Bounded Space Differentially Private Quantiles

Daniel Alabi
Harvard University
alabid@g.harvard.edu

Omri Ben-Eliezer
Massachusetts Institute of Technology
omrib@mit.edu

Anamay Chaturvedi
Northeastern University
chaturvedi.a@northeastern.edu

ABSTRACT

Estimating the quantiles of a large dataset is a fundamental problem in both the streaming algorithms literature and the differential privacy literature. However, all existing private mechanisms for distribution-independent quantile computation require space at least linear in the input size n. In this work, we devise a differentially private algorithm for the quantile estimation problem with strongly sublinear space complexity. Our main mechanism estimates any α-approximate quantile of a length-n stream over a data universe $\mathcal{X}$ with probability $1-\beta$ using $O\left(\frac{\log(|\mathcal{X}|/\beta)\log(\alpha\epsilon n)}{\alpha\epsilon}\right)$ space while satisfying ϵ-differential privacy. Our approach builds upon deterministic streaming algorithms for non-private quantile estimation instantiating the exponential mechanism using a utility function defined on sketch items, while (privately) sampling from intervals defined by the sketch. We also present another algorithm based on histograms that is especially suited to the multiple quantiles case. We implement our algorithms and experimentally evaluate them on synthetic and real-world datasets.

1 INTRODUCTION

Quantile estimation is a fundamental subroutine in data analysis and statistics. For $q \in [0,1]$, the q-quantile in a dataset of size n is the element ranked $\lceil qn \rceil$ when the elements are sorted from smallest to largest. Computing a small number of quantiles in a possibly huge collection of data elements can serve as a quick and effective sketch of the "shape" of the data. Quantile estimation also serves an essential role in robust statistics, where data is generated from some distribution but is contaminated by a non-negligible fraction of outliers, i.e., "out of distribution" elements that may sometimes even be adversarial. For example, the median (50th percentile) of a dataset is used as a robust estimator of the mean in such situations where the data may be contaminated. Location parameters can also be (robustly) estimated via truncation or winsorization, an operation that relies on quantile estimation as a subroutine [28, 49]. Rank-based nonparametric statistics can be used for hypothesis testing (e.g., the Kruskal-Wallis test statistic [34]). Thus, designing quantile-based or rank-based estimators, whether distribution-dependent or distribution-agnostic, is important in many scenarios.

Maintaining the privacy of individual users or data items, or even of groups, is an essential prerequisite in many modern data analysis and management systems. Differential privacy (DP) is a rigorous and now well-accepted definition of privacy for data analysis and machine learning. In particular, there is already a substantial amount of literature on differentially private quantile estimation (e.g., see [4, 24, 44, 50]).[1]

All of these previous works, however, either make certain distributional assumptions about the input, or assume the ability to access all input elements (thus virtually requiring a linear or worse space complexity). Such assumptions may be infeasible in many practical scenarios, where large scale databases have to quickly process streams of millions or billions of data elements without clear a priori distributional characteristics.

The field of streaming algorithms aims to provide space-efficient algorithms for data analysis tasks. These algorithms typically maintain good accuracy and fast running time while having space requirements that are substantially smaller than the size of the data. While distribution-agnostic quantile estimation is among the most fundamental problems in the streaming literature [2, 22, 25, 29, 31, 38, 43, 46, 52], no differentially private sublinear-space algorithms for the same task are currently known. Thus, the following question, essentially posed by [47] and [40], naturally arises:

> *Can we design differentially private quantile estimators that use space sublinear in the stream length, have efficient running time, provide high-enough utility, and do not rely on restrictive distributional assumptions?*

It is well known [43] that exact computation of quantiles cannot be done with sublinear space, even where there are no privacy considerations. Thus, one must resort to approximation. Specifically, for a dataset of n elements, an *α-approximate q-quantile* is any element which has rank $(q \pm \alpha)n$ when sorting the elements from smallest to largest, and it is known that the space complexity of α-approximating quantiles is $\tilde{\Omega}(1/\alpha)$ [43]. In our case, the general goal is to efficiently compute α-approximate quantiles in a (pure or approximate) differentially private manner.

1.1 Our Contributions

We answer the above question affirmatively by providing theoretically proven algorithms with accompanying experimental validation for quantile estimation with DP guarantees. The algorithms are suitable for private computation of either a single quantile or multiple quantiles. Concretely, the main contributions are:

(1) We devise `DPExpGK`, a differentially private sublinear-space algorithm for quantile estimation based on the exponential mechanism. In order to achieve sublinear space complexity, our algorithm carefully instantiates the exponential mechanism with the basic blocks being intervals from the

[1] Robust estimators are also known to be useful for accurate differentially private estimation; see, e.g., the work of Dwork and Lei [19] in the context of quantile estimation for the interquartile range and for medians.

Greenwald-Khanna [25] data structure for non-private quantile estimation, rather than single elements. We prove general distribution-agnostic utility bounds on our algorithm and show that the space complexity is logarithmic in n.

(2) We present DPHistGK, another differentially private mechanism for quantile estimation, which applies the Laplace mechanism to a histogram, again using intervals of the GK-sketch as the basic building block. We theoretically demonstrate that DPHistGK may be useful in cases where one has prior knowledge on the input.

(3) We empirically validate our results by evaluating DPExpGK, analyzing and comparing various aspects of performance on real-world and synthetic datasets.

DPExpGK can be interpreted as a more “general purpose” solution that performs well unconditionally of the data characteristics. It is especially suitable for the single quantile problem, and can be adapted to multiple quantiles by splitting the privacy budget and applying standard composition theorems in differential privacy. On the other hand, DPHistGK inherently solves the all-quantiles problem and may be suitable when there are not too many bins (small set of possible values), the variance of the target distribution is small, or the approximation parameter (i.e., α) is large.

2 RELATED WORK

2.1 Quantile Approximation of Streams and Sketches

Approximation of quantiles in large data streams (without privacy guarantees) is among the most well-investigated problems in the streaming literature [27, 52, 53]. A classical result of Munro and Paterson from 1980 [43] shows that computing the median exactly with p passes over a stream requires $\Omega(n^{1/p})$ space, thus implying the need for approximation to obtain very efficient (that is, at most polylogarithmic) space complexity. Manku, Rajagopalan and Lindsay [38] built on ideas from [43] to obtain a randomized algorithm with only $O((1/\alpha)\log^2(n\alpha))$ for α-approximating all quantiles; a deterministic variant of their approach with the same space complexity exists as well [2]. The best known deterministic algorithm is that of Greenwald and Khanna (GK) [25] on which we build on in this paper, with a space complexity of $O(\alpha^{-1}\log(\alpha n))$ to sketch all quantiles for n elements (up to rank approximation error of $\pm\alpha n$). A recent deterministic lower bound of Cormode and Veselý [16] (improving on an earlier lower bound of [29]) shows that the GK algorithm is in fact optimal among deterministic (comparison-based) sketches.

Randomization and sampling help for streaming quantiles, and in particular the space complexity is independent of n; an optimal $O((1/\alpha)\log\log(1/\beta))$-space algorithm was devised by Karnin, Lang and Liberty [31] (for failure probability β), concluding a series of work devoted to improving the randomized space complexity [2, 22, 37, 38].

The problem of biased or relative error quantiles, where one is interested in increased approximation accuracy for very small or very large quantiles, has also been investigated [14, 15]; it would be interesting to devise efficient differentially private algorithms for this problem.

Recall that our approach is based on the Greenwald-Khanna deterministic all-quantiles sketch [25]. While some of the aforementioned randomized algorithms have a slightly better space complexity, differential privacy mechanisms are inherently randomized by themselves, and the analysis seems somewhat simpler and more intuitive when combined with a deterministic sketch. This, of course, does not rule out improved private algorithms based on modern efficient randomized sketches (see Section 7).

2.2 Differential Privacy

Differentially private single quantile estimation: In the absence of distributional assumptions, in work by Nissim, Raskhodnikova and Smith [44] and Asi and Duchi [4], the trade-off between accuracy and privacy is improved by scaling the noise added for obfuscation in an instance-specific manner for median estimation. Another work by Dwork and Lei [19] uses a “propose-test-release" paradigm to take advantage of local sensitivity; however, as observed in [24], in practice the error incurred by this method is relatively large as compared to other works like [50]. The work [50] achieves the optimal trade-off between privacy and utility in the distributional setting, but again as observed by [24], with a time complexity of $O(n^4)$, this method does not scale well to large data sets.

A very recent work of Gillenwater et al. [24] shows how to optimize the division of the privacy budget to estimate m quantiles in a time-efficient manner. For estimation of m quantiles, their time and space complexity are $O(mn\log(n) + m^2 n)$ and $O(m^2 n)$, respectively. They do an extensive experimental analysis and find lower error compared to previous work. However, although they provide intuition for why their method should incur relatively low error, they do not achieve formal theoretical accuracy guarantees. Note that their results substantially differ from ours, as they hold for private *exact* quantiles and so cannot achieve sublinear space.

Differentially private statistical estimation: Estimation of global data statistics (or more generally, inference) is in general an important use-case of differential privacy [32]. The related private histogram release problem was studied by [5, 6, 10, 36] and in the continual observation privacy setting [11, 12, 21]. Perrier et al. [45] make improvements over prior work for private release of some statistics such as the moving average of a stream when the data distribution is light-tailed. They use private quantile estimation of the input stream as a sub-routine to achieve their improvements. Böhler and Kerschbaum [8] solve the problem of estimating the joint median of two private data sets with time complexity sub-linear in the size of the data-universe and provide privacy guarantees for small data sets as well as limited group privacy guarantees unconditionally against polynomially time-bounded adversaries.

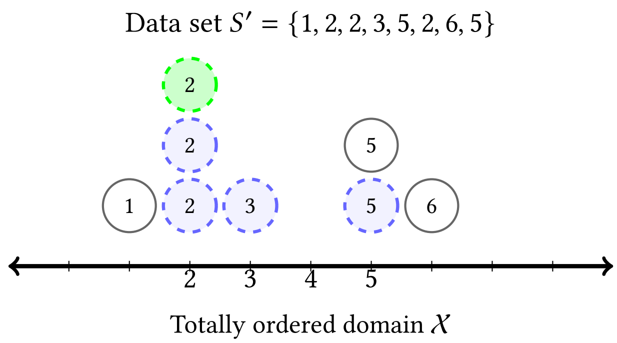


Figure 1: In the data set S' of 8 elements, the true 0.5 quantile is the value 2, and the values 2, 3, 4 and 5 are all acceptable 0.25-approximate answers. Note that although 1 is adjacent to 2 in the data universe $\mathcal{X}$, it is not an acceptable output.

