Confidence Intervals for Private Query Processing

Dajun Sun
Hong Kong University of Science and Technology
Hong Kong, China
dsunad@cse.ust.hk

Wei Dong
Hong Kong University of Science and Technology
Hong Kong, China
wdongac@cse.ust.hk

Ke Yi
Hong Kong University of Science and Technology
Hong Kong, China
yike@cse.ust.hk

ABSTRACT

Whenever randomness is involved in query processing, confidence intervals are commonly returned to the user to indicate the statistical significance of the query answer. However, this problem has not been explicitly addressed under differential privacy, which must use randomness by definition. For some classical mechanisms whose noise distribution does not depend on the input, such as the Laplace and the Gaussian mechanism, deriving confidence intervals is easy. But the problem becomes nontrivial for queries whose global sensitivity is large or unbounded, for which these classical mechanisms cannot be applied. There are three main techniques in the literature for dealing with such queries: the exponential mechanism, the sparse vector technique, and the smooth sensitivity. In this paper, for each of the three techniques we design mechanisms to produce confidence intervals that are (1) differentially private; (2) correct, i.e., the interval contains the true query answer with the specified confidence level; and (3) have a utility guarantee matching that of the original mechanism, up to constant factors. Then we show how to apply our techniques to a variety of problems ranging from simple statistics (e.g., mean, median, maximum) to graph pattern counting and conjunctive queries.

PVLDB Reference Format:
Dajun Sun, Wei Dong, and Ke Yi. Confidence Intervals for Private Query Processing. PVLDB, 17(3): 373 - 385, 2023.
doi:10.14778/3632093.3632102

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/PrivateCI/DP_CI.

1 INTRODUCTION

Differentially private (DP) query processing mechanisms have been developed for many important and fundamental problems such as mean/median [20, 28, 38], graph statistics [32, 38, 41], and (certain classes of) SQL queries [13, 29, 39]. However, all these mechanisms only return a noisy query answer. For instance, if a data analyst is interested in the number of orders completed this year such that the customer is from Asia, s/he may ask the following query (assuming the TPC-H schema):

```
SELECT count(*)
FROM Region, Nation, Orders, Customer
WHERE Orders.Orderdate > 2023-01-01
AND Region.Name = 'ASIA' AND Region.RK = Nation.RK
AND Nation.NK = Customer.NK AND Customer.CK = Orders.CK;
```

Current DP mechanisms only return a single value, say 101.2, which leaves the data analyst in limbo: S/he knows that the answer is inaccurate, but has no idea *how* inaccurate it is, thus cannot use the query answer reliably or make decisions with any confidence.

The most common technique for resolving the issue is to provide a *confidence interval (CI)*, namely, an interval that contains the true query answer with probability at least $1-\beta$ for some given β. Back to the example above, if we can return a CI, say, $[99, 102]$ with $\beta = 0.05$, then we know our result has high accuracy and this would allow downstream applications to interpret the result with more reliability and provenance. Indeed, in the non-private settings, CIs are a standard requirement for *approximate query processing (AQP)* systems, which often use sampling to reduce the query processing cost [1, 8, 27, 35]. When the returned CI is considered not accurate enough, the data analyst can instruct the AQP system to increase the sample size, hence a larger computational cost, to reduce the CI to a level of satisfaction.

In this paper, we study the CI problem in a DP query processing system. Here, the CI would play a role as important as that for AQP systems: It injects statistical significance into the noisy query answer, allows more confidence in decision-making, and when it is not accurate enough, the data analyst may decide to spend more privacy budget to improve its accuracy or abandon the task to avoid making risky decisions.

Our starting observation is that, if the CI can be derived from the output of the DP mechanism using only public information, then solving the CI problem is straightforward. This is the case for some classical DP mechanisms, such as the Laplace mechanism and the Gaussian mechanism. For instance, the Laplace mechanism outputs $\mathcal{M}(D) := Q(D) + \mathrm{GS}_Q/\varepsilon \cdot \mathrm{Lap}(1)$, where $Q(D)$ is the true query result, GS_Q is the *global sensitivity* (formal definition in Section 2) of the query $Q(\cdot)$, and $\mathrm{Lap}(1)$ denotes a random variable drawn from the unit Laplace distribution. Using a tail bound on the noise distribution (see Lemma 2.4), it immediately follows that $[\mathcal{M}(D) - \mathrm{GS}_Q/\varepsilon \cdot \log(1/\beta), \mathcal{M}(D) + \mathrm{GS}_Q/\varepsilon \cdot \log(1/\beta)]$ is a $(1-\beta)$-CI for $Q(D)$. More generally, a CI can be obtained by evaluating the cumulative density function (CDF) of the noise distribution at $\beta/2$ and $1-\beta/2$. This still satisfies DP, since it can be considered as a post-processing step on $\mathcal{M}(D)$. Most importantly, this works because the scale of the noise distribution only depends on public information, i.e., GS_Q, ε, and β.

However, the global sensitivity for many problems is large or even unbounded. A prototypical example is the median problem:

Proceedings of the VLDB Endowment, Vol. 17, No. 3 ISSN 2150-8097.
doi:10.14778/3632093.3632102

just consider the neighboring pairs of datasets $(\underbrace{0,\ldots,0}_{n/2+1},\underbrace{N,\ldots,N}_{n/2})$ and $(\underbrace{0,\ldots,0}_{n/2},\underbrace{N,\ldots,N}_{n/2+1})$. This means that $\mathrm{GS}_Q = N$ if all elements are taken from the domain $[N] = \{0, 1, \ldots, N\}$, and $\mathrm{GS}_Q = \infty$ if the domain is unbounded, say $\mathbb{N}$ or $\mathbb{Z}$. The issue is more prominent in relational databases: Any query with at least one join has unbounded GS_Q [13, 14].

To deal with such queries, the literature has identified three main techniques: the exponential mechanism (EM), the sparse vector technique (SVT), and the smooth sensitivity (SS). Each one can be applied to a class, but not all, of such queries, with different and sometimes incomparable utility guarantees. In particular, they can all be applied to the median problem, which is often used as a "testbed" for these techniques. In particular, all these non-GS based mechanisms use instance-specific noise distributions [11, 13, 38, 41], namely their accuracy depends on the input data, which is private. For example, on input D, the EM draws the output from a distribution where the probability of returning each y is proportional to $\exp\left(\frac{\varepsilon u(D,y)}{2\Delta u}\right)$, where $u(D, y)$ is a utility function that assigns a score to each y on input D (more details given in Section 3). It is clear that this noise distribution depends on D. For the EM mechanism, it is not possible to produce a CI with only public information. In fact, the fundamental reason why EM, as well as SVT and SS, can overcome the issue of a large GS_Q is to use a noise distribution whose scale (and possibly some other properties) varies according to the hardness of the input. For example, the pair of inputs for the median problem mentioned above are hard, and a noise scale $\Omega(N)$ is needed to make them indistinguishable, but most real-world instances do not require such a large noise. However, this introduces a challenge for the CI problem. On the one hand, the naive CI that simply adds $\pm\mathrm{GS}_Q$ to the output of the EM satisfies DP, but it loses all utility. On the other hand, setting the CI proportional to the scale of the noise distribution violates DP as it depends on D.

1.1 Our Contributions

In this paper, for each of the three general techniques, we design mechanisms to produce confidence intervals that are

(1) differentially private;
(2) correct, i.e., the CI contains $Q(D)$ with probability $1 - \beta$; and
(3) have a utility guarantee matching that of the original mechanism, up to constant factors.

Note that point (3) above depends on the particular form of guarantee. For example, EM has the utility guarantee that the utility score of the output is smaller than the highest score by at most $O(\Delta u/\varepsilon \cdot \log(N/\beta))$ with probability $1 - \beta$. Then, our EMCI mechanism (Section 3) returns a CI such that *every* point inside the CI has such a high score (up to a constant factor). The utility guarantees of our CIs for SVT and SS have a similar nature, and the exact statements can be found in Section 4 and 5, respectively. Meanwhile, it is easy to see that (3) is the best one can hope for: If the CI had a better utility, then we would be improving the original mechanism as well.

Next, for each technique, we show how our CI mechanisms can be applied to some representative applications, as listed in Table 1. In particular, they can all be applied to the median problem, but with different utility guarantees.

Finally, in Section 6 we present an experimental evaluation, in which we compare the utility of the CIs with the actual errors of the corresponding mechanisms. Note that the CI cannot be better than the actual error, by the definition of a CI. However, the actual errors cannot be published but the CIs can. Thus, the goal of the experiments is to gauge the (constant-factor) gap between the CI and best theoretical possible.

1.2 Empirical Setting and Statistical Setting

Differentially private confidence intervals have previously been studied in the *statistical setting* [5, 10, 17, 18, 26, 31] on some parameter estimation problems such as mean estimation. There, the input dataset D is assumed to be an i.i.d. sample drawn from some underlying distribution $\mathcal{P}$ with some unknown parameter θ, and the goal is to derive a CI for θ. The correctness and utility of the CI rely on statistical properties of $\mathcal{P}$ and the CI should consider randomness in both D and the mechanism.

On the other hand, the focus of this paper is in the *empirical setting*, namely, we do not assume that D is random and the goal is to provide a CI on $Q(D)$ for any given D. In this setting, the only randomness is from the DP mechanism. One can see the empirical CI problem will no longer exist without DP requirement since in that case we can output the real query result which is 100% accurate.

These two settings differ by how they view the data. For instance, if we want to estimate the average score of students in a school, where D consists of a random sample of the students and the scores follow some distribution, e.g., Gaussian, then it is more appropriate to use the statistical setting. However, if we compute the average stock price of a company in the past 5 years, where D consists of daily prices, then the statistical setting will no longer be meaningful since stock prices do not follow any distribution and are certainly not i.i.d. We emphasize that the empirical setting is more appropriate for relational databases; in fact, for most problems studied in this paper, e.g., graph pattern counting and SQL queries, DP mechanisms have not been studied in the statistical setting, because there are no generally agreed statistical models for these problems.

While the statistical setting and empirical setting are two different problems, we observe that some existing statistical mean CI mechanisms [31] can be modified to support the empirical setting. We describe the modification and compare it with our CI mechanisms experimentally in Section 6.2. However, the mechanism on median CI [10, 17] in the statistical setting cannot be easily modified to support the empirical setting.

1.3 Other Related Work

We have derived CIs for the three most important techniques for handling queries with large/unbounded global sensitivity: EM, SVT, and SS. Another popular technique is the *propose-test-release* framework [20]. It first proposes a bound, say b, on the local sensitivity, and test (in a DP fashion) if the input D is far from any instance that violates the bound. If yes, it releases the query output using the Laplace mechanism with noise scale b. Because b here is public

Table 1: Summary of different techniques, here $\mathrm{rad}(D)$ is the radius of D and $\mathrm{SS}(D)$ is the smooth sensitivity, detailed definitions are given later.

