tilde{\pi}_0 = 2^{-v} \leq \pi_0$. □

Relative to the sketches discussed in the coming sections, for any given value of k, a larger number of minimum items n_0 is needed to ensure this sketch is differentially private, because the constant factor $k_{max} \in \{32k, 64k\}$ is worse than that for the later sketches we study (all of which have $k_{max} = k$). However, the sketch is highly compressible as, for large n, each bitmap has entropy of approximately 4.7 bits [Lan17]. More recent works have placed this numerical result on firmer theoretical footing [PW21], and in fact shown that the resulting space-vs.-error tradeoff is essentially *optimal* amongst a large class of sketching algorithms. A practical implementation of the compressed sketch can be found in the Apache DataSketches library [dat], where it is referred to as CPC, short for compressed probabilistic counting. It achieves close to constant update time by buffering stream elements and only decompressing the sketch when the buffer is full.

Our results imply the above compressed sketches can yield a relaxed (ϵ, δ_n)-differential privacy guarantee when the number of inserted items is $n < n_0$ (Equation (21)). If the size of the sketch in bits is b, the sketch is ϵ-differentially private if $n > \frac{b-1}{1-\exp(-\epsilon)}$ or equivalently $b < n(1 - \exp(-\epsilon)) + 1$. Thus, $\delta_n = \Pr_r(b \geq n(1 - \exp(-\epsilon) + 1)$.

4.2 MinCount/KMV/Bottom-k

The KMV sketch stores the k smallest hash values. Removing an item changes one of these values if and only if 1) the item's hash value is one of these k and 2) it does not collide with another item's hash value. Thus, $k_{max} = k$. Typically, the output size of the hash function is large enough to ensure that the probability of collision is negligible, and for all practical purposes $K_r = k$ exactly.

For an (ϵ, δ)-DP guarantee, the generic coupon collecting strategy yields a loose bound for δ. We provide a tighter analysis here.

Theorem 17. *Consider KMV with k minimum values. Given $\epsilon > 0$, let π_0, n_0 be the corresponding subsampling and minimum cardinality to ensure the modified KMV sketch is $(\epsilon, 0)$-DP. When run on streams of cardinality $n \geq n_0$, then the unmodified sketch is (ϵ, δ)-DP, where $\delta = P(X \leq k) < \exp(-n\alpha_n)$ where $X \sim Binomial(n, \pi_0)$ and $\alpha_n = \frac{1}{2}\frac{(\pi_0 - k/n)^2}{\pi_0(1-\pi_0)+1/3n^2} \to \frac{1}{2}\frac{\pi_0}{1-\pi_0}$ as $n \to \infty$.*

Proof. The value $\pi(s)$ is equal to the k^{th} minimum value. If $X > k$, then the k^{th} minimum value is $< \pi_0$ and Condition 18 is satisfied. Thus, δ is the failure probability. The bound follows directly from Bernstein's inequality:

$$P(X \leq k) = P(n - X > n - k) = P((n - X) - n(1 - \pi_0) > n\pi_0 - k) \tag{26}$$

$$\leq \frac{1}{2}\frac{(n\pi_0 - k)^2}{n\pi_0(1-\pi_0) + 1/3} \tag{27}$$

$$= \frac{1}{2}\frac{(\pi_0 - k/n)^2}{\pi_0(1-\pi_0) + 1/3n^2} n \tag{28}$$

□

Since the KMV estimator $\hat{N}(s) = (k-1)/\pi(s)$ is a function of the update probability $\pi(s)$, Theorem 9 gives an (ϵ, δ)-DP guarantee in terms of the cardinality estimate.

4.3 HyperLogLog

HyperLogLog hashes each item to a bin and value (B_i, G_i). Within each bin, it takes the maximum value. Thus, each bin can be regarded as a form of bottom-1 sketch. If there are k bins, then $k_{max} = k$.

