The VLDB Journal (2023) 32:23–48
https://doi.org/10.1007/s00778-022-00728-2

REGULAR PAPER

Free gap estimates from the exponential mechanism, sparse vector, noisy max and related algorithms

Zeyu Ding[1] · Yuxin Wang[1] · Yingtai Xiao[1] · Guanhong Wang[1] · Danfeng Zhang[1] · Daniel Kifer[1]

Received: 30 November 2020 / Revised: 21 September 2021 / Accepted: 19 December 2021 / Published online: 16 February 2022

Abstract
Private selection algorithms, such as the exponential mechanism, noisy max and sparse vector, are used to select items (such as queries with large answers) from a set of candidates, while controlling privacy leakage in the underlying data. Such algorithms serve as building blocks for more complex differentially private algorithms. In this paper we show that these algorithms can release additional information related to the gaps between the selected items and the other candidates for free (i.e., at no additional privacy cost). This free gap information can improve the accuracy of certain follow-up counting queries by up to 66%. We obtain these results from a careful privacy analysis of these algorithms. Based on this analysis, we further propose novel hybrid algorithms that can dynamically save additional privacy budget.

Keywords Differential privacy · Exponential mechanism · Noisy max · Sparse vector

1 Introduction

Industry and government agencies are increasingly adopting differential privacy [18] to protect the confidentiality of users who provide data. Current and planned major applications include data gathering by Google [7,21], Apple [43] and Microsoft [13]; database querying by Uber [28]; and publication of population statistics at the US Census Bureau [2,9,26,34].

The accuracy of differentially private data releases is very important in these applications. One way to improve accuracy is to increase the value of the privacy parameter ϵ, known as the privacy loss budget, as it provides a tradeoff between an algorithm's utility and its privacy protections. However, values of ϵ that are deemed too high can subject a company to criticisms of not providing enough privacy [42]. For this reason, researchers invest significant effort in tuning algorithms [1,11,22,29,40,47] and privacy analyses [8,20,38,40] to provide better utility at the same privacy cost.

Differentially private algorithms are built on smaller components called *mechanisms* [37]. Popular mechanisms include the Laplace mechanism [18], geometric mechanism [24], Noisy Max [19], sparse vector technique (SVT) [19,33] and the exponential mechanism [36]. As we will explain in this paper, some of these mechanisms, such as the exponential mechanism, Noisy Max and SVT, inadvertently throw away information that is useful for designing accurate algorithms. Our contribution is to present novel variants of these mechanisms that provide more functionality at the same privacy cost (under pure differential privacy).

Given a set of queries, Noisy Max returns the identity (not value) of the query that is likely to have the largest value—it adds noise to each query answer and returns the index of the query with the largest noisy value. The exponential mechanism is a replacement for Noisy Max in situations where query answers have utility scores. Meanwhile, SVT is an online algorithm that takes a stream of queries and a predefined public threshold T. It tries to return the identities (not

✉ Zeyu Ding
zyding@psu.edu

Yuxin Wang
yxwang@psu.edu

Yingtai Xiao
yxx5224@psu.edu

Guanhong Wang
gpw5092@psu.edu

Danfeng Zhang
zhang@cse.psu.edu

Daniel Kifer
dkifer@cse.psu.edu

[1] Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA

values) of the first k queries that are likely larger than the threshold. To do so, it adds noise to the threshold. Then, as it sequentially processes each query, it outputs "$\top$" or "$\bot$", depending on whether the noisy value of the current query is larger or smaller than the noisy threshold. The mechanism terminates after k "$\top$" outputs.

In recent work [45], using program verification tools, Wang et al. showed that SVT can provide additional information *at no additional cost to privacy*. That is, when SVT returns "$\top$" for a query, it can also return the gap between its noisy value and the noisy threshold.[1] We refer to their algorithm as SVT with Gap.

Inspired by this program verification work, we propose novel variations of exponential mechanism, SVT and Noisy Max that add new functionality. For SVT, we show that in addition to releasing this gap information, even stronger improvements are possible—we present an adaptive version that can answer more queries than before by controlling how much privacy budget it uses to answer each query. The intuition is that we would like to spend less of our privacy budget for queries that are probably much larger than the threshold (compared to queries that are probably closer to the threshold). A careful accounting of the privacy impact shows that this is possible. Our experiments confirm that Adaptive SVT with Gap can answer many more queries than the prior versions [19,33,45] at the same privacy cost.

For Noisy Max, we show that it too inadvertently throws away information. Specifically, *at no additional cost to privacy*, it can release an estimate of the gap between the largest and second largest queries (we call the resulting mechanism Noisy Max with Gap). We generalize this result to Noisy Top-K—showing that one can release an estimate of the *identities* of the k largest queries and, at no extra privacy cost, release noisy estimates of the pairwise *gaps* (differences) among the top $k+1$ queries.

For exponential mechanism, we show that there is also a concept of a gap, which can be used to test whether a non-optimal query was returned. One of the challenges with the exponential mechanism is that for efficiency purposes it can use complex sampling algorithms to select the chosen candidate. We show that it is possible to release the noisy gap information even if the sampling algorithms are treated as black boxes (i.e., without access to its intermediate computations).

The extra noisy gap information opens up new directions in the construction of differentially private algorithms and can be used to improve accuracy of certain subsequent queries. For instance, one common task is to use Noisy Max to select the approximate top k queries and then use additional privacy loss budget to obtain noisy answers to these queries. We show that a postprocessing step can combine these noisy answers with gap information to improve accuracy by up to 66% for counting queries. We provide similar applications for the free gap information in SVT.

This paper is an extension of a conference paper [14]. For this extension we have added the following results: (a) free gap results for the exponential mechanism, (b) free gap results when Noisy Max and SVT are used with one-sided noise, which improves on the accuracy reported in [14] for two-sided noise, (c) novel hybrid algorithms that combine SVT and Noisy Max into an offline selection procedure; these algorithms return the *identities* of the approximate top-k queries, but only if they are larger than a pre-specified threshold. These algorithms save privacy budget if fewer than k queries are approximately over the threshold, in which case they also provide free estimates of the query answers. (If all k queries are approximately over the threshold, then we obtain information about the gaps between them).

We prove most of our results using the alignment of random variables framework [11,33,45,46], which is based on the following question: If we change the input to a program, how must we change its random variables so that output remains the same? This technique is used to prove the correctness of almost all pure differential privacy mechanisms [19] but needs to be used in sophisticated ways to prove the correctness of the more advanced algorithms [11,19,33,45,46]. Nevertheless, alignment of random variables is often used incorrectly (as discussed by Lyu et al. [33]). Thus a secondary contribution of our work is to lay out the precise steps and conditions that must be checked and to provide helpful lemmas that ensure these conditions are met. The exponential mechanism does not fit in this framework and requires its own proof techniques, which we explain in Sect. 8. To summarize, our contributions are as follows:

1. We provide a simplified template for writing correctness proofs for intricate differentially private algorithms.
2. Using this technique, we propose and prove the correctness of two new mechanisms: Noisy Top-K with Gap and Adaptive SVT with Gap.
   These algorithms improve on the original versions of Noisy Max and SVT by taking advantage of *free* information (i.e., information that can be released at no additional privacy cost) that those algorithms inadvertently throw away. We also show that the free gap information can be maintained even when these algorithms use one-sided noise. This variation improves the accuracy of the gap information.
3. We demonstrate some of the uses of the gap information that is provided by these new mechanisms. When an algorithm needs to use Noisy Max or SVT to select some queries and then measure them (i.e., obtain their

[1] This was a surprising result given the number of incorrect attempts at improving SVT based on flawed manual proofs [33] and shows the power of automated program verification techniques.

noisy answers), we show how the gap information from our new mechanisms can be used to improve the accuracy of the noisy measurements. We also show how the gap information in SVT can be used to estimate the confidence that a query's true answer really is larger than the threshold.

4. We show that the exponential mechanism can also release free gap information. Noting that the free gap extensions of Noisy Max and SVT required access to the internal state of those algorithms, we show that this is unnecessary for exponential mechanism. This is useful because implementations of exponential mechanism can be very complex and use a variety of different sampling routines.
5. We propose two novel hybridizations of Noisy Max and SVT. These algorithms can release the identities of the approximate top-k queries as long as they are larger than a pre-specified threshold. If fewer than k queries are returned, the algorithms save privacy budget and the gap information they release directly turns into estimates of the query answers (i.e., the algorithm returns the query identities and their answers for free). If k queries are returned then the algorithms still return the gaps between their answers.
6. We empirically evaluate the mechanisms on a variety of datasets to demonstrate their improved utility.

In Sect. 2, we discuss related work. We present background and notation in Sect. 3. We present simplified proof templates for randomness alignment in Sect. 4. We present Adaptive SVT with Gap in Sect. 5 and Noisy Top-K with Gap in Sect. 6. We present the novel algorithms that combine elements of Noisy Max and SVT in 7. We present exponential mechanism with gap algorithms in Sect. 8. We present experiments in Sect. 9, proofs underlying the alignment of randomness framework in Sect. 10 and conclusions in Sect. 11. Other proofs appear in "Appendix."

2 Related works

Selection algorithms, such as exponential mechanism [36,41], sparse vector technique (SVT) [19,33] and Noisy Max [19] are used to select a set of items (typically queries) from a much larger set. They have applications in hyperparameter tuning [11,32], iterative construction of microdata [27], feature selection [44], frequent itemset mining [6], exploring a privacy/accuracy tradeoff [31], data preprocessing [12], etc. Various generalizations have been proposed [5,10,31,32,41,44]. Liu and Talwar [32] and Raskhodnikova and Smith [41] extend the exponential mechanism for arbitrary sensitivity queries. Beimel et al. [5] and Thakurta and Smith [44] use the propose-test-release framework [17] to find a gap between the best and second best queries and, if the gap is large enough, release the identity of the best query. These two algorithms rely on a relaxation of differential privacy called approximate (ϵ, δ)-differential privacy [16] and can fail to return an answer (in which case they return $\perp$). Our algorithms work with pure ϵ-differential privacy. Chaudhuri et al. [10] also proposed a large margin mechanism (with approximate differential privacy) which finds a large gap separating top queries from the rest and returns one of them.

There have also been unsuccessful attempts to generalize selection algorithms such as SVT (incorrect versions are catalogued by Lyu et al. [33]), which has sparked innovations in program verification for differential privacy (e.g., [3,4,45,46]) with techniques such as probabilistic coupling [4] and a simplification based on randomness alignment [46]. These are similar to ideas behind handwritten proofs [11,19,33]—they consider what changes need to be made to random variables in order to make two executions of a program, with different inputs, produce the same output. It is a powerful technique that is behind almost all proofs of differential privacy, but is very easy to apply incorrectly [33]. In this paper, we state and prove a more general version of this technique in order to prove correctness of our algorithms and also provide the additional results that simplify the application of this technique.

Table 1 Noise distributions

Symbol	Support	Density/mass	Mean	Variance
Lap(β)	$\mathbb{R}$	$\frac{1}{2\beta}\exp\left(-\frac{\lvert x\rvert}{\beta}\right)$	0	$2\beta^2$
Exp(β)	$[0,\infty)$	$\frac{1}{\beta}\exp\left(-\frac{x}{\beta}\right)$	β	β^2
Geo(p)	$\{0,1,\ldots\}$	$p(1-p)^n$	$\frac{1}{p}$	$\frac{1-p}{p^2}$

3 Background and notation

In this paper, we use the following notation. D and D' refer to databases. We use the notation $D \sim D'$ to represent adjacent databases.[2] M denotes a randomized algorithm whose input is a database. Ω denotes the range of M and $\omega \in \Omega$ denotes a specific output of M. We use $E \subseteq \Omega$ to denote a set of possible outputs. Because M is randomized, it also relies on a random noise vector $H \in \mathbb{R}^\infty$. This noise sequence is infinite, but of course M will only use a finite-length prefix of H. Some of the commonly used noise distributions for this vector H include the Laplace distribution, the exponential distribution and the geometric distribution. Their properties are summarized in Table 1.

[2] The notion of adjacency depends on the application. Some papers define it as D can be obtained from D' by modifying one record [18] or by adding/deleting one record [15].

Table 2 Notation

Symbol	Meaning
M	Randomized algorithm
D, D'	Database
$D \sim D'$	D is adjacent to D'
$H = (\eta_1, \eta_2, \ldots)$	Input noise vector
Ω	The space of all output of M
ω	A possible output; $\omega \in \Omega$
E	A set of possible outputs; $E \subseteq \Omega$
$\mathcal{H}_{D:E} = \mathcal{H}^M_{D:E}$	$\{H \mid M(D, H) \in E\}$
$\mathcal{H}_{D:\omega} = \mathcal{H}^M_{D:\omega}$	$\{H \mid M(D, H) = \omega\}$

When we need to draw attention to the noise, we use the notation $M(D, H)$ to indicate the execution of M with database D and randomness coming from H. Otherwise we use the notation $M(D)$. We define $\mathcal{H}^M_{D:E} = \{H \mid M(D, H) \in E\}$ to be the set of noise vectors that allow M, on input D, to produce an output in the set $E \subseteq \Omega$. To avoid overburdening the notation, we write $\mathcal{H}_{D:E}$ for $\mathcal{H}^M_{D:E}$ and $\mathcal{H}_{D':E}$ for $\mathcal{H}^M_{D':E}$ when M is clear from the context. When E consists of a single point ω, we write these sets as $\mathcal{H}_{D:\omega}$ and $\mathcal{H}_{D':\omega}$. This notation is summarized in Table 2.

3.1 Formal privacy

Differential privacy [15,18,19] is currently the gold standard for releasing privacy-preserving information about a database. It has a parameter $\epsilon > 0$ known as the privacy loss budget. The smaller it is, the more privacy is provided. Differential privacy bounds the effect of one record on the output of the algorithm (for small ϵ, the probability of any output is barely affected by any person's record).

Definition 1 (*Pure differential privacy* [15]) Let $\epsilon > 0$. A randomized algorithm M with output space Ω satisfies (pure) ϵ-differential privacy if for all $E \subseteq \Omega$ and all pairs of adjacent databases $D \sim D'$, we have

$$\mathbb{P}[M(D, H) \in E] \leq e^{\epsilon} \mathbb{P}[M(D', H') \in E] \tag{1}$$

where the probability is only over the randomness of H. With the notation in Table 2, the differential privacy condition from Eq. (1) is $\mathbb{P}[\mathcal{H}_{D:E}] \leq e^{\epsilon} \mathbb{P}[\mathcal{H}_{D':E}]$.

Differential privacy enjoys the following properties:

1. Resilience to postprocessing. If we apply an algorithm A to the output of an ϵ-differentially private algorithm M, then the composite algorithm $A \circ M$ still satisfies ϵ-differential privacy. In other words, privacy is not reduced by postprocessing.
2. Composition. If $M_1, M_2, \ldots, M_k$ satisfy differential privacy with privacy loss budgets $\epsilon_1, \ldots, \epsilon_k$, the algorithm that runs all of them and releases their outputs satisfies $(\sum_i \epsilon_i)$-differential privacy.

Many differentially private algorithms take advantage of the Laplace mechanism [36], which provides a noisy answer to a vector-valued query $\boldsymbol{q}$ based on its ℓ_1 *global sensitivity* $\Delta_{\boldsymbol{q}}$, defined as follows:

Definition 2 (*Global sensitivity* [19]) The ℓ_1 global sensitivity of a query $\boldsymbol{q}$ is $\Delta_{\boldsymbol{q}} = \sup_{D \sim D'} \left\| \boldsymbol{q}(D) - \boldsymbol{q}(D') \right\|_1$.

Theorem 1 (Laplace mechanism [18]) *Given a privacy loss budget ϵ, consider the mechanism that returns $\boldsymbol{q}(D) + H$, where H is a vector of independent random samples from the* $\text{Lap}(\Delta_{\boldsymbol{q}}/\epsilon)$ *distribution. This Laplace mechanism satisfies ϵ-differential privacy.*

Other kinds of additive noise distributions that can be used in place of Laplace in Theorem 1 include discrete Laplace [24] (when all query answers are integers or multiples of a common base) and Staircase [23].

In some cases, queries may have additional structure, such as *monotonicity*, which can allow algorithms to provide privacy with less noise (such as one-sided noisy max [19]).

Definition 3 (*Monotonicity*) A list of queries $\boldsymbol{q} = (q_1, q_2, \ldots)$ with numerical values is monotonic if for all pair of adjacent databases $D \sim D'$ we have either $\forall i : q_i(D) \leq q_i(D')$, or $\forall i : q_i(D) \geq q_i(D')$.

Monotonicity is a natural property that is satisfied by counting queries—when a person is added to a database, the value of each query either stays the same or increases by 1.

4 Randomness alignment

To establish that the algorithms we propose are differentially private, we use an idea called *randomness alignment* that previously had been used to prove the privacy of a variety of sophisticated algorithms [11,19,33] and incorporated into verification/synthesis tools [3,45,46]. While powerful, this technique is also easy to use incorrectly [33], as there are many technical conditions that need to be checked. In this section, we present the results (namely Lemma 1) that significantly simplify this process and make it easy to prove the correctness of our proposed algorithms.

In general, to prove ϵ-differential privacy for an algorithm M, one needs to show $\mathbb{P}[M(D, H) \in E] \leq e^{\epsilon} \mathbb{P}[M(D', H') \in E]$ for all pairs of adjacent databases $D \sim D'$ and sets of possible outputs $E \subseteq \Omega$. In our notation, this inequality is represented as $\mathbb{P}[\mathcal{H}_{D:E}] \leq e^{\epsilon} \mathbb{P}[\mathcal{H}_{D':E}]$.

Establishing such inequalities is often done with the help of a function $\phi_{D,D'}$, called a *randomness alignment* (there is a function $\phi_{D,D'}$ for every pair $D \sim D'$), which maps noise vectors H into noise vectors H' so that $M(D', H')$ produces the same output as $M(D, H)$. Formally,

Definition 4 (*Randomness alignment*) Let M be a randomized algorithm. Let $D \sim D'$ be a pair of adjacent databases. A *randomness alignment* is a function $\phi_{D,D'} : \mathbb{R}^\infty \to \mathbb{R}^\infty$ such that

1. The alignment does not output invalid noise vectors (e.g., it cannot produce negative numbers for random variables that should have the exponential distribution).
2. For all H on which $M(D, H)$ terminates, $M(D, H) = M(D', \phi_{D,D'}(H))$.

Example 1 Let D be a database that records the salary of every person, which is guaranteed to be between 0 and 100. Let $q(D)$ be the sum of the salaries in D. The sensitivity of q is thus 100. Let $H = (\eta_1, \eta_2, \dots)$ be a vector of independent Lap$(100/\epsilon)$ random variables. The Laplace mechanism outputs $q(D) + \eta_1$ (and ignores the remaining variables in H). For every pair of adjacent databases $D \sim D'$, one can define the corresponding randomness alignment $\phi_{D,D'}(H) = H' = (\eta'_1, \eta'_2, \dots)$, where $\eta'_1 = \eta_1 + q(D) - q(D')$ and $\eta'_i = \eta_i$ for $i > 1$. Note that $q(D) + \eta_1 = q(D') + \eta'_1$, so the output of M remains the same.