CI Technique	Median	Other applications
EM	Bounded domain with $O(\frac{\log N}{\varepsilon})$ rank error	Graph pattern counting with special patterns (e.g., triangle, k-star)
SVT	Unbounded domain with $O(\frac{\log \mathrm{rad}(D)}{\varepsilon})$ rank error	Radius (maximum), mean (sum)
SS	Bounded domain with $O(\frac{\mathrm{SS}(D)}{\varepsilon})$ value error	Conjunctive queries

information, one can still derive a CI easily in the same manner as for the Laplace mechanism.

The machine learning community is interested in many high-dimensional problems [7, 25, 28]. A confidence interval, in high dimensions, becomes generally a *confidence set*. Our mechanisms can still be applied by deriving a CI in each dimension. However, this may not yield the tightest confidence set, as it is restricted to axis-parallel boxes. How to derive tighter confidence sets in high dimensions is an interesting open problem.

2 PRELIMINARIES

2.1 Notation

A database instance D is a multiset of n tuples. Depending on the problem, the tuples may take different forms. For simple aggregation queries like mean, sum, or median, each tuple is an integer, and we write $D = \{x_1, \ldots, x_n\}$ and reorder D such that $x_1 \le \cdots \le x_n$. We further distinguish the bounded-domain case, where each x_i is taken from $[N] = \{0, 1, \ldots, N\}$, as well as the unbounded domain $\mathbb{Z}$. We will use the following notation: $\mathrm{Count}(D, [a, b]) := |D \cap [a, b]|$ (multiplicity is considered during counting); $\mathrm{rad}(D) := \max_i |x_i|$ is the radius of D; $\gamma(D) = x_n - x_1$ is the width; $\mathrm{abs}(D) := \{|x_i| : x_i \in D\}$ (multiplicity is preserved); $D + a$ is the multiset obtained by adding a to all elements of D; $D - a$ is defined similarly.

For graph pattern counting queries or conjunctive queries, D is a graph with n edges or a database containing multiple relations with a total of n tuples, respectively. Note that the former is a special case of the latter where we use just one relation to store all the edges.

We use $\mathcal{D}$ to denote the input domain of all database instances, i.e., $\mathcal{D} = [N]^n$ ($\mathcal{D} = \mathbb{Z}^n$ for the unbounded-domain case), all graphs with n edges, or all database instances with n tuples, respectively. While we consider these varieties of input domains, we require the output domain of the query to be one-dimensional, where the confidence interval is really an "interval". Specifically, for mechanisms that require a finite output domain (e.g., the exponential mechanism), we use $[N]$; for mechanisms on an infinite but discrete domain (e.g., SVT), we use $\mathbb{Z}$; otherwise we take the output domain to be $\mathbb{R}$ (e.g., the Laplace mechanism and smooth sensitivity). All log's have base e unless specified otherwise.

2.2 Differential Privacy

Definition 2.1 (Differential privacy (DP [21, 22])). For $\varepsilon > 0$, an algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ is ε-differentially private, or simply ε-DP, if for any neighboring database instances $D \sim D' \in \mathcal{D}$ and any $y \in \mathcal{Y}$,

$$\Pr[\mathcal{M}(D) = y] \le e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') = y].$$

In this paper, two database instances D and D' are said to be *neighbors*, denoted $D \sim D'$, if they differ by one tuple. This is known as replacement-DP where the number of tuples n is regarded as public information.

For the CI problem, the mechanism will return the two endpoints of the interval, and the joint distribution of the two endpoints must satisfy the DP requirement above. The (adaptive) composition property below simplifies the analysis, so that we can focus on designing a mechanism for each endpoint while splitting the privacy budget:

Lemma 2.2 (Adaptive Composition [22]). *Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{Y}_1$ and $\mathcal{M}_2 : \mathcal{D} \times \mathcal{Y}_1 \to \mathcal{Y}_2$ be two mechanisms. If $\mathcal{M}_1(\cdot)$ satisfies ε_1-DP and $\mathcal{M}_2(\cdot, y)$ satisfies ε_2-DP for any $y \in \mathcal{Y}_1$, then the mechanism $\mathcal{M}(D) := (\mathcal{M}_1(D), \mathcal{M}_2(D, \mathcal{M}_1(D)))$ satisfies $(\varepsilon_1 + \varepsilon_2)$-DP.*

A useful special case of this lemma is when $\varepsilon_2 = 0$, which implies that $\mathcal{M}_2(\cdot, y)$ only depends on y. In this case, the lemma degenerates into the *post-processing* property of DP. Furthermore, the lemma easily extends to the case where any number of DP mechanisms are composed.

For any query $Q : \mathcal{D} \to \mathbb{R}$, its *local sensitivity* on database instance D is

$$\mathrm{LS}_Q(D) = \sup_{D' : D \sim D'} |Q(D) - Q(D')|,$$

and its *global sensitivity* is

$$\mathrm{GS}_Q = \sup_D \mathrm{LS}_Q(D).$$

When the query Q is clear from the context, we omit the subscript Q and simply write $\mathrm{LS}(D)$ and GS.

One standard technique for achieving DP is to add a Laplace noise with scale proportional to GS before releasing $Q(D)$:

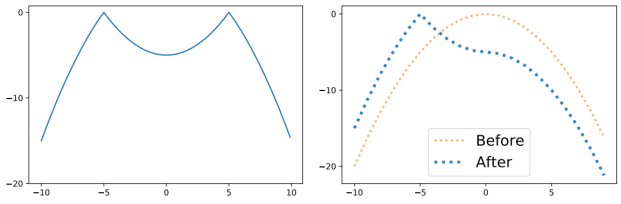
Lemma 2.3 (Laplace Mechanism [24]). *The mechanism*

$$\mathcal{M}(D) := Q(D) + \mathrm{GS}/\varepsilon \cdot \mathrm{Lap}(1)$$

preserves ε-DP, where $\mathrm{Lap}(1)$ denotes a random variable drawn from the unit Laplace distribution.

The following lemma gives the high probability bound on the magnitude of Laplace noise, immediately yielding a CI for the Laplace mechanism:

Lemma 2.4 (Laplace tail bound). *If $X \sim \mathrm{Lap}(1)$, then $\Pr(|X| > \log \frac{1}{\beta}) \le \beta$.*



(a) A counter-example of well-behaved utility function.

(b) Partially flipping the utility function assuming $u_{\max}(D) = 0, s = 4, l = 1$.

Figure 1

3 EXPONENTIAL MECHANISM

The *exponential mechanism (EM)* [36] is a general DP mechanism. For a utility function $u : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ that assigns a utility score to each output $y \in \mathcal{Y}$ on instance $D \in \mathcal{D}$, EM returns a y with high utility in a DP manner. More precisely, it samples $y \in \mathcal{Y}$ with probability proportional to $\exp(\frac{\varepsilon u(D,y)}{2\Delta u})$, where $\Delta u := \sup_{y \in \mathcal{Y}} \sup_{D', D'' \in \mathcal{D}, D' \sim D''} |u(D', y) - u(D'', y)|$. The utility function is often designed in a way such that $u(D, y)$ maximizes at $y = Q(D)$, so that EM returns $Q(D)$ with the highest probability.

Denote by $\mathrm{EM}(D, u, \varepsilon)$ the output of the exponential mechanism on input D with utility function u. It is known to satisfy ε-DP and has the following utility guarantee:

Lemma 3.1 ([36]). *With probability at least $1 - \beta$,*

$$u(D, \mathrm{EM}(D, u, \varepsilon)) \geq u(D, y^*(D)) - \frac{2\Delta u}{\varepsilon} \log \frac{|\mathcal{Y}|}{\beta},$$

where $y^*(D) = \arg\max_{y \in \mathcal{Y}} u(D, y)$.

Note that Lemma 3.1 requires a finite $\mathcal{Y}$. Thus we take $\mathcal{Y} = [N]$ in this section.

3.1 Private CI for General EM

In this subsection, we describe EMCI, a mechanism for producing a private confidence interval for the general EM. EMCI will return a CI $[L(D), R(D)]$ that (1) satisfies ε-DP, (2) is correct, i.e., $\Pr[y^*(D) \in [L(D), R(D)] \geq 1 - \beta$, and (3) has a utility guarantee that is as good as the EM itself (up to a constant factor), i.e., every $y \in [L(D), R(D)]$ has high utility.

EMCI works with EM instantiated using any *well-behaved* utility function:

Definition 3.2 (Well-behaved utility function). A utility function u is well-behaved if it has the following properties:

(1) Unimodality: For any D, $u(D, y)$ is unimodal reaching maximum value at $y = y^*(D)$, i.e., $u(D, y)$ is non-decreasing in $[0, y^*(D)]$ and non-increasing in $[y^*(D), N]$.
(2) Non-triviality: For any D, there exist $y_1 < y^*(D) < y_2$ such that $u(D, y_1)$ and $u(D, y_2)$ are smaller than $u(D, y^*(D)) - \frac{9\Delta u}{\varepsilon} \log \frac{2N}{\beta}$. Otherwise simply returning $[0, N]$ already meets all three desiderata above.
(3) Continuity: $u(D, y)$ is l-Lipschitz in y for some l, i.e., $|u(D, y+1) - u(D, y)| \leq l$ for any D, y.

Figure 1a provides a counter-example of well-behaved utility function. However, we are not aware of any instantiation of the EM where the utility function has more than one maximum like in this counter-example. Furthermore, note that the constraints (2) and (3) in the above definition only affect the CI width and do not affect its correctness of returned CI.

In general, given an arbitrary utility function u, it may not be easy to verify whether it is well-behaved. However, most utility functions used for EM in reality have nice properties. For the two specific instantiations considered below, we can easily prove that they are well-behaved.

The intuition behind EMCI is as follows. To find $L(D)$, we run another EM with a different utility function $u_L(D, y)$ such that it (1) has small sensitivity; and (2) peaks somewhere on the left side of but close to $y^*(D)$, while taking small values for $y \geq y^*(D)$. Thus, this EM can return an $L(D) = \mathrm{EM}(D, u_L, \frac{\varepsilon}{2})$ that is a close lower bound of $y^*(D)$. Then symmetrically, we use the other half of the privacy budget to find $R(D)$.