Inherent privacy: Another line of work [7, 13, 48] demonstrates that sketching algorithms for streaming problems might have inherent privacy guarantees under minimal assumptions on the dataset in some cases. For such algorithms, relatively little noise needs to be added to preserve privacy unconditionally.

Robust estimators: A substantial number of works (e.g., [3, 19, 44, 45, 47]) in the differential privacy literature use robust estimators to design statistical inference estimators. However, most estimators use space and time that is polynomial in the length of the input. For very large datasets, the computational resources required by these estimators may be prohibitive, motivating the need for methods to improve the space and time efficiency.

3 PRELIMINARIES AND NOTATION

In this section we give standard differential privacy notation and formally describe the quantile estimation problem. We also present the Greenwald-Khanna sketch guarantees in a form that is adapted for our use.

3.1 Differential Privacy

Definition 3.1 (Differential Privacy [20]). Let $Q : \mathcal{X}^n \to \mathcal{R}$ be a (randomized) mechanism. For any $\epsilon \geq 0, \delta \in [0, 1]$, Q satisfies (ϵ, δ)-**differential privacy** if for any neighboring databases $\mathbf{x} \sim \mathbf{x}' \in \mathcal{X}^n$ and any $S \subseteq \mathcal{R}$,

$$\mathbb{P}[Q(\mathbf{x}) \in S] \leq e^{\epsilon}\mathbb{P}[Q(\mathbf{x}') \in S] + \delta.$$

The probability is taken over the coin tosses of Q. We say that Q satisfies pure differential privacy (ϵ-DP) if $\delta = 0$ and approximate differential privacy ((ϵ, δ)-DP) if $\delta > 0$. We can set ϵ to be a small constant (e.g., between 0.01 and 2) but will require that $\delta \leq n^{-\omega(1)}$ be cryptographically small. Stability-based histograms can be used to obtain approximate DP guarantees. However, most of our results apply only to the pure DP setting.

3.2 Quantile Approximation

Definition 3.2. Let $X = ((x_1, 1), \dots, (x_n, n))$ (sometimes implicitly referred to as $X = (x_1, \dots, x_n)$) be a stream of elements drawn from some finite totally ordered data universe $\mathcal{X}$, i.e. $x_i \in \mathcal{X}$ for all $i \in [n]$.

(1) For $(x_i, i) \in X$, let $\mathrm{val}((x_i, i)) = x_i$ and $\mathrm{ix}((x_i, i)) = \sum_{j \leq i} |\{(x_j, j) : x_j < x_i \text{ or } x_j = x_i, j < i\}|$.
(2) For $v_1, v_2 \in X$, we say that $v_1 \leq v_2$ if $\mathrm{ix}(v_1) \leq \mathrm{ix}(v_2)$.
(3) The q-quantile of X is $\mathrm{val}(v)$ for $v \in X$ such that $\mathrm{ix}(v) = \lceil qn \rceil$.
(4) For $x \in \mathcal{X}$, we define $r_{\min}(x) = |\{v \in X : \mathrm{val}(v) < x\}|$, $r_{\max}(x) = |\{v \in X : \mathrm{val}(v) \leq x\}|$ and $\mathrm{rank}(X, x)$ to be the interval $[r_{\min}(x), r_{\max}(x)]$.
(5) We say that $x \in \mathcal{X}$ is an α-approximate q-quantile for X if $\mathrm{rank}(X, x) \cap [\lceil qn \rceil - \alpha n, \lceil qn \rceil + \alpha n] \neq \emptyset$.

With this notation, the data set X is naturally identified as a multiset of elements drawn from $\mathcal{X}$.

Example 3.3. Given the data set $\{1, 2, 2, 3, 5, 2, 6, 5\}$ (refer to figure 1), the 0.5 quantile is 2, which we distinguish from the median (which would be the average of the elements ranked 4 and 5, i,e, 2.5 for this data set. A straightforward way to obtain quantiles is to sort the dataset and the pick the element at the $\lceil q \cdot n \rceil$ position. This method only works in the offline (non-streaming) setting. For $\alpha = 0.25$, the α-approximate 0.5 quantiles are 2, 3, 4 and 5. Note that 4, which does not occur in the data set, is still a valid response, but 1, which occurs in the data set and is even adjacent to the true 0.5 quantile 2 in the data universe $\mathcal{X}$, is not a valid response.

Definition 3.4 (Single Quantile). Given sample $S = (x_1, \dots, x_n)$ in a streaming fashion in arbitrary order, construct a data structure for computing the quantile $Q^q_{\mathcal{D}}$ such that for any $q \in (0, 1)$, with probability at least $1 - \beta$,

$$|Q^q_{\mathcal{D}} - \tilde{Q}^q_S| \leq \alpha.$$

Definition 3.5 (All Quantiles). Given sample $S = (x_1, \dots, x_n)$ in a streaming fashion in arbitrary order, construct a data structure for computing the quantile $Q^q_{\mathcal{D}}$ such that with probability at least $1 - \beta$, for all values of $q \in M$ where $M \subset (0, 1)$,

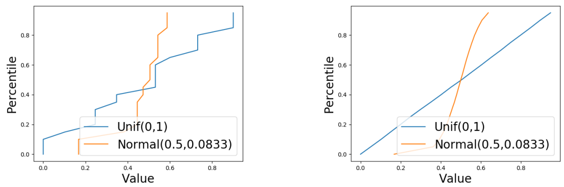
$$|Q^q_{\mathcal{D}} - \tilde{Q}^q_S| \leq \alpha.$$

In Figures 2a and 2b, we show the performance of the (non-private) Greenwald-Khanna sketch. As can be seen in Figure 2a, for higher α (corresponding to bigger approximation and smaller space), the sketch is less accurate than for lower α (smaller approximation and larger space) (Figure 2b). In the private algorithms, the loss in approximation must also be incurred.

3.3 Non-private Quantile Streaming

Lemma 3.6 (GK guarantees). *Given a data stream $x_1, \dots, x_n$ of elements drawn from $\mathcal{X}$, the GK sketch outputs a list $S(X)$ of $s(= O((1/\alpha)\log(\alpha n)))$ many tuples $(v_i, g_i, \Delta_i) \in (\mathcal{X} \times \mathbb{N}) \times \mathbb{N} \times \mathbb{N}$ for $i = 1, \dots, s$ such that if X is the data multi-set then*

(1) $\mathrm{rank}(X, \mathrm{val}(v_i)) \subset [\sum_{j \leq i} g_j, \Delta_i + \sum_{j \leq i} g_j]$.



(a) (Non-Private) α-Approximate Quantiles for $\alpha = 0.1$ and $q \in (0, 1)$.

(b) (Non-Private) α-Approximate Quantiles for $\alpha = 0.0001$ and $q \in (0, 1)$.

Figure 2

(2) $g_i + \Delta_i \leq 2\alpha n$.
(3) *The first tuple is* $(\min\{x \in X\}, 1, 0)$ *and the last tuple is* $(\max\{x \in X\}, 1, 0)$.
(4) *The* v_i *are sorted in ascending order. WLOG, the lower confidence interval bounds* $\sum_{j \leq i} g_j$ *and upper confidence interval bounds* $\Delta_i + \sum_{j \leq i} g_j$ *are respectively also sorted in increasing order.*

Proof. The first three statements are part of the GK sketch guarantee. For the third statement, i.e. to see that the v_i are sorted in ascending order, we see that the GK sketch construction ensures that $\mathrm{val}(v_i) \leq \mathrm{val}(v_{i+1})$ for all i. Since the INSERT operation always inserts a repeated value after all previous occurrences and the tuple order is always preserved, it follows that $\mathrm{ix}(v_i) \leq \mathrm{ix}(v_{i+1})$ as well, so in sum $\mathrm{rank}(X, v_i) \leq \mathrm{rank}(X, v_{i+1})$. In other words, the sort order in the GK sketch is *stable*.

The fact that the sequence $\sum_{j \leq i} g_j$ is sorted in increasing order follows from the non-negativity of the g_i. To ensure that $\Delta_i + \sum_{j \leq i} g_j$ are sorted in increasing order note that we always have that $\mathrm{rank}(X, v_i) \leq \mathrm{rank}(X, v_{i+1})$ so we can decrement Δ_i and ensure that $\Delta_i + \sum_{j \leq i} g_j \leq \Delta_{i+1} + \sum_{j \leq i+1} g_j$ without violating the guarantees of the GK sketch. □

Remark 3.7. We can add two additional tuples $(-\infty, 0, 0)$ and $(\infty, 0, 0)$ to the sketch, which corresponds to respective rank intervals $[0, 0]$ and $[n + 1, n + 1]$. The bounds $g_i + \Delta_i \leq 2\alpha n$ are preserved. This will ensure that the sets $\{i : \mathrm{val}(v_i) < x\}$ and $\{i : \mathrm{val}(v_i) > x\}$ for any $x \in \mathcal{X}$ are always non-empty.

4 DIFFERENTIALLY PRIVATE ALGORITHMS

In this section we present our two DP mechanisms for quantile estimation. Throughout, we assume that α is a user-defined approximation parameter. The goal is to obtain α-approximate q-quantiles.

The new algorithms we introduce are:

(1) `DPExpGKGumb` (Algorithm 2): An exponential mechanism based $(\epsilon, 0)$-DP algorithm for computing a single q-quantile. To solve the all-quantiles problem with approximation factor α, one can run this algorithm iteratively with target quantile $0, \alpha, 2\alpha, \ldots$. Doing so requires scaling the privacy parameter in each call by an additional α factor which increases the sample complexity by a factor of $1/\alpha$.
(2) `DPHistGK`: A histogram based $(\epsilon, 0)$-DP algorithm (Algorithm 3) for the α-approximate all-quantiles problem. The privacy guarantee of this algorithm is unconditional, but there is no universal theoretical utility bound as in the previous algorithm. However, in some cases the utility is provably better: for example, we show that if the data set is drawn from a normal distribution (with unknown mean and variance), we can avoid the quadractic $1/\alpha^2$ factor in the sample complexity that we incur when using `DPExpGKGumb` for the same all-quantiles task.

The rest of the section contains the descriptions of the two algorithms, as well as a detailed theoretical analysis. The next section demonstrates how the histogram-based mechanism can be effectively applied to data generated from a normal distribution.

4.1 `DPExpGKGumb`: Exponential Mechanism Based Approach

We first establish how the exponential mechanism and the GK sketch may be used in conjunction to solve the single quantile problem. Concretely, the high level idea is to call the privacy preserving exponential mechanism with a utility function derived from the GK sketch. The exponential mechanism is a fundamental privacy primitive which when given a public set of choices and a private score for each choice outputs a choice that with high probability has a score close to optimal whilst preserving privacy. In the course of constructing our algorithms, we have to resolve two problems; one, how to usefully construct a utility function to pass to the exponential mechanism so that the private value derived is a good approximation to the q-quantile, and two, how to execute the exponential mechanism efficiently on the (possibly massive) data universe $\mathcal{X}$. On resolving the first issue we get the (not necessarily efficient) routine Algorithm 1, and on resolving the second issue we get an essentially equivalent but far more efficient routine i.e. Algorithm 2.