In practice, $\phi_{D,D'}$ is constructed locally (piece by piece) as follows. For each possible output $\omega \in \Omega$, one defines a function $\phi_{D,D',\omega}$ that maps noise vectors H into noise vectors H' with the following properties: if $M(D, H) = \omega$ then $M(D', H') = \omega$ (that is, $\phi_{D,D',\omega}$ only cares about what it takes to produce the specific output ω). We obtain our randomness alignment $\phi_{D,D'}$ in the obvious way by piecing together the $\phi_{D,D',\omega}$ as follows: $\phi_{D,D'}(H) = \phi_{D,D',\omega^*}(H)$, where ω^* is the output of $M(D, H)$. Formally,

Definition 5 (*Local alignment*) Let M be a randomized algorithm. Let $D \sim D'$ be a pair of adjacent databases and ω a possible output of M. A *local alignment* for M is a function $\phi_{D,D',\omega} : \mathcal{H}_{D:\omega} \to \mathcal{H}_{D':\omega}$ (see notation in Table 2) such that for all $H \in \mathcal{H}_{D:\omega}$, we have $M(D, H) = M(D', \phi_{D,D',\omega}(H))$.

Example 2 Continuing the setup from Example 1, consider the mechanism M_1 that, on input D, outputs $\top$ if $q(D)+\eta_1 \geq 10{,}000$ (i.e., if the noisy total salary is at least 10,000) and $\perp$ if $q(D) + \eta_1 < 10{,}000$. Let D' be a database that differs from D in the presence/absence of one record. Consider the local alignments $\phi_{D,D',\top}$ and $\phi_{D,D',\perp}$ defined as follows. $\phi_{D,D',\top}(H) = H' = (\eta'_1, \eta'_2, \dots)$ where $\eta'_1 = \eta_1 + 100$ and $\eta'_i = \eta_i$ for $i > 1$; and $\phi_{D,D',\perp}(H) = H'' = (\eta''_1, \eta''_2, \dots)$ where $\eta''_1 = \eta_1 - 100$ and $\eta''_i = \eta_i$ for $i > 1$. Clearly, if $M_1(D, H) = \top$ then $M_1(D', H') = \top$ and if $M_1(D, H) = \perp$ then $M_1(D', H'') = \perp$. We piece these two local alignments together to create a randomness alignment $\phi_{D,D'}(H) = H^* = (\eta^*_1, \eta^*_2, \dots)$ where:

$$\eta^*_1 = \begin{cases} \eta_1 + 100 & \text{if } M(D, H) = \top \text{ (i.e. } q(D) + \eta_1 \geq 10{,}000) \\ \eta_1 - 100 & \text{if } M(D, H) = \perp \text{ (i.e. } q(D) + \eta_1 < 10{,}000) \end{cases}$$

$$\eta^*_i = \eta_i \text{ for } i > 1.$$

Special properties of alignments Not all alignments can be used to prove differential privacy. In this section we discuss some additional properties that help prove differential privacy. We first make two mild assumptions about the mechanism M: (1) it terminates with probability[3] one and (2) based on the output of M, we can determine how many random variables it used. The vast majority of differentially private algorithms in the literature satisfy these properties.

We next define two properties of a local alignment: whether it is *acyclic* and what its *cost* is.

Definition 6 (*Acyclic*) Let M be a randomized algorithm. Let $\phi_{D,D',\omega}$ be a local alignment for M. For any $H = (\eta_1, \eta_2, \dots)$, let $H' = (\eta'_1, \eta'_2, \dots)$ denote $\phi_{D,D',\omega}(H)$. We say that $\phi_{D,D',\omega}$ is acyclic if there exists a permutation π and piecewise differentiable functions $\psi^{(j)}_{D,D',\omega}$ such that:

$$\eta'_{\pi(1)} = \eta_{\pi(1)} + \text{constant that only depends on } D, D', \omega$$

$$\eta'_{\pi(j)} = \eta_{\pi(j)} + \psi^{(j)}_{D,D',\omega}(\eta_{\pi(1)}, \dots, \eta_{\pi(j-1)}) \text{ for } j \geq 2$$

Essentially, a local alignment $\phi_{D,D',\omega}$ is acyclic if there is some ordering of the variables so that η'_j is the sum of η_j and a function of the variables that came earlier in the ordering. The local alignments $\phi_{D,D',\top}$ and $\phi_{D,D',\perp}$ from Example 2 are both acyclic. (In general, each local alignment function is allowed to have its own specific ordering and differentiable functions $\psi^{(j)}_{D,D',\omega}$.) The pieced-together randomness alignment $\phi_{D,D'}$ itself need not be acyclic.

Definition 7 (*Alignment cost*) Let M be a randomized algorithm that uses H as its source of randomness. Let $\phi_{D,D',\omega}$ be a local alignment for M. For any $H = (\eta_1, \eta_2, \dots)$, let $H' = (\eta'_1, \eta'_2, \dots)$ denote $\phi_{D,D',\omega}(H)$. Suppose each η_i is generated independently from a distribution f_i with the property that, for some α_i, $\ln(\frac{f_i(x)}{f_i(y)}) \leq \frac{|x-y|}{\alpha_i}$ for all x, y in the domain of f_i, then the cost of $\phi_{D,D',\omega}$ is defined as: $\text{cost}(\phi_{D,D',\omega}) = \sum_i \left|\eta_i - \eta'_i\right| / \alpha_i$. Distributions that we use in this paper (see Table 1) with this property include the Laplace (i.e., Lap(α_i)), exponential (i.e., Exp(α_i)), and geometric (i.e., Geo$(1 - e^{-1/\alpha_i})$).

[3] That is, for each input D, there might be some random vectors H for which M does not terminate, but the total probability of these vectors is 0, so we can ignore them.

The following lemma uses those properties to establish that M satisfies ϵ-differential privacy.

Lemma 1 *Let M be a randomized algorithm with input randomness $H = (\eta_1, \eta_2, \dots)$. If the following conditions are satisfied, then M satisfies ϵ-differential privacy.*

1. *M terminates with probability 1.*
2. *The number of random variables used by M can be determined from its output.*
3. *Each η_i is generated independently from a distribution f_i with the property that $\ln(f_i(x)/f_i(y)) \leq |x - y|/\alpha_i$ for all x, y in the domain of f_i.*
4. *For every $D \sim D'$ and ω there exists a local alignment $\phi_{D,D',\omega}$ that is acyclic with $\text{cost}(\phi_{D,D',\omega}) \leq \epsilon$.*
5. *For each $D \sim D'$ the number of distinct local alignments is countable. That is, the set $\{\phi_{D,D',\omega} \mid \omega \in \Omega\}$ is countable (i.e., for many choices of ω we get the same exact alignment function).*

We defer the proof to Sect. 10.

Example 3 Consider the randomness alignment $\phi_{D,D'}$ from Example 1. We can define all of the local alignments $\phi_{D,D',\omega}$ to be the same function: $\phi_{D,D',\omega}(H) = \phi_{D,D'}(H)$. Clearly $\text{cost}(\phi_{D,D',\omega}) = \sum_{i=0}^{\infty} \frac{\epsilon}{100} |\eta_i' - \eta_i| = \frac{\epsilon}{100} |q(D') - q(D)| \leq \epsilon$. For Example 2, there are two acyclic local alignments $\phi_{D,D'\top}$ and $\phi_{D,D'\bot}$, both have cost $= 100 \cdot \frac{\epsilon}{100} = \epsilon$. The other conditions in Lemma 1 are trivial to check. Thus both mechanisms satisfy ϵ-differential privacy by Lemma 1.

5 Improving sparse vector

In this section we propose an adaptive variant of SVT that can answer more queries than both the original SVT [19,33] and the SVT with Gap of Wang et al. [45]. We explain how to tune its privacy budget allocation. We further show that using other types of random noise, such as exponential and geometric random variables, in place of the Laplace, makes the free gap information more accurate at the same cost to privacy. Finally, we discuss how the free gap information can be used for improved utility of data analysis.

5.1 Adaptive SVT with Gap

The sparse vector technique (SVT) is designed to solve the following problem in a privacy-preserving way: given a stream of queries (with sensitivity 1), find the first k queries whose answers are larger than a public threshold T. This is done by adding noise to the queries and threshold and finding the first k queries whose noisy answers exceed the noisy threshold. Sometimes this procedure creates a feeling of regret—if these k queries are much larger than the threshold, we could have used more noise (hence consumed less privacy budget) to achieve the same result. In this section, we show that sparse vector can be made adaptive—so that it will *probably* use more noise (less privacy budget) for the larger queries. This means if the first k queries are very large, it will still have privacy budget left over to find additional queries that are likely to be over the threshold. Adaptive SVT is shown in Algorithm 1.

Algorithm 1: Adaptive SVT with Gap. The hyperparameter $\theta \in (0, 1)$ controls the budget allocation between threshold and queries.

input : $\boldsymbol{q}$: a list of queries of global sensitivity 1
D: database, ϵ: privacy budget, T: threshold
k: minimum number of above-threshold queries algorithm is able to output

1 **function** AdaptiveSparse ($\boldsymbol{q}$, D, T, k, ϵ):
2 $\quad\epsilon_0 \leftarrow \theta\epsilon$; $\epsilon_1 \leftarrow (1-\theta)\epsilon/k$; $\epsilon_2 \leftarrow \epsilon_1/2$
3 $\quad\sigma \leftarrow 2\sqrt{2}/\epsilon_2$ // std dev of Lap($2/\epsilon_2$)
4 $\quad\eta \leftarrow \text{Lap}(1/\epsilon_0)$; $\widetilde{T} \leftarrow T + \eta$
5 $\quad\text{cost} \leftarrow \epsilon_0$
6 $\quad$**foreach** i $\in \{1, \cdots, \text{len}(\boldsymbol{q})\}$ **do**
7 $\quad\quad\xi_i \leftarrow \text{Lap}(2/\epsilon_2)$; $\tilde{q}_i \leftarrow q_i(D) + \xi_i$
8 $\quad\quad\eta_i \leftarrow \text{Lap}(2/\epsilon_1)$; $\hat{q}_i \leftarrow q_i(D) + \eta_i$
9 $\quad\quad$**if** $\tilde{q}_i - \widetilde{T} \geq 2\sigma$ **then**
10 $\quad\quad\quad$**output:** $(\top, \tilde{q}_i - \widetilde{T}, \text{bud\_used} = \epsilon_2)$
11 $\quad\quad\quad\text{cost} \leftarrow \text{cost} + \epsilon_2$
12 $\quad\quad$**else if** $\hat{q}_i - \widetilde{T} \geq 0$ **then**
13 $\quad\quad\quad$**output:** $(\top, \hat{q}_i - \widetilde{T}, \text{bud\_used} = \epsilon_1)$
14 $\quad\quad\quad\text{cost} \leftarrow \text{cost} + \epsilon_1$
15 $\quad\quad$**else**
16 $\quad\quad\quad$**output:** $(\bot, \text{bud\_used} = 0)$
17 $\quad\quad$**if** cost $> \epsilon - \epsilon_1$ **then break**

The main idea behind this algorithm is that, given a target privacy budget ϵ and an integer k, the algorithm will create three budget parameters: ϵ_0 (budget for the threshold), ϵ_1 (baseline budget for each query) and ϵ_2 (smaller alternative budget for each query, $\epsilon_2 < \epsilon_1$). The privacy budget allocation between threshold and queries is controlled by a hyperparameter $\theta \in (0, 1)$ on Line 2. These budget parameters are used as follows. First, the algorithm adds $\text{Lap}(1/\epsilon_0)$ noise to the threshold and consumes ϵ_0 of the privacy budget. Then, when a query comes in, the algorithm first adds a lot of noise (i.e., $\text{Lap}(2/\epsilon_2)$) to the query. The first "if" branch checks if this value is much larger than the noisy threshold (i.e., checks if the gap is $\geq 2\sigma$ for some[4] σ). If so, then it outputs the following three items: (1) $\top$, (2) the noisy gap, and (3) the amount of privacy budget used for this query

[4] In our algorithm, we set σ to be the standard deviation of the noise distribution.

(which is ϵ_2). The use of alignments will show that failing this "if" branch consumes no privacy budget. If the first "if" branch fails, then the algorithm adds more moderate noise (i.e., $\text{Lap}(2/\epsilon_1)$) to the query answer. If this noisy value is larger than the noisy threshold, the algorithm outputs: $(1')$ $\top$, $(2')$ the noisy gap, and $(3')$ the amount of privacy budget consumed (i.e., ϵ_1). If this "if" condition also fails, then the algorithm outputs: $(1'')$ $\perp$ and $(2'')$ the privacy budget consumed (0 in this case).

To summarize, there is a one-time cost for adding noise to the threshold. Then, for each query, if the top branch succeeds the privacy budget consumed is ϵ_2, if the middle branch succeeds, the privacy cost is ϵ_1, and if the bottom branch succeeds, there is no additional privacy cost. These properties can be easily seen by focusing on the local alignment—if $M(D, H)$ produces a certain output, how much does H need to change to get a noise vector H' so that $M(D', H')$ returns the same exact output.

Local alignment To create a local alignment for each pair $D \sim D'$, let $H = (\eta, \xi_1, \eta_1, \xi_2, \eta_2, \ldots)$ where η is the noise added to the threshold T, and ξ_i (resp. η_i) is the noise that should be added to the ith query q_i in Line 7 (resp. Line 8), if execution ever reaches that point. We view the output $\omega = (w_1, \ldots, w_s)$ as a variable-length sequence where each w_i is either $\perp$ or a nonnegative gap (we omit the $\top$ as it is redundant), together with a tag $\in \{0, \epsilon_1, \epsilon_2\}$ indicating which branch w_i is from (and the privacy budget consumed to output w_i). Let $\mathcal{I}_\omega = \{i \mid \text{tag}(w_i) = \epsilon_2\}$ and $\mathcal{J}_\omega = \{i \mid \text{tag}(w_i) = \epsilon_1\}$. That is, $\mathcal{I}_\omega$ is the set of indexes where the output is a gap from the top branch, and $\mathcal{J}_\omega$ is the set of indexes where the output is a gap from the middle branch. For $H \in \mathcal{H}_{D:\omega}$ define $\phi_{D,D',\omega}(H) = H' = (\eta', \xi_1', \eta_1', \xi_2', \eta_2', \ldots)$ where

$$\eta' = \eta + 1,$$
$$(\xi_i', \quad \eta_i') = \begin{cases} (\xi_i + 1 + q_i - q_i', \quad \eta_i), & i \in \mathcal{I}_\omega \\ (\xi_i, \quad \eta_i + 1 + q_i - q_i'), & i \in \mathcal{J}_\omega \\ (\xi_i, \quad \eta_i), & \text{otherwise} \end{cases} \tag{2}$$

In other words, we add 1 to the noise that was added to the threshold. (Thus if the noisy $q(D)$ failed a specific branch, the noisy $q(D')$ will continue to fail it because of the higher noisy threshold.) If a noisy $q(D)$ succeeded in a specific branch, we adjust the query's noise so that the noisy version of $q(D')$ will succeed in that same branch.

Lemma 2 *Let M be the Adaptive SVT with Gap algorithm. For all $D \sim D'$ and ω, the functions $\phi_{D,D',\omega}$ defined above are acyclic local alignments for M. Furthermore, for every pair $D \sim D'$, there are countably many distinct $\phi_{D,D',\omega}$.*

Proof Pick an adjacent pair $D \sim D'$ and an $\omega = (w_1, \ldots, w_s)$. For a given $H = (\eta, \xi_1, \eta_1, \ldots)$ such that $M(D, H) = \omega$, let $H' = (\eta', \xi_1', \eta_1', \ldots) = \phi_{D,D',\omega}(H)$. Suppose $M(D', H') = \omega' = (w_1', \ldots, w_t')$. Our goal is to show $\omega' = \omega$. Choose an $i \le \min(s, t)$.

- If $i \in \mathcal{I}_\omega$, then by (2) we have

$$\begin{aligned} &q_i' + \xi_i' - (T + \eta') \\ &\quad = q_i' + \xi_i + 1 + q_i - q_i' - (T + \eta + 1) \\ &\quad = q_i + \xi_i - (T + \eta) \ge \sigma. \end{aligned}$$

  This means the first "if" branch succeeds in both executions and the gaps are the same. Therefore, $w_i' = w_i$.
- If $i \in \mathcal{J}_\omega$, then by (2) we have

$$\begin{aligned} &q_i' + \xi_i' - (T + \eta') = q_i' + \xi_i - (T + \eta + 1) \\ &\quad = q_i' - 1 + \xi_i - (T + \eta) \le q_i + \xi_i - (T + \eta) < \sigma, \\ &q_i' + \eta_i' - (T + \eta') = q_i' + \eta_i + 1 \\ &\qquad + q_i - q_i' - (T + \eta + 1) \\ &\quad = q_i + \eta_i - (T + \eta) \ge 0. \end{aligned}$$

  The first inequality is due to the sensitivity restriction: $|q_i - q_i'| \le 1 \implies q_i' - 1 \le q_i$. These two equations mean that the first "if" branch fails and the second "if" branch succeeds in both executions, and the gaps are the same. Hence $w_i' = w_i$.
- If $i \notin \mathcal{I}_\omega \cup \mathcal{J}_\omega$, then by a similar argument we have

$$\begin{aligned} &q_i' + \xi_i' - (T + \eta') \le q_i + \xi_i - (T + \eta) < \sigma, \\ &q_i' + \eta_i' - (T + \eta') \le q_i + \eta_i - (T + \eta) < 0. \end{aligned}$$

  Hence both executions go to the last "else" branch and $w_i' = (\perp, 0) = w_i$.

Therefore for all $1 \le i \le \min(s, t)$, we have $w_i' = w_i$. That is, either ω' is a prefix of ω, or vice versa. Let $\boldsymbol{q}$ be the vector of queries passed to the algorithm and let $\text{len}(\boldsymbol{q})$ be the number of queries it contains (which can be finite or infinity). By the termination condition of Algorithm 1 we have two possibilities.

1. $s = \text{len}(\boldsymbol{q})$: in this case there is still enough privacy budget left after answering $s - 1$ above-threshold queries, and we must have $t = \text{len}(\boldsymbol{q})$ too because $M(D', H')$ will also run through all the queries. (It cannot stop until it has exhausted the privacy budget or hits the end of the query sequence.)
2. $s < \text{len}(\boldsymbol{q})$: in this case the privacy budget is exhausted after outputting w_s and we must also have $t = s$.

Thus $t = s$ and hence $\omega' = \omega$. The local alignments are clearly acyclic (e.g., use the identity permutation). Note that

$\phi_{D,D',\omega}$ only depends on ω through $\mathcal{I}_\omega$ and $\mathcal{J}_\omega$ (the sets of queries whose noisy values were larger than the noisy threshold). There are only countably many possibilities for $\mathcal{I}_\omega$ and $\mathcal{J}_\omega$ and thus countably many distinct $\phi_{D,D',\omega}$. □

Alignment cost and privacy Now we establish the alignment cost and the privacy property of Algorithm 1.