We construct such an u_L by partially "flipping" the original u. Let $s := \frac{9\Delta u}{\varepsilon} \log \frac{2N}{\beta}$ and $u_{\max}(D) := u(D, y^*(D))$. We define

$$u_L(D, y) = \begin{cases} -|u(D, y) - u_{\max}(D) + s + l|, & \text{if } y \leq y^*(D); \\ u(D, y) - u_{\max}(D) - s - l, & \text{if } y > y^*(D). \end{cases} \quad (1)$$

See Figure 1b for an illustration.

Symmetrically, to find $R(D)$, we use the utility function

$$u_R(D, y) = \begin{cases} u(D, y) - u_{\max}(D) - s - l, & \text{if } y \leq y^*(D); \\ -|u(D, y) - u_{\max}(D) + s + l|, & \text{if } y > y^*(D). \end{cases} \quad (2)$$

It can be shown that u_L and u_R have low sensitivity. For the proof please refer to the full version [40].

Lemma 3.3. *For any $\beta, \varepsilon < \frac{9 \log(2N/\beta)}{2}$ and any well-behaved utility function $u(D, y)$, we have $\Delta u_L, \Delta u_R \leq 2\Delta u$.*

Then EMCI simply returns the confidence interval $[L(D), R(D)]$, where $L(D) = \mathrm{EM}(D, u_L, \frac{\varepsilon}{2})$ and $R(D) = \mathrm{EM}(D, u_R, \frac{\varepsilon}{2})$. Now we prove the main result on EMCI:

Theorem 3.4. *Given $\beta, \varepsilon < \frac{9 \log(2N/\beta)}{2}$, and any well-behaved utility function $u(D, y)$ that is l-Lipschitz, EMCI is ε-DP, and on any input D, it returns a confidence interval $[L(D), R(D)]$ such that with probability at least $1 - \beta$,*

$$y^*(D) \in [L(D), R(D)],$$

and for all $y \in [L(D), R(D)]$,

$$u(D, y) \geq u(D, y^*(D)) - \frac{17\Delta u}{\varepsilon} \log \frac{2N}{\beta} - 2l.$$

Proof. Privacy directly follows from the privacy of the EM and basic composition. Next, we prove correctness and utility.

Let $y_L^*(D)$ be the smallest integer such that

$$u_{\max}(D) - s - l \leq u(D, y_L^*(D)) \leq u_{\max}(D) - s.$$

Note that such a $y_L^*(D)$ must exist for any well-behaved u. According to lemma 3.1 and 3.3, with probability at least $1 - \frac{\beta}{2}$:

$$\begin{aligned} u_L(D, L(D)) &\geq \max_{y \in \mathcal{Y}} u_L(D, y) - \frac{8\Delta u}{\varepsilon} \log \frac{2N}{\beta} \\ &\geq u_L(D, y_L^*(D)) - \frac{8\Delta u}{\varepsilon} \log \frac{2N}{\beta} \\ &\geq -\frac{8\Delta u}{\varepsilon} \log \frac{2N}{\beta} - l. \end{aligned}$$

On the other hand, according to our design, when $y \geq y^*(D)$, $u_L(D, y) \leq -\frac{9\Delta u}{\varepsilon} \log \frac{2N}{\beta} - l$. So we know $L(D) < y^*(D)$. What's more,

$$\begin{aligned} u(D, L(D)) &\geq u(D, y^*(D)) + u_L(D, L(D)) - s - l \\ &\geq u(D, y^*(D)) - \frac{17\Delta u}{\varepsilon} \log \frac{2N}{\beta} - 2l. \end{aligned}$$

Similarly, $R(D) > y^*(D)$ and

$$u(D, R(D)) \geq u(D, y^*(D)) - \frac{17\Delta u}{\varepsilon} \log \frac{2N}{\beta} - 2l$$

with probability at least $1 - \frac{\beta}{2}$. In summary, $y^*(D)$ falls inside the interval $[L(D), R(D)]$ with probability at least $1-\beta$ and both ending points have utility score at least $u(D, y^*(D)) - \frac{17\Delta u}{\varepsilon} \log \frac{2N}{\beta} - 2l$. Then all y in $[L(D), R(D)]$ have high utility due to the unimodality of u. □

Here we allocate the privacy budget evenly to find the left/right bounding point. Without further assumption, this is the simplest choice and maximizes the total utility score on both ending points of the CI.

3.2 Private CI for Median

We describe two applications of EMCI. The first is the median problem over a finite domain $[N]$, namely, the query $Q(D)$ returns the $\lceil n/2 \rceil$-th smallest element in D. We use the negative path length as the utility function : $u(D, y) = -\operatorname{len}(Q, D, y)$, where

$$\operatorname{len}(Q, D, y) := \min_{D'} \left\{ d\left(D, D'\right) : Q\left(D'\right) = y \right\}.$$

Here $d(D, D')$ denotes the minimal number of tuples that have to be changed to transform D to D'. This instantiation of EM had been a folklore since [9, 36], and was generalized in [2], who termed it the *inverse sensitivity (INV)* mechanism. For this utility function, we have $\Delta u = 1$. Then Lemma 3.1 immediately implies a *rank error* of $\frac{2}{\varepsilon} \log(N/\beta)$, i.e., the rank of the returned median is between $\lceil n/2 \rceil \pm \frac{2}{\varepsilon} \log(N/\beta)$ with probability $1 - \beta$ [3]. However, a technical issue with this $u(D, y)$ is that it has $l = \Omega(n)$. For example, if D consists of n 0's, then $u(D, 0) = 0$ but $u(D, 1) = -\lceil n/2 \rceil$.

There is a simple solution to address this issue. First, if it is guaranteed that D does not contain duplicates, then we have $l = 1$, and Theorem 3.4 implies that the rank error of any element in the CI is $O(\log(N/\beta)/\varepsilon)$. If D may contain duplicates, then we de-duplicate it by expanding the domain from $[N]$ to $[n \cdot N]$. Specifically, each $D \in [N]^n$ is mapped to a $\hat{D} \in [n \cdot N]^n$, where the repetitions of a value $x \in D$ are mapped to $nx, nx + 1, \ldots, nx + k - 1$, if x has k repetitions in D. Note that any pair of neighboring instances remain neighbors after the mapping, so privacy is not affected. Therefore, EMCI can be applied on $\hat{D}$ with $l = 1$. Finally, we transform the CI back to the original domain after dividing by n and rounding down. The rank error becomes $O(\log(nN/\beta)/\varepsilon)$ as $|\mathcal{Y}| = nN$ after the mapping.

EMCI can be implemented efficiently in $O(n)$ time given a sorted D. First, the mapping from D to $\hat{D}$ can be easily done in $O(n)$ time. The utility function u can be then be constructed in $O(n)$ time: Note that there is no need to compute $u(D, y)$ for each $y \in [nN]$. The n elements of $\hat{D}$ partition the domain into $n + 1$ segments, and the utility score remains the same within each segment. Similarly, u_L and u_R can also be constructed in $O(n)$ time. To run EM using u_L or u_R, we first sample a segment with probability proportional to the segment length times its utility score, which takes $O(n)$ time. Then, we return an element inside the segment uniformly at random.

3.3 Graph Pattern Counting

Zhang et al. [41] applied EM on the graph pattern counting problem, i.e., given a graph G and a pattern P, say, a triangle, count the number of subgraphs of G that are isomorphic to P. Two graphs are neighbors if they differ by one edge. The utility function used, which they call the *ladder function*, is based on $\mathrm{LS}(G, t)$, *local sensitivities at distance t*. More precisely, $\mathrm{LS}(G, t)$ is defined as $\max_{G': d(G,G') \leq t} \mathrm{LS}(G')$, i.e., the largest local sensitivity attained on any graph having at most t different edges from G. Then the utility function (ladder function) is defined as follows:

- $u(G, Q(G)) := 0$, where $Q(G)$ is the true count;
- for every $s \geq 0$, $u(G, y) := -s - 1$ for all y such that $|y - Q(G)| \in \left(\sum_{t=0}^{s-1} \mathrm{LS}(G, t), \sum_{t=0}^{s} \mathrm{LS}(G, t)\right]$.

It has been shown that this utility function has sensitivity $\Delta u = 1$ [41]. Additionally, it is easy to see u is $|P|$-Lipschitz, where $|P|$ denotes the number of edges in the pattern P.

For the graph pattern counting problem, the output domain is $[N]$ for $N = n^{|P|}$. Then Theorem 3.4 means that the utility of every y in the returned CI is at least

$$u(G, y^*(G)) - O\left(|P| \log \frac{n}{\beta}\right) - O(|P|) \geq u(G, y^*(G)) - O\left(|P| \log \frac{n}{\beta}\right),$$

which is as good as the utility guarantee of EM itself, up to constant factors.

Unfortunately, computing $\mathrm{LS}(G, t)$, hence the utility function above, takes exponential time for an arbitrary pattern (even taking the pattern size as a constant). Past work has identified some special patterns for which $\mathrm{LS}(G, t)$ can be computed in polynomial time, such as as triangle [38] and k-star [30]. Thus, EMCI also runs in polynomial time for these patterns.

4 SPARSE VECTOR TECHNIQUE

Suppose each query $Q_i : \mathcal{D} \to \mathbb{R}$ has global sensitivity 1. The *sparse vector technique (SVT)* [23] runs over a sequence of such queries, $Q_1, Q_2, \ldots$ with a threshold T. It aims to find the first index $i^*(D)$ such that $Q_{i^*(D)}(D) \geq T$ (assuming that such an $i^*(D)$ exists). The algorithm is shown in Algorithm 1. Note that SVT even supports an infinite query sequence; in this case, the output domain is $\mathcal{Y} = \mathbb{Z}^+$.

It is known that SVT satisfies ε-DP with the following utility guarantees:

Algorithm 1: SVT.

Input: $T, \varepsilon, Q_1(D), Q_2(D), \ldots$

$\tilde{T} \leftarrow T + \text{Lap}(2/\varepsilon)$;
for $i \leftarrow 1, 2, \ldots$ **do**
  $\tilde{Q}_i(D) \leftarrow Q_i(D) + \text{Lap}(4/\varepsilon)$;
  if $\tilde{Q}_i(D) > \tilde{T}$ **then** Break ;
end
return i;

Lemma 4.1 ([24]). *Let k_1 be the largest such that $Q_i(D) \le T - \frac{8}{\varepsilon}\log(2k_1/\beta)$ for all $1 \le i \le k_1$, assuming it exists. With probability at least $1-\beta$, SVT returns an $i > k_1$.*

Lemma 4.2 ([16]). *Let k_2 be the smallest such that $Q_{k_2}(D) \ge T + \frac{6}{\varepsilon}\log(2/\beta)$, assuming it exists. With probability at least $1-\beta$, SVT returns an $i \le k_2$.*

Note that these utility guarantees depend on k_1, k_2, which are highly sensitive: One can easily design neighboring instances on which k_1, k_2 could change arbitrarily. Thus, they cannot be published as a confidence interval for $i^*(D)$.