Our results uniformly improve upon existing DP results on the HLL sketch and its variants. Choi et al. [CDSKY20] provide an (ϵ, δ) guarantee for streams of cardinality $n \geq n_0'$, for an n_0' that is larger than our n_0 by a factor of roughly (at least) 8, with δ falling exponentially with n. In contrast, for streams with cardinality $n > n_0$, we provide a *pure* ϵ-DP guarantee.

A variation of the HLL sketch was studied by [SSGT20] and was also shown to provide an ϵ-DP guarantee. However, the variant analyzed is far slower than HLL itself, as it requires every item to be independently hashed k times, once for each of the k bins, rather than just one time. In other words, [SSGT20] needs $O(k)$ update time compared to $O(1)$ for our algorithms.

HLL also has the following (ϵ, δ) guarantee.

Theorem 18. *If $n > n_0$, then HLL satisfies an (ϵ, δ)-DP guarantee where $\delta \leq k \exp(-\pi_0 n/k)$*

Proof. In HLL, the sampling probability $\pi(s) = k^{-1}\sum_{i=1}^{k} 2^{-s_i}$ here s_i is the value in each bin. Thus, if all bins have value $s_i > -\log_2 \pi_0$, then $\pi(s) < \pi_0$. Let C_i be the event that $s_i > -\log_2 \pi_0$. Then $P(\neg C_i | n) \leq (1 - \pi_0/k)^n \leq \exp(-\pi_0 n/k)$. A union bound gives $\Pr_r(\pi(S_r(\mathcal{D})) \geq \pi_0) \leq k \exp(-\pi_0 n/k)$. □

HLL's estimator is only a function of the sampling probability $\pi(s)$ for medium to large cardinalities. That is, the estimator has the form $\hat{N}(s) = \tilde{N}(\pi(s))$ when $\tilde{N}(\pi(s)) > 5k/2$. Thus, if π_0 is sufficiently small so that $\tilde{N}(\pi_0(s)) > 5k/2$, then Theorem 9 can still be applied, and HLL satisfies an (ϵ, δ) guarantee with $\delta = P(\hat{N}(S_r(\mathcal{D})) < \tilde{N}(\pi_0))$.

4.4 Linear Probabilistic Counting

The Linear Probabilistic Counting Algorithm (LPCA) is equivalent to a Bloom filter where the number of hashes is 1. Thus, the sketch is a length-k bitmap. Each item is hashed to an index and sets the corresponding bit to 1. If B is the number of bits set to 1, the LPCA cardinality estimate is $\hat{N}_{\texttt{LPCA}} = -k\log(1-B/k) = k\log\pi(S_r(\mathcal{D}))$. Trivially, $k_{max} = k$.

Since all bits are expected to be set to 1 after processing roughly $k\log k$ distinct items, the capacity of the sketch is bounded. To estimate larger cardinalities, typically one first downsamples the distinct items with some sampling probability p. To ensure the sketch satisfies an ϵ-DP guarantee, one simply ensures $p \geq \pi_0$. In this case, our analysis shows that LPCA is differentially private with no modifications. Otherwise, since the estimator $\hat{N}(s)$ is a function of the sampling probability $\pi(s)$, Theorem 9 provides an (ϵ, δ) guarantee in terms of $\hat{N}$.

We also provide a guarantee in terms of the true cardinality n.

Theorem 19. *Consider an LPCA sketch with k bits and downsampling probability p, and assume that $n > \frac{k-1}{1-e^{-\epsilon}}$ so that Condition 19 is satisfied. If $p < \pi_0$ then LPCA is ϵ-DP. Otherwise, let $b_0 = \lceil k(1-\pi_0/p)\rceil$, $\tilde{\pi}_0 = b_0/k$, and μ_0 be the expected number of items inserted to fill b_0 bits in the sketch. Then, if $n > \mu_0$, LPCA is (ϵ, δ)-DP with*

$$\delta = P(B < b_0) \tag{29}$$

$$< \frac{\mu_0}{n}\exp\left(-\frac{\tilde{\pi}_0}{\mu_0}n\right)\exp(-\tilde{\pi}_0) \tag{30}$$

where B is the number of filled bits in the sketch. Furthermore,

$$\mu_0 < \tilde{N}(\tilde{\pi}_0)$$

where $\tilde{N}(\tilde{\pi}) = -\frac{k}{p}\log(1-\tilde{\pi})$ is the cardinality estimate of the sketch when the sampling probability is $\tilde{\pi}$.