Theorem 2 *The Adaptive SVT with Gap satisfies ϵ-differential privacy.*

Proof First, we bound the cost of the alignment function defined by Eq. (2). We use the $\epsilon_0, \epsilon_1, \epsilon_2$ and ϵ defined in Algorithm 1. From (2) we have

$$\begin{aligned}
&\text{cost}(\phi_{D,D',\omega}) \\
&= \epsilon_0 \left|\eta' - \eta\right| + \sum_{i=1}^{\infty} \left(\frac{\epsilon_2}{2} \left|\xi_i' - \xi_i\right| + \frac{\epsilon_1}{2} \left|\eta_i' - \eta_i\right| \right) \\
&= \epsilon_0 + \sum_{i \in \mathcal{I}_\omega} \frac{\epsilon_2}{2} \left|1 + q_i - q_i'\right| + \sum_{i \in \mathcal{J}_\omega} \frac{\epsilon_1}{2} \left|1 + q_i - q_i'\right| \\
&\le \epsilon_0 + \epsilon_2 \left|\mathcal{I}_\omega\right| + \epsilon_1 \left|\mathcal{J}_\omega\right| \le \epsilon.
\end{aligned}$$

The first inequality is from the assumption on sensitivity: $\left|1 + q_i - q_i'\right| \le 1 + \left|q_i - q_i'\right| \le 2$. The second inequality is from loop invariant on Line 17: $\epsilon_0 + \epsilon_2 \left|\mathcal{I}_\omega\right| + \epsilon_1 \left|\mathcal{J}_\omega\right| = \texttt{cost} \le \epsilon - \epsilon_1 + \max(\epsilon_1, \epsilon_2) = \epsilon$.

Conditions 1, 2, 3 of Lemma 1 are trivial to check, 4 and 5 follow from Lemma 2 and the above bound on cost. Thus Theorem 2 follows from Lemma 1. □

Algorithm 1 can be easily extended with multiple additional "if" branches. For simplicity we do not include such variations. In our setting, $\epsilon_2 = \epsilon_1/2$ so, theoretically, if queries are very far from the threshold, our adaptive version of Sparse Vector will be able to find twice as many of them as the non-adaptive version. Lastly, if all queries are monotonic queries, then Algorithm 1 can be further improved: We can use $\text{Lap}(1/\epsilon_2)$ in Line 7 and $\text{Lap}(1/\epsilon_1)$ noises in 8 instead.[5]

Choice of θ. We can optimize the budget allocation between threshold noise and query noises by following the methodology of [33], which is equivalent to minimizing the variance of the gap between a noisy query and the threshold. If the majority of gaps are expected to be returned from the top branch, then we optimize $\text{Var}(\tilde{q}_i - \widetilde{T}) = \frac{2}{\epsilon_0^2} + \frac{8}{\epsilon_2^2} = \frac{2}{\epsilon^2}(\frac{1}{\theta^2} + \frac{16k^2}{(1-\theta)^2})$. This variance attains its minimum value of $2(1+\sqrt[3]{16k^2})^3/\epsilon^2$ when $\theta = 1/(1+\sqrt[3]{16k^2})$. If on the other hand the majority of gaps are expected to be returned from the middle branch, then we optimize $\text{Var}(\hat{q}_i - \widetilde{T}) = \frac{2}{\epsilon_0^2} + \frac{8}{\epsilon_1^2} = \frac{2}{\epsilon^2}(\frac{1}{\theta^2} + \frac{4k^2}{(1-\theta)^2})$. In this case, the minimum value is $2(1+\sqrt[3]{4k^2})^3/\epsilon^2$ when $\theta = 1/(1+\sqrt[3]{4k^2})$. If all queries are monotone, then the optimal variance further reduces to $2(1+\sqrt[3]{4k^2})^3/\epsilon^2$ in the top branch when $\theta = 1/(1+\sqrt[3]{4k^2})$, and $2(1+\sqrt[3]{k^2})^3/\epsilon^2$ in the middle branch when $\theta = 1/(1+\sqrt[3]{k^2})$.

These allocation strategies also extend to SVT with Gap (originally proposed in [45]). SVT with Gap can be obtained by removing the first branch of Algorithm 1 (Line 9 through 11) or setting $\sigma = \infty$. For reference, we show its pseudocode as Algorithm 2. In [45], θ is set to 0.5, which is suboptimal. The optimal value is $\theta = 1/(1+\sqrt[3]{4k^2})$.

Algorithm 2: SVT with Gap [45]

input : same as Algorithm 1

function GapSparse ($\boldsymbol{q}$, D, T, k, ϵ):
- $\epsilon_0 \leftarrow \theta\epsilon$; $\epsilon_1 \leftarrow (1-\theta)\epsilon/k$;
- $\eta \leftarrow \text{Lap}(1/\epsilon_0)$; $\widetilde{T} \leftarrow T + \eta$
- cost $\leftarrow \epsilon_0$
- **foreach** i $\in \{1, \cdots, \text{len}(\boldsymbol{q})\}$ **do**
  - $\eta_i \leftarrow \text{Lap}(2/\epsilon_1)$; $\tilde{q}_i \leftarrow q_i(D) + \eta_i$
  - **if** $\tilde{q}_i - \widetilde{T} \ge 0$ **then**
    - **output:** $(\top, \tilde{q}_i - \widetilde{T}$, bud_used $= \epsilon_1)$
    - cost $\leftarrow$ cost $+ \epsilon_1$
  - **else**
    - **output:** $(\perp$, bud_used $= 0)$
  - **if** cost $> \epsilon - \epsilon_1$ **then break**

5.2 Using exponential or geometric noise

In this section, we show that Adaptive SVT with Gap also satisfies differential privacy if the Laplace noise is replaced by the exponential distribution or the geometric distribution (when query answers are guaranteed to be integers). Both of these are one-sided distributions that result in a gap estimate with lower variance (see Table 1 for information about those distributions). The same result carries over to SVT with Gap [45].

Exponential noise When using random noise from the exponential distribution, we need to subtract off the expected value of the noise from the queries and threshold. We make the following changes to Lines 3, 4, 7 and 8 of Algorithm 1:

3 $\sigma \leftarrow 2/\epsilon_2$ //std dev of $\text{Exp}(2/\epsilon_2)$

4 $\eta \leftarrow \text{Exp}(1/\epsilon_0)$; $\widetilde{T} \leftarrow T + \eta - 1/\epsilon_0$

7 $\xi_i \leftarrow \text{Exp}(2/\epsilon_2)$; $\tilde{q}_i \leftarrow q_i(D) + \xi_i - 2/\epsilon_2$

8 $\eta_i \leftarrow \text{Exp}(2/\epsilon_1)$; $\hat{q}_i \leftarrow q_i(D) + \eta_i - 2/\epsilon_1$

[5] In the case of monotonic queries, if $\forall i : q_i \ge q_i'$, then the alignment changes slightly: We set $\eta' = \eta$ (the random variable added to the threshold) and set the adjustment to noise in the winning "if" branches to $q_i - q_i'$ instead of $1 + q_i - q_i'$. (Hence cost terms become $|q_i - q_i'|$ instead of $|1 + q_i - q_i'|$.) If $\forall i : q_i \le q_i'$ then we keep the original alignment but in the cost calculation we note that $|1 + q_i - q_i'| \le 1$ (due to the monotonicity and sensitivity).

In more detail, the changes are:

1. Line 3: the algorithm changes the value of σ from $2\sqrt{2}/\epsilon_2$, the standard deviation of Lap$(2/\epsilon_2)$, to $2/\epsilon_2$, the standard deviation of Exp$(2/\epsilon_2)$. It is worth repeating that the one-sided exponential noise results in a reduction of variance.
2. Lines 4, 7 and 8: change Laplace noises to exponential noises of the same scale, and then subtracts the expected values of the noises.

If all queries are counting queries, we further replace ϵ_1 and ϵ_2 in Line 3, 7 and 8 with $2\epsilon_1$ and $2\epsilon_2$ respectively.

Geometric noise When all queries have integer values (e.g., counting queries), we could utilize geometric noise to ensure that the gap is also an integer. To do so we make the following changes to Algorithm 1:

3 $\sigma \leftarrow e^{\frac{\epsilon_2}{4}}/(e^{\frac{\epsilon_2}{2}}-1)$ `//std dev of` Geo$\left(1-e^{-\frac{\epsilon_2}{2}}\right)$

4 $\eta \leftarrow$ Geo$\left(1-e^{-\epsilon_0}\right)$; $\widetilde{T} \leftarrow T+\eta-1/\left(1-e^{-\epsilon_0}\right)$

7 $\xi_i \leftarrow$ Geo$\left(1-e^{-\frac{\epsilon_2}{2}}\right)$; $\tilde{q}_i \leftarrow q_i(D)+\xi_i-1/\left(1-e^{-\frac{\epsilon_2}{2}}\right)$

8 $\eta_i \leftarrow$ Geo$\left(1-e^{-\frac{\epsilon_1}{2}}\right)$; $\hat{q}_i \leftarrow q_i(D)+\eta_i-1/\left(1-e^{-\frac{\epsilon_1}{2}}\right)$

If all queries are counting queries, we further replace ϵ_1 and ϵ_2 in Line 3, 7 and 8 with $2\epsilon_1$ and $2\epsilon_2$ respectively.

Local alignment and privacy The alignment in Eq. 2 for the Adaptive SVT with Gap with Laplace noise also works for both exponential noise and geometric noise, because $\eta'-\eta = 1$ and $\xi_i'-\xi_i, \eta_i'-\eta_i \in \{0, 1+q_i-q_i'\}$. The value $1+q_i-q_i'$ is always ≥ 0 and is an integer when q_i, q_i' are integers.

Recall that if $f(x)$ is the probability density function of Exp(β), then $\ln \frac{f(x)}{f(y)} \leq \frac{1}{\beta}|x-y|$. Similarly, if $g(x)$ is the probability mass function of Geo(p), then $\ln \frac{g(x)}{g(y)} = \ln \frac{p(1-p)^x}{p(1-p)^y} \leq -\ln(1-p)\,|x-y|$. Therefore, our choice of the parameters ensures that the alignment cost is the same as that of Laplace noise, which is bounded by ϵ. Thus both variants are ϵ-differentially private.

Choice of θ. As before, we choose the θ that minimizes the variance of the gap to make the result most accurate. Note that exponential distribution has half the variance of the Laplace distribution of the same scale. Thus, when exponential noise is used, the minimum variance of the gap is $(1+\sqrt[3]{16k^2})^3/\epsilon^2$ in the top branch when $\theta = 1/(1+\sqrt[3]{16k^2})$, and $(1+\sqrt[3]{4k^2})^3/\epsilon^2$ in the middle branch when $\theta = 1/(1+\sqrt[3]{4k^2})$. If all queries are monotone, then the optimal variance further reduces to $(1+\sqrt[3]{4k^2})^3/\epsilon^2$ in the top branch when $\theta = 1/(1+\sqrt[3]{4k^2})$, and $(1+\sqrt[3]{k^2})^3/\epsilon^2$ in the middle branch when $\theta = 1/(1+\sqrt[3]{k^2})$.

Since the geometric distribution is the discrete analogue of the exponential distribution, the above results apply to geometric noise as well. For example, when all queries are counting queries and geometric noise is used, then $\mathrm{Var}(\hat{q}_i - \widetilde{T}) = \frac{e^{\epsilon_0}}{(e^{\epsilon_0}-1)^2} + \frac{e^{\epsilon_1}}{(e^{\epsilon_1}-1)^2} = \frac{e^{\theta\epsilon}}{(e^{\theta\epsilon}-1)^2} + \frac{e^{(1-\theta)\epsilon/k}}{(e^{(1-\theta)\epsilon/k}-1)^2}$ in the middle branch. The variance of the gap, albeit complicated, is a convex function of θ on $(0,1)$. We used the LBFGS algorithm [39] from SciPy to find the θ where the variance is minimum, and found that those values are almost the same as those for exponential noise (see Fig. 1). Therefore, we can use the budget allocation strategy for exponential noise as the strategy for geometric noise too.

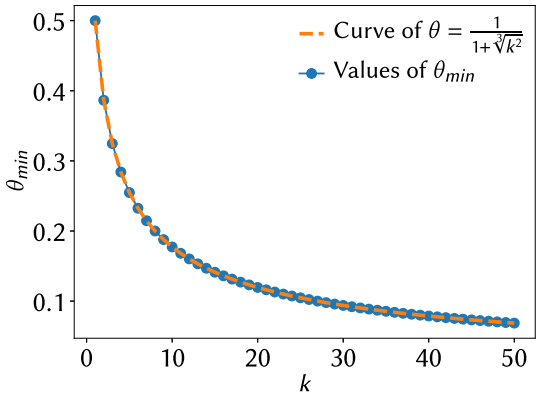


Fig. 1 The blue dots are values of $\theta_{\min} = \mathrm{argmin}(\frac{e^{\theta\epsilon}}{(e^{\theta\epsilon}-1)^2} + \frac{e^{(1-\theta)\epsilon/k}}{(e^{(1-\theta)\epsilon/k}-1)^2})$ for k from 1 to 50. The orange curve is the function $\theta = 1/(1+\sqrt[3]{k^2})$ (colour figure online)

5.3 Utilizing gap information

When SVT with Gap or Adaptive SVT with Gap returns a gap γ_i for a query q_i, we can add to it the public threshold T. This means $\gamma_i + T$ is an estimate of the value of $q_i(D)$. We can ask two questions: how can we improve the accuracy of this estimate and how can we be confident that the true answer $q_i(D)$ is really larger than the threshold T?

Lower confidence interval Recall that the randomness in the gap in Adaptive SVT with Gap (Algorithm 1) is of the form $\eta_i - \eta$ where η and η_i are independent zero mean Laplace variables with scale $1/\epsilon_0$ and $1/\epsilon_*$, where ϵ_* is either ϵ_1 or ϵ_2, depending on the branch. The random variable $\eta_i - \eta$ has the following lower tail bound:

Lemma 3 *For any $t \geq 0$ we have*

$$\mathbb{P}(\eta_i - \eta \geq -t) = \begin{cases} 1 - \frac{\epsilon_0^2 e^{-\epsilon_* t} - \epsilon_*^2 e^{-\epsilon_0 t}}{2(\epsilon_0^2 - \epsilon_*^2)} & \epsilon_0 \neq \epsilon_* \\ 1 - \left(\frac{2+\epsilon_0 t}{4}\right) e^{-\epsilon_0 t} & \epsilon_0 = \epsilon_* \end{cases}$$

For proof see "Appendix." For any confidence level, say 95%, we can use this result to find a number $t_{0.95}$ such that $\mathbb{P}((\eta_i - \eta) \geq -t_{0.95}) = 0.95$. This is a lower confidence bound, so that the true value $q_i(D)$ is $\geq$ our estimated value $\gamma_i + T$ minus $t_{0.95}$ with probability 0.95.

Improving accuracy To improve accuracy, one can split the privacy budget ϵ in half. The first half $\epsilon' \equiv \epsilon/2$ can be used to run SVT with Gap (or Adaptive SVT with Gap) and the second half $\epsilon'' \equiv \epsilon/2$ can be used to provide an independent noisy measurement of the selected queries (i.e., if we selected k queries, we add $\text{Lap}(k/\epsilon'')$ noise to each one). Denote the k selected queries by $q_1, \ldots, q_k$, the noisy gaps by $\gamma_1, \ldots, \gamma_k$ and the independent noisy measurements by $\alpha_1, \ldots, \alpha_k$. The noisy estimates can be combined together with the gaps to get improved estimates β_i of $q_i(D)$ in the standard way (inverse-weighting by variance):

$$\beta_i = \left(\frac{\alpha_i}{\text{Var}(\alpha_i)} + \frac{\gamma_i + T}{\text{Var}(\gamma_i)}\right) \Big/ \left(\frac{1}{\text{Var}(\alpha_i)} + \frac{1}{\text{Var}(\gamma_i)}\right).$$

Note that $\frac{\text{Var}(\beta_i)}{\text{Var}(\alpha_i)} = \frac{\text{Var}(\gamma_i)}{\text{Var}(\alpha_i)+\text{Var}(\gamma_i)} < 1$.

As discussed in Sect. 5.1, the optimal budget allocation between threshold noise and query noises *within* SVT with Gap is the ratio $1{:}\sqrt[3]{4k^2}$. Under this setting, we have $\text{Var}(\gamma_i) = 8(1 + \sqrt[3]{4k^2})^3/\epsilon^2$. Also, we know $\text{Var}(\alpha_i) = 8k^2/\epsilon^2$. Therefore, $\frac{E(|\beta_i - q_i|^2)}{E(|\alpha_i - q_i|^2)} = \frac{\text{Var}(\beta_i)}{\text{Var}(\alpha_i)} = \frac{(1+\sqrt[3]{4k^2})^3}{(1+\sqrt[3]{4k^2})^3+k^2}$.

Since $\lim_{k\to\infty} \frac{(1+\sqrt[3]{4k^2})^3}{(1+\sqrt[3]{4k^2})^3+k^2} = \frac{4}{5}$, the improvement in accuracy approaches 20% as k increases. For monotonic queries, the optimal budget allocation *within* SVT with Gap is $1 : \sqrt[3]{k^2}$. Then we have $\text{Var}(\gamma_i) = 8(1+\sqrt[3]{k^2})^3/\epsilon^2$ and therefore $\frac{\text{Var}(\beta_i)}{\text{Var}(\alpha_i)} = \frac{(1+\sqrt[3]{k^2})^3}{(1+\sqrt[3]{k^2})^3+k^2}$ which is close to 50% when k is large. When the algorithm uses exponential noise, the variance of the gap further reduces to $\text{Var}(\gamma_i) = 4(1 + \sqrt[3]{k^2})^3/\epsilon^2$ and therefore $\frac{\text{Var}(\beta_i)}{\text{Var}(\alpha_i)} = \frac{(1+\sqrt[3]{k^2})^3}{(1+\sqrt[3]{k^2})^3+2k^2}$ which is close to a 66% reduction of mean squared errors when k is large. Our experiments in Sect. 9 confirm this improvement.

6 Improving report noisy max

In this section, we present novel variations of the Noisy Max mechanism [19]. Given a list of queries with sensitivity 1, the purpose of Noisy Max is to estimate the identity (i.e., index) of the largest query. We show that, in addition to releasing this index, it is possible to release a numerical estimate of the gap between the values of the largest and second largest queries. This extra information comes at no additional cost to privacy, meaning that the original Noisy Max mechanism threw away useful information. This result can be generalized to the setting in which one wants to estimate the identities of the top k queries—we can release (for free) all of the gaps between each top k query and the next best query (i.e., the gap between the best and second best queries, the gap between the second and third best queries, etc.). When a user subsequently asks for a noisy answer to each of the returned queries, we show how the gap information can be used to reduce squared error by up to 66% (for counting queries).

6.1 Noisy Top-K with Gap

Our proposed Noisy Top-K with Gap mechanism is shown in Algorithm 3. (The function $\arg\max_c$ returns the top c items.) We can obtain the classical noisy max algorithm [19] from it by setting $k = 1$ and throwing away the gap information (the boxed items on Lines 6 and 7). The Noisy Top-K with Gap algorithm takes as input a sequence of n queries $q_1, \ldots, q_n$, each having sensitivity 1. It adds Laplace noise to each query. It returns the indexes $j_1, \ldots, j_k$ of the k queries with the largest noisy values in descending order. Furthermore, for each of these top k queries q_{j_i}, it releases the noisy gap between the value of q_{j_i} and the value of the next best query. Our key contribution in this section is the observation that these gaps can be released for free. That is, the classical Top-K algorithm, which does not release the gaps, satisfies ϵ-differential privacy. But, our improved version has exactly the same privacy cost yet is strictly better because of the extra information it can release.