4.1 Private CI for SVT

We now present the mechanism SVTCI that returns a privatized CI $[i_L(D), i_R(D)]$ for $i^*(D)$ with high utility. The methods for finding the lower bound $i_L(D)$ and the upper bound $i_R(D)$ are slightly different. Besides, some applications of SVT that we will discuss later need only one of the two bounds, so we present the two subroutines in Algorithm 2 and 3 respectively, each using the full privacy budget ε and failure probability β. For applications that require both the lower and upper bound, each should be invoked with a privacy budget of $\varepsilon/2$ and failure probability $\beta/2$ instead.

Algorithm 2: SVTCI-L

Input: $T, \varepsilon, \beta, Q_1(D), Q_2(D), \ldots$

$T_{\text{noise}} = \text{Lap}(2/\varepsilon)$;
for $i \leftarrow 1, 2, \ldots$ **do**
  $\tilde{T}_i = T - \frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta} - \frac{2}{\varepsilon}\log\frac{2}{\beta} + T_{\text{noise}}$;
  $\text{Noise}_i = \text{Lap}(4/\varepsilon)$;
  $\tilde{Q}_i(D) \leftarrow Q_i(D) + \text{Noise}_i$;
  if $\tilde{Q}_i(D) \ge \tilde{T}_i$ **then** Break ;
end
return $i_L(D) := i$;

Similar to the original SVT, for each query Q_i, SVTCI-L (resp. SVTCI-R) compares a noisy $\tilde{Q}_i(D)$ with a noisy $\tilde{T}_i$, except that in SVTCI-L (resp. SVTCI-R), we compare with a $\tilde{T}_i$ that is less (resp. more) than $\tilde{T}$ by a $\Theta(\log(i/\beta)/\varepsilon)$ term. Intuitively, this allows the SVT to stop earlier (later) than $i^*(D)$ with high probability, thus obtaining a valid CI. This gap term is carefully designed in a way such that $i^*(D)$ falls inside $[i_L(D), i_R(D)]$ with probability $1-\beta$ while achieving high utility, as shown in the following theorem:

Algorithm 3: SVTCI-R

Input: $T, \varepsilon, \beta, Q_1(D), Q_2(D), \ldots$

$T_{\text{noise}} = \text{Lap}(2/\varepsilon)$;
for $i \leftarrow 1, 2, \ldots$ **do**
  $\tilde{T}_i = T + \frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta} + \frac{2}{\varepsilon}\log\frac{2}{\beta} + T_{\text{noise}}$;
  $\text{Noise}_i = \text{Lap}(4/\varepsilon)$;
  $\tilde{Q}_i(D) \leftarrow Q_i(D) + \text{Noise}_i$;
  if $\tilde{Q}_i(D) \ge \tilde{T}_i$ **then** Break ;
end
return $i_R(D) := i$;

Theorem 4.3. *Given any T, ε, β, and any sequence of sensitivity-1 queries $Q_1, Q_2, \ldots$, SVTCI-L (resp. SVTCI-R) is ε-DP, and on any input D, it returns $i_L(D)$ (resp. $i_R(D)$) such that with probability at least $1-\beta$, for all $i < i_L(D)$ (resp. $i_R(D)$),*

$$Q_i(D) < T, \text{ which implies } i_L(D) \le i^*(D); \text{ and}$$
$$Q_{i_L(D)}(D) \ge T - \frac{8}{\varepsilon}\log\frac{i_L(D)^2\pi^2}{3\beta} - \frac{4}{\varepsilon}\log\frac{2}{\beta}. \quad (3)$$

$$\Big(\text{resp. } Q_{i_R(D)}(D) \ge T, \text{ which implies } i_R(D) \ge i^*(D); \text{ and}$$
$$Q_i(D) < T + \frac{8}{\varepsilon}\log\frac{i_R(D)^2\pi^2}{3\beta} + \frac{4}{\varepsilon}\log\frac{2}{\beta}. \Big) \quad (4)$$

Proof. We will only prove the theorem for SVTCI-L. A symmetric proof works for SVTCI-R.

For privacy, we observe that although SVTCI-L uses a different $\tilde{T}_i$ for each i, the same privacy proof for SVT in [23] still works. Fundamentally, the proof only uses the fact that the sequence of $\tilde{T}_i$'s are $(\varepsilon/2)$-indistinguishable on two neighboring datasets, which is still the case in SVTCI-L.

Below prove correctness and utility. First, we know with probability at least $1 - \frac{\beta}{2}$, $|T_{\text{noise}}| \le \frac{2}{\varepsilon}\log\frac{2}{\beta}$. And also for any $i \in \mathbb{Z}^+$, we know with probability at least $1 - \frac{3\beta}{\pi i^2}$, the noise on $Q_i(D)$ has magnitude bounded by $\frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta}$. Since $\sum_{i=1}^{\infty}\frac{1}{i^2} = \frac{\pi^2}{6}$, combined with a union bound, we know that with probability at least $1 - \frac{\beta}{2}$ we have

$$|\text{Noise}_i| = |Q_i(D) - \tilde{Q}_i(D)| \le \frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta}$$

holds simultaneously for all $i \in \mathbb{Z}^+$. So if the algorithm returns $i_L(D)$, we know for all $i \in [i_L(D) - 1]$, $\tilde{Q}_i(D) < \tilde{T}_i$, which implies with probability at least $1-\beta$

$$\begin{aligned} Q_i(D) &< T - \frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta} - \frac{2}{\varepsilon}\log\frac{2}{\beta} + T_{\text{noise}} - \text{Noise}_i \\ &\le T - \frac{4}{\varepsilon}\log\frac{i^2\pi^2}{3\beta} - \frac{2}{\varepsilon}\log\frac{2}{\beta} + |T_{\text{noise}}| + |\text{Noise}_i| \\ &\le T \end{aligned}$$

Since i are integers, we know $i_L(D) \le i^*(D)$. Similarly, we have $Q_{i_L(D)}(D) \ge T - \frac{8}{\varepsilon}\log\frac{i_L(D)^2\pi^2}{3\beta} - \frac{4}{\varepsilon}\log\frac{2}{\beta}$. □

Algorithm 4: SVTRadius.

Input: ε, β, D

$D' \leftarrow \text{abs}(D)$;

$\tilde{i} \leftarrow$ SVTCI-L $(n, \varepsilon, \beta, Q_1(D'), Q_2(D'), \ldots$ as defined$)$;

if $\tilde{i} = 1$ **then**

  $\widetilde{\text{rad}}(D) = 0$

else

  $\widetilde{\text{rad}}(D) = 2^{\tilde{i}-1}$

end

return $\widetilde{\text{rad}}(D)$;

To compare the utility guarantees of SVT-CI with those of the SVT, we compare (3) with Lemma 4.1 and (4) with Lemma 4.2. Lemma 4.1 says that SVT will stop only after $Q_i(D)$ gets within a distance of $O(\log(i/\beta)/\varepsilon)$ to T, which is essentially the same as (3) up to constant factors. On the other hand, Lemma 4.2 says that SVT must have stopped before $Q_i(D)$ reaches $T + O(\log(1/\beta)/\varepsilon)$. For this case, (4) is slightly weaker by a logarithmic factor when $i_R(D) \gg 1/\beta$. Again, without additional public information, here the best choice is to allocate the privacy budget evenly on the left/right bounding point which minimizes the total error on both ending points of the CI.

4.2 Private CI for Radius

We present three applications of SVTCI. The first is to find the radius $\text{rad}(D) := \max_i |x_i|$ over an unbounded domain $\mathcal{D} = \mathbb{Z}^n$. Note that in the case where the elements in D are all non-negative, the problem is just the maximum problem. Specifically, we would like to find a privatized constant-factor approximation of $\text{rad}(D)$ (i.e., a confidence interval whose width is $O(\text{rad}(D))$), which in turn will be a building block for the subsequent applications.

The idea is to run SVTCI-L on the queries

$$Q_1(D) = \text{Count}(\text{abs}(D), [0, 0]),$$
$$Q_2(D) = \text{Count}(\text{abs}(D), [0, 2^0]),$$
$$Q_3(D) = \text{Count}(\text{abs}(D), [0, 2^1]), \ldots$$

with the threshold $T = n$. It is clear that the query sequence first reaches T at $i^*(D) = \lceil \log_2 \text{rad}(D) \rceil + 2$, so that $2^{i^*(D)-2}$ is a 2-approximation of $\text{rad}(D)$. According to Theorem 4.3, with high probability, SVTCI-L will return an $i_L(D) \le i^*(D)$, thus $2^{i_L(D)-3}$ is a valid lower bound on $\text{rad}(D)$. Meanwhile, the utility guarantee in Theorem 4.3 implies that this lower bound is not too loose in the sense that there are only $O(\log\log \text{rad}(D))$ elements (we take ε and β as constants in this intuitive explanation of the algorithm) in D are not covered by $[-2^{i_L(D)-2}, 2^{i_L(D)-2}]$. Unfortunately, this does not necessarily lead to a constant-factor approximation of $\text{rad}(D)$: If the data is very sparse near $\text{rad}(D)$, $2^{i_L(D)-2}$ can be arbitrarily smaller than $\text{rad}(D)$. We formalize this intuition into a density requirement in Theorem 4.4; later, we also prove that such density requirement is necessary for obtaining a constant-factor approximation of the radius privately.

The details of our private radius algorithm are shown in Algorithm 4, with its utility guarantee given in Theorem 4.4.

Algorithm 5: SVTMedianCI.

Input: ε, β, D

$L(D) \leftarrow$ SVTCI-L $\left(\lceil n/2 \rceil, \frac{\varepsilon}{2}, \frac{\beta}{2}, Q_1(D), Q_2(D), \ldots \text{as defined}\right)$;

$R(D) \leftarrow$ SVTCI-R $\left(\lceil n/2 \rceil, \frac{\varepsilon}{2}, \frac{\beta}{2}, Q_1(D), Q_2(D), \ldots \text{as defined}\right)$;

return $[L(D) - 1, R(D) - 1]$;

Theorem 4.4 (Density Requirement). *For any ε, β,* SVTRadius *satisfies ε-DP. For any $D \in \mathbb{Z}^n$ such that*

$$\text{Count}\left(\text{abs}(D), \left[\frac{\text{rad}(D)}{2}, \text{rad}(D)\right]\right) > \frac{24}{\varepsilon} \log\left(\frac{8\lceil \log_2 \text{rad}(D) \rceil}{\beta}\right), \tag{5}$$

with probability at least $1 - \beta$, SVTRadius *returns a $\widetilde{\text{rad}}(D)$ such that $\text{rad}(D) \le \widetilde{\text{rad}}(D) \le 4 \cdot \text{rad}(D)$. Then $[\frac{\widetilde{\text{rad}}(D)}{4}, \widetilde{\text{rad}}(D)]$ is a $1 - \beta$ confidence interval of $\text{rad}(D)$.*

Proof. The returned value $\widetilde{\text{rad}}(D)$ is simply a post-processing of the result of SVTCI-L, so privacy follows directly.