Proof. $S_r(D)$ is not privacy violating (i.e., $\pi(s) < \pi_0$) if $\pi(S_r(\mathcal{D})) = p(1-B/k) < \pi_0$ or, equivalently, $B > k(1-\pi_0/p)$. Note that $G_i \sim Geometric(p(1-i/k))$ items must be added for the number of filled bits to go from i to $i+1$.

We can use a tail bound for the sum of geometric random variables [Jan18]. Assume that $n \geq \frac{k-1}{1-\exp(-\epsilon)} \geq n_0$ so that Condition 19 is satisfied. If $n > \mu_0$ then

$$\delta \leq P\left(\sum_{i=0}^{b_0} G_i > n\right) \tag{31}$$

$$\leq \exp\left(-\tilde{\pi}_0(n/\mu_0 - 1 - \log(n/\mu_0))\right) \tag{32}$$

$$= \frac{\mu_0}{n}\exp\left(-\tilde{\pi}_0(n/\mu_0 - 1)\right). \tag{33}$$

$$\tag{34}$$

The number of expected items needed to fill b_0 bits if $b_0 \geq 1$ is

$$\mu_0 := \sum_{i=0}^{b_0-1} \frac{1}{p} \frac{1}{1-i/k} \tag{35}$$

$$< \frac{k}{p}\left(\log\left(\frac{k}{k-b_0}\right) + 1/(2k) - 1/(2(k-b_0)) + \frac{1}{12(k-b_0)^2}\right) \tag{36}$$

$$= \frac{k}{p}\log\left(\frac{k}{k-b_0}\right) - \frac{1}{p}\frac{b_0}{2(k-b_0)} + \frac{1}{p}\frac{k}{12(k-b_0)^2} \tag{37}$$

$$= \frac{k}{p}\log\left(\frac{k}{k-b_0}\right) - \frac{1}{p}\frac{6b_0(k-b_0)-k}{12(k-b_0)^2} \tag{38}$$

$$< \frac{k}{p}\log\left(\frac{k}{k-b_0}\right). \tag{39}$$

$$\tag{40}$$

□

4.5 Adaptive Sampling

Wegman's adaptive sampling is similar to the bottom-k sketch but does not require the sketch to store exactly k hashes. Instead, it maintains a threshold p and stores all hash values less than p. Whenever the sketch size exceeds k, then the threshold is cut in half and only values under the threshold are retained. This ensures that processing a stream of length n takes expected $O(n)$ time rather than $O(n \log k)$ as in a max-heap-based implementation of Bottom-k.

It is order invariant since the sketch only depends on the number of hash values under each of the potential thresholds and not the insertion order. Since at most k hashes are stored, and the sketch is modified only if one of these hashes is removed, like KMV a.k.a. Bottom-k, the maximum number of items that can modify the sketch by removal is $k_{max} = k$.

Corollary 20. *For any size k and cardinality n, if a size k KMV sketch is (ϵ, δ)-DP, then a maximum size k adaptive sampling sketch is (ϵ, δ)-DP with the same ϵ and δ.*

Proof. Consider sketches $S_r^{AT}(\mathcal{D})$, $S_r^{KMV}(\mathcal{D})$ using the same hash function. Since the threshold in adaptive sampling is at most the k^{th} minimum value, $\pi(S_r^{AT}(\mathcal{D})) \leq \pi(S_r^{KMV}(\mathcal{D}))$. So $\pi(S_r^{KMV}(\mathcal{D})) < \pi_0 \implies \pi(S_r^{AT}(\mathcal{D})) < \pi_0$. □

5 Conclusion

We have studied the (differential) privacy of a class of cardinality estimation sketches that includes most popular algorithms. Two examples are the HLL and KMV (bottom-k) sketches that have been deployed in large systems [HNH13, dat]. We have shown that the sketches returned by these algorithms are ϵ-differentially private when run on streams of cardinality greater than $n_0 = \frac{k_{max}-1}{1-e^{-\epsilon}}$ and when combined with a simple downsampling procedure. Moreover, even without downsampling, these algorithms satisfy (ϵ, δ)-differential privacy where δ falls exponentially with the stream cardinality n once n is larger than the threshold n_0. Our results are more general and yield better privacy guarantees than prior work for small space cardinality estimators that preserve differential privacy. Future work will experimentally validate the accuracy of our algorithms.