Algorithm 3: Noisy Top-K with Gap

input: $\boldsymbol{q}$: a list of n queries of global sensitivity 1
D: database, k: # of indexes, ϵ: privacy budget

1 **function** `NoisyTopK` $(\boldsymbol{q}, D, k, \epsilon)$**:**
2 **foreach** $i \in \{1, \cdots, n\}$ **do**
3 $\eta_i \leftarrow \text{Lap}(2k/\epsilon)$; $\widetilde{q}_i \leftarrow q_i(D) + \eta_i$
4 $(j_1, \ldots, j_{k+1}) \leftarrow \arg\max_{k+1}(\widetilde{q}_1, \ldots, \widetilde{q}_n)$
5 **foreach** $i \in \{1, \cdots, k\}$ **do**
6 $\boxed{g_i \leftarrow \widetilde{q}_{j_i} - \widetilde{q}_{j_{i+1}}}$ // ith gap
7 **return** $((j_1 \boxed{, g_1}), \ldots, (j_k \boxed{, g_k}))$

We emphasize that keeping the noisy gaps hidden does not decrease the privacy cost. Furthermore, this algorithm gives estimates of the pairwise gaps between any pair of the k queries it selects. For example, suppose we are interested in estimating the gap between the ath largest and bth largest queries (where $a < b \le k$). This is equal to $\sum_{i=a}^{b-1} g_i$ because: $\sum_{i=a}^{b-1} g_i = \sum_{i=a}^{b-1} (\widetilde{q}_{j_i} - \widetilde{q}_{j_{i+1}}) = \widetilde{q}_{j_a} - \widetilde{q}_{j_b}$ and hence its variance is $\text{Var}(\widetilde{q}_{j_a} - \widetilde{q}_{j_b}) = 16k^2/\epsilon^2$.

The original Noisy Top-K mechanism satisfies ϵ-differential privacy. In the special case that all the q_i are counting queries then it satisfies $\epsilon/2$-differential privacy [19]. We will show the same properties for Noisy Top-K with Gap. We prove the privacy property in this section and then in Sect. 6.3 we show how to use this gap information.

Local alignment To prove the privacy of Algorithm 3, we need to create a local alignment function for each possible

pair $D \sim D'$ and output ω. Note that our mechanism uses precisely n random variables. Let $H = (\eta_1, \eta_2, \dots)$ where η_i is the noise that should be added to the ith query. We view the output $\omega = ((j_1, g_1), \dots, (j_k, g_k))$ as k pairs where in the ith pair (j_i, g_i), the first component j_i is the index of ith largest noisy query and the second component g_i is the gap in noisy value between the ith and $(i+1)$th largest noisy queries. As in prior work [19], we will base our analysis on continuous noise so that the probability of ties among the top $k+1$ noisy queries is 0. Thus each gap is positive: $g_i > 0$.

Let $\mathcal{I}_\omega = \{j_1, \dots, j_k\}$ and $\mathcal{I}_\omega^c = \{1, \dots, n\} \setminus \mathcal{I}_\omega$, i.e., $\mathcal{I}_\omega$ is the index set of the k largest noisy queries selected by the algorithm and $\mathcal{I}_\omega^c$ is the index set of all unselected queries. For $H \in \mathcal{H}_{D:\omega}$ define $\phi_{D,D',\omega}(H) = H' = (\eta_1', \eta_2', \dots)$ as

$$\eta_i' = \begin{cases} \eta_i & i \in \mathcal{I}_\omega^c \\ \eta_i + q_i - q_i' + \max\limits_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max\limits_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) & i \in \mathcal{I}_\omega \end{cases} \tag{3}$$

The idea behind this local alignment is simple: We want to keep the noise of the losing queries the same (when the input is D or its neighbor D'). But, for each of the k selected queries, we want to align its noise to make sure it wins by the same amount when the input is D or its neighbor D'.

Lemma 4 *Let M be the Noisy Top-K with Gap algorithm. For all $D \sim D'$ and ω, the functions $\phi_{D,D',\omega}$ defined above are acyclic local alignments for M. Furthermore, for every pair $D \sim D'$, there are countably many distinct $\phi_{D,D',\omega}$.*

Proof Given $D \sim D'$ and $\omega = ((j_1, g_1), \dots, (j_k, g_k))$, for any $H = (\eta_1, \eta_2, \dots)$ such that $M(D, H) = \omega$, let $H' = (\eta_1', \eta_2', \dots) = \phi_{D,D',\omega}(H)$. We show that $M(D', H') = \omega$. Since $\phi_{D,D',\omega}$ is identity on components $i \in \mathcal{I}_\omega^c$, we have $\max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l') = \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l)$. From (3) we have that when $i \in \mathcal{I}_\omega$,

$$\begin{aligned} \eta_i' &= \eta_i + q_i - q_i' + \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) \\ &\implies q_i' + \eta_i' - \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) = q_i \\ &\quad + \eta_i - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) \\ &\implies q_i' + \eta_i' - \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l') = q_i \\ &\quad + \eta_i - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) \end{aligned}$$

So, for the kth selected query, $(q_{j_k}' + \eta_{j_k}') - \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l') = (q_{j_k} + \eta_{j_k}) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) = g_k > 0$. This means on D' the noisy query with index j_k is larger than the best of the unselected noisy queries by the same margin as it is on D. Furthermore, for all $1 \le i < k$, we have

$$\begin{aligned} &\left(q_{j_i}' + \eta_{j_i}'\right) - \left(q_{j_{i+1}}' + \eta_{j_{i+1}}'\right) \\ &= \left(q_{j_i} + \eta_{j_i} + \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right) \\ &- \left(q_{j_{i+1}} + \eta_{j_{i+1}} + \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right) \\ &= \left(q_{j_i} + \eta_{j_i}\right) - \left(q_{j_{i+1}} + \eta_{j_{i+1}}\right) = g_i > 0. \end{aligned}$$

In other words, the query with index j_i is still the ith largest query on D' by the same margin. Therefore, $M(D', H') = \omega$.

The local alignments are clearly acyclic (any permutation that puts $\mathcal{I}_\omega^c$ before $\mathcal{I}_\omega$ does the trick). Also, note that $\phi_{D,D',\omega}$ only depends on ω through $\mathcal{I}_\omega$ (the indexes of the k largest queries). There are n queries and therefore $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ distinct $\phi_{D,D',\omega}$. □

Alignment cost and privacy To establish the alignment cost, we need the following lemma.

Lemma 5 *Let $(x_1, \dots, x_m), (x_1', \dots, x_m') \in \mathbb{R}^m$ be such that $\forall i, \left|x_i - x_i'\right| \le 1$. Then $\left|\max_i(x_i) - \max_i(x_i')\right| \le 1$.*

Proof Let s be an index that maximizes x_i and let t be an index that maximizes x_i'. Without loss of generality, assume $x_s \ge x_t'$. Then $x_s \ge x_t' \ge x_s' \ge x_s - 1$. Hence $\left|x_s - x_t'\right| = x_s - x_t' \le x_s - (x_s - 1) = 1$.

Theorem 3 *The Noisy Top-K with Gap mechanism satisfies ϵ-differential privacy. If all of the queries are counting queries, then it satisfies $\epsilon/2$-differential privacy.*

Proof First we bound the cost of the alignment function defined in (3). Recall that the η_i's are independent $\mathrm{Lap}(2k/\epsilon)$ random variables. By Definition 7

$$\begin{aligned} \mathrm{cost}(\phi_{D,D',\omega}) &= \sum_{i=1}^{\infty} \left|\eta_i' - \eta_i\right| \frac{\epsilon}{2k} \\ &= \frac{\epsilon}{2k} \sum_{i \in \mathcal{I}_\omega} \left|q_i - q_i' + \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right|. \end{aligned}$$

By the global sensitivity assumption we have $\left|q_i - q_i'\right| \le 1$. Apply Lemma 5 to the vectors $(q_l + \eta_l)_{l \in \mathcal{I}_\omega^c}$ and $(q_l' + \eta_l)_{l \in \mathcal{I}_\omega^c}$, we have $\left|\max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right| \le 1$. Therefore,

$$\begin{aligned} &\left|q_i - q_i' + \max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right| \\ &\le \left|q_i - q_i'\right| + \left|\max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l)\right| \\ &\le 1 + 1 = 2. \end{aligned}$$

Furthermore, if $\boldsymbol{q}$ is monotonic, then

– either $\forall i : q_i \le q_i'$ in which case $q_i - q_i' \in [-1, 0]$ and $\max_{l \in \mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l \in \mathcal{I}_\omega^c}(q_l + \eta_l) \in [0, 1]$,

– or $\forall i : q_i \geq q_i'$ in which case $q_i - q_i' \in [0,1]$ and $\max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l) \in [-1, 0]$.

In both cases we have $q_i - q_i' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l) \in [-1, 1]$ so $|q_i - q_i' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)| \leq 1$. Therefore,

$$
\begin{aligned}
&\text{cost}(\phi_{D,D',\omega}) \\
&= \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega}\left|q_i - q_i' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)\right| \\
&\leq \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega} 2 \quad \left(\text{or } \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega} 1 \text{ if } \boldsymbol{q} \text{ is monotonic}\right) \\
&= \frac{\epsilon}{2k}\cdot 2\,|\mathcal{I}_\omega| \quad \left(\text{or } \frac{\epsilon}{2k}\cdot|\mathcal{I}_\omega| \text{ if } \boldsymbol{q} \text{ is monotonic}\right) \\
&= \epsilon \quad (\text{or } \epsilon/2 \text{ if } \boldsymbol{q} \text{ is monotonic})\,.
\end{aligned}
$$

Conditions 1 through 3 of Lemma 1 are trivial to check, 4 and 5 follow from Lemma 4 and the above bound on cost. Therefore, Theorem 3 follows from Lemma 1. □

6.2 Noisy top-K with exponential noise

The original noisy max algorithm also works with one-sided exponential noise [19] with smaller variance than the Laplace noise. In this subsection, we show that this result extends to the Noisy Top-K with Gap algorithm by simply changing Line 3 of Algorithm 3 to $\eta_i \leftarrow \text{Exp}(2k/\epsilon)$ and privacy is maintained while the variance of the gap decreases. However, the proof relies on a different local alignment.

Local alignment The alignment used in Sect. 6.1 will not work here because it might set our noise random variables to negative numbers. Thus we need a new alignment. As before, let $H = (\eta_1, \eta_2, \dots)$ where η_i is the noise that should be added to the ith query. We view the output $\omega = ((j_1, g_1), \dots, (j_k, g_k))$ as k pairs where in the ith pair (j_i, g_i), the first component j_i is the index of ith largest noisy query and the second component $g_i > 0$ is the gap in noisy value between the ith and $(i+1)$th largest noisy queries.

Let $\mathcal{I}_\omega = \{j_1, \dots, j_k\}$ and $\mathcal{I}_\omega^c = \{1, \dots, n\} \setminus \mathcal{I}_\omega$, i.e., $\mathcal{I}_\omega$ is the index set of the k largest noisy queries selected by the algorithm and $\mathcal{I}_\omega^c$ is the index set of all unselected queries. For $H \in \mathcal{H}_{D:\omega}$ we will use $\phi_{D,D',\omega}(H) = H' = (\eta_1', \eta_2', \dots)$ to refer to the aligned noise. In order to define the alignment, we need the following quantities:

$$
\begin{aligned}
s &= \underset{l\in\mathcal{I}_\omega^c}{\text{argmax}}(q_l + \eta_l), \quad t = \underset{l\in\mathcal{I}_\omega^c}{\text{argmax}}(q_l' + \eta_l) \\
i_* &= \text{argmin}_{i\in\mathcal{I}_\omega}\left\{q_i - q_i' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)\right\} \\
&= \text{argmin}_{i\in\mathcal{I}_\omega}\left\{q_i - q_i'\right\} \quad (\text{the other terms have no } i)
\end{aligned}
$$

$$
\begin{aligned}
\delta_* &= \min_{i\in\mathcal{I}_\omega}\left\{q_i - q_i' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)\right\} \\
&= q_{i_*} - q_{i_*}' + (q_t' + \eta_t) - (q_s + \eta_s)
\end{aligned}
$$

Note that $i_* \in \mathcal{I}_\omega$ and $s, t \in \mathcal{I}_\omega^c$. We define the alignment according to the value of δ_*. When $\delta_* \geq 0$, we use the same alignment as in the Laplace version of the algorithm:

$$
\eta_i' = \begin{cases} \eta_i & i \in \mathcal{I}_\omega^c \\ \eta_i + q_i - q_i' + (q_t' + \eta_t) - (q_s + \eta_s) & i \in \mathcal{I}_\omega \end{cases} \tag{4}
$$

When $\delta_* < 0$ that alignment could result in a negative η_i' for some $i \in \mathcal{I}_\omega$. So instead, we take that alignment and further add the positive quantity $-\delta_*$ in several places so that overall we are adding nonnegative numbers to each η_i to get η_i'. (This ensures that η_i' is nonnegative for each i.) Thus, when $\delta_* < 0$, define

$$
\begin{aligned}
\eta_i' &= \begin{cases} \eta_i & i \in \mathcal{I}_\omega^c \setminus \{t\} \\ \eta_i - \delta_* & i = t \\ \eta_i + q_i - q_i' + (q_t' + \eta_t) - (q_s + \eta_s) - \delta_* & i \in \mathcal{I}_\omega \end{cases} \\
&= \begin{cases} \eta_i & i \in \mathcal{I}_\omega^c \setminus \{t\} \\ \eta_i - \delta_* & i = t \\ \eta_i + q_i - q_i' - q_{i_*} + q_{i_*}' & i \in \mathcal{I}_\omega \end{cases}
\end{aligned} \tag{5}
$$

Lemma 6 *Let M be the Noisy Top-K with Gap algorithm that uses exponential noise. For all $D \sim D'$ and ω, the functions $\phi_{D,D',\omega}$ defined above are acyclic local alignments for M. Furthermore, for every pair $D \sim D'$, there are countably many distinct $\phi_{D,D',\omega}$.*

Proof First, we show that $\forall i, \eta_i' \geq \eta_i$. Recall that $\delta_* = \min_{i\in\mathcal{I}_\omega}\left\{q_i - q_i' + (q_t' + \eta_t) - (q_s + \eta_s)\right\}$. When $\delta_* \geq 0$, we have $\eta_i' - \eta_i = q_i - q_i' + (q_t' + \eta_t) - (q_s + \eta_s) \geq \delta_* \geq 0$ for all $i \in \mathcal{I}_\omega$. When $\delta_* < 0$, we have $\eta_t' - \eta_t = -\delta_* > 0$ and $\eta_i' - \eta_i = (q_i - q_i') - (q_{i_*} - q_{i_*}') \geq 0$ for $i \in \mathcal{I}_\omega$. Therefore, all η_i' are nonnegative.

The proof that (4) is an alignment when $\delta_* \geq 0$ is the same as in the Laplace noise case. To show that (5) is an alignment when $\delta_* < 0$, first note that since $t = \text{argmax}_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l)$ and $-\delta_* > 0$, we have $t = \text{argmax}_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l')$. Then from (5), we have that when $i \in \mathcal{I}_\omega$,

$$
\begin{aligned}
&\eta_i' = \eta_i + q_i - q_i' + (q_t' + \eta_t) - (q_s + \eta_s) - \delta_* \\
&\quad\implies q_i' + \eta_i' - (q_t' + (\eta_t - \delta_*)) = q_i \\
&\qquad + \eta_i - (q_s + \eta_s) \\
&\quad\implies q_i' + \eta_i' - (q_t' + \eta_t') = q_i + \eta_i - (q_s + \eta_s) \\
&\quad\implies q_i' + \eta_i' - \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l') = q_i \\
&\qquad + \eta_i - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)
\end{aligned}
$$

Thus by a similar argument in Lemma 4, all relative orders among the k largest noisy queries and their associated gaps are preserved. The facts that $\phi_{D,D',\omega}$ is acyclic and there are finitely many $\phi_{D,D',\omega}$ are clear. □

Alignment cost and privacy Recall from Table 1 that if $f(x)$ is the density of $\text{Exp}(\beta)$, then for $x, y \geq 0$, $\ln \frac{f(x)}{f(y)} = \frac{y-x}{\beta} \leq \frac{|y-x|}{\beta}$. When $\delta_* \geq 0$, the alignment cost computation is the same as with the Laplace version of the algorithm. When $\delta_* < 0$, we have

$$\begin{aligned}\text{cost}(\phi_{D,D',\omega}) &= \sum_{i=1}^{\infty} \left|\eta_i' - \eta_i\right| \frac{\epsilon}{2k} \\ &= \frac{\epsilon}{2k}\left|\delta_*\right| + \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega}\left|q_i - q_i' - q_{i_*} + q_{i_*}'\right| \\ &= \frac{\epsilon}{2k}\left|\delta_*\right| + \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega\setminus\{i_*\}}\left|q_i - q_i' - q_{i_*} + q_{i_*}'\right|.\end{aligned}$$

and note that there are $k-1$ terms in the right-most summation. It is clear that $\left|q_i - q_i' - q_{i_*} + q_{i_*}'\right| \leq 2$ (or 1 if $\boldsymbol{q}$ is monotone). Also, it is shown in the proof of Theorem 3 that $|\delta_*| = |q_{i_*} - q_{i_*}' + \max_{l\in\mathcal{I}_\omega^c}(q_l' + \eta_l) - \max_{l\in\mathcal{I}_\omega^c}(q_l + \eta_l)| \leq 2$ (or 1 if $\boldsymbol{q}$ is monotone). Therefore,

$$\begin{aligned}&\text{cost}(\phi_{D,D',\omega}) \\ &= \frac{\epsilon}{2k}\left|\delta_*\right| + \frac{\epsilon}{2k}\sum_{i\in\mathcal{I}_\omega\setminus\{i_*\}}\left|q_i - q_i' - q_{i_*} + q_{i_*}'\right| \\ &\quad \text{(note that there are } 1 + (k-1) = k \text{ terms above)} \\ &\leq \frac{\epsilon}{2k}\cdot 2\cdot k \quad \left(\text{or } \frac{\epsilon}{2k}\cdot 1\cdot k \text{ if } \boldsymbol{q} \text{ is monotonic}\right) \\ &= \epsilon \quad (\text{or } \epsilon/2 \text{ if } \boldsymbol{q} \text{ is monotonic}).\end{aligned}$$

Thus, Algorithm 3 with $\text{Exp}(2k/\epsilon)$ noise on Line 3 instead of $\text{Lap}(2k/\epsilon)$ noise, satisfies ϵ-differential privacy. If all of the queries are counting queries, then it satisfies $\epsilon/2$-differential privacy.

6.3 Utilizing gap information

Let us consider one scenario that takes advantage of the gap information. Suppose a data analyst is interested in the identities and values of the top k queries. A typical approach would be to split the privacy budget ϵ in half—use $\epsilon/2$ of the budget to identify the top k queries using Noisy Top-K with Gap. The remaining $\epsilon/2$ budget is evenly divided between the selected queries and is used to obtain noisy measurements (i.e., add $\text{Lap}(2k/\epsilon)$ noise to each query answer). These measurements will have variance $\sigma^2 = 8k^2/\epsilon^2$. In this section we show how to use the gap information from Noisy Top-K with Gap and postprocessing to improve the accuracy of these measurements.

Problem statement Let $q_1, \ldots, q_k$ be the true answers of the top k queries that are selected by Algorithm 3. Let $\alpha_1, \ldots, \alpha_k$ be their noisy measurements. Let $g_1, \ldots, g_{k-1}$ be the noisy gaps between $q_1, \ldots, q_k$ that are obtained from Algorithm 3 for free. Then $\alpha_i = q_i + \xi_i$ where each ξ_i is a $\text{Lap}(2k/\epsilon)$ random variable and $g_i = q_i + \eta_i - q_{i+1} - \eta_{i+1}$ where each η_i is a $\text{Lap}(4k/\epsilon)$ random variable, or a $\text{Lap}(2k/\epsilon)$ random variable if the query list is monotonic (recall the mechanism was run with a privacy budget of $\epsilon/2$). Our goal is then to find the *best linear unbiased estimate* (BLUE) [30] β_i of q_i in terms of the measurements α_i and gap information g_i.