Constructing a score function: We recall that the GK sketch returns a short sequence of elements from the data set with a deterministic confidence interval for their ranks and the promise that for any target quantile $q \in [0, 1]$, there is some sketch element that lies within αn units in rank of $\lceil qn \rceil$. One technicality that we run into when trying to construct a score function on the data universe $\mathcal{X}$ is that when a single value occurs with very high frequency in the data set, the ranks of the set of occurrences can span a large interval in $[0, n]$, and there is no one rank we can ascribe to it so as to compare it with the target rank $\lceil qn \rceil$. This can be resolved by defining the score for any data domain value in terms of the distance of its respective rank interval $[r_{\min}(x), r_{\max}(x)]$ (formalized in Definition 3.2) from $\lceil qn \rceil$; elements whose intervals lie closer to the target have a higher score than those whose intervals lie further away.

Efficiently executing the exponential mechanism: The exponential mechanism samples one of the public choices (in our case some element from the data universe $\mathcal{X}$) with probability that increases with the quality of the choice according to the utility function. In general every element can have a possibly different score and the efficiency of the exponential mechanism can vary widely depending on the context. In our setting, the succinctness of the GK sketch leads to a crucial observation: by defining the score function via the sketch, the data domain is partitioned into a relatively small number of sets such that the utility function is constant on each partition. Concretely, for any two successive elements in the GK sketch, the range of values in the data universe that lie between them will have the same score according to our utility function. We can hence first sample a partition from which to output a value, and then choose a value from within that interval uniformly at random. To make our implementation even more efficient and easy to use, we also make use of the *Gumbel-max* trick that allows us to iterate through the set of choices instead of storing them in memory.

Outline: From Definition 4.1 to Lemma 4.3, we formalize how the GK sketch may be used to construct rank interval estimates for any data domain value. We then recall and apply the exponential mechanism with a utility function derived from the GK sketch (Definition 4.4 to Lemma 4.7 and Algorithm 1), and derive the error guarantee Lemma 4.9. We conclude this subsection with a detailed description of an efficient implementation of the exponential mechanism (Algorithm 2 and Lemma 4.10), and summarize our final accuracy and space complexity guarantees in Theorem 4.11.

Definition 4.1. Let $\hat{r}_{\min}(x) = \max\{\sum_{j\leq i} g_j : \text{val}(v_i) < x\}$ and $\hat{r}_{\max}(x) = \min\{\Delta_i + \sum_{j\leq i} g_j : \text{val}(v_i) > x\}$. Note that for every $v \in X$ such that $\text{val}(v) = x$, $\text{ix}(v) \in [\hat{r}_{\min}(x), \hat{r}_{max}(x)]$.

We formalize the rank interval estimation in a partition-wise manner as below.

Lemma 4.2. *Given a GK sketch* $(v_1, g_1, \Delta_1), \ldots, (v_s, g_s, \Delta_s)$, *for every* $x \in \mathcal{X}$ *one of the following two cases holds:*

(1) $x = v_i$ *for some* $i \in [s]$ *and* $i = \min\{j : v_j = x\}$,
$$\hat{r}_{\min}(x) = \sum_{j\leq i} g_j$$
$$\hat{r}_{\max}(x) = \min\{\Delta_{i^*} + \sum_{j\leq i^*} g_j : \exists i^*, \text{val}(v_{i^*}) > \text{val}(v_i)\}$$

(2) $x \in (v_{i-1}, v_i)$, *i.e.,* $x > v_{i-1}$ *and* $x < v_i$ *for some* $i \in [s]$,
$$\hat{r}_{\min}(x) = \sum_{j\leq i} g_j$$
$$\hat{r}_{\max}(x) = \Delta_{i+1} + \sum_{j\leq i+1} g_j$$

Proof. Recall that by Remark 3.7 we always have that the first tuple and the last tuple are formal elements at $-\infty$ and ∞, ensuring that every data universe element either explicitly occurs in the GK sketch or lies between two values that occur in the GK sketch. Both statements now follow directly from Definition 4.1 and the fact that the values v_i occur in increasing order in the sketch (Lemma 3.6). □

The quality of the rank interval estimate $[\hat{r}_{\min}(x), \hat{r}_{\max}(x)]$ compared to the true rank interval $[r_{\min}(x), r_{\max}(x)]$ is formalized as follows.

Lemma 4.3. $|r_{\min}(x) - \hat{r}_{\min}(x)| \leq 2\alpha n$ *and* $|r_{\max}(x) - \hat{r}_{\max}(x)| \leq 2\alpha n$.

Proof. Let $i^* = \text{argmax}_{i:\text{val}(v_i)<x} \sum_{j\leq i} g_j$. Then by Definition of i^*, we have that $\text{val}(v_{i^*}) < x \leq \text{val}(v_{i^*+1})$. It follows that
$$\begin{aligned}[\text{ix}(v_{i^*}), \text{ix}(v_{i^*+1})] &\subset [\sum_{j\leq i^*} g_j, \Delta_{i^*+1} + g_{i^*+1} + \sum_{j\leq i^*} g_j] \\ &\subset [\hat{r}_{\min}(x), \Delta_{i^*+1} + g_{i^*+1} + \hat{r}_{\min}(x)]\end{aligned}$$
Since $r_{\min}(x) \in [\text{ix}(v_{i^*}), \text{ix}(v_{i^*+1})]$ and $g_{i^*+1} + \Delta_{i^*+1} \leq 2\alpha n$, it follows that $|r_{\min}(x) - \hat{r}_{\min}(x)| \leq 2\alpha n$. The other inequality follows analogously. □

Definition 4.4 (Exponential Mechanism [39])**.** Let $u : \mathcal{S}^S \times \mathcal{R} \to \mathbb{R}$ be an arbitrary utility function with global sensitivity Δ_u. For any database summary $d \in \mathcal{S}^S$ and privacy parameter $\epsilon > 0$, the exponential mechanism $\mathcal{E}_u^\epsilon : \mathcal{S}^S \to \mathcal{R}$ outputs $r \in \mathcal{R}$ with probability $\propto \exp(\frac{\epsilon \cdot u(S(X),r)}{2\Delta_u})$ where
$$\Delta_u = \max_{X\sim X',r} |u(S(X), r) - u(S(X), r)|.$$

The following statement formalizes the trade-off between the privacy parameter ϵ and the tightness of the tail bound on the score attained by the exponential mechanism.

Theorem 4.5 ([39])**.** *The exponential mechanism (Definition 4.4) satisfies* ϵ*-differential privacy. Further, the following tail bound on the utility holds:*
$$P\left(u(S(X), \mathcal{E}_u^\epsilon(S(X))) < \max_{r\in R} u(S(X), r) - \frac{2\Delta_u(t + \ln s)}{\epsilon}\right) \leq e^{-t},$$
where s *is the size of the universe from which we are sampling from.*

To run the exponential mechanism using our approximate rank interval estimates, we define a utility function as follows.

Definition 4.6. Let $d(\cdot,\cdot)$ denote the ℓ_1 metric on $\mathbb{R}$. Given a sketch $S(X)$, we define a utility function on $\mathcal{X}$:
$$\begin{aligned}u(S(X), x) &= -\min\{|y - \lceil qn \rceil| : y \in [\hat{r}_{\min}(x), \hat{r}_{\max}(x)]\} \\ &= -d(\lceil qn \rceil, [\hat{r}_{\min}(x), \hat{r}_{\max}(x)])\end{aligned}$$

The magnitude of the noise that is added in the course of the exponential mechanism depends on the sensitivity of the score function, which we bound from above as follows.

Lemma 4.7. *For all* $n > 1/\alpha$, *the sensitivity of* u *(i.e.,* Δ_u*) is at most* $4\alpha n + 2$ *units.*

Proof. Fix any data set X' neighbouring X under swap DP and let $[r'_{\min}(\cdot), r'_{\max}(\cdot)]$ be the rank ranges with respect to X' for values in $\mathcal{X}$. Let $[\hat{r'}_{\min}(\cdot), \hat{r'}_{\max}(\cdot)]$ denote the confidence interval derived from the GK sketch $S(X')$ for values in $\mathcal{X}$.

Claim 4.8. $|r_{\min}(x) - r'_{\min}(x)| \leq 2$, $|r_{\max}(x) - r'_{\max}(x)| \leq 2$.

Proof. These bounds follow directly from the Definition of $r_{\min}$ and $r_{\max}$; under swap DP at most two elements of the stream are changed which implies that the count of the sets defining these terms changes by at most 1 unit each for a total shift of 2 units (in fact, this can be bounded by 1 unit). □

We now prove the sensitivity bound.

$$\begin{aligned} u(S(X), x) &= -d(\lceil qn \rceil, [\hat{r}_{\min}(x), \hat{r}_{\max}(x)]) \\ &\leq -d(\lceil qn \rceil, [r_{\min}(x), r_{\max}(x)]) + 2\alpha n \\ &\leq -d(\lceil qn \rceil, [r'_{\min}(x), r'_{\max}(x)]) + 2\alpha n + 2 \\ &\leq -d(\lceil qn \rceil, [\hat{r'}_{\min}(x), \hat{r'}_{\max}(x)]) + 4\alpha n + 2 \\ &\leq u(S(X'), x) + 4\alpha n + 2. \end{aligned}$$

Swapping the positions of X and X', we get the reverse bound to complete the sensitivity analysis. □

Algorithm 1: `DPExpGK`: Exponential Mechanism DP Quantiles : High Level Description

---

Data: $X = (x_1, x_2, \ldots, x_n)$
Input: ϵ, α (approximation parameter), $q \in [0, 1]$ (quantile parameters), Δ_u

$S(X) = \{(v_i, g_i, \Delta_i) : i \in [s]\} \leftarrow GK(X, \alpha)$

Define utility function

$$u(S(X), x) = -\min\{|y - \lceil qn \rceil| : y \in [\hat{r}_{\min}(x), \hat{r}_{\max}(x)]\}. \quad (1)$$

where

$$\hat{r}_{\min}(x) = \max\{\sum_{j \leq i} g_j : \text{val}(v_i) < x\}$$

$$\hat{r}_{\max}(x) = \min\{\Delta_i + \sum_{j \leq i} g_j : \text{val}(v_i) > x\}$$

Choose and output $e \in \mathcal{X}$ with probability

$$\propto \exp\left(\frac{\epsilon}{2\Delta_u} \cdot u(S(X), e)\right).$$

We can now derive a high probability bound on the utility that is achieved by Algorithm 1.