Acknowledgements. We are grateful to Graham Cormode for valuable comments on an earlier version of this manuscript. Justin Thaler was supported by NSF SPX award CCF-1918989 and NSF CAREER award CCF-1845125.

References

[BGH+09] Kevin Beyer, Rainer Gemulla, Peter J Haas, Berthold Reinwald, and Yannis Sismanis. Distinct-value synopses for multiset operations. *Communications of the ACM*, 52(10):87–95, 2009.

[BYJK+02] Ziv Bar-Yossef, TS Jayram, Ravi Kumar, D Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 1–10. Springer, 2002.

[CDSKY20] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. Differentially-private multi-party sketching for large-scale statistics. *Proceedings of Privacy Enhancing Technologies*, 2020(3):153–174, 2020.

[CK07] Edith Cohen and Haim Kaplan. Summarizing data using bottom-k sketches. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 225–234, 2007.

[dat] Apache DataSketches. `https://datasketches.apache.org/`.

[DLB19] Damien Desfontaines, Andreas Lochbihler, and David Basin. Cardinality estimators do not preserve privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(2):26–46, 2019.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[EVF03] Cristian Estan, George Varghese, and Mike Fisk. Bitmap algorithms for counting active flows on high speed links. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 153–166, 2003.

[FFGM07] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science*, pages 137–156. Discrete Mathematics and Theoretical Computer Science, 2007.

[Fla90] Philippe Flajolet. On adaptive sampling. *Computing*, 43(4):391–400, 1990.

[FM85] Philippe Flajolet and G Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31(2):182–209, 1985.

[Gir09] Frédéric Giroire. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2):406–427, 2009.

[HNH13] Stefan Heule, Marc Nunkesser, and Alexander Hall. Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 683–692, 2013.

[Jan18] Svante Janson. Tail bounds for sums of geometric and exponential variables. *Statistics & Probability Letters*, 135:1–6, 2018.

[Lan17] Kevin J Lang. Back to the future: an even more nearly optimal cardinality estimation algorithm. *arXiv preprint arXiv:1708.06839*, 2017.

[MMNW11] Darakhshan Mir, Shan Muthukrishnan, Aleksandar Nikolov, and Rebecca N Wright. Pan-private algorithms via statistics on sketches. *PODS*, 2011.

[PS21] Rasmus Pagh and Nina Mesing Stausholm. Efficient differentially private F_0 linear sketching. In *24th International Conference on Database Theory (ICDT 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[PW21] Seth Pettie and Dingyu Wang. Information theoretic limits of cardinality estimation: Fisher meets shannon. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 556–569, 2021.

[SAC+79] P. Griffiths Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*, SIGMOD '79, page 23–34, New York, NY, USA, 1979. Association for Computing Machinery.

[SSGT20] Adam Smith, Shuang Song, and Abhradeep Guha Thakurta. The Flajolet-Martin sketch itself preserves differential privacy: Private counting with minimal space. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19561–19572. Curran Associates, Inc., 2020.

[TS13] Florian Tschorsch and Björn Scheuermann. An algorithm for privacy-preserving distributed user statistics. *Computer Networks*, 57(14):2775–2787, 2013.

[vVT19] Saskia Nuñez von Voigt and Florian Tschorsch. RRTxFM: Probabilistic counting for differentially private statistics. In *Conference on e-Business, e-Services and e-Society*, pages 86–98. Springer, 2019.

[WVZT90] Kyu-Young Whang, Brad T Vander-Zanden, and Howard M Taylor. A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems (TODS)*, 15(2):208–229, 1990.