Theorem 4 *With notations as above let* $\boldsymbol{q} = [q_1, \ldots, q_k]^{\mathrm{T}}$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k]^{\mathrm{T}}$ *and* $\boldsymbol{g} = [g_1, \ldots, g_{k-1}]^{\mathrm{T}}$. *Suppose the ratio* $\text{Var}(\xi_i) : \text{Var}(\eta_i)$ *is equal to* $1 : \lambda$. *Then the BLUE of* $\boldsymbol{q}$ *is* $\boldsymbol{\beta} = \frac{1}{(1+\lambda)k}(X\boldsymbol{\alpha} + Y\boldsymbol{g})$ *where*

$$X = \begin{bmatrix} 1+\lambda k & 1 & \cdots & 1 \\ 1 & 1+\lambda k & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1+\lambda k \end{bmatrix}_{k\times k}$$

$$Y = \left(\begin{bmatrix} k-1 & k-2 & \cdots & 1 \\ k-1 & k-2 & \cdots & 1 \\ k-1 & k-2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ k-1 & k-2 & \cdots & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ k & 0 & \cdots & 0 \\ k & k & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ k & k & \cdots & k \end{bmatrix} \right)_{k\times(k-1)}$$

For proof, see "Appendix." Even though this is a matrix multiplication, it is easy to see that it translates into the following algorithm that is linear in k:

1. Compute $\alpha = \sum_{i=1}^{k}\alpha_i$ and $p = \sum_{i=1}^{k-1}(k-i)g_i$.
2. Set $p_0 = 0$. For $i = 1, \ldots, k-1$ compute the prefix sum $p_i = \sum_{j=1}^{i} g_j = p_{i-1} + g_i$.
3. For $i = 1, \ldots, k$, set $\beta_i = (\alpha + \lambda k\alpha_i + p - kp_{i-1})/(1+\lambda)k$.

Now, each β_i is an estimate of the value of q_i. How does it compare to the direct measurement α_i (which has variance $\sigma^2 = 8k^2/\epsilon^2$)? The following result compares the expected error of β_i (which used the direct measurements and the gap information) with the expected error of using only the direct measurements (i.e., α_i only).

Corollary 1 *For all* $i = 1, \ldots, k$, *we have*

$$\frac{E(|\beta_i - q_i|^2)}{E(|\alpha_i - q_i|^2)} = \frac{1+\lambda k}{k+\lambda k} = \frac{\text{Var}(\xi_i) + k\,\text{Var}(\eta_i)}{k(\text{Var}(\xi_i) + \text{Var}(\eta_i))}.$$

For proof, see Appendix. In the case of counting queries, we have $\text{Var}(\xi_i) = \text{Var}(\eta_i) = 8k^2/\epsilon^2$ and thus $\lambda = 1$. The error reduction rate is $\frac{k-1}{2k}$ which is close to 50% when k is

large. If we use exponential noise instead, i.e., replace $\eta_i \leftarrow \text{Lap}(2k/\epsilon)$ with $\eta_i \leftarrow \text{Exp}(2k/\epsilon)$ at Line 3 of Algorithm 3, then $\text{Var}(\eta_i) = 4k^2/\epsilon^2 = \text{Var}(\xi_i)/2$ and thus $\lambda = 1/2$. In this case, the error reduction rate is $\frac{2k-2}{3k}$ which is close to 66% when k is large. Our experiments in Sect. 9 confirm these theoretical results.

7 SVT/noisy max hybrids with gap

In this section, we present two hybrids of SVT with Gap and Noisy Top-K with Gap. Recall that SVT with Gap is an *online* algorithm that returns the identities and noisy gaps (with respect to the threshold) of the first k noisy queries it sees that are larger than the noisy threshold. Its benefits are: (i) Privacy budget is saved if fewer than k queries are returned. (ii) The queries that are returned come with estimates of their noisy answers (obtained by adding the public threshold to the noisy gap), while the drawbacks are that the returned queries are likely *not* to resemble the k largest queries. (Queries that come afterward are ignored, no matter how large their values are.)

Meanwhile, Noisy Top-K with Gap returns the identities and gaps (with respect to the runner-up query) of the top k noisy queries. Its benefits are: (i) The queries returned are approximately the top k. (ii) The gap tells us how large the queries are compared to the best non-selected noisy query. The drawbacks are: (i) k queries are always returned, even if their values are small. (ii) Only gap information is returned (not estimates of the query answers).

For users who are interested in identifying the top k queries that are likely to be over a threshold, we present two hybrid algorithms that try to combine the benefits of both algorithms while minimizing the drawbacks. Both algorithms take as input a number k, a list of answers to queries having sensitivity 1, and a public threshold T. They both return a subset of the top k noisy queries *that are larger than the noisy threshold* T; hence, the privacy cost is dynamic and is smaller if fewer than k queries are returned. The difference is in the gap information.

The first hybrid (Algorithm 4) is a variant of Noisy Top-K with gap. It adds the public threshold T to the list of queries (it becomes Query 0), adds the same noise to them (Lines 2 and 4). In line 6, it takes the top k noisy queries (*sorted* in decreasing order) and their gaps with the next best query. It filters out any that are smaller than the noisy Query 0. For the queries that did not get removed, it returns their identities (recall the threshold is Query 0) and their gap with the next best query. If the last returned item is Query 0, this means that the gap information tells us how much larger the other returned queries are compared to the noisy threshold Query 0, and this allows us to get numerical estimates for those query answers by adding in the public threshold.

Algorithm 4: Hybrid Noisy Top-K with Gap

```
   input: q: a list of n queries of global sensitivity 1
          D: database, ε: privacy budget
          T: public threshold, k: # of indexes
1  function NoisyTopK (q, D, T, k, ε):
2      η_0 ← Exp(2k/ε); q̃_0 ← T + η_0
3      foreach i ∈ {1, ··· , n} do
4          η_i ← Exp(2k/ε); q̃_i ← q_i(D) + η_i
5      (j_1, ..., j_{k+1}) ← arg max_{k+1}(q̃_0, q̃_1, ..., q̃_n)
6      foreach i ∈ {1, ··· , k} do
7          g_i ← q̃_{j_i} − q̃_{j_{i+1}}; t ← i
8          if j_i = 0 then
9              break
10     return ((j_1, g_1), ..., (j_t, g_t))
```

Alignment and privacy cost for Algorithm 4. By replacing the index sets $\mathcal{I}_\omega$ in Eqs. (4) and (5) with $\mathcal{I}_\omega = \{j_1, \ldots, j_t\}$, the same formula can be used as the alignment function for Algorithm 4. Note that since $|\mathcal{I}_\omega| = t \leq k$, the privacy cost is $(t/k)\epsilon$.

Lemma 7 *If Algorithm 4 is run with privacy budget ϵ and returns t queries (and their associated gaps), then the actual privacy cost is $(t/k)\epsilon$.*

The second hybrid (Algorithm 5) is essentially SVT with Gap applied to the list of queries that is sorted in descending order by their noisy answers. We note that it adds more noise to each query than Algorithm 4 but always returns the noisy gap between the noisy query answer and the noisy threshold, just like SVT with Gap.

Algorithm 5: Hybrid Sparse Vector with Gap

```
   input: same as Algorithm 4
1  function GapSparse (q, D, T, k, ε):
2      ε_0 ← θε; ε_1 ← (1 − θ)ε/k;
3      η ← Exp(1/ε_0); T̃ ← T + η − 1/ε_0
4      foreach i ∈ {1, ··· , n} do
5          η_i ← Exp(2/ε_1); q̃_i ← q_i(D) + η_i − 2/ε_1
6      (j_1, ..., j_k) ← arg max_k(q̃_1, ..., q̃_n)
7      t ← 0
8      foreach i ∈ {1, ··· , k} do
9          if q̃_{j_i} ≥ T̃ then
10             g_i ← q̃_{j_i} − T̃; t ← i
11         else
12             break
13     return ((j_1, g_1), ..., (j_t, g_t)) // ∅ if t = 0
```

Alignment and privacy cost for Algorithm 5 The alignment for Algorithm 5 is the same as the one for SVT with Gap and is hence omitted here. Note that the privacy cost is $\epsilon_0 + t\epsilon_1 = (\theta + (t/k)(1-\theta))\epsilon$ where t is the number of queries returned. As discussed in Sect. 5.1, the optimal θ is $1/(1 + \sqrt[3]{4k^2})$.

Lemma 8 *If Algorithm 5 is run with privacy budget ϵ and returns t queries (and their associated gaps), then the actual privacy cost is $(\theta + (t/k)(1-\theta))\epsilon$.*

Benefits of the Hybrid Algorithms Compared with Noisy Top-K with Gap, the hybrid algorithms have these advantages: (i) saving privacy budget: The actual privacy budget consumption for the hybrid algorithms is dynamic—it depends on the number of queries returned. Thus if the threshold T is set high, the hybrid algorithms will likely return fewer than k queries and consume less privacy budget; (ii) providing query estimates: Algorithm 5 always returns the noisy gap with the threshold. (Hence, by adding in the public threshold value, this gives an estimate of the query answer.) Meanwhile, Algorithm 4 only returns the noisy gap with the threshold if the last query returned is the noisy threshold Query 0. (Otherwise it functions like Noisy Top-K with Gap and returns the gaps with the runner up query.)

Compared with SVT with Gap, the hybrid algorithms are trying to select the *overall* top k queries that are above the threshold, whereas SVT with Gap tries pick the *first* k queries it sees that are above the threshold. So the queries returned by the hybrid algorithms are expected to have much higher values. There is an important distinction though: SVT with Gap is an online algorithm that can process queries as they arrive, whereas the hybrid algorithms require all queries to be known beforehand.

The first hybrid (Algorithm 4) is more likely to provide accurate identity information than the second hybrid (Algorithm 5). That is, the queries it returns are more likely to be the actual queries whose true values are largest (because the first algorithm adds less noise to the query answers). However, as mentioned before Algorithm 5 always provides estimates of query answers, whereas Algorithm 4 only provides such estimates if the last query returned is the noisy threshold Query 0. Therefore, if it is more desirable to always have query answer estimates then one should use Algorithm 5. Otherwise Algorithm 4 is a good default choice.

8 Improving the exponential mechanism

The exponential mechanism [36] was designed to answer non-numeric queries in a differentially private way. In this setting, $\mathcal{D}$ is the set of possible input databases and $\mathcal{R} = \{\omega_1, \omega_2, \ldots, \omega_n\}$ is a set of possible outcomes. There is a utility function $\mu : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ where $\mu(D, \omega_i)$ gives us the utility of outputting ω_i when the true input database is D. The exponential mechanism randomly selects an output ω_i with probabilities that are defined by the following theorem:

Theorem 5 (The Exponential Mechanism [36]) *Given $\epsilon > 0$ and a utility function $\mu : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$, the mechanism $M(D, \mu, \epsilon)$ that outputs $\omega_i \in \mathcal{R}$ with probability proportional to $\exp(\frac{\epsilon\mu(D,\omega_i)}{2\Delta_\mu})$ satisfies ϵ-differential privacy where Δ_μ, the sensitivity of μ, is defined as*

$$\Delta_\mu = \max_{D \sim D'} \max_{\omega_i \in \mathcal{R}} \left|\mu(D, \omega_i) - \mu(D', \omega_i)\right|.$$

We show that the exponential mechanism can also output (for free) a type of gap information in addition to the selected index. This gap provides noisy information about the difference between the utility scores of the selected output and non-selected outputs. What is surprising about this result is that we can treat the exponential mechanism as a black box (i.e., it does not matter how the sampling is implemented). In contrast, the internal state of the noisy max algorithm was needed (i.e., the gap was computed from the noisy query answers). The details are shown in Algorithm 6, which makes use of the Logistic(θ) distribution having pdf $f(x;\theta) = \frac{e^{-(x-\theta)}}{(1+e^{-(x-\theta)})^2}$.

Algorithm 6: Exponential Mechanism w. Gap

input: μ: utility function with sensitivity Δ_μ
D: database, ϵ: privacy budget

function `GapExpMech` (D, μ, ϵ)**:**
  $\omega_s \leftarrow$ `ExpMech`(D, μ, ϵ) `// Selected query`
  $\theta \leftarrow \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \ln \sum_{j \neq s} \exp(\frac{\epsilon\mu(D,\omega_j)}{2\Delta_\mu})$
  while `true` **do**
    $g_s \leftarrow$ Logistic(θ) `// Location=`θ`, scale=1`
    if $g_s > 0$ **then**
      break
  return ω_s, g_s

Theorem 6 *Algorithm 6 satisfies ϵ-differential privacy and the expected value of g_s is $(1+e^{-\theta})\ln(1+e^{\theta})$ where s is the index of the query returned by the exponential mechanism and $\theta = \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \ln \sum_{j \neq s} \exp(\frac{\epsilon\mu(D,\omega_j)}{2\Delta_\mu})$ is the location parameter of the sampling distribution.*

Utilizing the Gap Information. From Theorem 6 and Algorithm 6, we see that θ is a kind of gap (scaled by $\epsilon/2\Delta_\mu$) between the selected query ω_s and a softmax of the remaining items. While θ can be numerically estimated from g_s, one can also use g_s for the following purpose.

The exponential mechanism is randomized, so an important question is whether it returned a query that has the highest utility. We can use the noisy gap information g_s from Algorithm 6 to answer this question in a hypothesis testing framework. Specifically, let H_0 be the null hypothesis that the returned query ω_s *does not* have the highest utility score. Then g_s can tell us how unlikely this null hypothesis is—the quantity $\mathbb{P}[g_s \geq \gamma \mid H_0]$ is the significance level (also known

Table 3 Statistics of datasets

Dataset	# of Records	# of Unique items
BMS-POS	515,597	1657
Kosarak	990,002	41,270
T40I10D100K	100,000	942

as a p value), and small values indicate the null hypothesis is unlikely. Its computation is given in Theorem 7.

Theorem 7 $\mathbb{P}[g_s \geq \gamma \mid H_0] \leq 2/(1+e^{\gamma})$.

We note that if we want a significance level of $\alpha = 0.05$ (i.e., there is less than a 5% chance that a non-optimal query could have produced a large noisy gap) then we need $g_s \geq \ln(\frac{2}{\alpha} - 1) \approx 3.66$.

9 Experiments

We evaluate the algorithms proposed in this paper using the two real datasets from [33]: BMP-POS, Kosarak and a synthetic dataset T40I10D100K created by the generator from the IBM Almaden Quest research group. These datasets are collections of transactions. (Each transaction is a set of items.) In our experiments, the queries correspond to the counts of each item (i.e., how many transactions contained item #23?)

The statistics of the datasets are listed below (Table 3).

9.1 Improving query estimates with gap information

The first set of experiments is to measure how gap information can help improve estimates in selected queries. We use the setup of Sects. 5.3 and 6.3. That is, a data analyst splits the privacy budget ϵ into half. She uses the first half to select k queries using Noisy Top-K with Gap or SVT with Gap (or Adaptive SVT with Gap) and then uses the second half of the privacy budget to obtain independent noisy measurements of each selected query.

If one were unaware that gap information came for free, one would just use those noisy measurements as estimates for the query answers. The error of this approach is the gap-free baseline. However, since the gap information does come for free, we can use the postprocessing described in Sects. 5.3 and 6.3 to improve accuracy (we call this latter approach SVT with Gap with Measures and Noisy Top-K with Gap with Measures).

We first evaluate the percentage reduction of mean squared error (MSE) of the postprocessing approach compared to the gap-free baseline and compare this improvement to our theoretical analysis. As discussed in Sect. 5.3, we set the budget allocation ratio *within* the SVT with Gap algorithm (i.e., the budget allocation between the threshold and queries) to be $1 : k^{\frac{2}{3}}$ for monotonic queries and $1 : (2k)^{\frac{2}{3}}$ otherwise—such a ratio is recommended in [33] for the original SVT. The threshold used for SVT with Gap is randomly picked from the top $2k$ to top $8k$ in each dataset for each run.[6] All numbers plotted are averaged over 10,000 runs. Due to space constraints, we only show experiments for counting queries (which are monotonic).

Our theoretical analysis in Sects. 5.3 and 6.3 suggested that in the case of monotonic queries, the error reduction rate can reach up to 50% when Laplace noise is used, and 66% when exponential or geometric noise is used, as k increases. This is confirmed in Fig. 2a, for SVT with Gap and Fig. 2b, for our Top-K algorithm using the BMS-POS dataset. (the results for the other datasets are nearly identical.) These figures plot the theoretical and empirical percent reduction of MSE as a function of k and show the power of the free gap information.

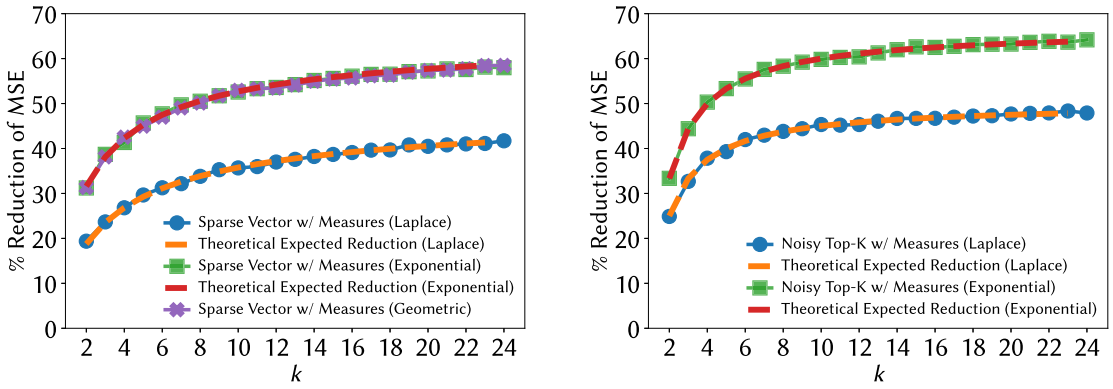
We also generated corresponding plots where k is held fixed and the total privacy budget ϵ is varied. We only present the result for the kosarak dataset as the results for the other datasets are nearly identical. For SVT with Gap, Fig. 3a confirms that this improvement is stable for different ϵ values. For our Top-K algorithm, Fig. 3b confirms that this improvement is also stable for different values of ϵ.

9.2 Benefits of adaptivity

In this section we present an evaluation of the budget-saving properties of our novel Adaptive SVT with Gap algorithm to show that it can answer more queries than SVT and SVT with Gap at the same privacy cost (or, conversely, answer the same number of queries but with leftover budget that can be used for other purposes). First note that SVT and SVT with Gap both answer exactly the same amount of queries, so we only need to compare Adaptive SVT with Gap to the original SVT [19,33]. In both algorithms, the budget allocation between the threshold noise and query noise is set according to the ratio $1 : k^{\frac{2}{3}}$ (i.e., the hyperparameter θ in Adaptive SVT with Gap is set to $1/(1+k^{\frac{2}{3}})$), following the discussion in Sect. 5.1. The threshold is randomly picked from the top $2k$ to top $8k$ in each dataset and all reported numbers are averaged over 10,000 runs.