Lemma 4.9. *If $\hat{x}$ is the value returned* `DPExpGK` *then with probability $1 - \beta$,*

$$d(\lceil qn \rceil, [\hat{r}_{\min}(\hat{x}), \hat{r}_{\max}(\hat{x})]) \leq 2\alpha n + \frac{2(4\alpha n + 2)\log(|\mathcal{X}|/\beta)}{\epsilon}.$$

Proof. By construction, Algorithm 1 is simply a call to the exponential mechanism with utility function $u(S(X), \cdot)$, Since for any target q-quantile, $\lceil qn \rceil$ lies in $[0, n]$ it follows that there is some $i^* \in s$ such that $\lceil qn \rceil \in [\sum_{j \leq i^*} g_j, \Delta_{i^*+1} + g_{i^*+1} + \sum_{j \leq i^*} g_j]$. It follows that $d(\lceil qn \rceil, [r_{\min}(\text{val}(v_{i^*}), r_{\max}(\text{val}(v_{i^*}))]) \leq 2\alpha n$ and that hence $\max(u(S(X), x)) \geq -2\alpha n$. If x^* is the output of the exponential mechanism, then applying the utility tail bound we get that with probability $1 - \beta$,

$$u(S(X), x^*) \geq -2\alpha n - \frac{2(4\alpha n + 2)\log(|\mathcal{X}|/\beta)}{\epsilon}.$$

By definition of u, the desired bound follows. □

Algorithm 2: `DPExpGKGumb`: Implementing the Exponential Mechanism on $S(X)$ using the Gumbel Distribution

---

Data: $X = (x_1, x_2, \ldots, x_n)$
Input: ϵ, α (approximation parameter), $q \in [0, 1]$ (quantile parameters)

Build summary sketch $S(X)$ and let $s = |S(X)|$.
Let $(v_i, g_i, \Delta_i) = S(X)[i]$ for all $i \in [s]$
maxIndex = −1
maxValue = $-\infty$
`/* Iterating over tuple values` v_i `*/`
Let $i = 1$
while $i <= s$ **do**
  $\hat{r}_{\min} = \sum_{j \leq i} g_j$
  $\hat{r}_{\max} = \min\{\Delta_{i^*} + \sum_{j \leq i^*} g_j : \text{val}(v_{i^*}) > \text{val}(v_i)\}$
  $u_i = -\min\{|y - \lceil qn \rceil| : y \in [\hat{r}_{\min}, \hat{r}_{\max}]\}$.
  $f = \frac{\epsilon}{2} u_i$
  $\tilde{f} = f + \text{Gumb}(0, 1)$
  if $\tilde{f} >$ *maxValue* **then**
    maxIndex = (i, tuple)
    maxValue = $\tilde{f}$
  $i \leftarrow \min\{j : v_j > v_i\}$
`/* Iterating over intervals between tuples` $\mathcal{X}(v_{i-1}, v_i) \subset \mathcal{X}$ `*/`
Let $i = 1$
while $i <= s$ **do**
  if $\mathcal{X}(v_{i-1}, v_i)$ *is not empty* **then**
    $\hat{r}_{\min} = \sum_{j \leq i} g_j$
    $\hat{r}_{\max} = \Delta_{i+1} + \sum_{j \leq i+1} g_j$
    $u_{i-1,i} = -\min\{|y - \lceil qn \rceil| : y \in [\hat{r}_{\min}, \hat{r}_{\max}]\}$.
    $f = \log(|\mathcal{X}(v_{i-1}, v_i)|) + \frac{\epsilon}{2} u_{i-1,i}$
    $\tilde{f} = f + \text{Gumb}(0, 1)$
    if $\tilde{f} >$ *maxValue* **then**
      maxIndex = $(i, \text{interval})$
      maxValue = $\tilde{f}$
  $i \leftarrow i + 1$
if *maxIndex* = $(i$, *tuple*$)$ *for some* $i \in 1, \ldots, s$ **then**
  return v_i
else if *maxIndex* = $(i$, *interval*$)$ *for some* $i \in 1, \ldots, s$ **then**
  Pick $v \in \mathcal{X}(v_{i-1}, v_i)$ uniformly at random
  return v

As discussed before, in general a naive implementation of the exponential mechanism as in Algorithm 1 would in general not be efficient. To resolve this issue, in Algorithm 2 we take advantage of the partition of the data domain by the score function and the *Gumbel-max* trick to implement the exponential mechanism without any higher-order overhead and return an α-approximate q quantile. This trick has become a standard way to implement the exponential mechanism over intervals/tuples.

Lemma 4.10. *Algorithm 2 implements the exponential mechanism with utility function $u(S(X), \cdot)$ on the data universe $\mathcal{X}$ with space complexity $O(|S(X)|)$ (where $S(X)$ is the GK sketch) and additional time complexity $O(|S(X)| \log |S(X)|)$.*

Proof. As noted in previous work [1], if $Z_1, \ldots, Z_N$ are drawn i.i.d. from standard Gumbel distribution,

$$\mathbb{P}\left[f_i + Z_i = \max_{j \in [N]} \{f_j + Z_j\}\right] = \frac{\exp(f_i)}{\sum_{j \in [N]} \exp(f_j)}, \forall i \in [N].$$

We recall that when running the exponential mechanism on $\mathcal{X}$, we want to sample the element $x \in \mathcal{X}$ with probability $\propto \exp(\epsilon u(S(X), x))$. To implement the exponential mechanism via the identification with Gumbel argmax distribution above, we will simply compute the scores $u(S(X), x)$ and let $f_i = \epsilon \cdot u(S(X), x)$.

For $x \in \mathcal{X}$ such that $x = v_i$ for some $i \in [s]$, Algorithm 2 directly computes the scores according to Definition 4.6 and Lemma 4.2; this is formalized by lines 5 to 15 in the pseudo code.

For $x \in \mathcal{X}$ which lie strictly between the tuple values $\{v_i : i \in [s]\}$, we proceed as follows. Fixing i, from Lemma 4.2 we have that that for $\mathcal{X}(v_{i-1}, v_i) := \{x \in \mathcal{X} : x > v_{i-1}, x < v_i\}$, the rank confidence interval estimate is the same, i.e. $[\sum_{j \leq i-1} g_j, \Delta_i + \sum_{j \leq i} g_j]$. It follows from Definition 4.6 that for all such domain values the the utility function score $u(S(X), \cdot)$ is equal; this is denoted $u_{i-1,i}$ in the pseudo code. By summing the probabilities for sampling individual domain elements, it follows that the likelihood of the exponential mechanism outputting some value from the set $\mathcal{X}(v_{i-1}, v_i)$ is $\propto |\mathcal{X}(v_{i-1}, v_i)| \exp(\epsilon u_{i-1,i}/2) = \exp(\epsilon u_{i-1,i}/2 + \log(|\mathcal{X}(v_{i-1}, v_i)|))$. This is formalized by lines 16 to 27 in the pseudo code.

Finally, if some interval is selected, then by outputting elements chosen uniformly at random, we ensure that the likelihood of $x \in \mathcal{X}(v_{i-1}, v_i)$ being output is $\propto \frac{1}{|\mathcal{X}(v_{i-1}, v_i)|} \cdot \exp(\epsilon u_{i-1,i}/2 + \log(|\mathcal{X}(v_{i-1}, v_i)|) = \exp(\epsilon u_{i-1,i})$. Note that we do not need to account for ties in the Gumbel scores as the event $f_i + Z_i = f_j + Z_j$ for any $j \neq i$ has measure 0. [2]

To bound the space and time complexity; we note that by the guarantees of the GK sketch, the size of the sketch $S(X)$ is $O((1/\alpha) \log \alpha n)$; we compute Gumbel scores by iterating over tuples and intervals of which there are at most $O(S(X))$-many of each, each computation takes at most $O(\log |S(X)|)$ time, and only the max score and index seen at any point is tracked in the course of the algorithm. □

We can now state and prove our main theorem in this section, proving utility bounds for α-approximating quantiles through DPExpGK with sublinear space.

Theorem 4.11. *Algorithm 2 is ϵ-differentially private. Let $\hat{x}$ be the value returned by Algorithm 2 when initialized with target quantile q. The following statements hold:*

(1) *Algorithm 2 can be run with space complexity $O(1/\alpha \log \alpha n)$, such that with probability $1 - \beta$*

$$d(\lceil qn \rceil, [\hat{r}_{\min}(\hat{x}), \hat{r}_{\max}(\hat{x})]) \leq 2\alpha n + \frac{2(4\alpha n + 2) \log(|\mathcal{X}|/\beta)}{\epsilon} \quad (2)$$

(2) *For $n > \frac{\log |\mathcal{X}|/\beta}{12\alpha\epsilon}$, Algorithm 2 can be run with space complexity $O\left((\alpha\epsilon)^{-1} \log(|\mathcal{X}|/\beta) \log(\alpha\epsilon n)\right)$ such that with probability $1 - \beta$, $\hat{x}$ is an α approximate q-quantile.*

Proof. The privacy guarantee of Algorithm 2 follows from the privacy guarantee of the exponential mechanism and Lemma 4.10. The accuracy bound in equation 2 is simply a restatement of Lemma 4.9. To derive the second statement, we substitute $\frac{\alpha \min\{\epsilon,1\}}{24 \log(|\mathcal{X}|/\beta)}$ for the approximation parameter α in equation 2 and get

$$\begin{aligned}
&d(\lceil qn \rceil, [\hat{r}_{\min}(\hat{x}), \hat{r}_{\max}(\hat{x})]) \\
&\leq 2 \cdot \frac{\alpha \min\{\epsilon, 1\} n}{24 \log(|\mathcal{X}|/\beta)} + \frac{2(4(\frac{\alpha \min\{\epsilon,1\}}{24 \log(|\mathcal{X}|/\beta)})n + 2) \log(|\mathcal{X}|/\beta)}{\epsilon} \\
&\leq \frac{\alpha n}{12 \log(|\mathcal{X}|/\beta)} + \frac{\alpha n}{3} + \frac{4 \log |\mathcal{X}|/\beta}{\epsilon} \\
&\leq \frac{\alpha n}{12} + \frac{\alpha n}{3} + \frac{\alpha n}{3} \\
&\leq \alpha n.
\end{aligned}$$

The space complexity bound now follows directly from the space complexity bound derived in Lemma 4.10, the space complexity bound $O(1/\alpha \log \alpha n)$ for the GK sketch, and by substituting $\frac{\alpha' \min\{\epsilon,1\}}{24 \log(|\mathcal{X}|/\beta)}$ for α. □

4.2 DPHistGK: Histogram Based Approach

For methods in this section, we assume that we have $K \geq 1$ *disjoint* bins each of width w (e.g., $w = \alpha/2$). These bins are used to construct a histogram.