For utility, the case when $\widetilde{\text{rad}}(D) = 0$ is trivial. We consider when $\widetilde{\text{rad}}(D) \neq 0$. According to Theorem 4.3, with probability at least $1 - \beta$ we have $\tilde{i} \le \lceil \log_2(\text{rad}(D)) \rceil + 2$ and:

$$\begin{aligned}\text{Count}(\text{abs}(D), [0, 2^{\tilde{i}-2}]) &\ge n - \frac{8}{\varepsilon} \log \frac{(\tilde{i}-2)^2 \pi^2}{3\beta} - \frac{4}{\varepsilon} \log \frac{2}{\beta} \\ &\ge n - \frac{24}{\varepsilon} \log \frac{8\lceil \log_2 \text{rad}(D) \rceil}{\beta}\end{aligned}$$

On the other hand, by our density assumption (5), for any i such that $2^i < \frac{\text{rad}(D)}{2}$:

$$\text{Count}(\text{abs}(D), [0, 2^i]) < n - \frac{24}{\varepsilon} \log \frac{8\lceil \log_2 \text{rad}(D) \rceil}{\beta},$$

which means $\tilde{i} - 2 \ge \lceil \log_2 \frac{\text{rad}(D)}{2} \rceil$. Combine the arguments above, we have $\widetilde{\text{rad}}(D) = 2^{\tilde{i}-1} \in [\text{rad}(D), 4\text{rad}(D)]$. □

Remark. It is tempting to use SVTCI-R to get an upper bound on $\text{rad}(D)$. This will not work: Although Theorem 4.3 guarantees that $i_R(D)$ is a valid upper bound on $i^*(D)$, the utility guarantee is vacuous since all the queries in the query sequence have $Q_i(D) \le T$ by definition. So it may return an arbitrarily bad upper bound.

4.3 Private CI for Median

The second application is still the median problem, but now over an unbounded domain. We first consider the domain $\mathcal{D} = \mathbb{N}^n$, and discuss how to handle $\mathcal{D} = \mathbb{Z}^n$ later.

We simply invoke both SVTCI-L and SVTCI-R with the query sequence

$$Q_1(D) = \text{Count}(D, [0, 0]),$$
$$Q_2(D) = \text{Count}(D, [0, 1]),$$
$$Q_3(D) = \text{Count}(D, [0, 2]), \ldots$$

Details are given in Algorithm 5.

Theorem 4.5. *For any ε, β, n,* SVTMedianCI *satisfies ε-DP. Let $\alpha = \frac{40}{\varepsilon}\log(8\mathrm{rad}(D)/\beta)$ and assume $n > 2\alpha$. On any $D = \{x_1, \dots, x_n\} \in \mathbb{N}^n$, with probability at least $1-\beta$, we have $x_{\lceil n/2\rceil} \in [L(D)-1, R(D)-1]$ and $[L(D)-1, R(D)-1] \subseteq [x_{\lceil n/2\rceil-\alpha}, x_{\lceil n/2\rceil+\alpha}]$.*

Proof. Privacy follows from that of SVTCI-L and SVTVI-R and basic composition. Below we prove correctness and utility.

According to Theorem 4.3, correctness is obvious. Namely with probability at least $1-\beta$ we have $x_{\lceil n/2\rceil} \in [L(D)-1, R(D)-1]$ since $i^*(D) - 1 = x_{\lceil n/2\rceil}$.

For utility, when $n > 2\alpha$, we should have $L(D)-1, R(D)-1 \le \mathrm{rad}(D)$ so according to Theorem 4.3,

$$\begin{aligned}\mathrm{Count}(D, [0, L(D)-1]) &\ge T - \frac{16}{\varepsilon}\log\frac{2L(D)^2\pi^2}{3\beta} - \frac{8}{\varepsilon}\log\frac{4}{\beta}\\ &\ge \lceil n/2\rceil - \frac{16}{\varepsilon}\log\frac{2(\mathrm{rad}(D)+1)^2\pi^2}{3\beta} - \frac{8}{\varepsilon}\log\frac{4}{\beta}\\ &\ge \lceil n/2\rceil - \frac{40}{\varepsilon}\log(8\mathrm{rad}(D)/\beta) = \lceil n/2\rceil - \alpha\end{aligned}$$

thus $L(D) - 1 \ge x_{\lceil n/2\rceil-\alpha}$ and similarly,

$$\begin{aligned}\mathrm{Count}(D, [0, R(D)-2]) &< T + \frac{16}{\varepsilon}\log\frac{2R(D)^2\pi^2}{3\beta} + \frac{8}{\varepsilon}\log\frac{4}{\beta}\\ &\le \lceil n/2\rceil + \alpha\end{aligned}$$

Thus $R(D)-2 < x_{\lceil n/2\rceil+\alpha}$. Since D is in integer domain, although the value of $\mathrm{Count}(D, [0, R(D)-1])$ can be arbitrarily large, the rank of $R(D)-1$ is bounded. Namely, we have:

$$R(D) - 1 \le x_{\lceil n/2\rceil+\alpha}.$$

Combining the arguments above proves the theorem. □

To deal with $\mathcal{D} = \mathbb{Z}^n$, we can find a privatized radius $\widetilde{\mathrm{rad}}(D) \le 4\mathrm{rad}(D)$. Then we can call SVTMedianCI on $D' := D + \widetilde{\mathrm{rad}}(D)$. Finally, we shift the returned CI to the left by $\widetilde{\mathrm{rad}}(D)$.

Remark. Compared with EMCI, the rank error of SVTMedianCI is always no worse (up to constant factors), since $\mathrm{rad}(D) \le N$. Furthermore, unlike EMCI, SVTMedianCI supports an unbounded domain (i.e., $N = \infty$).

4.4 Private CI for Mean

The mean problem is to estimate $\mu(D) := \frac{1}{n}\sum_{i=1}^n x_i$. When the domain is bounded $\mathcal{D} = [N]^n$, the global sensitivity of $\mu(D)$ is also bounded: $\mathrm{GS} = N/n$. Then the Laplace mechanism immediately yields a $\tilde{\mu}(D)$ with error $O(N/n \cdot \log(1/\beta)/\varepsilon)$ with probability $1-\beta$, together with a CI of the same length. However, this has low utility for large N (e.g., $N = 2^{32}$); for an unbounded domain $\mathcal{D} = \mathbb{Z}^n$, this does not work at all.

To get around the issue, a standard technique is to clip all values in D into a bounded range $[l, r]$. More precisely, define

$$\mathrm{Clip}(x, [l, r]) = \max(\min(x, r), l)$$

Let

$$\mathrm{Clip}(D, [l, r]) = \{\mathrm{Clip}(x_i, [l, r]) \mid x_i \in D\}.$$

Then the clipped mean estimator is

$$\mathrm{ClippedMean}(D, [l, r]) = \mu(\mathrm{Clip}(D, [l, r])).$$

Algorithm 6: SVTMeanCI.

Input: ε, β, D

$\widetilde{\mathrm{rad}}(D) \leftarrow \mathrm{SVTRadius}(\frac{\varepsilon}{4}, \frac{\beta}{4}, \mathrm{abs}(D))$;

$D' \leftarrow D + \widetilde{\mathrm{rad}}(D)$;

$x' \leftarrow \text{SVTCI-L}\left(\frac{n}{2}, \frac{\varepsilon}{4}, \frac{\beta}{4}, Q_1(D'), \dots \text{ as defined in Sec 4.3}\right)$;

$D'' \leftarrow D' - x'$;

$\widetilde{\mathrm{rad}}(D'') \leftarrow \mathrm{SVTRadius}(\frac{\varepsilon}{4}, \frac{\beta}{4}, \mathrm{abs}(D''))$;

$\tilde{\mu}(D'') \leftarrow \mathrm{ClippedMean}\left(D'', [-\widetilde{\mathrm{rad}}(D''), \widetilde{\mathrm{rad}}(D'')]\right) + \mathrm{Lap}(8\widetilde{\mathrm{rad}}(D'')/\varepsilon n)$

$\alpha \leftarrow \frac{8\widetilde{\mathrm{rad}}(D'')\log\frac{4}{\beta}}{\varepsilon n}$;

return $[\tilde{\mu}(D'') + x' - \widetilde{\mathrm{rad}}(D) - \alpha, \tilde{\mu}(D'') + x' + \widetilde{\mathrm{rad}}(D) + \alpha]$;

It is obvious that $\mathrm{ClippedMean}(\cdot, [l, r])$ has global sensitivity $\mathrm{GS} = (r-l)/n$, on which the Laplace mechanism can be applied to achieve an error and CI of $\tilde{O}((r-l)/n)$. The optimal choices for l and r are $l = x_1, r = x_n$, resulting in an error of $O(\gamma(D)/n \cdot \log(1/\beta)/\varepsilon)$ where $\gamma(D) = x_n - x_1$ is the width of D. However, this violates DP since x_1, x_n are sensitive to D. Existing private mean estimators [16, 28] have achieved this error (up to some logarithmic factors), but they cannot return a privatized CI.

To find a privatized CI, we take the following steps (assuming the domain is $\mathcal{D} = \mathbb{Z}^n$):

(1) Use SVTRadius to find a privatized $\widetilde{\mathrm{rad}}(D)$.
(2) Shift D to $D' := D + \widetilde{\mathrm{rad}}(D)$, so that $D' \in \mathbb{N}^n$ (these two steps can be skipped if the domain is already $\mathbb{N}^n$).
(3) Use SVTCI-L to find a point x' inside D'.
(4) Shift D' to $D'' := D' - x'$, thus $\gamma(D) = \gamma(D'') = \Theta(\mathrm{rad}(D''))$.
(5) Use SVTRadius to find $\widetilde{\mathrm{rad}}(D'')$, use $\widetilde{\mathrm{rad}}(D'')$ to clip D''.
(6) Apply the Laplace mechanism and shift back the returned CI correspondingly.