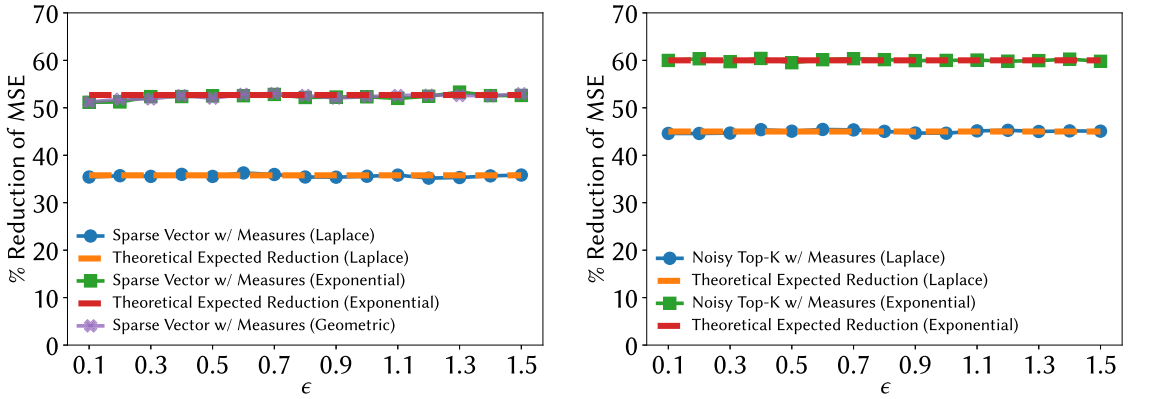
Number of queries answered. We first compare the number of queries answered by each algorithm as the parameter k is varied from 2 to 24 with a privacy budget of $\epsilon = 0.7$. (The results for other settings of the total privacy budget are similar.) The results are shown in Fig. 4a–c. In each of

[6] Selecting thresholds for SVT in experiments is difficult, but we feel this may be fairer than averaging the answer to the top kth and $k+1$th queries as was done in prior work [33].



(a) SVT with Gap with Measures, BMS-POS. **(b)** Noisy Top-K with Gap with Measures, BMS-POS.

Fig. 2 Percent reduction of mean squared error on monotonic queries, for different k, for SVT with Gap and Noisy Top-K with Gap when half the privacy budget is used for query selection and the other half is used for measurement of their answers. Privacy budget $\epsilon = 0.7$



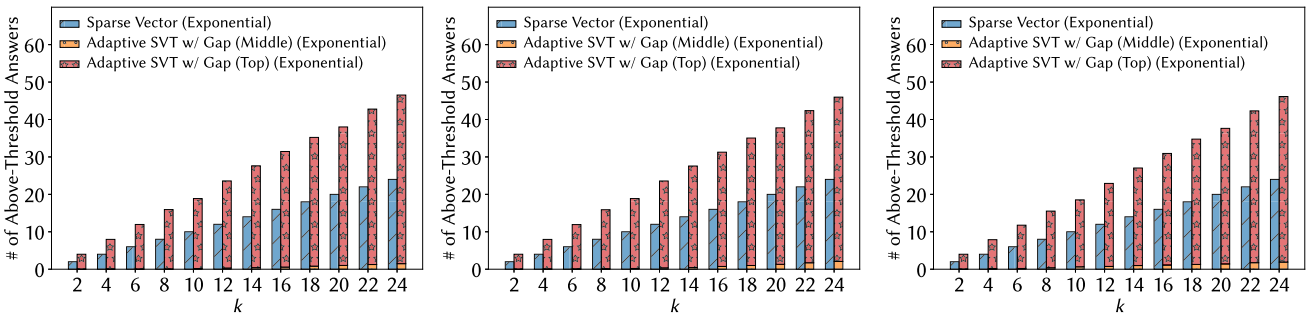
(a) SVT with Gap with Measures, kosarak. (b) Noisy Top-K with Gap with Measures, kosarak.

Fig. 3 Percent reduction of mean squared error on monotonic queries, for different ϵ, for SVT with Gap and Noisy Top-K with Gap when half the privacy budget is used for query selection and the other half is used for measurement of their answers. The value of k is set to 10

these bar graphs, the left (blue) bar is the number of answers returned by SVT and the right bar is the number of answers returned by Adaptive SVT with Gap. This right bar is broken down into two components: the number of queries returned from the top "if" branch (corresponding to queries that were significantly larger than the threshold even after a lot of noise was added) and the number of queries returned from the middle "if" branch. Queries returned from the top branch of Adaptive SVT with Gap have less privacy cost than those returned by SVT. Queries returned from the middle branch of Adaptive SVT with Gap have the same privacy cost as in SVT. We see that most queries are answered in the top branch of Adaptive SVT with Gap, meaning that the above-threshold queries were generally large (much larger than the threshold). Since Adaptive SVT with Gap uses more noise in the top branch, it uses less privacy budget to answer those queries and uses the remaining budget to provide additional answers (up to an average of 20 more answers when k was set to 24).

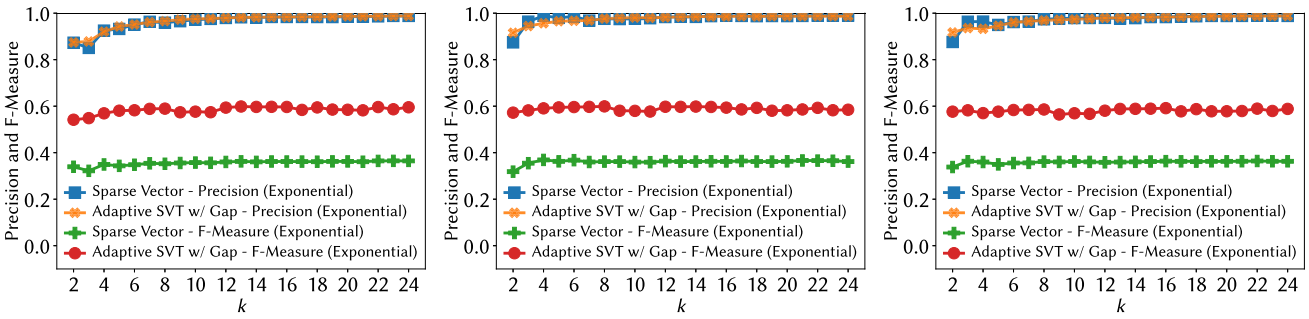
Precision and F-Measure. Although the adaptive algorithm can answer more above-threshold queries than the original, one can still ask the question of whether the returned queries really are above the threshold. Thus we can look at the precision of the returned results (the fraction of returned queries that are actually above the threshold) and the widely used F-Measure (the harmonic mean of precision and recall). One would expect that the precision of Adaptive SVT with Gap should be less than that of SVT, because the adaptive version can use more noise when processing queries. In Fig. 5a–c we compare the precision and F-Measure of the two algorithms. Generally we see very little difference in precision. On the other hand, since Adaptive SVT with Gap answers more queries while maintaining high precision, the recall of Adaptive SVT with Gap would be much larger than SVT, thus leading to the F-Measure being roughly 1.5 times that of SVT.

Remaining Privacy Budget. If a query is large, Adaptive SVT with Gap may only need to use a small part of the privacy



(a) BMS-POS. **(b)** kosarak. **(c)** T40I10D100K.

Fig. 4 # of queries answered by SVT and Adaptive SVT with Gap under different k's for monotonic queries. Privacy budget $\epsilon = 0.7$ and x-axis: k



(a) BMS-POS. **(b)** kosarak. **(c)** T40I10D100K.

Fig. 5 Precision and F-measure of SVT and Adaptive SVT with Gap under different k's for monotonic queries. Privacy budget $\epsilon = 0.7$ and x-axis: k

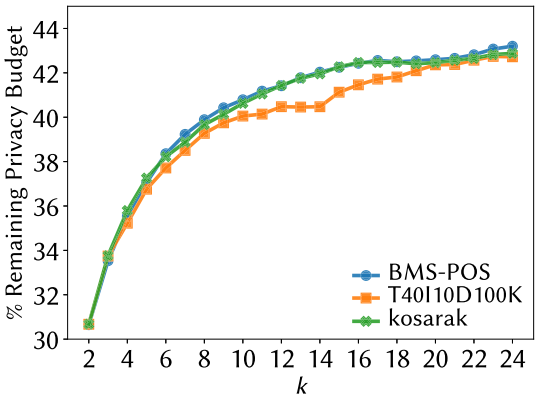


Fig. 6 Remaining privacy budget when Adaptive SVT with Gap is stopped after answering k queries using different datasets. Privacy budget $\epsilon = 0.7$

budget to determine that the query is likely above the noisy threshold. That is, it may produce an output in its top branch, where a lot of noise (hence less privacy budget) is used. If we stop Adaptive SVT with Gap after k returned queries, it may still have some privacy budget left over (in contrast to standard versions of sparse vector, which use up all of their privacy budget). This remaining privacy budget can then be used for other data analysis tasks. For all three datasets, Fig. 6 shows the percentage of privacy budget that is left over when Adaptive SVT with Gap is run with parameter k and stopped after k queries are returned. We see that roughly 40% of the privacy budget is left over, confirming that Adaptive SVT with Gap is able to save a significant amount of privacy budget.

9.3 Benefits of the hybrid algorithms

We next evaluate whether our hybrid algorithms combine the best properties of SVT (saving budget if few queries are over the threshold) and Noisy Top-K (selecting queries with higher values than SVT).

To evaluate the budget-saving properties, we set the threshold T to be the 12th largest query and let k vary from 2 to 24. This creates settings where fewer than k queries may be returned (i.e., when $k > 12$). The remaining privacy budget for different k is shown in Fig. 7. When $k > 12$, SVT and the hybrid algorithms use less privacy budget because they return fewer than k queries. However, Noisy Top-K uses the full budget because it returns k queries, even when we do not want the ones below the threshold. Hybrid Noisy Top-K saves more privacy budget than hybrid SVT because hybrid SVT spends a fixed amount of budget $\theta\epsilon$ on the threshold whereas Hybrid Noisy Top-K treats the threshold as a query

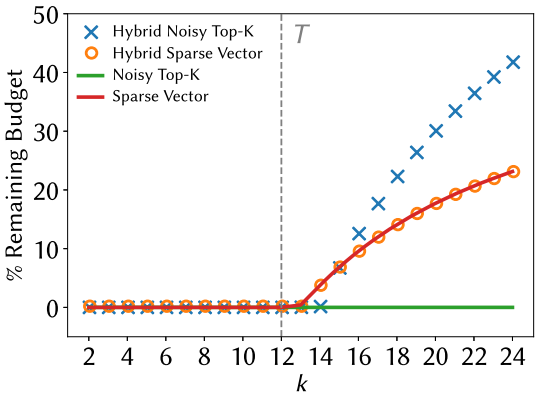


Fig. 7 Percentage of remaining privacy budget of hybrid algorithms, Noisy Top-K and SVT on the BMS-POS dataset. Results on Kosarak and T40I10D100K are similar. Privacy budget $\epsilon = 0.7$

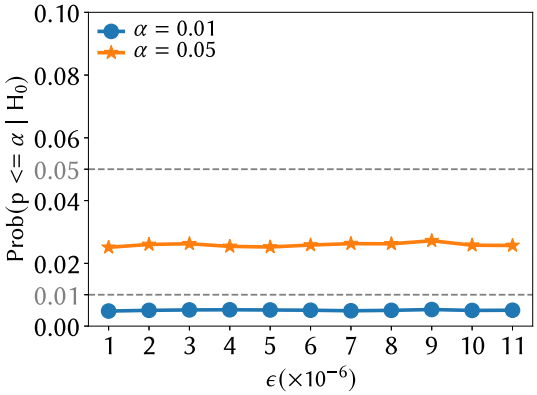


Fig. 9 The estimated probability of $p \leq \alpha$ when the output index from Exponential Mechanism with Gap is not optimal. Utility scores are sampled from BMS-POS

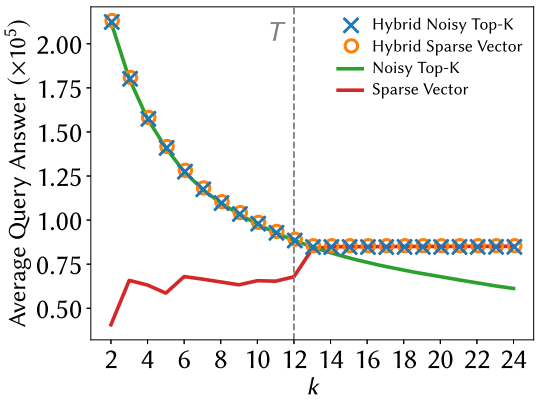


Fig. 8 Average query answers of the hybrid algorithms, Noisy Top-K and SVT on the BMS-POS dataset. Results on Kosarak and T40I10D100K are similar. Privacy budget $\epsilon = 0.7$

and only spends ϵ/k on it. SVT behaves similarly to hybrid SVT in terms of budget consumption.

Next, we compare how well the algorithms return queries whose answers are large. Using the same settings as before, we show how the average of the answers to the returned queries (as k varies) in Fig. 8.

Since the threshold is set at the value of the 12th largest query, when $k \leq 12$, the algorithms tend to return k queries. Here Noisy Top-K and the hybrid algorithms return much better queries than SVT. However, when $k > 12$, we are only interested in the queries that are larger than the threshold. Noisy Top-K has no ability to filter out the queries below the threshold and so the average query quality decreases. Meanwhile, SVT and our hybrid algorithms filter out the queries that are likely to be below the threshold, resulting in higher average quality. Thus we see that the hybrid algorithms indeed inherit the best properties of SVT and Noisy Top-K.

9.4 *p* Values from Exponential Mechanism with Gap

Algorithm 6 returns a selected query ω_s and a gap estimate g_s that we can use for hypothesis testing. Let H_0 be the null hypothesis that ω_s is not the query with the highest utility. Theorem 7 shows how to convert g_s into a p value and one would reject the null hypothesis if the p value is below a pre-specified significance level α (such as 0.01 or 0.05). As a simple experiment to verify the validity of this procedure, we simulate the utility scores of 100 queries by sampling 100 numbers from the datasets and we vary ϵ from 1×10^{-6} to 11×10^{-6} to ensure a decent chance of a non-optimal query being returned. We run the Exponential Mechanism with Gap for $n = 100{,}000$ times and record as c_1 the number of times the returned ω_s is not optimal (H_0 is true). Thus c_1/n is an estimate of $\mathbb{P}[H_0]$. Among the c_1 occurrences where H_0 is true, we record as c_2 the number of times Theorem 7 gives a p value $\leq \alpha$ (for $\alpha = 0.05$ and for $\alpha = 0.01$), causing the null hypothesis to be erroneously rejected. The quantity c_2/c_1 is an estimate of how frequently this happens. (This is called the Type I error and must be $\leq \alpha$ in order for the hypothesis testing framework to be considered valid.) As shown in Fig. 9, the errors of the hypothesis test using Theorem 7 are indeed less than the significance levels.

10 General randomness alignment and Proof of Lemma 1

In this section, we prove Lemma 1, which was used to establish the privacy properties of the algorithms we proposed. The proof of the lemma requires a more general theorem for working with randomness alignment functions. We explicitly list all of the conditions needed for the sake of reference. (Many prior works had incorrect proofs because they did not

have such a list to follow.) In the general setting, the method of randomness alignment requires the following steps.

1. For each pair of adjacent databases $D \sim D'$ and $\omega \in \Omega$, define a randomness alignment $\phi_{D,D'}$ or local alignment functions $\phi_{D,D',\omega} : \mathcal{H}_{D:\omega} \to \mathcal{H}_{D':\omega}$ (see notation in Table 2). In the case of local alignments this involves proving that if $M(D, H) = \omega$ then $M(D', \phi_{D,D',\omega}(H)) = \omega$.
2. Show that $\phi_{D,D'}$ (or all the $\phi_{D,D',\omega}$) is one-to-one (it does not need to be onto). That is, if we know D, D', ω and we are given the value $\phi_{D,D'}(H)$ (or $\phi_{D,D',\omega}(H)$), we can obtain the value H.
3. For each pair of adjacent databases $D \sim D'$, bound the *alignment cost* of $\phi_{D,D'}$ ($\phi_{D,D'}$ is either given or constructed by piecing together the local alignments). Bounding the alignment cost means the following: If f is the density (or probability mass) function of H, find a constant a such that $\frac{f(H)}{f(\phi_{D,D'}(H))} \leq a$ for all H (except a set of measure 0). In the case of local alignments, one can instead show the following. For all ω, and adjacent $D \sim D'$ the ratio $\frac{f(H)}{f(\phi_{D,D',\omega}(H))} \leq a$ for all H (except on a set of measure 0).
4. Bound the *change-of-variables cost* of $\phi_{D,D'}$ (only necessary when H is not discrete). One must show that the Jacobian of $\phi_{D,D'}$, defined as $J_{\phi_{D,D'}} = \frac{\partial \phi_{D,D'}}{\partial H}$, exists (i.e., $\phi_{D,D'}$ is differentiable) and is continuous except on a set of measure 0. Furthermore, for all pairs $D \sim D'$, show the quantity $\left|\det J_{\phi_{D,D'}}\right|$ is lower bounded by some constant $b > 0$. If $\phi_{D,D'}$ is constructed by piecing together local alignments $\phi_{D,D',\omega}$ then this is equivalent to showing the following (i) $\left|\det J_{\phi_{D,D',\omega}}\right|$ is lower bounded by some constant $b > 0$ for every $D \sim D'$ and ω, and (ii) for each $D \sim D'$, the set Ω can be partitioned into countably many disjoint measurable sets $\Omega = \bigcup_i \Omega_i$ such that whenever ω and ω^* are in the same partition, then $\phi_{D,D',\omega}$ and ϕ_{D,D',ω^*} are the same function. Note that this last condition (ii) is equivalent to requiring that the local alignments must be defined without using the axiom of choice (since non-measurable sets are not constructable otherwise) and for each $D \sim D'$, the number of distinct local alignments is countable. That is, the set $\{\phi_{D,D',\omega} \mid \omega \in \Omega\}$ is countable (i.e., for many choices of ω we get the same exact alignment function).

Theorem 8 *Let M be a randomized algorithm that terminates with probability 1 and suppose the number of random variables used by M can be determined from its output. If, for all pairs of adjacent databases $D \sim D'$, there exist randomness alignment functions $\phi_{D,D'}$ (or local alignment functions $\phi_{D,D',\omega}$ for all $\omega \in \Omega$ and $D \sim D'$) that satisfy conditions 1 though 4 above, then M satisfies $\ln(a/b)$-differential privacy.*

Proof We need to show that for all $D \sim D'$ and $E \subseteq \Omega$, $\mathbb{P}[\mathcal{H}_{D:E}] \leq (a/b)\mathbb{P}[\mathcal{H}_{D':E}]$.

First, we note that if we have a randomness alignment $\phi_{D,D'}$, we can define corresponding local alignment functions as follows $\phi_{D,D',\omega}(H) = \phi_{D,D'}(H)$. (In other words, they are all the same.) The conditions on local alignments are a superset of the conditions on randomness alignments, so for the rest of the proof we work with the $\phi_{D,D',\omega}$.