Essentially, Algorithm 3 builds an empirical histogram based on the GK sketch, adds noise so that the bin values satisfy $(\epsilon, 0)$-DP, and converts this empirical histogram to an approximate empirical CDF, from which the quantiles can be approximately calculated.

Lemma 4.12. *Algorithm 3 satisfies $(\epsilon, 0)$-DP.*

Proof. For any $i \in [n]$, any item x_i can belong in at most one bin. Plus, the global sensitivity of the function that computes the empirical histogram is 2, since changing a single item can change the contents of at most two bins.

As a result, adding noise of $\mathrm{Lap}(0, 2/\epsilon)$ to each bin satisfies ϵ-DP by Theorem 4.13. □

Theorem 4.13 (Laplace Mechanism [20])**.** *Fix $\epsilon > 0$ and any function $f : \mathcal{Y}^n \to \mathbb{R}^K$. The Laplace mechanism outputs*

$$f(y) + (L_1, \ldots, L_K),$$

$L_1, \ldots, L_K \sim \mathrm{Lap}(0, GS_f/\epsilon)$ where GS_f is the global sensitivity of the function f. Furthermore, the mechanism satisfies $(\epsilon, 0)$-DP.

[2] As is usual in the privacy literature, we assume that the sampling of the $Gumb(0, 1)$ distribution can be done on finite-precision computers [6]. While the problem of formally dealing with rounding has not been settled in the privacy literature [41], for any practical purpose it easily suffices to store the output of the Gumbel distribution using a few computer words.

Algorithm 3: DPHistGK: Computing DP Quantiles in Bounded Space

Data: $X = (x_1, x_2, \ldots, x_n)$
Input: ϵ, α (approximation parameter), $q \in [0, 1]$ (quantile parameters), w
Build summary sketch $S(X)$ where
$S(X) = \{(v_i, g_i, \Delta_i) : i \in [s]\} \leftarrow GK(X, \alpha)$
/* cell labels a_i and counts $c_i = 0$ */
Initialize data-agnostic (empty) histogram $Hist = \langle (a_i, c_i), \ldots \rangle$ with cell widths w
for $(v_i, g_i, \Delta_i) \in S(X)$ **do**
  Insert g_i counts of v_i into histogram $Hist$
$c = 0$
$H = []$
for $(a_i, c_i) \in Hist$ **do**
  $\tilde{c}_i = \max(0, c_i + \text{Lap}(0, 2/\epsilon))$
  Append $(a_i, c + \tilde{c}_i)$ to H
  $c = c + \tilde{c}_i$
$r = \lceil q \cdot n \rceil$
for $(b, rank) \in H$ **do**
  if $r < rank$ **then**
    return b
/* return last element of H */
return $H[|H| - 1]$

5 LEARNING DIFFERENTIALLY PRIVATE QUANTILES OF A NORMAL DISTRIBUTION WITH UNKNOWN MEAN

We demonstrate one use case of DPHistGK where the space complexity required to achieve a certain degree of accuracy improves upon the worst-case bound for DPExpGK, Theorem 4.11. While the histogram based mechanism does not have universal utility bounds in the spirit of the above theorem, the results in this section serve as one simple example where it may yield desirable accuracy while using less space.

Suppose that we are given an i.i.d. sample $S = (X_1, X_2, \ldots, X_n)$ such that for all $i \in [n]$, $X_i \sim \mathcal{N}(\mu, \sigma^2 I_{d\times d})$, $\mu \in \mathbb{R}^d$. The goal is to estimate DP quantiles of the distribution $\mathcal{N}(\mu, \sigma^2 I_{d\times d})$ without knowledge of μ or σ^2. We will show how to estimate the quantiles assuming that σ^2 is known. Note that it is easy to generalize the work to the case where σ^2 is unknown as follows:

For any sample S drawn from i.i.d. from $\mathcal{N}(\mu, \sigma^2 I_{d\times d})$, the $1 - \beta$ confidence interval is

$$\bar{X} \pm \frac{\sigma}{\sqrt{n}} \cdot z_{1-\beta/2},$$

where $z_{1-\beta/2}$ is the $1 - \beta/2$ quantile of the standard normal distribution and $\bar{X}$ is the empirical mean. The length of this interval is fixed and equal to

$$\frac{2\sigma z_{1-\beta/2}}{\sqrt{n}} = \Theta\left(\frac{\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\beta}}\right).$$

In the case where σ^2 is unknown, the confidence interval becomes

$$\bar{X} \pm \frac{s}{\sqrt{n}} \cdot t_{n-1,1-\beta/2},$$

where $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$ is the sample variance (sample estimate of σ^2) and $t_{n-1,1-\beta/2}$ is the $1-\beta/2$ quantile of the t-distribution with $n - 1$ degrees of freedom. The length of the interval can be shown to be

$$\frac{2\sigma}{\sqrt{n}} \cdot k_n \cdot t_{n-1,1-\beta/2} = \Theta\left(\frac{\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\beta}}\right),$$

where $k_n = 1 - O(1/n)$ is an appropriately chosen constant. See [32, 33, 35] for more details and discussion. We will assume that σ^2 is known and proceed to show sample and space complexity bounds. One could also estimate the variance in a DP way and then prove the complexity bounds.

For any $q \in (0, 1)$, we denote the q-quantile of the sample as Q_S^q and the q-quantile of the distribution as $Q_{\mathcal{D}}^q$.

That is, for any $q \in (0, 1)$ and sample $S = (X_1, X_2, \ldots, X_n)$, we wish to obtain a DP q-quantile $\tilde{Q}_S^q$ with the following guarantee:

$$\mathbb{P}[\|Q_{\mathcal{D}}^q - \tilde{Q}_S^q\| \geq \alpha] \leq \beta,$$

for any $\beta \in (0, 1], \alpha > 0$.

We shall proceed to use a three-step approach: (1) Estimate a DP range of the population in sub-linear space; (2) Use this range of the population to construct a DP histogram using the stream S; (3) Use the DP histogram to estimate one or more quantiles via the sub-linear data structure of Greenwald and Khanna.

In the case where $d = 1$, by Theorem 5.1, there exists an (ϵ, δ)-DP algorithm $\tilde{Q}_S^q$ such that if $\mu \in (-R, R)$ then using space of $O(\max\{\frac{R}{\sigma}, \frac{1}{\alpha}\log \alpha n\})$ (with probability 1) as long as the stream length is at least

$$n \geq O\left(\max\left\{\min\left\{O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{R}{\sigma\beta}\right), O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{1}{\beta\delta}\right)\right\}, O\left(\frac{R^2}{\sigma^2\alpha^2}\log\frac{1}{\beta}\right)\right\}\right),$$

we get the guarantee

$$\mathbb{P}[\|Q_{\mathcal{D}}^q - \tilde{Q}_S^q\| \geq \alpha] \leq \beta,$$

for any $\beta \in (0, 1], \alpha > 0$.

Intuitively, this means that: (1) **Space**: We need less space to estimate any quantile with DP guarantees if the distribution is less concentrated (i.e., σ can be large) or if we do not require a high degree of accuracy for our queries (i.e., α can be large). (2) **Stream Length**: We need a large stream length to estimate any quantile if we require a high degree of accuracy (i.e., smaller β, α), or do not have a good public estimate of μ (i.e., large R), or have small privacy parameters (i.e., small ϵ, δ), or have concentrated datasets (i.e., small σ).

Theorem 5.1. *For the 1-D normal distribution $\mathcal{N}(\mu, \sigma^2)$, let $S = (X_1, X_2, \ldots, X_n)$ be a data stream through which we wish to obtain $\tilde{Q}_S^q$, a DP estimate of the q-quantile of the distribution.*

For any $q \in (0,1)$, there exists an (ϵ,δ)-DP algorithm such that, with probability at least $1-\beta$, we obtain $|Q^q_{\mathcal{D}} - \tilde{Q}^q_S| \le \alpha$ for any $\alpha > 0$, $\beta \in (0,1]$, $\epsilon, \delta \in (0, 1/n)$ and for stream length

$$n \ge \max\{\min\{A, B\}, C\}, \quad \text{where}$$

$$A = O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{R}{\sigma\beta}\right), B = O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{1}{\beta\delta}\right), C = O\left(\frac{R^2}{\sigma^2\alpha^2}\log\frac{1}{\beta}\right)$$

as long as $\mu \in (-R, R)$ and using space of $O(\max\{\frac{R}{\sigma}, \frac{1}{\alpha}\log \alpha n\})$.

Proof. For any stream $S = (X_1, \dots, X_n)$, we use the triangle inequality so that

$$|Q^q_{\mathcal{D}} - \tilde{Q}^q_S| \le |Q^q_{\mathcal{D}} - Q^q_S| + |Q^q_S - \tilde{Q}^q_S| \tag{3}$$

$$\le \alpha/2 + \alpha/2. \tag{4}$$

$|Q^q_{\mathcal{D}} - Q^q_S| \le \alpha/2$ follows with probability $1-\beta/2$ by Corollary 5.3 and $|Q^q_S - \tilde{Q}^q_S| \le \alpha/2$ follows with probability $1-\beta/2$ by Lemma 5.4. The space complexity follows with probability 1 via the deterministic nature of the Greenwald-Khanna sketch. ◻

Lemma 5.2 (Dvoretzky-Kiefer-Wolfowitz inequality [18]). *For any $n \in \mathbb{Z}_+$, let $X_1, \dots, X_n$ be i.i.d. random variables with cumulative distribution function F so that $F(x)$ is the probability that a single random variable X is less than x for any $x \in \mathbb{R}$. Let the corresponding empirical distribution function be $F_n(x) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}[X_i \le x]$ for any $x \in \mathbb{R}$.*

Then for any $\gamma > 0$,

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| > \gamma\right) \le 2\exp(-2n\gamma^2).$$

Corollary 5.3. *For any $q \in (0,1)$, let $Q^q_{\mathcal{D}}$ be the q-quantile estimate for the distribution $\mathcal{D}$ and Q^q_S be the q-quantile estimate for the estimate.*

Then, $|Q^q_{\mathcal{D}} - Q^q_S| \le \alpha/2$ with probability $1-\beta/2$ when $n \ge \frac{2}{\alpha^2}\log 4/\beta$.