The detailed algorithm SVTMeanCI is shown in Algorithm 6, and the following theorem shows that the length of the CI is optimal, under two mild conditions: A size requirement on n, and a density requirement that D is not too sparse in the first or the last quarter region of $[x_1, x_n]$.

Theorem 4.6. *Given ε, β, for any $D \in \mathbb{Z}^n$,* SVTMeanCI *satisfies ε-DP. If $n > \frac{80}{\varepsilon}\log(16\mathrm{rad}(D)/\beta)$ and*

$$\mathrm{Count}\left(D, \left[x_1, x_1 + \frac{\gamma(D)}{4}\right]\right) > \frac{96}{\varepsilon}\log\left(\frac{32\lceil\log_2 \mathrm{rad}(D)\rceil}{\beta}\right), \quad (6)$$

$$\mathrm{Count}\left(D, \left[x_n - \frac{\gamma(D)}{4}, x_n\right]\right) > \frac{96}{\varepsilon}\log\left(\frac{32\lceil\log_2 \mathrm{rad}(D)\rceil}{\beta}\right), \quad (7)$$

then with probability at least $1-\beta$, it returns a CI containing $\mu(D)$ and the interval length is $O(\gamma(D)/n \cdot \log(1/\beta)/\varepsilon)$.

For the proof please refer to the full version [40].

Note that SVTMeanCI needs two very mild density conditions (6), (7), that $\Omega(\log\log(\mathrm{rad}(D))/\varepsilon)$ points in D should fall in either quarter of the data range. It is attempting to modify SVTMeanCI by checking these conditions first, and simply output $(-\infty, \infty)$ if these conditions are not satisfied. However, we show that this violates DP. In fact, in the next subsection we prove a strong negative result (see

Theorem 4.7): There is no DP mechanism that is both always correct (i.e., it returns a correct CI on every dataset D) and nontrivial (i.e., it does not always return $(-\infty, \infty)$). If we modify SVTMeanCI so that it returns a good CI when (6), (7) are satisfied, and returns $(-\infty, \infty)$ when they are not, then this would be a mechanism that is both nontrivial and always correct. This contradicts with Theorem 4.7, so the modified mechanism must not be private.

4.5 Private CI for Mean: Lower bounds

Recall that Theorem 4.6 has two requirements. The first requirement $n = \Omega(\log(\mathrm{rad}(D))/\varepsilon)$ is in fact needed even for just obtaining any value in $[x_1, x_n]$ under DP, known as the *interior point* problem [4, 6]. Below, we argue that the density requirement is also necessary.

We first prove a strong negative result, which says that for any DP mechanism, if it returns a CI with a confidence level $\geq 2/3$ on every $D \in [N]^n$, then the length of the CI must be $\Omega(N/n)$, i.e., the same as the naive Laplace mechanism. In particular, this implies that for an unbounded domain, the CI returned can only be the trivial one $[-\infty, \infty]$.

Theorem 4.7. *For any ε-DP mechanism $\mathcal{M}$ with $\varepsilon < \log 2$, if $\mathcal{M}(D)$ outputs an interval $[L(D), R(D)]$ satisfying*

$$\Pr(\mu(D) \in [L(D), R(D)]) \geq \frac{2}{3},$$

for every $D \in [N]^n$, then

$$\Pr\left(R(D) - L(D) \geq \frac{N}{2n}\right) \geq \frac{2 - e^{\varepsilon}}{3}$$

for every D.

Theorem 4.7 means that some condition on D is needed for obtaining any nontrivial CI. Our next lower bound is more quantitative: It shows that in order to achieve the optimal CI width, the density requirement is also necessary. It is stated for the bounded domain case $\mathcal{D} = [N]^n$; in the unbounded domain case, it still holds by replacing N with $\mathrm{rad}(D)$.

Theorem 4.8. *Let $\mathcal{D}_{a,n} = \left\{D \in [N]^n \mid \mathrm{Count}(D, \frac{x_n}{2}, x_n) \geq a\right\}$. If there exists an ε-DP mechanism $\mathcal{M}$ such that for any $D \in \mathcal{D}_{a,n}$, $\mathcal{M}(D) = (L(D), R(D))$ satisfies*

$$\Pr(\mu(D) \in [L(D), R(D)]) \geq \frac{11}{12} \tag{8}$$

and

$$\Pr\left(R(D) - L(D) > \frac{\gamma(D)}{3\varepsilon n} \log\log_2 N\right) < \frac{1}{6}, \tag{9}$$

then $a = \Omega\left(\frac{\log\log_2 N}{\varepsilon}\right)$.

The proofs are omitted here due to space constraints, for details please refer to the full version [40].

5 SMOOTH SENSITIVITY

While GS might be large or unbounded, $\mathrm{LS}(D)$ is usually small for most real-world instances. However, Nissim et al. [38] pointed out that adding a noise scaled to $\mathrm{LS}(D)$ violates privacy, as $\mathrm{LS}(D)$ can be highly sensitive to D. In addressing this issue, they proposed *smooth sensitivity*, which is the third general technique for dealing with queries with a large or unbounded GS.

The idea is to smooth out $\mathrm{LS}(D)$ with an upper bound:

Algorithm 7: SSCI.

Input: ε, β, γ, $\widetilde{\mathrm{SS}}(\cdot)$, D

$\alpha = \frac{\varepsilon}{4(1+\gamma)}$;

calculate $\widetilde{\mathrm{SS}}(D)$;

$\tilde{Q}(D) = Q(D) + \frac{4(1+\gamma)\widetilde{\mathrm{SS}}(D)}{\varepsilon}\mathrm{Cauchy}(\gamma)$;

$\overline{\mathrm{SS}}(D) = \widetilde{\mathrm{SS}}(D) \cdot e^{\mathrm{Lap}(\frac{2\alpha}{\varepsilon}) + \frac{2\alpha}{\varepsilon}\log\frac{2}{\beta}}$;

$w = \frac{4(1+\gamma)\overline{\mathrm{SS}}(D)}{\varepsilon} F_{\gamma}^{-1}(1 - \beta/4)$;

return $[\tilde{Q}(D) - w, \tilde{Q}(D) + w]$;

Definition 5.1 ([38]). For $\alpha > 0$, a function $\widetilde{\mathrm{SS}} : \mathcal{D} \to \mathbb{R}$ is an *α-smooth upper bound* on $\mathrm{LS}(\cdot)$ if

$$\widetilde{\mathrm{SS}}(D) \geq \mathrm{LS}(D), \text{for all } D \in \mathcal{D}; \text{ and} \tag{10}$$

$$\widetilde{\mathrm{SS}}(D) \leq e^{\alpha} \cdot \widetilde{\mathrm{SS}}(D'), \text{ for all } D \sim D' \in \mathcal{D}. \tag{11}$$

It is shown [38] that any such $\widetilde{\mathrm{SS}}(\cdot)$ can be used to achieve DP, as follows.

Lemma 5.2 ([38]). *Fix any constant $\gamma > 1$, and let $\widetilde{\mathrm{SS}}(\cdot)$ be defined as above with $\alpha = \frac{\varepsilon}{2(1+\gamma)}$. The mechanism that outputs $\tilde{Q}(D) = Q(D) + 2(1+\gamma)\widetilde{\mathrm{SS}}(D)/\varepsilon \cdot \mathrm{Cauchy}(\gamma)$ is ε-differentially private, where $\mathrm{Cauchy}(\gamma)$ denotes a random variable drawn from the generalized Cauchy distribution with parameter γ.*

The generalized Cauchy distribution has a probability density function $f(x) \propto \frac{1}{1+|x|^{\gamma}}$, which is a well-defined distribution for any $\gamma > 1$. However, we often choose $\gamma > 3$ for it to have a bounded variance. The tail bound for the Cauchy distribution leads to the following utility guarantee:

Lemma 5.3. *On any D and any $\beta > 0, \gamma > 3$, the smooth sensitivity mechanism outputs a $\tilde{Q}(D)$ such that*

$$\Pr\left[\left|\tilde{Q}(D) - Q(D)\right| \geq \Omega\left(\frac{\widetilde{\mathrm{SS}}(D)}{\varepsilon \cdot \beta^{1/(\gamma-1)}})\right)\right] \leq \beta.$$

Since the noise scale is proportional to $\widetilde{\mathrm{SS}}(D)$, the *smooth sensitivity* $\mathrm{SS}(D)$ is defined to be the minimum over all $\widetilde{\mathrm{SS}}(D)$, which is the smallest noise scale achievable under this framework. However, computing $\mathrm{SS}(D)$ can be computationally expensive, so one often settles for some larger $\widetilde{\mathrm{SS}}(D)$ that can be computed efficiently [13, 29].

5.1 Private CI for SS

Note that Lemma 5.3 does not lead to a private CI, since $\widetilde{\mathrm{SS}}(D)$ cannot be released. The idea to generate a private CI is to use property (11) to release a privatized upper bound $\overline{\mathrm{SS}}(D)$ of $\widetilde{\mathrm{SS}}(D)$. We need $\overline{\mathrm{SS}}(D)$ to be an upper bound of $\widetilde{\mathrm{SS}}(D)$ (w.h.p.) to ensure correctness, while this upper bound should be as close to $\widetilde{\mathrm{SS}}$ as possible for good utility. The details of the algorithm, called SSCI, are shown in Algorithm 7, where $F_{\gamma}(\cdot)$ denotes the cumulative distribution function (CDF) of the generalized Cauchy distribution with parameter γ.

Theorem 5.4. *Given any $\varepsilon, \beta, \gamma > 3$, any $\widetilde{\mathrm{SS}}(\cdot)$ as defined above,* SSCI *preserves ε-DP. On any $D \in \mathcal{D}$, with probability at least $1-\beta$, it returns a CI containing $Q(D)$ and the length of the CI is $O\left(\frac{\widetilde{\mathrm{SS}}(D)}{\varepsilon \cdot \beta^{(3\gamma-1)/(\gamma^2-1)}}\right)$.*

Proof. For privacy, first, by definition, the sensitivity of $\log \widetilde{\mathrm{SS}}(D)$ is bounded by α thus both $\overline{\mathrm{SS}}(D)$ and w preserve $\frac{\varepsilon}{2}$-DP. Besides, by Lemma 5.2, $\tilde{Q}(D)$ preserves $\frac{\varepsilon}{2}$-DP. Finally, according to composition rule, Algorithm 7 preserves ε-DP.