Let $\phi_1, \phi_2, \ldots$ be the distinct local alignment functions (there are countably many of them by Condition 4). Let $E_i = \{\omega \in E \mid \phi_{D,D',\omega} = \phi_i\}$. By Conditions 1 and 2 we have that for each $\omega \in E_i$, ϕ_i is one-to-one on $\mathcal{H}_{D:\omega}$ and $\phi_i(\mathcal{H}_{D:\omega}) \subseteq \mathcal{H}_{D':\omega}$. Note that $\mathcal{H}_{D:E_i} = \cup_{\omega \in E_i} \mathcal{H}_{D:\omega}$ and $\mathcal{H}_{D':E_i} = \cup_{\omega \in E_i} \mathcal{H}_{D':\omega}$. Furthermore, the sets $\mathcal{H}_{D:\omega}$ are pairwise disjoint for different ω and the sets $\mathcal{H}_{D':\omega}$ are pairwise disjoint for different ω. It follows that ϕ_i is one-to-one on $\mathcal{H}_{D:E_i}$ and $\phi_i(\mathcal{H}_{D:E_i}) \subseteq \mathcal{H}_{D':E_i}$. Thus for any $H' \in \phi_i(\mathcal{H}_{D:E_i})$ there exists $H \in \mathcal{H}_{D:E_i}$ such that $H = \phi_i^{-1}(H')$. By Conditions 3 and 4, we have $\frac{f(H)}{f(\phi_i(H))} = \frac{f(\phi_i^{-1}(H'))}{f(H')} \leq a$ for all $H \in \mathcal{H}_{D:E_i}$, and $\left|\det J_{\phi_i}\right| \geq b$ (except on a set of measure 0). Then the following is true:

$$
\begin{aligned}
&\mathbb{P}[\mathcal{H}_{D:E_i}] \\
&= \int_{\mathcal{H}_{D:E_i}} f(H)dH = \int_{\phi_i(\mathcal{H}_{D:E_i})} f(\phi_i^{-1}(H'))\frac{dH'}{\left|\det J_{\phi_i}\right|} \\
&\leq \int_{\phi_i(\mathcal{H}_{D:E_i})} af(H')\frac{1}{b}\mathrm{d}H' = \frac{a}{b}\int_{\phi_i(\mathcal{H}_{D:E_i})} f(H')\mathrm{d}H' \\
&\leq \frac{a}{b}\int_{\mathcal{H}_{D':E_i}} f(H')\mathrm{d}H' = \frac{a}{b}\mathbb{P}[\mathcal{H}_{D':E_i}].
\end{aligned}
$$

The second equation is the change of variables formula in calculus. The last inequality follows from the containment $\phi_i(\mathcal{H}_{D:E_i}) \subseteq \mathcal{H}_{D':E_i}$ and the fact that the density f is non-negative. In the case that H is discrete, simply replace the density f with a probability mass function, change the integral into a summation, ignore the Jacobian term and set $b = 1$. Finally, since $E = \cup_i E_i$ and $E_i \cap E_j = \emptyset$ for $i \neq j$, we conclude that

$$
\begin{aligned}
\mathbb{P}[\mathcal{H}_{D:E}] &= \sum_i \mathbb{P}[\mathcal{H}_{D:E_i}] \leq \frac{a}{b}\sum_i \mathbb{P}[\mathcal{H}_{D':E_i}] \\
&= \frac{a}{b}\mathbb{P}[\mathcal{H}_{D':E}].
\end{aligned}
$$

□

We now present the proof of Lemma 1.

Proof Let $\phi_{D,D',\omega}(H) = H' = (\eta'_1, \eta'_2, \ldots)$. By acyclicity there is some permutation π under which $\eta_{\pi(1)} = \eta'_{\pi(1)} - c$ where c is some constant depending on $D \sim D'$ and ω. Thus $\eta_{\pi(1)}$ is uniquely determined by H'. Now (as an induction hypothesis) assume $\eta_{\pi(1)}, \ldots, \eta_{\pi(j-1)}$ are uniquely determined by H' for some $j > 1$, then $\eta_{\pi(j)} = \eta'_{\pi(j)} - \psi^{(j)}_{D,D',\omega}(\eta_{\pi(1)}, \ldots, \eta_{\pi(j-1)})$, so $\eta_{\pi(j)}$ is also uniquely determined by H'. Thus by strong induction H is uniquely determined by H', i.e., $\phi_{D,D',\omega}$ is one-to-one. It is easy to see that with this ordering, $J_{\phi_{D,D',\omega}}$ is an upper triangular matrix with 1's on the diagonal. Since permuting variables does not change $\left|\det J_{\phi_{D,D',\omega}}\right|$, we have $\left|\det J_{\phi_{D,D',\omega}}\right| = 1$ since that is the determinant of upper triangular matrices. Furthermore, (recalling the definition of the cost of $\phi_{D,D',\omega}$) we have $\ln \frac{f(H)}{f(\phi_\omega(H))} = \sum_i \ln \frac{f_i(\eta_i)}{f_i(\eta'_i)} \le \sum_i \frac{|\eta_i - \eta'_i|}{\alpha_i} \le \epsilon$. The first inequality follows from Condition 3 of Lemma 1 and the second from Condition 4. □

11 Conclusions and future work

In this paper we introduced variations of SVT, Noisy Max, and exponential mechanism that provide additional noisy gap information for free (without affecting the privacy cost). We also presented applications of how to use the gap information. Future work includes applying this gap information in larger differentially private algorithms to increase the accuracy of privacy-preserving data analysis.

A Proofs

A.1 Proof of Theorem 4 (BLUE)

Proof Let $q_1, \ldots, q_k$ be the *true* answers to the k queries selected by Noisy Top-K with gap algorithm. Let α_i be the estimate of q_i using Laplace mechanism, and g_i be the estimate of the gap between q_i and q_{i+1} from Noisy Top-K with gap.

Recall that $\alpha_i = q_i + \xi_i$ and $g_i = q_i + \eta_i - q_{i+1} - \eta_{i+1}$ where ξ_i and η_i are independent Laplacian random variables. Assume without loss of generality that $\mathrm{Var}(\xi_i) = \sigma^2$ and $\mathrm{Var}(\eta_i) = \lambda\sigma^2$. Write in vector notation

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_k \end{bmatrix}, \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_k \end{bmatrix}, \boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_k \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix},$$

$$\boldsymbol{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_{k-1} \end{bmatrix},$$

then $\boldsymbol{\alpha} = \boldsymbol{q} + \boldsymbol{\xi}$ and $\boldsymbol{g} = N(\boldsymbol{q} + \boldsymbol{\eta})$ where

$$N = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}_{(k-1)\times k}.$$

Our goal is then to find the *best linear unbiased estimate* (BLUE) $\boldsymbol{\beta}$ of $\boldsymbol{q}$ in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{g}$. In other words, we need to find a $k \times k$ matrix X and a $k \times (k-1)$ matrix Y such that

$$\boldsymbol{\beta} = X\boldsymbol{\alpha} + Y\boldsymbol{g} \tag{6}$$

with $E(\|\boldsymbol{\beta} - \boldsymbol{q}\|^2)$ as small as possible. Unbiasedness implies that $\forall \boldsymbol{q}$, $E(\boldsymbol{\beta}) = X\boldsymbol{q} + YN\boldsymbol{q} = \boldsymbol{q}$. Therefore $X + YN = I_k$ and thus

$$X = I_k - YN. \tag{7}$$

Plugging this into (6), we have $\boldsymbol{\beta} = (I_k - YN)\boldsymbol{\alpha} + Y\boldsymbol{g} = \boldsymbol{\alpha} - Y(N\boldsymbol{\alpha} - \boldsymbol{g})$. Recall that $\boldsymbol{\alpha} = \boldsymbol{q} + \boldsymbol{\xi}$ and $\boldsymbol{g} = N(\boldsymbol{q} + \boldsymbol{\eta})$, we have $N\boldsymbol{\alpha} - \boldsymbol{g} = N(\boldsymbol{q} + \boldsymbol{\xi} - \boldsymbol{q} - \boldsymbol{\eta}) = N(\boldsymbol{\xi} - \boldsymbol{\eta})$. Thus

$$\boldsymbol{\beta} = \boldsymbol{\alpha} - YN(\boldsymbol{\xi} - \boldsymbol{\eta}). \tag{8}$$

Write $\boldsymbol{\theta} = N(\boldsymbol{\xi} - \boldsymbol{\eta})$, then we have $\boldsymbol{\beta} - \boldsymbol{q} = \boldsymbol{\alpha} - \boldsymbol{q} - Y\boldsymbol{\theta} = \boldsymbol{\xi} - Y\boldsymbol{\theta}$. Therefore, finding the BLUE is equivalent to solving the optimization problem $Y = \arg\min \Phi$ where

$$\begin{aligned} \Phi &= E(\|\boldsymbol{\xi} - Y\boldsymbol{\theta}\|^2) = E((\boldsymbol{\xi} - Y\boldsymbol{\theta})^T(\boldsymbol{\xi} - Y\boldsymbol{\theta})) \\ &= E(\boldsymbol{\xi}^T\boldsymbol{\xi} - \boldsymbol{\xi}^T Y\boldsymbol{\theta} - \boldsymbol{\theta}^T Y^T\boldsymbol{\xi} + \boldsymbol{\theta}^T Y^T Y\boldsymbol{\theta}) \end{aligned}$$

Taking the partial derivatives of Φ w.r.t Y, we have

$$\frac{\partial \Phi}{\partial Y} = E(\mathbf{0} - \boldsymbol{\xi}\boldsymbol{\theta}^T - \boldsymbol{\xi}\boldsymbol{\theta}^T + Y(\boldsymbol{\theta}\boldsymbol{\theta}^T + \boldsymbol{\theta}\boldsymbol{\theta}^T))$$

By setting $\frac{\partial \Phi}{\partial Y} = 0$ we have $YE(\boldsymbol{\theta}\boldsymbol{\theta}^T) = E(\boldsymbol{\xi}\boldsymbol{\theta}^T)$ thus

$$Y = E(\boldsymbol{\xi}\boldsymbol{\theta}^T)E(\boldsymbol{\theta}\boldsymbol{\theta}^T)^{-1}. \tag{9}$$

Recall that $(\boldsymbol{\xi}\boldsymbol{\theta}^T)_{ij} = \xi_i(\xi_j - \xi_{j+1} - \eta_j + \eta_{j+1})$, we have

$$E(\boldsymbol{\xi}\boldsymbol{\theta}^T)_{ij} = \begin{cases} E(\xi_i^2) = \mathrm{Var}(\xi_i) = \sigma^2 & i = j \\ -E(\xi_i^2) = -\mathrm{Var}(\xi_i) = -\sigma^2 & i = j+1 \\ 0 & \text{otherwise} \end{cases}$$

Hence

$$E(\boldsymbol{\xi}\boldsymbol{\theta}^T) = \sigma^2 \begin{bmatrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \\ & & -1 \end{bmatrix}_{k\times(k-1)} = \sigma^2 N^T.$$

Similarly, we have

$$\begin{aligned}(\boldsymbol{\theta}\boldsymbol{\theta}^T)_{ij} &= (\xi_i - \xi_{i+1} - \eta_i + \eta_{i+1})(\xi_j - \xi_{j+1} - \eta_j + \eta_{j+1}) \\ &= \xi_i\xi_j + \xi_{i+1}\xi_{j+1} - \xi_i\xi_{j+1} - \xi_{i+1}\xi_j \\ &\quad + \eta_i\eta_j + \eta_{i+1}\eta_{j+1} - \eta_i\eta_{j+1} - \eta_{i+1}\eta_j \\ &\quad - (\xi_i - \xi_{i+1})(\eta_j - \eta_{j+1}) \\ &\quad - (\eta_i - \eta_{i+1})(\xi_j - \xi_{j+1})\end{aligned}$$

Thus

$$E(\boldsymbol{\theta}\boldsymbol{\theta}^T)_{ij} = \begin{cases} E(\xi_i^2 + \xi_{i+1}^2 + \eta_i^2 + \eta_{i+1}^2) = 2(1+\lambda)\sigma^2 & i = j \\ E(-\xi_i^2 - \eta_i^2) = -(1+\lambda)\sigma^2 & i = j+1 \\ E(-\xi_j^2 - \eta_j^2) = -(1+\lambda)\sigma^2 & i = j-1 \\ 0 & \text{otherwise} \end{cases}$$

Hence

$$E(\boldsymbol{\theta}\boldsymbol{\theta}^T) = (1+\lambda)\sigma^2 \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}_{(k-1)\times(k-1)}.$$

It can be directly computed that $E(\boldsymbol{\theta}\boldsymbol{\theta}^T)^{-1}$ is a symmetric matrix whose lower triangular part is

$$\frac{1}{k(1+\lambda)\sigma^2} \begin{bmatrix} (k-1)\cdot 1 & \cdots & \cdots & \cdots & \cdots \\ (k-2)\cdot 1 & (k-2)\cdot 2 & \cdots & \cdots & \cdots \\ (k-3)\cdot 1 & (k-3)\cdot 2 & (k-3)\cdot 3 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1\cdot 1 & 1\cdot 2 & 1\cdot 3 & \cdots & 1\cdot(k-1), \end{bmatrix}$$

i.e., $E(\boldsymbol{\theta}\boldsymbol{\theta}^T)^{-1}_{ij} = E(\boldsymbol{\theta}\boldsymbol{\theta}^T)^{-1}_{ji} = \frac{1}{k(1+\lambda)\sigma^2} \cdot (k-i)\cdot j$ for all $1 \le i \le j \le k-1$. Therefore, $Y = E(\boldsymbol{\xi}\boldsymbol{\theta}^T)E(\boldsymbol{\theta}\boldsymbol{\theta}^T)^{-1} =$

$$\frac{1}{k(1+\lambda)} \left(\begin{bmatrix} k-1 & k-2 & \cdots & 1 \\ k-1 & k-2 & \cdots & 1 \\ k-1 & k-2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ k-1 & k-2 & \cdots & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ k & 0 & \cdots & 0 \\ k & k & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ k & k & \cdots & k \end{bmatrix} \right)_{k\times(k-1)}$$

Hence

$$X = I_k - YN = \frac{1}{k(1+\lambda)} \begin{bmatrix} 1+k\lambda & 1 & \cdots & 1 \\ 1 & 1+k\lambda & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1+k\lambda \end{bmatrix}_{k\times k}.$$

□

A.2 Proof of Corollary 1

Recall that $\alpha_i = q_i + \xi_i$ and $g_i = q_i + \eta_i - q_{i+1} - \eta_{i+1}$ where ξ_i and η_i are independent Laplacian random variables. Assume without loss of generality that $\mathrm{Var}(\xi_i) = \sigma^2$ and $\mathrm{Var}(\eta_i) = \lambda\sigma^2$ as before. From the matrices X and Y in Theorem 4 we have that $\beta_i = \frac{x_i + y_i}{k(1+\lambda)}$ where

$$\begin{aligned} x_i &= \alpha_1 + \cdots + (1+k\lambda)\alpha_i + \cdots + \alpha_k \\ &= (q_1 + \xi_1) + \cdots + (1+k\lambda)(q_i + \xi_i) + \cdots + (q_k + \xi_k) \end{aligned}$$

and

$$\begin{aligned} y_i &= -g_1 - 2g_2 - \cdots - (i-1)g_{i-1} \\ &\quad + (k-i)g_i + \ldots + 2g_{k-2} + g_{k-1} \\ &= -(q_1 + \eta_1) - (q_2 + \eta_2) - \cdots - (q_{i-1} + \eta_{i-1}) \\ &\quad + (k-1)(q_i + \eta_i) - (q_{i+1} + \eta_{i+1}) - \cdots - (q_k + \eta_k). \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{Var}(x_i) &= \sigma^2 + \cdots + (1+k\lambda)^2\sigma^2 + \cdots + \sigma^2 \\ &= (k^2\lambda^2 + 2k\lambda + k)\sigma^2 \\ \mathrm{Var}(y_i) &= \lambda\sigma^2 + \cdots + (k-1)^2\lambda\sigma^2 + \cdots + \lambda\sigma^2 \\ &= (k^2 - k)\lambda\sigma^2 \end{aligned}$$

and thus $\mathrm{Var}(\beta_i) = \frac{\mathrm{Var}(x_i) + \mathrm{Var}(y_i)}{k^2(1+\lambda)^2} = \frac{1+k\lambda}{k+k\lambda}\sigma^2$. Recall that $\mathrm{Var}(\alpha_i) = \mathrm{Var}(\xi_i) = \sigma^2$, we have $\frac{\mathrm{Var}(\beta_i)}{\mathrm{Var}(\alpha_i)} = \frac{1+k\lambda}{k+k\lambda}$.

A.3 Proof of Lemma 3

The density function of $\eta_i - \eta$ is $f_{\eta_i - \eta}(z) = \int_{-\infty}^{\infty} f_{\eta_i}(x) f_{\eta}(x-z)\,\mathrm{d}x = \frac{\epsilon_0\epsilon_*}{4}\int_{-\infty}^{\infty} e^{-\epsilon_*|x|}e^{-\epsilon_0|x-z|}\,\mathrm{d}x$. First consider the case $\epsilon_0 \neq \epsilon_*$. When $z \ge 0$, we have

$$\begin{aligned} &f_{\eta_i - \eta}(z) \\ &= \frac{\epsilon_0\epsilon_*}{4}\int_{-\infty}^{\infty} e^{-\epsilon_*|x|}e^{-\epsilon_0|x-z|}\,\mathrm{d}x \\ &= \frac{\epsilon_0\epsilon_*}{4}\Big(\int_{-\infty}^{0} e^{\epsilon_* x}e^{\epsilon_0(x-z)}\,\mathrm{d}x \\ &\quad + \int_0^z e^{-\epsilon_* x}e^{\epsilon_0(x-z)}\,\mathrm{d}x + \int_z^{\infty} e^{-\epsilon_* x}e^{-\epsilon_0(x-z)}\,\mathrm{d}x\Big) \\ &= \frac{\epsilon_0\epsilon_*}{4}\Big(\frac{e^{-\epsilon_0 z}}{\epsilon_0 + \epsilon_*} + \frac{e^{-\epsilon_* z} - e^{-\epsilon_0 z}}{\epsilon_0 - \epsilon_*} + \frac{e^{-\epsilon_* z}}{\epsilon_0 + \epsilon_*}\Big) \\ &= \frac{\epsilon_0\epsilon_*(\epsilon_0 e^{-\epsilon_* z} - \epsilon_* e^{-\epsilon_0 z})}{2(\epsilon_0^2 - \epsilon_*^2)} \end{aligned}$$

Thus by symmetry we have that for all $z \in \mathbb{R}$, $f_{\eta_i-\eta}(z) = \frac{\epsilon_0\epsilon_*(\epsilon_0 e^{-\epsilon_*|z|} - \epsilon_* e^{-\epsilon_0|z|})}{2(\epsilon_0^2-\epsilon_*^2)}$, and

$$\mathbb{P}(\eta_i - \eta \geq -t) = \int_{-t}^{\infty} f_{\eta_i-\eta}(z)\,\mathrm{d}z = \int_{-t}^{0} f_{\eta_i-\eta}(z)\,\mathrm{d}z + \frac{1}{2}$$
$$= 1 - \frac{\epsilon_0^2 e^{-\epsilon_* t} - \epsilon_*^2 e^{-\epsilon_0 t}}{2(\epsilon_0^2 - \epsilon_*^2)}.$$

Now if $\epsilon_0 = \epsilon_*$, by similar computations we have $f_{\eta_i-\eta}(z) = (\frac{\epsilon_0}{4} + \frac{\epsilon_0^2|z|}{4})e^{-\epsilon_0|z|}$ and $\mathbb{P}(\eta_i - \eta \geq -t) = 1 - (\frac{2+\epsilon_0 t}{4})e^{-\epsilon_0 t}$.