Proof. Follows by the DKW inequality (Lemma 5.2) where $n \ge \frac{1}{2\gamma^2}\log 4/\beta$ and $\gamma = \alpha/2$. ◻

Lemma 5.4. *For any $q \in (0,1)$, $\alpha > 0$, $\beta \in (0,1]$, $\epsilon, \delta \in (0,1/n)$, there exists an (ϵ,δ)-differentially private algorithm $\tilde{Q}^q_S$ for computing the q-quantile such that*

$$|Q^q_S - \tilde{Q}^q_S| \le \alpha/2,$$

with probability $\ge 1-\beta$ for stream length

$$n \ge O\left(\min\{O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{R}{\sigma\beta}\right), O\left(\frac{R}{\epsilon\sigma\alpha}\log\frac{1}{\beta\delta}\right)\}\right).$$

Furthermore, with probability 1, $\tilde{Q}^q_S$ uses space of $O(\max\{\frac{R}{\sigma}, \frac{1}{\alpha}\log \alpha n\})$.

Proof. First, by the tail bounds of the Gaussian distribution (Claim 5.6), we can obtain that for any $i \in [n]$,

$$\mathbb{P}[|X_i - \mu| > c] \le 2e^{-c^2/2\sigma^2},$$

so that by the union bound,

$$\mathbb{P}[\exists i, |X_i - \mu| \ge c] \le 2ne^{-c^2/2\sigma^2},$$

which implies that for any $\beta \in (0,1]$,

$$\mathbb{P}[\forall i, |X_i - \mu| \le \sigma\sqrt{2\log 4n/\beta}] \ge 1-\beta/2,$$

which holds by our sample complexity (stream length) guarantees.

Next, let $r = \lceil R/\sigma \rceil$. [3] Divide $[-R-\sigma/2, R+\sigma/2]$ into $2r+1$ bins of length at most σ each. Each bin B_j should equal $((j-0.5)\sigma, (j+0.5)\sigma]$ for any $j \in \{-r, \dots, r\}$. Next run the histogram learner of Lemma 5.5 with per-bin accuracy parameter of α/K, high-probability parameter of $\beta/2$, privacy parameters $\epsilon, \delta \in (0, 1/n)$, and number of bins $K = 2\lceil R/\sigma \rceil + 1$. We can do this because of our sample complexity (stream length) bounds. Then we obtain noisy estimates $\tilde{p}_{-r}, \dots, \tilde{p}_r$ with per-bin accuracy of α/K. Then any quantile estimate would have accuracy of α (by summing noisy estimates for at most K bins).

Next, we use these bins to construct a sketch (private by DP post-processing) based on the deterministic algorithms of [26] to, with probability 1, obtain space of $O(\max\{\frac{R}{\sigma}, \frac{1}{\alpha}\log \alpha n\})$. ◻

Lemma 5.5 (Histogram Learner [9, 32, 51]). *For every $K \in \mathbb{N} \cup \{\infty\}$ and every collection of disjoint bins $B_1, \dots, B_K$ defined on the domain $\mathcal{X}$. For any $n \in \mathbb{N}$, $\epsilon, \delta \in (0,1/n)$, $\alpha > 0$, and $\beta \in (0,1)$, there exists an (ϵ,δ)-DP algorithm $M : \mathcal{X}^n \to \mathbb{R}^K$ such that for every distribution $\mathbb{D}$ on the domain $\mathcal{X}$, if*

1. $X_1, \dots, X_n \sim \mathbb{D}$, $p_k = \mathbb{P}[X_i \in B_k]$ *for any* $k \in [K]$,
2. $(\tilde{p}_1, \dots, \tilde{p}_K) \leftarrow M(X_1, \dots, X_n)$,
3. $n \ge \max\left\{\min\left\{\frac{8}{\epsilon\alpha}\log\frac{2K}{\beta}, \frac{8}{\epsilon\alpha}\log\frac{4}{\beta\delta}\right\}, \frac{1}{2\alpha^2}\log\frac{4}{\beta}\right\}$,

then (over the randomness of the data $X_1, \dots, X_n$ and of M)

1. $\mathbb{P}_{\underline{X}\sim\mathbb{D},M}[|\tilde{p}_k - p_k| \le \alpha] \ge 1-\beta$,
2. $\mathbb{P}[\text{argmax}_k \tilde{p}_k = j] \le np_j$ *if* $K \ge 2/\delta$,
3. $\mathbb{P}[\text{argmax}_k \tilde{p}_k = j] \le np_j + 2\exp(-(\epsilon n/8)\cdot(\max_k p_k))$ *if* $K < 2/\delta$.

Claim 5.6 (Gaussian Tail Bound). *Let Z be a random variable distributed according to a standard normal distribution (with mean 0 and variance 1). For every $t > 0$,*

$$\mathbb{P}[|Z| > t] \le 2\exp(-t^2/2).$$

6 EXPERIMENTAL EVALUATION

In this section, we experimentally evaluate our sublinear-space exponential-based mechanism, `DPExpGKGumb`. We study how well our algorithm performs in terms of accuracy and space usage. Note that we proved space complexity and accuracy bounds for the algorithm; part of this section validates that the space complexity of the algorithm is indeed very small in practice, and that the accuracy is typically closely tied to the approximation parameter α.

Our main baselines will be `DPExpFull` (without use of the GK sketch but with the use of DP) and the true quantile value. The true quantile values are used to compute absolute errors.

[3] Note that this argument is similar to the arguments for Algorithm 1 in [32].

DPExpFull. The use of the exponential mechanism to ϵ-privately compute q-quantiles without the use of the sketching data structure. The space used by this algorithm is of order $\Theta(n)$. See Section 6.3 for further implementation details. We experimentally validate DPExpGKGumb varying the following parameters: $\epsilon > 0$ (the privacy parameter), n (the stream length), and α (the approximation parameter). We show results on both synthetically-generated datasets and time-series real-world datasets.

The real-world time-series datasets [17] are: (1) **Taxi Service Trajectory**: A dataset from the UCI machine learning repository describing trajectories performed by all 442 taxis (at the time) in the city of Porto in Portugal [42]. This dataset contains real-valued attributes with at least 1 million instances. (2) **Gas Sensor Dataset**: A UCI repository dataset containing recordings of 16 chemical sensors exposed to, at varying concentrations, two gas mixtures [23]. The sensor measurements are acquired continuously during a 12-hour time range and contains at least 4 millions instances.

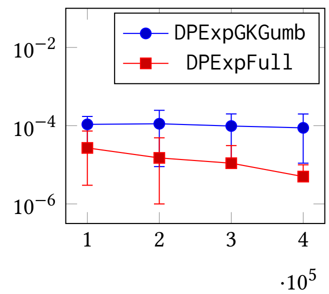
6.1 Synthetic Datasets

In this section, we compare our methods on synthetically generated datasets. We vary the following parameters:

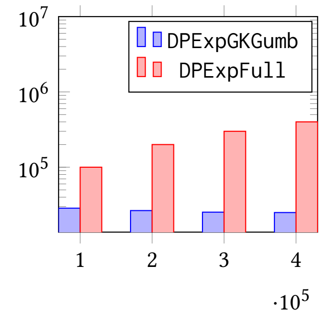
(1) **Privacy Parameter** ϵ: For each choice of $\epsilon \geq 0$, we run the DP algorithms over 100 trials and output means and confidence intervals over these trials.
(2) **Stream Length** n: The size of the stream $x_1, \ldots, x_n$ received by the sketching algorithm. When we vary n, we fix $\epsilon = 1$.
(3) **Approximation Parameter** α: The approximation factor used to construct the GK sketch for our α-approximate q quantiles.
(4) **Data Distribution**: We generate data from the uniform and Gaussian distributions. We use a uniform distribution in range $[0, 1]$ (i.e., $U(0, 1)$) or a normal distribution with mean 0 and variance 1 (i.e., $N(0, 1)$), clipped to lie within the interval $[-10, 10]$.[4]

We show results on space usage and absolute error (both plotted mostly on logarithmic scales) from non-DP estimates, as we vary the parameters listed above. Here, the *absolute error* is defined as the absolute difference between the DP estimate and the true non-DP quantile value. For plots that compare the absolute error to the stream length (e.g., Figures 3a and 4a), the *absolute error* will be on the y-axis while the stream length or the privacy parameter will be on the x-axis. For plots that compare the data structure size to the stream length (e.g., Figures 3b and 4b), the sizes will be on the y-axis while the stream length will be on the x-axis. We graph the mean absolute error over 100 trials of the exponential mechanism per experiment, as well as the 95% confidence interval computed by taking the 5th and 95th percentile. While the quantile to be estimated can also be varied, we fix it to $q = 0.5$ (the median)

[4]Clipping is required for the exponential mechanism, as it must operate on some bounded interval of values. In any case, we never expect to see samples from $N(0, 1)$ that lie outside $[-10, 10]$ for any practical purpose; the probability for any given sample to satisfy this is minuscule, at about $\approx 10^{-21}$.



(a) Empirical mean absolute error and 90% confidence interval vs stream length over 100 trials



(b) Data structure sizes vs stream length

Figure 3: DPExpGKGumb versus DPExpFull for uniformly random data, $\alpha = 10^{-4}$, $\epsilon = 1$, $q = 0.5$

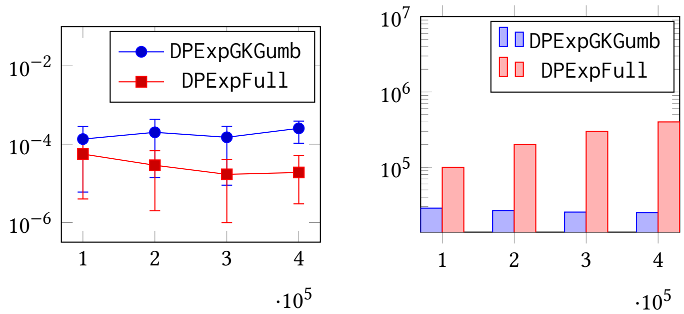
throughout as the absolute error and the space usage generally do not seem to vary much with choice of q.

In Figures 3 and 4 we vary the stream length n for an approximation factor of $\alpha = 10^{-4}$. The streams are either normally or uniformly distributed. In Figures 3a and 3b, we compare DPExpGKGumb (space strongly sublinear in n) vs. DPExpFull (uses space of $O(n)$) in terms of space usage and accuracy. We verify experimentally that DPExpFull will use more space than DPExpGKGumb. On the other hand, the space savings incurred by DPExpGKGumb are expected to create some drop in accuracy; recall that such a drop must occur even for non-private streaming algorithms. Figures 4a and 4b the streams are from a normal distribution instead of uniform.

In general we find that although our method DPExpGKGumb incurs higher error, in absolute terms it remains quite small and the 95% confidence intervals tend to be adjacent for DPExpGKGumb and DPExpFull. However, there is a clear trend of an exponential gap developing between their respective space usages which is a natural consequence of the space complexity guarantee of the GK sketch. This holds for both distributions studied (Figures 3 and Figure 4).