For utility, with the tail bound of Laplace distribution, with probability at least $1 - \frac{\beta}{2}$ we have:

$$\widetilde{\mathrm{SS}} \le \overline{\mathrm{SS}}(D) \le \widetilde{\mathrm{SS}}(D) \cdot e^{\frac{4\alpha}{\varepsilon} \log \frac{2}{\beta}} = \left(\frac{2}{\beta}\right)^{1/(\gamma+1)} \cdot \widetilde{\mathrm{SS}}(D). \tag{12}$$

Combining with the tail bound of the Cauchy distribution, we have that with probability $1-\beta$, $Q(D) \in [\tilde{Q}(D) - w, \tilde{Q}(D) + w]$ and the length of the CI is $O\left(\frac{\widetilde{\mathrm{SS}}(D)}{\varepsilon \cdot \beta^{1/(\gamma+1)+1/(\gamma-1)}}\right) = O\left(\frac{\widetilde{\mathrm{SS}}(D)}{\varepsilon \cdot \beta^{2\gamma/(\gamma^2-1)}}\right)$. □

5.2 Median

We discuss two applications of SSCI. The first problem is again the median problem over a bounded domain $\mathcal{D} = [N]^n$. For the median problem, it is known that the real smooth sensitivity is

$$\mathrm{SS}(D) = \max_{k=0,\dots,n} \left(e^{-k\alpha} \max_{t=0,\dots,k+1} \left(x_{\lceil n/2 \rceil + t} - x_{\lceil n/2 \rceil + t - k - 1} \right) \right),$$

where we define $x_i := 0$ for $i < 0$ and $x_i := N$ for $i > n$. And it can be computed in $O(n \log n)$ time [38]. We can thus just plug $\widetilde{\mathrm{SS}}(D) = \mathrm{SS}(D)$ into SSCI.

Remark. For the median problem, SSCI returns a CI of length $O(\mathrm{SS}(D))$, while EMCI and SVTCI offer a CI with rank error $O(\log N)$ (taking ε, β as constants). These two utility guarantees are generally incomparable. For example, on $D = \{1, 2, \dots, n\}$, we have $\mathrm{SS}(D) = O(1)$, so SSCI returns a better CI than EMCI/SVTCI. However, on $D = \{1, 2, \dots, \lceil n/2 \rceil, \lceil n/2 \rceil + k, \dots\}$ for some large k, we have $\mathrm{SS}(D) = O(k)$ and SSCI would return a CI $[\lceil n/2 \rceil - O(k), \lceil n/2 \rceil + O(k)]$, while EMCI/SVTCI would return $[\lceil n/2 \rceil - O(\log N), \lceil n/2 \rceil + k + O(\log N)]$. Finally, EMCI/SVTCI has a better, logarithmic dependency on $1/\beta$, while that of SSCI is polynomial.

5.3 Conjunctive Queries

The smooth sensitivity framework has been applied to conjunctive queries [13, 14], which includes graph pattern counting as a special case. In particular, the *residual sensitivity* [13, 14], which is a constant-factor upper bound of the smooth sensitivity, can serve as the $\widetilde{SS}(\cdot)$ and is computable in polynomial time for any conjunctive query. Thus significantly broadening the scope EMCI, which can only support some special graph patterns efficiently.

6 EXPERIMENTS

In this section, we present an experimental evaluation of our mechanisms on various problems and datasets. We mainly compare our CI width with the actual error of the original DP mechanism, which only returns a privatized query answer. Note that, however, the actual error is private information and cannot be released, while the CI can. Also note that the CI cannot be better than the actual error by definition. Thus, the goal of the experiments is to gauge the (constant-factor) gap between the CI and best theoretical possible.

We conduct two series of experiments. Firstly, we compare three different median CI techniques (namely, SS, EM, and SVT based median CI) on various datasets and analyze their performances. Then for each CI technique, we consider some additional problems to demonstrate the generality of our approaches. For the SS-based method, we study conjunctive query CI, including both TPC-H query and graph counting query. For the EM-based method, we study graph counting query CI. As both SS and EM can provide graph counting CI, we also make a comparison between them. For the SVT-based algorithm, we study empirical mean CI on real datasets and also compare it with the current state-of-the-art statistical mean CI algorithm [31].

6.1 Setup

For median CI, we try two real datasets: the Bank Marketing dataset (Bank) [37] and the Adult dataset (Adult) [19], with 45,211 and 48,842 records, respectively. We focus on the median of clients' account balances for the Bank dataset, which ranges from −8,109 to 102,127 and has a median value of 448. For the Adult dataset, we calculate the median of final weight, which is the number of people an entry represents, and ranging from 12,285 to 1,490,400, with a median value of 178,147.

For mean CI, we also employ the Bank and Adult datasets, focusing on the same attribute as median. Due to the presence of outliers, we remove the top/bottom 5% data to satisfy the density requirement (7), which is a common practice in real-world data analysis. Following the removal of the outliers, the actual means are 903.3 and 183,014.0, respectively.

For general conjunctive query CI, we use the TPC-H dataset with scale factor 1. Since TPC-H schema has many foreign key constraints, when a tuple is deleted in one relation, we might obtain a neighbouring instance violating the foreign key constraints. In order to define a meaningful DP policy, we treat only Customer, Order, Supplier, PartSupp and Lineitem as private relations and the remaining as public relations. There are no foreign key constraints among these private relations so our DP policy is well defined. Under this setting, two datasets are neighbours if they differ by a tuple in the private relations. We run the counting query |Nation ⋈ Customer ⋈ Orders ⋈ Lineitems ⋈ Supplier|, which is an analog of Q7 from the benchmark, and the query result is 6,001,215. Computing the smooth sensitivity for general SQL queries is challenging; therefore, we adopt the residual sensitivity-based technique [13] mentioned in Section 5.3.

For graph pattern counting CI, we use the HepPh dataset [34], which contains 12,008 nodes and 236,978 edges. We treat it as a directed graph and run both triangle counting query ($q_\triangle$) and 3-star counting query ($q_{3\star}$). The real counts are 20,150,994 and 7,661,801,994 respectively. We use the formulas in [30, 38] to compute the ladder function [41].

Experimental Parameters. All experiments are done on a desktop PC equipped with a 2.8 GHz Intel Core i7 CPU and 16GB memory. For the privacy budgets, we try $\varepsilon = 0.125, 0.25, 0.5, 1, 2, 4$ and the default value is 1. For confidence levels, we try $\beta = 0.01, 0.025, 0.05,$

Table 2: Results for median CI on real datasets under default setting $\varepsilon = 1, \beta = 0.1$.

Problem Type	Data	Query Result	Technique	Average CI Width	Error Quantile	Relative CI Width(CI/Error ratio)	Observed Coverage
Median	Bank	448	SS	42.8	23.8	1.8	0.97
			SVT	14.2	1.0	14.2	1
			EM	13.8	1.0	13.8	1
	Adult	178,147	SS	3,075.6	1,681.4	1.8	0.99
			SVT	1,280.5	105.0	12.2	1
			EM	1,024.9	47.0	21.8	1

Table 3: Results for other queries, we use default setting $\varepsilon = 1, \beta = 0.1$.

Problem Type	Data	Query Result	Technique	Average CI Width	Error Quantile	Relative CI Width(CI/Error ratio)	Observed Coverage
Conjunctive Query	TPC-H	6,001,215	SS	33,491.8	20,720.7	1.6	0.98
$q_{\triangle}$	HepPh	20,150,994	SS	64,655.5	41,885.9	1.5	0.98
			EM	418,626	6,310	66.3	1
$q_{3\star}$	HepPh	7,661,801,994	SS	34,559,295.7	22,630,117.2	1.5	0.98
			EM	299,647,401	4,196,135	71.4	1
Mean	Bank	930.3	SVT	11.9	7.4	1.6	0.98
			Statistical [31]	17.3	13.2	1.3	0.95
	Adult	183,014.0	SVT	352.0	266.7	1.3	0.96
			Statistical [31]	1030.4	729.7	1.4	0.94

0.1,0.2 and the default value is 0.1. For the SS-based method, we always fix $\gamma = 4$. In median CI problem, we always set $N = 10,000,000$ while in graph counting problem we set $N = n^{|P|}$. Each experiment is repeated 100 times.

Utility Measures. We consider several utility measures in our experiments:

- Average CI Width: The average width of the confidence interval is a commonly used metric to evaluate the effectiveness of a CI technique. Such measurement is applied in the works regarding statistical CI [10, 17, 26]. In our experiments, we record half of the average confidence interval width of 100 trials. Here the width is divided by 2 because given any CI, we can use the midpoint of this interval as a point estimation and its error is bounded by half of the interval width.
- Error Quantile: The $1-\beta$ quantile of corresponding point estimation error (absolute value), here β is the desired confidence level of our CI. This is an analog to the Mean Absolute Error(MAE) but fits our task better. Since we'll compare the error with the width of a $1-\beta$ CI, it's better to also use the $1-\beta$ error quantile.
- Relative CI Width: The relative CI width is the ratio between the average CI width and the corresponding error quantile. This ratio indicates the overhead caused by our CI algorithms and a smaller value means smaller overhead. A similar metric is also used in the works regarding statistical CI [10, 17, 26].
- Observed Coverage: The probability that the real query result indeed lies in our confidence interval, it should be greater than $1-\beta$ as required. This value indicates the correctness of our CI technique, namely, it indeed returns a CI as claimed.

We do not record execution time here because all our algorithms are efficient. Say, each trail can be done in a few seconds.

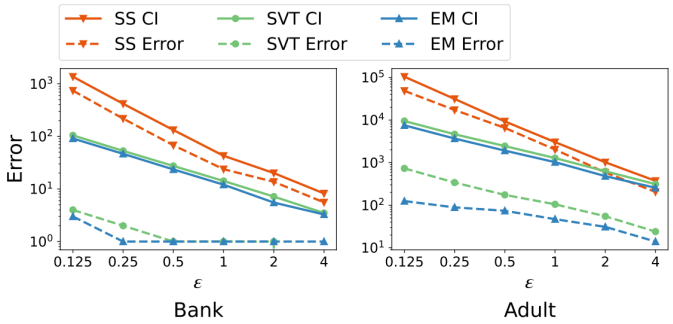
6.2 Results

Datasets and Queries. The experimental results under default setting are summarized in Table 2 and 3. In the case of median/graph counting CI problem, we also investigated the effect of varying ε, β and results are presented in Figure 2 and 3. Please note that the y-axis in these figures represents the CI width or estimation error, but for simplicity, we only labeled it as error.

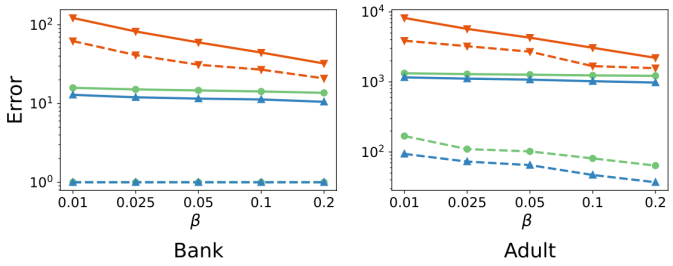
Median. Across all parameter settings and datasets, the EM/SVT-based mechanisms achieve smaller average CI width, while SS-based mechanism provides a smaller relative CI width, which means it has smaller overheads. These results are in line with our analysis. It is important to note that both EM and SVT have rank error guarantees, with EM guarantees being better than those of SVT. However, a rank error guarantee does not imply any bound on CI width and error magnitude, so the ratio between them may be large. In contrast, SS does not provide any error guarantee. Instead it bounds the ratio between $\mathrm{SS}(D)$ and its privatized upper bound $\overline{\mathrm{SS}}(D)$. Such a bound transforms into a bound on the ratio between CI width and error.

Figure 2a shows the effect of varying ε. When increasing the value of ε, the error of all methods decreases while the CI/Error ratio roughly remains unchanged. One interesting finding is that as ε grows, the error (and CI width) of SS decreases faster compared to EM and SVT. This is because the value of smooth sensitivity has a strong (roughly exponential) dependency on ε, so increasing ε has a double effect on the utility of SS-based methods. On the contrast, increasing ε only decreases the rank error in EM/SVT linearly.

Figure 2b shows how error and CI width change with β. For EM and SVT, the effect of β is much smaller than ε since β only affects the rank error by a log factor. So as β grows, CI width only slightly decreases. Table 2 summarizes results under default setting, all the errors are small compared with the large N and the relative error is smaller than 1%. We can see in all cases, the observed coverage of our CI is higher than the value indicated by β.

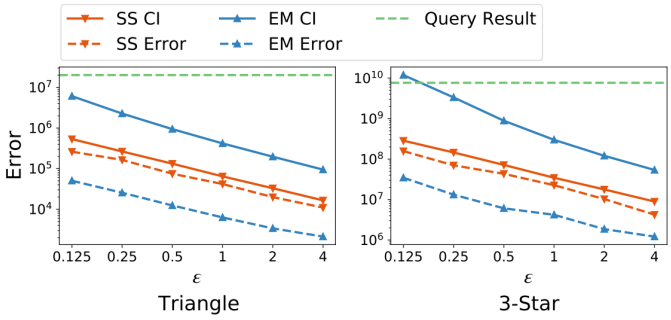


(a) Three techniques with $\beta = 0.1$ and different ε

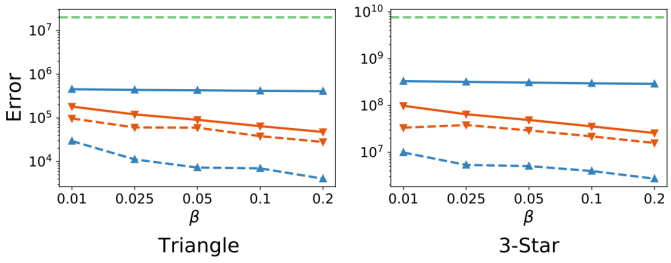


(b) Three techniques with $\varepsilon = 1$ and different β.

Figure 2: Median CI/estimation errors with varying ε, β.



(a) SS and EM with $\beta = 0.1$ and different ε.



(b) SS and EM with $\varepsilon = 1$ and different β.

Figure 3: Graph counting CI/estimation errors with varying ε, β.

Graph Pattern Counting. For graph pattern counting CI, the separation between SS and EM becomes clearer as shown in Figure 3a, 3b. We can see EM always achieves smaller error, but it provides worse confidence interval due to the large CI/Error ratio. That's because the standard utility guarantee in lemma 3.1 is quite loose, which is an intrinsic problem in the analysis of EM, rather our CI technique. This is also confirmed by the high correct rate (100%), showing the interval length is more than necessary. In contrast, SS always provide reasonable result as while as good CI. The reason is that SS is a sensitivity-based technique. Say, it achieves DP by adding a noise scaled with the smooth sensitivity. Thus the CI should be tight as long as we use proper tail bounds.

The effect of varying ε, β is similar to the case of median CI problem, details are shown in Figure 3.

Summary of Other Queries. Results of other queries under default setting are listed in Table 3. For TPC-H query, SS based method achieves good utility. with less than 1% relative error. For mean estimation, SVT based CI technique has good utility. This time it also provides good relavtive CI width (1.6×), unlike in median/graph counting CI (more than 10×). That's because the underlying mean estimation algorithm is sensitivity-based, so it can provide good CI as long as it can give accurate point estimation. Additionally, we should note that in all cases the observed coverage of our CI is higher than the value indicated by β.

Statistical Mean CI Mechanism. The CI problems has been studied in the statistical setting. Nevertheless, we observe that the statistical mean CI mechanism in [31] can be modified to support the empirical as well. More precisely, the CI in [31] consists of a bound on the sampling error plus a bound on the DP noise. By removing the sampling error bound, the mechanism of [31] provides a valid CI for empirical mean. We compared this mechanism with our SVT-MeanCI algorithm on two real datasets, and the results are shown in Table 3. We can see that our empirical method outperforms the (modified) statistical method on both datasets. The reason is that the mechanism of [31] is tailored for Gaussian data, which is not the case in practice. Secondly, the statistical method spends some additional privacy budget to estimate the variance, thus leading to worse utility. Additionally, the statistical method has a larger constant hidden by $O()$ notation.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose three general CI techniques with small utility overhead compared to the point estimation algorithms. And here we point out two possible future research directions. First, the DP policy adopted in this paper is commonly known as *tuple-DP*, i.e., two datasets are neighboring if they differ by one tuple. In relational databases, another popular DP policy is known as *user-DP* modeled by foreign-key constraints [11, 12, 15, 32, 33], which includes *node-DP* for private graph analysis as a special case. How to derive CIs for such mechanisms is an interesting open problem. Another future direction is to derive tighter confidence sets in high dimensions, as mentioned in Section 1.3.

ACKNOWLEDGMENTS

This work has been supported by HKRGC under grants 16205420, 16205422, and 16204223. We would also like to thank the anonymous reviewers who have made valuable suggestions on improving the presentation of the paper.

REFERENCES

[1] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European conference on computer systems*. 29–42.

[2] Hilal Asi and John C Duchi. 2020. Instance-optimality in differential privacy via approximate inverse sensitivity mechanisms. *Advances in neural information processing systems* 33 (2020), 14106–14117.

[3] Hilal Asi and John C Duchi. 2020. Near instance-optimality in differential privacy. *arXiv preprint arXiv:2005.10630* (2020).

[4] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2014. Bounds on the sample complexity for private learning and private data release. *Machine learning* 94 (2014), 401–437.

[5] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. 2020. CoinPress: Practical Private Mean and Covariance Estimation. In *Advances in Neural Information Processing Systems*, Vol. 33. 14475–14485.

[6] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. 2015. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 634–649.

[7] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[8] Surajit Chaudhuri, Bolin Ding, and Srikanth Kandula. 2017. Approximate query processing: No silver bullet. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 511–519.

[9] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially private spatial decompositions. In *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 20–31.

[10] Christian Covington, Xi He, James Honaker, and Gautam Kamath. 2021. Unbiased statistical estimation and valid confidence intervals under differential privacy. *arXiv preprint arXiv:2110.14465* (2021).

[11] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2022. R2T: Instance-optimal Truncation for Differentially Private Query Evaluation with Foreign Keys. In *Proc. ACM SIGMOD International Conference on Management of Data*.

[12] Wei Dong, Dajun Sun, and Ke Yi. 2023. Better than Composition: How to Answer Multiple Relational Queries under Differential Privacy. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–26.

[13] Wei Dong and Ke Yi. 2021. Residual Sensitivity for Differentially Private Multi-Way Joins. In *Proceedings of the 2021 International Conference on Management of Data*. 432–444.

[14] Wei Dong and Ke Yi. 2022. A Nearly Instance-optimal Differentially Private Mechanism for Conjunctive Queries. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 213–225.

[15] Wei Dong and Ke Yi. 2023. Query Evaluation under Differential Privacy. *ACM SIGMOD Record* 52, 3 (2023), 6–17.

[16] Wei Dong and Ke Yi. 2023. Universal private estimators. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 195–206.

[17] Jörg Drechsler, Ira Globus-Harris, Audra Mcmillan, Jayshree Sarathy, and Adam Smith. 2022. Nonparametric Differentially Private Confidence Intervals for the Median. *Journal of Survey Statistics and Methodology* 10 (2022), 804–829.

[18] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. 2020. Differentially private confidence intervals. *arXiv preprint arXiv:2001.02285* (2020).

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

[20] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 371–380.

[21] Cynthia Dwork and Jing Lei. 2009. Differential Privacy and Robust Statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. 371–380.

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

[23] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 381–390.

[24] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[25] Cynthia Dwork and Guy N Rothblum. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887* (2016).

[26] Cecilia Ferrando, Shufan Wang, and Daniel Sheldon. 2022. Parametric Bootstrap for Differentially Private Confidence Intervals. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. 1598–1618.

[27] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. 1997. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 171–182.

[28] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. *Advances in Neural Information Processing Systems* (2021).

[29] Noah Johnson, Joseph P Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. *Proceedings of the VLDB Endowment* 11, 5 (2018), 526–539.

[30] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. 2011. Private analysis of graph structure. *Proceedings of the VLDB Endowment* 4, 11 (2011), 1146–1157.

[31] Vishesh Karwa and Salil Vadhan. 2017. Finite Sample Differentially Private Confidence Intervals.

[32] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*. Springer, 457–476.

[33] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2019. Privatesql: a differentially private sql query engine. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1371–1384.

[34] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[35] Feifei Li, Bin Wu, Ke Yi, and Zhuoyue Zhao. 2016. Wander join: Online aggregation via random walks. In *Proceedings of the 2016 International Conference on Management of Data*. 615–629.

[36] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 94–103.

[37] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[38] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth Sensitivity and Sampling in Private Data Analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. 75–84.

[39] Davide Proserpio, Sharon Goldberg, and Frank McSherry. 2014. Calibrating data to sensitivity in private data analysis: A platform for differentially-private analysis of weighted datasets. *Proceedings of the VLDB Endowment* 7, 8 (2014), 637–648.

[40] Dajun Sun, Wei Dong, and Ke Yi. 2023. Confidence Intervals for Private Query Processing. https://drive.google.com/file/d/1nk5HqZQT5J4hFEwGx1TOZOs0KRSYd1gh/view

[41] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2015. Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 731–745.