In Figures 6 and 7, we vary the stream length for a relatively large approximation factor of 0.1. Here we see that compared to the non-approximate method we incur far higher error, although there is also a concomitant increase in the space savings. This is not a typical use-case since the non-private error can itself be large, but we get a complete picture of how space usage and performance vary with this user-defined parameter.

In Figures 9a and 9b, we also vary the privacy parameter. In the small approximation factor setting we see the inverse tendency of accuracy with privacy which is characteristic of most DP algorithms. On the other hand, in the large approximation setting (and indeed even in the small approximation setting for intermediate privacy parameter values), there is no such clear drop in performance with more privacy. This motivates the question of determining the true interplay between the approximation factor α and the private parameter ϵ, as discussed further in Section 7.



(a) Mean absolute error and 90% confidence interval vs stream length over 100 trials

(b) Data structure sizes vs stream length

Figure 4: `DPExpGKGumb` versus `DPExpFull` for normally distributed data, $\alpha = 10^{-4}$, $\epsilon = 1$, $q = 0.5$

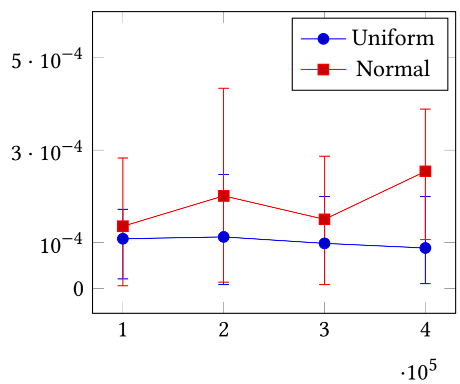
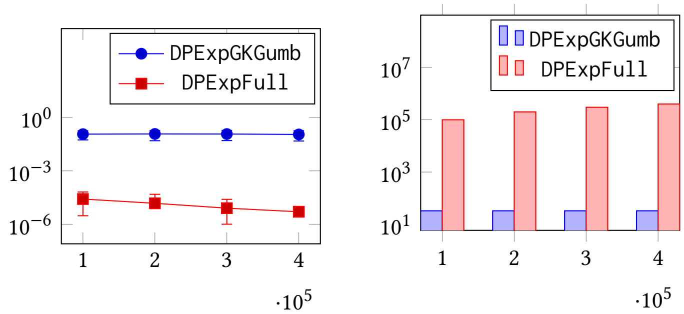


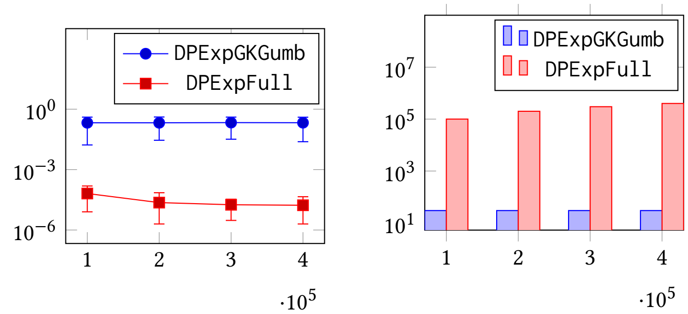
Figure 5: Error vs datastream size for $\alpha = 10^{-4}$, $\epsilon = 1$, $q = 0.5$



(a) Mean absolute error and 90% confidence interval vs stream length over 100 trials

(b) Data structure sizes vs stream length

Figure 6: `DPExpGKGumb` versus `DPExpFull` for uniformly random data, $\alpha = 10^{-1}$, $\epsilon = 1$, $q = 0.5$



(a) Mean absolute error and 90% confidence interval vs stream length over 100 trials

(b) Data structure sizes vs stream length

Figure 7: `DPExpGKGumb` versus `DPExpFull` for normally distributed data, $\alpha = 10^{-1}$, $\epsilon = 1$, $q = 0.5$

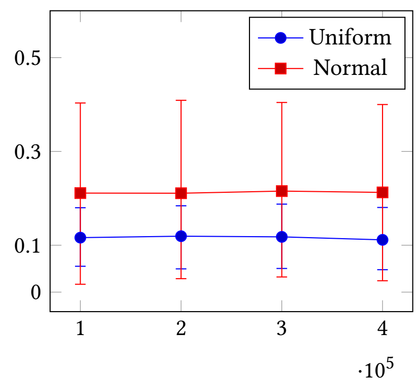


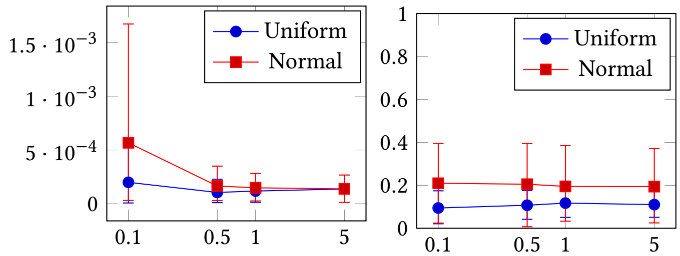
Figure 8: `DPExpGKGumb` error vs datastream size for $\alpha = 10^{-1}$, $\epsilon = 1$, $q = 0.5$

As the main takeaway, and as predicted by the theoretical results, our algorithm performs well in practical settings where one wishes to estimate some quantity across all data items privately and using small space. For example, our results indicate that choosing an approximation factor of $\alpha = 10^{-4}$ induces an error which is also of order about 10^{-4} for privately computing parameters chosen according to a uniform or normal distribution, all while saving orders of magnitude in the space complexity. Another takeaway is that choosing a large approximation factor like $\alpha = 0.1$ can cause large errors and is therefore not advisable in practice.

6.2 Real-World Datasets

In Table 1, we show the properties of attributes available from the taxi service and gas sensor datasets. We pick a real-valued attribute from each dataset and calculate the median on these datasets.

In Table 2, for different values of the approximation parameter α, we show the multiplicative factors in space usage incurred on these datasets. The larger the approximation factor, the larger the space savings with `DPExpGK`. Comparing `DPExpFull` to `DPExpGK`,



(a) Approximation factor $\alpha = 10^{-4}$

(b) Approximation factor $\alpha = 10^{-1}$

Figure 9: Error versus privacy parameter

	# of Attributes	# Instances	Attribute Type
Taxi Service	9	1,710,671	Real
Gas Sensor	19	4,178,504	Real

Table 1: Comparing properties of the datasets.

α	Gas Sensor	Taxi Service
0.01	10654x	3054x
0.001	917x	122x

Table 2: Space savings of DPExpGK on real-world datasets.

we see space savings of 122 times up to 10,000 times as we vary the approximation factor. These results are consistent with our expectations that the space savings are inversely proportional to the allowed approximation factor.

6.3 Full Space Quantile Computation

Without the bounded space requirement (i.e., space sublinear in the stream length), we can use the exponential mechanism with a utility function that is not defined on the sketch $S(X)$ but on the entire stream of values X. In that case, the sensitivity of the utility function is at most 1. We will use this as one of the baselines for our experimental validation.

Lemma 6.1. *Given any insertion only stream*

$$X = (x_1, x_2, \ldots, x_{n-1}, x_n),$$

the sensitivity of the utility function u (under swap differential privacy) is at most 1. i.e., $\Delta_u \leq 1$. The function u is defined as $u(X, e) = -|\operatorname{rank}(X, e) - r|$ where r is the approximate $\lfloor q \cdot n \rfloor$ rank of the sketch and $\operatorname{rank}(X, e)$ is the rank of e amongst all values in the stream X.

Proof. Let $|X| = n$. The utility function becomes $-|\operatorname{rank}(X, e) - n_q|$ where $n_q = \lfloor q \cdot n \rfloor$. Consider two streams with only one element changed: X, X'. Further suppose that X, X' differ in element x_d. Then at time $d \leq n$, in the second stream x'_d is inserted instead of x_d. In both cases, n_q changes by at most q (in the case of add-remove DP) and for swap DP, n_q remains the same. And for any e, $\operatorname{rank}(X', e)$ would differ from $\operatorname{rank}(X, e)$ by at most 1 since the rank of any element can change by at most 1 after adding, deleting, or replacing an item in the stream. Furthermore, for any $n \geq d$, the rank of any e will differ in X, X' by at most 1 replacing x_d with x'_d can displace the rank of any element by at most 1. Also, the term n_q will remain the same. (Note that in the add-remove privacy definition $n_q = \lfloor q \cdot n \rfloor$ would change to either $\lfloor q \cdot (n+1) \rfloor$ or $\lfloor q \cdot (n-1) \rfloor$.)

Next, recall the "reverse triangle inequality" which says that for any real numbers x and y, $|x - y| \geq ||x| - |y||$. As a result, $-|\operatorname{rank}(X, e) - n_q| + |\operatorname{rank}(X', e) - n_q| \leq |\operatorname{rank}(X', e) - \operatorname{rank}(X, e)| \leq 1$ for any e. □

7 CONCLUSION & FUTURE WORK

In this work, we presented sublinear-space and differentially private algorithms for approximately estimating quantiles in a dataset. Our solutions are two-part: one based on the exponential mechanism and efficiently implemented via the use of the Gumbel distribution; the other based on constructing histograms. Our algorithms are supplemented with theoretical utility guarantees. Furthermore, we experimentally validate our methods on both synthetic and real-world datasets. Our work leaves room for further exploration in various directions:

- **Interplay between α and ϵ:** The space complexity bounds we obtain are (up to lower order terms) inversely linear in α and in ϵ. While it is either known or easy to show that such linear dependence in each of these parameters in itself is necessary, it is not clear whether the $\alpha^{-1}\epsilon^{-1}$ term in Theorem 4.11 can be replaced with, say, $\alpha^{-1} + \epsilon^{-1}$. Such an improvement, if possible, seems to require substantially modifying the baseline Greenwald-Khanna sketch or adding randomness of some sort.
- **Alternative streaming baselines:** We base our mechanisms upon the GK-sketch, which is known to be space-optimal among deterministic streaming algorithms for quantile approximation. The use of a deterministic baseline simplifies the analysis and the overall solution, but better randomized streaming algorithms for the same problem are known to exist. What would be the benefit of working with the (optimal among randomized algorithms) KLL-sketch [31], for example?
- **Dependence in universe size:** The dependence of our space complexity bounds in the size of the universe, $\mathcal{X}$, is logarithmic. Recent work of Kaplan et al. [30] (see also [9]) on the sample (not space) complexity of privately learning thresholds in one dimension, a fundamental problem at the intersection of learning theory and privacy, demonstrate a bound polynomial in $\log^* |\mathcal{X}|$ on the sample complexity. As quantile estimation and threshold learning are closely related problems, this raises the question of whether techniques developed in the aforementioned papers can improve the dependence on $|\mathcal{X}|$ in our bounds.
- **Continual release:** Our work applies to the "one-shot" setting where the quantiles — whether for single or multiple queries

— are computed once after observing the entire stream. However, many relevant applications require real-time release of statistics. Can our results be adapted to function under continual releases (e.g., in the settings of [21] and [11])?

Random order: The results presented here (except for those about normally distributed data) all assume that the data stream is presented in worst case order, an assumption that may be too strong for some scenarios. Can improved bounds be proved when the data elements are chosen in advance but their order is chosen randomly? This can serve as a middle ground between the most general case (which we address in this paper) and the case where data is assumed to be generated according to a certain distribution.

REFERENCES

[1] Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. 2017. *Perturbation Techniques in Online Learning and Optimization.* 233–264.

[2] Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, Jeff M. Phillips, Zhewei Wei, and Ke Yi. 2013. Mergeable summaries. *ACM Trans. Database Syst.* 38, 4 (2013), 26:1–26:28.

[3] Daniel Alabi, Audra McMillan, Jayshree Sarathy, Adam Smith, and Salil Vadhan. 2020. Differentially Private Simple Linear Regression. *CoRR* abs/2007.05157 (2020). https://arxiv.org/abs/2007.05157

[4] Hilal Asi and John C. Duchi. 2020. Near Instance-Optimality in Differential Privacy. *CoRR* abs/2005.10630 (2020). arXiv:2005.10630 https://arxiv.org/abs/2005.10630

[5] Martin Aumüller, Christian Janos Lebeda, and Rasmus Pagh. 2021. Differentially private sparse vectors with low error, optimal space, and fast access. *CoRR* abs/2106.10068 (2021). arXiv:2106.10068 [cs.CR] https://arxiv.org/abs/2106.10068

[6] Victor Balcer and Salil P. Vadhan. 2018. Differential Privacy on Finite Computers. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA (LIPIcs)*, Anna R. Karlin (Ed.), Vol. 94. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 43:1–43:21.

[7] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012.* IEEE Computer Society, 410–419. https://doi.org/10.1109/FOCS.2012.67

[8] Jonas Böhler and Florian Kerschbaum. 2020. Secure Sublinear Time Differentially Private Median Computation. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020.* https://www.ndss-symposium.org/ndss-paper/secure-sublinear-time-differentially-private-median-computation/

[9] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. 2015. Differentially Private Release and Learning of Threshold Functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015.* 634–649. https://doi.org/10.1109/FOCS.2015.45

[10] Adrian Rivera Cardoso and Ryan Rogers. 2021. Differentially Private Histograms under Continual Observation: Streaming Selection into the Unknown. arXiv:2103.16787 [cs.DS]

[11] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 14, 3 (2011), 26:1–26:24. https://doi.org/10.1145/2043621.2043626

[12] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2017. PeGaSus: Data-Adaptive Differentially Private Stream Processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017.* 1375–1388. https://doi.org/10.1145/3133956.3134102

[13] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. 2020. Differentially-Private Multi-Party Sketching for Large-Scale Statistics. *Proc. Priv. Enhancing Technol.* 2020, 3 (2020), 153–174. https://doi.org/10.2478/popets-2020-0047

[14] Graham Cormode, Zohar S. Karnin, Edo Liberty, Justin Thaler, and Pavel Veselý. 2020. Relative Error Streaming Quantiles. *CoRR* abs/2004.01668 (2020). https://arxiv.org/abs/2004.01668

[15] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. 2006. Space- and time-efficient deterministic algorithms for biased quantiles over data streams. In *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, Illinois, USA.* 263–272.

[16] Graham Cormode and Pavel Veselý. 2020. A Tight Lower Bound for Comparison-Based Quantile Summaries. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Portland, OR, USA) *(PODS'20)*. Association for Computing Machinery, New York, NY, USA, 81–93. https://doi.org/10.1145/3375395.3387650

[17] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[18] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. 1956. Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *Ann. Math. Statist.* 27, 3 (09 1956), 642–669. https://doi.org/10.1214/aoms/1177728174

[19] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009.* 371–380.

[20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings.* 265–284.

[21] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, Leonard J. Schulman (Ed.). ACM, 715–724. https://doi.org/10.1145/1806689.1806787

[22] David Felber and Rafail Ostrovsky. 2017. A Randomized Online Quantile Summary in $O((1/\epsilon)\log(1/\epsilon))$ Words. *Theory Comput.* 13, 1 (2017), 1–17.

[23] Jordi Fonollosa, Sadique Sheik, Ramón Huerta, and Santiago Marco. 2015. Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators B: Chemical* 215 (2015), 618–629. https://www.sciencedirect.com/science/article/pii/S0925400515003524

[24] Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. 2021. Differentially Private Quantiles. *CoRR* abs/2102.08244 (2021). https://arxiv.org/abs/2102.08244

[25] Michael Greenwald and Sanjeev Khanna. 2001. Space-Efficient Online Computation of Quantile Summaries. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, CA, USA, May 21-24, 2001.* 58–66. https://doi.org/10.1145/375663.375670

[26] Michael Greenwald and Sanjeev Khanna. 2004. Power-Conserving Computation of Order-Statistics over Sensor Networks. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 14-16, 2004, Paris, France.* 275–285. https://doi.org/10.1145/1055558.1055597

[27] Michael B. Greenwald and Sanjeev Khanna. 2016. Quantiles and Equi-depth Histograms over Streams. In *Data Stream Management - Processing High-Speed Data Streams.* 45–86.

[28] Peter J. Huber. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* 35, 1 (1964), 73 – 101. https://doi.org/10.1214/aoms/1177703732

[29] Regant Y. S. Hung and Hing-Fung Ting. 2010. An $\Omega(1/\epsilon\log(1/\epsilon)$ Space Lower Bound for Finding *epsilon*-Approximate Quantiles in a Data Stream. In *Frontiers in Algorithmics, 4th International Workshop, FAW 2010, Wuhan, China, August 11-13, 2010. Proceedings (Lecture Notes in Computer Science)*, Der-Tsai Lee, Danny Z. Chen, and Shi Ying (Eds.), Vol. 6213. Springer, 89–100. https://doi.org/10.1007/978-3-642-14553-7_11

[30] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. 2020. Privately Learning Thresholds: Closing the Exponential Gap. In *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria] (Proceedings of Machine Learning Research)*, Jacob D. Abernethy and Shivani Agarwal (Eds.), Vol. 125. PMLR, 2263–2285. http://proceedings.mlr.press/v125/kaplan20a.html

[31] Zohar S. Karnin, Kevin J. Lang, and Edo Liberty. 2016. Optimal Quantile Approximation in Streams. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA.* 71–78.

[32] Vishesh Karwa and Salil P. Vadhan. 2017. Finite Sample Differentially Private Confidence Intervals. *CoRR* abs/1711.03908 (2017). http://arxiv.org/abs/1711.03908

[33] R.W. Keener. 2010. *Theoretical Statistics: Topics for a Core Course.* Springer New York.

[34] William H. Kruskal and W. Allen Wallis. 1952. Use of Ranks in One-Criterion Variance Analysis. *J. Amer. Statist. Assoc.* 47, 260 (1952), 583–621. https://doi.org/10.1080/01621459.1952.10483441 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1952.10483441

[35] Erich L. Lehmann and Joseph P. Romano. 2005. *Testing Statistical Hypotheses* (3. ed.). Springer New York, New York, NY.

[36] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. 2009. Optimizing Histogram Queries under Differential Privacy. *CoRR* abs/0912.4742 (2009). http://arxiv.org/abs/0912.4742

[37] Ge Luo, Lu Wang, Ke Yi, and Graham Cormode. 2016. Quantiles over data streams: Experimental comparisons, new analyses, and further improvements. , 449–472 pages.

[38] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. 1999. Random Sampling Techniques for Space Efficient Online Computation of Order Statistics of Large Datasets. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*. 251–262. https://doi.org/10.1145/304182.304204

[39] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*. 94–103. https://doi.org/10.1109/FOCS.2007.41

[40] Darakhshan J. Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. 2011. Pan-private algorithms via statistics on sketches. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*. 37–48. https://doi.org/10.1145/1989284.1989290

[41] Ilya Mironov. 2012. On significance of the least significant bits for differential privacy. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*. 650–661.

[42] Luís Moreira-Matias, João Gama, Michel Ferreira, João Mendes-Moreira, and Luís Damas. 2013. Predicting Taxi-Passenger Demand Using Streaming Data. *IEEE Trans. Intell. Transp. Syst.* 14, 3 (2013), 1393–1402. https://doi.org/10.1109/TITS.2013.2262376

[43] J.I. Munro and M.S. Paterson. 1980. Selection and sorting with limited storage. *Theoretical Computer Science* 12, 3 (1980), 315–323. https://doi.org/10.1016/0304-3975(80)90061-4

[44] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*. 75–84. https://doi.org/10.1145/1250790.1250803

[45] Victor Perrier, Hassan Jameel Asghar, and Dali Kaafar. 2019. Private Continual Release of Real-Valued Data Streams. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. https://www.ndss-symposium.org/ndss-paper/private-continual-release-of-real-valued-data-streams/

[46] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. 2004. Medians and beyond: New Aggregation Techniques for Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems* (Baltimore, MD, USA) *(SenSys '04)*. Association for Computing Machinery, New York, NY, USA, 239–249. https://doi.org/10.1145/1031495.1031524

[47] Adam D. Smith. 2011. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. 813–822. https://doi.org/10.1145/1993636.1993743

[48] Adam D. Smith, Shuang Song, and Abhradeep Thakurta. 2020. The Flajolet-Martin Sketch Itself Preserves Differential Privacy: Private Counting with Minimal Space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/e3019767b1b23f82883c9850356b71d6-Abstract.html

[49] J. W. Tukey. 1960. A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics* (1960), 448–485. https://ci.nii.ac.jp/naid/20000755025/en/

[50] Christos Tzamos, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Ilias Zadik. 2020. Optimal Private Median Estimation under Minimal Distributional Assumptions. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. https://proceedings.neurips.cc/paper/2020/hash/21d144c75af2c3a1cb90441bbb7d8b40-Abstract.html

[51] Salil P. Vadhan. 2017. The Complexity of Differential Privacy. In *Tutorials on the Foundations of Cryptography*. 347–450.

[52] Lu Wang, Ge Luo, Ke Yi, and Graham Cormode. 2013. Quantiles over data streams: an experimental study. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*. 737–748.

[53] Zhuolun Xiang, Bolin Ding, Xi He, and Jingren Zhou. 2020. Linear and Range Counting under Metric-based Local Differential Privacy. In *IEEE International Symposium on Information Theory, ISIT 2020, Los Angeles, CA, USA, June 21-26, 2020*. 908–913. https://doi.org/10.1109/ISIT44484.2020.9173952