A.4 Proofs in Sect. 8 (Exp. Mech. with Gap)

A well-known, but inefficient, folklore algorithm for the exponential mechanism is based on the Gumbel-Max trick [25,35]: Given numbers $\mu_1, \ldots, \mu_n$, add independent Gumbel(0) noise to each and select the *index* of the largest noisy value. This is the same as sampling the ith item with probability proportional to e^{μ_i}. Let $\mathrm{Cat}(\mu_1, \ldots, \mu_n)$ denote the categorical distribution that returns item ω_i with probability $\frac{\exp(\mu_i)}{\sum_{j=1}^n \exp(\mu_j)}$. The Gumbel-Max theorem provides distributions for the identity of the noisy maximum and the value of the noisy maximum:

Theorem 9 (The Gumbel-Max Trick [25,35]) *Let $G_i, \ldots, G_n$ be i.i.d.* Gumbel(0) *random variables and let $\mu_1, \ldots, \mu_n$ be real numbers. Define $X_i = G_i + \mu_i$. Then*

1. *The distribution of* $\arg\max_i(X_1, \ldots, X_n)$ *is the same as* $\mathrm{Cat}(\mu_1, \ldots, \mu_n)$.
2. *The distribution of* $\max_i(X_1, \ldots, X_n)$ *is the same as the* $\mathrm{Gumbel}(\ln \sum_{i=1}^n \exp(\mu_i))$ *distribution.*

Using the Gumbel-Max trick, one can propose an Exponential Mechanism with Gap by replacing Laplace or exponential noise in Noisy Max with Gap with the Gumbel distribution as shown in Algorithm 7. (Boxed items represent gap information.) We first prove the correctness of this algorithm and then show how to replace the Gumbel-max trick with any efficient black box algorithm for the exponential mechanism.

Algorithm 7: Exponential Mechanism w. Gap

input: μ: utility function with sensitivity Δ_μ
D: database, ϵ: privacy budget

function `GapExpMech` (D, μ, ϵ):
  foreach $i \in \{1, \cdots, n\}$ **do**
    $x_i \leftarrow \epsilon\mu(D, \omega_i)/2\Delta_\mu + \mathrm{Gumbel}(0)$
  $s, \boxed{t} \leftarrow \arg\max_2(x_1, \ldots, x_n)$
  return $\omega_s, \boxed{x_s - x_t}$

We first need the following results.

Lemma 9 *Let $\epsilon > 0$. Let $\mu : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ be a utility function of sensitivity Δ_μ. Define $\nu : \mathcal{D} \to \mathbb{R}$ and its sensitivity Δ_ν as*

$$\nu(D) = \ln \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D,\omega)}{2\Delta_\mu}}, \quad \Delta_\nu = \max_{D \sim D'} \left|\nu(D) - \nu(D')\right|.$$

Then Δ_ν, the sensitivity of ν, is at most $\frac{\epsilon}{2}$.

Proof of Lemma 9 From the definition of ν we have

$$\left|\nu(D) - \nu(D')\right| = \left|\ln \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D,\omega)}{2\Delta_\mu}} - \ln \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D',\omega)}{2\Delta_\mu}}\right|$$
$$= \left|\ln \left(\sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D,\omega)}{2\Delta_\mu}}\right) \Big/ \left(\sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D',\omega)}{2\Delta_\mu}}\right)\right|$$

By definition of sensitivity, we have

$$\mu(D', \omega) - \Delta_\mu \leq \mu(D, \omega) \leq \mu(D', \omega) + \Delta_\mu, \text{ and therefore}$$

$$e^{-\frac{\epsilon}{2}} \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D',\omega)}{2\Delta_\mu}} \leq \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D,\omega)}{2\Delta_\mu}} \leq e^{\frac{\epsilon}{2}} \sum_{\omega \in \mathcal{R}} e^{\frac{\epsilon\mu(D',\omega)}{2\Delta_\mu}}$$

Thus $\left|\nu(D) - \nu(D')\right| \leq \frac{\epsilon}{2}$, and hence $\Delta_\nu \leq \frac{\epsilon}{2}$. □

Lemma 10 *Let $f(x;\theta) = \frac{e^{-(x-\theta)}}{(1+e^{-(x-\theta)})^2}$ be the density of the logistic distribution, then $\left|\ln \frac{f(x;\theta)}{f(x;\theta')}\right| \leq \left|\theta - \theta'\right|$.*

Proof of Lemma 10 Note that $\left|\ln \frac{f(x;\theta)}{f(x;\theta')}\right| = \left|\ln \frac{f(x;\theta')}{f(x;\theta)}\right|$ so without loss of generality, we can assume that $\theta \geq \theta'$ (i.e., the location parameter in the numerator is $\geq$ the parameter in the denominator). From the formula of f we have $\frac{f(x;\theta)}{f(x;\theta')} = e^{\theta-\theta'} \cdot \left(\frac{1+e^{-x}e^{\theta'}}{1+e^{-x}e^{\theta}}\right)^2$. Clearly $e^{\theta} \geq e^{\theta'} \implies \frac{1+e^{-x}e^{\theta'}}{1+e^{-x}e^{\theta}} \leq 1$. Also,

$$\frac{1+e^{-x}e^{\theta'}}{1+e^{-x}e^{\theta}} = \frac{e^{\theta'-\theta}(e^{\theta-\theta'}+e^{-x}e^{\theta})}{1+e^{-x}e^{\theta}} \geq \frac{e^{\theta'-\theta}(1+e^{-x}e^{\theta})}{1+e^{-x}e^{\theta}} = e^{\theta'-\theta}.$$

Therefore, $e^{\theta'-\theta} = e^{\theta-\theta'} \cdot (e^{\theta'-\theta})^2 \leq \frac{f(x;\theta)}{f(x;\theta')} \leq e^{\theta-\theta'}$. Thus $\left|\ln \frac{f(x;\theta)}{f(x;\theta')}\right| \leq \left|\theta - \theta'\right|$. □

Theorem 10 *Algorithm 7 satisfies ϵ-differential privacy. Its output distribution is equivalent to selecting ω_s with probability proportional to* $\exp\left(\frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu}\right)$ *and then independently sampling the gap from the logistic distribution (conditional on only sampling nonnegative values) with location parameter $\theta = \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \ln \sum_{j \neq s} \exp(\frac{\epsilon\mu(D,\omega_j)}{2\Delta_\mu})$.*

Proof of Theorem 10 For $\omega_i \in \mathcal{R}$, let $\mu_i = \frac{\epsilon\mu(D,\omega_i)}{2\Delta_\mu}$ and $\mu_i' = \frac{\epsilon\mu(D',\omega_i)}{2\Delta_\mu}$. Let $X_i \sim \text{Gumbel}(\mu_i)$ and $X_i' \sim \text{Gumbel}(\mu_i')$.

We consider the probability of outputting the selected ω_s with gap $\gamma \geq 0$ when D is the input database:

$$
\begin{aligned}
&P(\omega_s \text{ is chosen with gap} \geq \gamma \mid D) \\
&= \int_{\mathbb{R}} P(X_s = z+\gamma) \prod_{i \neq s} P(X_i \leq z)\, \mathrm{d}z \\
&= \int_{\mathbb{R}} \exp(-(z+\gamma-\mu_s) - e^{-(z+\gamma-\mu_s)}) \prod_{i \neq s} e^{-e^{-(z-\mu_i)}}\, \mathrm{d}z \\
&= \int_{\mathbb{R}} e^{\mu_s-\gamma} \exp(-z - e^{\mu_s-\gamma} e^{-z}) \prod_{i \neq s} \exp(-e^{\mu_i} e^{-z})\, \mathrm{d}z
\end{aligned}
$$

(let $\mu^* = \ln(\sum_{i \neq s} e^{\mu_i})$ and $\theta = \mu_s - \mu^*$)

$$
\begin{aligned}
&= \int_{\mathbb{R}} e^{\mu_s-\gamma} \exp(-z - e^{\mu_s-\gamma} e^{-z}) \exp(-e^{\mu^*} e^{-z})\, \mathrm{d}z \\
&= \int_{\mathbb{R}} e^{\mu_s-\gamma} \exp(-z - (e^{\mu_s-\gamma} + e^{\mu^*}) e^{-z})\, \mathrm{d}z \\
&= \frac{e^{\mu_s-\gamma}}{e^{\mu_s-\gamma} + e^{\mu^*}} \exp(-(e^{\mu_s-\gamma} + e^{\mu^*}) e^{-z}) \Big|_{-\infty}^{+\infty} \\
&= \frac{e^{\mu_s-\gamma}}{e^{\mu_s-\gamma} + e^{\mu^*}} = \frac{1}{1 + e^{-(\mu_s-\gamma-\mu^*)}} = \frac{1}{1+e^{-(\theta-\gamma)}}
\end{aligned}
$$

and so

$$
\begin{aligned}
&P(\omega_s \text{ is chosen with gap} \in [0,\gamma] \mid D) \\
&\quad = P(\omega_s \text{ is chosen} \mid D) \\
&\qquad - P(\omega_s \text{ is chosen with gap} \geq \gamma \mid D) \\
&\quad = \frac{e^{\mu_s}}{e^{\mu_s} + e^{\mu^*}} - \frac{1}{1 + e^{-(\mu_s-\gamma-\mu^*)}} = \frac{1}{1+e^{-\theta}} \\
&\qquad - \frac{1}{1+e^{-(\theta-\gamma)}}
\end{aligned}
$$

Taking the derivative with respect to γ, we get the density $f(\omega_s, \gamma \mid D)$ of ω_s being chosen with gap equal to γ:

$$
\begin{aligned}
f(\omega_s, \gamma \mid D) &= \frac{\mathrm{d}}{\mathrm{d}\gamma} \left(\frac{1}{1+e^{-\theta}} - \frac{1}{1+e^{-(\theta-\gamma)}} \right) \\
&= \frac{e^{-(\gamma-\theta)}}{(e^{-(\gamma-\theta)}+1)^2} \mathbf{1}_{[\gamma \geq 0]} \qquad (10) \\
&= \frac{e^{\mu_s}}{e^{\mu_s} + e^{\mu^*}} \left(\frac{e^{-(\gamma-\theta)}}{(e^{-(\gamma-\theta)}+1)^2} \mathbf{1}_{[\gamma \geq 0]} \right) \Big/ \frac{e^{\mu_s}}{e^{\mu_s} + e^{\mu^*}} \\
&= \frac{e^{\mu_s}}{e^{\mu_s} + e^{\mu^*}} \left(\frac{e^{-(\gamma-\theta)}}{(e^{-(\gamma-\theta)}+1)^2} \mathbf{1}_{[\gamma \geq 0]} \right) \Big/ \frac{1}{1+e^{-\theta}} \qquad (11)
\end{aligned}
$$

Now, in Eq. 11, the term $\frac{e^{\mu_s}}{e^{\mu_s}+e^{\mu^*}} = \frac{e^{\mu_s}}{e^{\mu_s}+\sum_{i \neq s} e^{\mu_i}} = \frac{e^{\mu_s}}{\sum_i e^{\mu_i}}$ is the probability of selecting ω_s.

The term $\frac{e^{-(\gamma-\theta)}}{(e^{-(\gamma-\theta)}+1)^2} \mathbf{1}_{[\gamma \geq 0]}$ is the density of the event that a logistic random variable with location θ has value γ and is nonnegative.

Finally, the term $\frac{1}{1+e^{-\theta}}$ is the probability that a logistic random variable with location θ is nonnegative.

Thus $\left(\frac{e^{-(\gamma-\theta)}}{(e^{-(\gamma-\theta)}+1)^2} \mathbf{1}_{[\gamma \geq 0]} \right) \Big/ \frac{1}{1+e^{-\theta}}$ is the probability of a logistic random variable having value γ conditioned on it being nonnegative.

Therefore Eq. 11 is the probability of selecting ω_s and independently sampling a nonnegative value γ from the conditional logistic distribution location parameter $\theta = \mu_s - \mu^*$ (i.e., conditional on it only returning nonnegative values).

Now, recall that $\mu_i = \frac{\epsilon\mu(D,i)}{2\Delta_\mu}$, we apply Lemmas 10 and 9 with the help of Eq. 10 to finish the proof:

$$
\begin{aligned}
\left| \ln \frac{f(\omega_s, \gamma \mid D)}{f(\omega_s, \gamma \mid D')} \right| &\leq \left| (\mu_s - \mu^*) - (\mu_s' - \mu^{*\prime}) \right| \\
&\leq |\mu_s - \mu_s'| + |\ln \sum_{i \neq s} e^{\mu_i} - \ln \sum_{i \neq s} e^{\mu_i'}| \\
&\leq \epsilon/2 + \epsilon/2 = \epsilon.
\end{aligned}
$$

□

Proof of Theorem 6 The first part follows directly from Theorem 10. Also, from the proof of Theorem 10 the gap g_s has density $f(x;\theta) = \left(\frac{e^{-(x-\theta)}}{(e^{-(x-\theta)}+1)^2} \mathbf{1}_{[x \geq 0]} \right) \Big/ \frac{1}{1+e^{-\theta}}$. Since

$$
\begin{aligned}
&\int_0^t \frac{e^{-x+\theta}}{(e^{-x+\theta}+1)^2} \cdot x\, \mathrm{d}x = \int_0^t \frac{e^{x-\theta}}{(1+e^{x-\theta})^2} \cdot x\, \mathrm{d}x \\
&= \int_0^t x \cdot \left(\frac{-1}{1+e^{x-\theta}} \right)' \mathrm{d}x = \frac{-x}{1+e^{x-\theta}} \Big|_0^t \\
&\quad + \int_0^t \frac{1}{1+e^{x-\theta}}\, \mathrm{d}x \\
&= \frac{-t}{1+e^{t-\theta}} + (x - \ln(1+e^{x-\theta})) \Big|_0^t \\
&= \frac{-t}{1+e^{t-\theta}} + t - \ln(1+e^{t-\theta}) + \ln(1+e^{-\theta}) \\
&= \frac{-t}{1+e^{t-\theta}} + \ln \frac{e^t}{1+e^{t-\theta}} + \ln(1+e^{-\theta})
\end{aligned}
$$

We have

$$
\begin{aligned}
&\int_0^\infty \frac{e^{-x+\theta}}{(e^{-x+\theta}+1)^2} \cdot x\, \mathrm{d}x = \lim_{t \to \infty} \int_0^t \frac{e^{-x+\theta}}{(e^{-x+\theta}+1)^2} \cdot x\, \mathrm{d}x \\
&= \lim_{t \to \infty} \left(\frac{-t}{1+e^{t-\theta}} + \ln \frac{e^t}{1+e^{t-\theta}} + \ln(1+e^{-\theta}) \right) \\
&= 0 + \ln(e^\theta) + \ln(1+e^{-\theta}) = \ln(1+e^\theta)
\end{aligned}
$$

Hence $\mathbb{E}(g_s) = (1 + e^{-\theta}) \ln(1 + e^{\theta})$. □

Proof of Theorem 7 Assume H_0 is true, i.e., there exists a $t \neq s$ such that $\mu(D, \omega_s) < \mu(D, \omega_t)$. Then

$$\theta = \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \ln \sum_{j \neq s} \exp(\frac{\epsilon\mu(D,\omega_j)}{2\Delta_\mu})$$
$$\leq \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \ln \exp \frac{\epsilon\mu(D,\omega_t)}{2\Delta_\mu} = \frac{\epsilon\mu(D,\omega_s)}{2\Delta_\mu} - \frac{\epsilon\mu(D,\omega_t)}{2\Delta_\mu} < 0$$

Using be the density of the gap from above, we have

$$\mathbb{P}[x \geq \gamma \mid H_0] = (1 + e^{-\theta}) \int_\gamma^\infty \frac{e^{-x+\theta}}{(1 + e^{-x+\theta})^2} \, \mathrm{d}x$$
$$= (1 + e^{-\theta}) \int_\gamma^\infty \frac{e^{x-\theta}}{(1 + e^{x-\theta})^2} \, \mathrm{d}x$$
$$= (1 + e^{-\theta}) \cdot \left(\frac{-1}{1 + e^{x-\theta}} \Big|_\gamma^\infty \right)$$
$$= \frac{1 + e^{-\theta}}{1 + e^{\gamma-\theta}} = \frac{e^{\theta} + 1}{e^{\theta} + e^{\gamma}} < \frac{2}{1 + e^{\gamma}}$$

because $\frac{e^{\theta}+1}{e^{\theta}+e^{\gamma}}$ is an increasing function of θ and $\theta < 0$. □

Acknowledgements This work was supported by NSF Awards CNS-1702760 and CNS-1931686.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: CCS (2016)
2. Abowd, J.M.: The us census bureau adopts differential privacy. In: KDD (2018)
3. Albarghouthi, A., Hsu, J.: Synthesizing coupling proofs of differential privacy. In: POPL (2017)
4. Barthe, G., Gaboardi, M., Gregoire, B., Hsu, J., Strub, P.Y.: Proving differential privacy via probabilistic couplings. In: LICS (2016)
5. Beimel, A., Nissim, K., Stemmer, U.: Private learning and sanitization: pure vs. approximate differential privacy. Theory Comput. **12**(1), 1–61 (2016)
6. Bhaskar, R., Laxman, S., Smith, A., Thakurta, A.: Discovering frequent patterns in sensitive data. In: KDD (2010)
7. Bittau, A., Erlingsson, U., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: Prochlo: strong privacy for analytics in the crowd. In: SOSP (2017)
8. Bun, M., Steinke, T.: Concentrated differential privacy: simplifications, extensions, and lower bounds. In: TCC (2016)
9. Bureau, U.S.C.: On the map: longitudinal employer-household dynamics. https://lehd.ces.census.gov/applications/help/onthemap.html#!confidentiality_protection
10. Chaudhuri, K., Hsu, D., Song, S.: The large margin mechanism for differentially private maximization. In: NIPS (2014)
11. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. J. Mach. Learn. Res. **12**(Mar), 1069–1109 (2011)
12. Chen, Y., Machanavajjhala, A., Reiter, J.P., Barrientos, A.F.: Differentially private regression diagnostics. In: ICDM (2016)
13. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting telemetry data privately. In: NIPS (2017)
14. Ding, Z., Wang, Y., Zhang, D., Kifer, D.: Free gap information from the differentially private sparse vector and noisy max mechanisms. In: PVLDB (2019)
15. Dwork, C.: Differential privacy. In: ICALP (2006)
16. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 486–503. Springer (2006)
17. Dwork, C., Lei, J.: Differential privacy and robust statistics. In: STOC (2009)
18. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. Springer (2006)
19. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
20. Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: from local to central differential privacy via anonymity. In: SODA (2019)
21. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: CCS (2014)
22. Fanaeepour, M., Rubinstein, B.I.P.: Histogramming privately ever after: Differentially-private data-dependent error bound optimisation. In: ICDE (2018)
23. Geng, Q., Viswanath, P.: The optimal mechanism in differential privacy. In: ISIT (2014)
24. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms. In: STOC. pp. 351–360 (2009)
25. Gumbel, E.: Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures. Applied Mathematics Series, U.S. Government Printing Office, Washington (1954)
26. Haney, S., Machanavajjhala, A., Abowd, J.M., Graham, M., Kutzbach, M., Vilhuber, L.: Utility cost of formal privacy for releasing national employer–employee statistics. In: SIGMOD (2017)
27. Hardt, M., Ligett, K., McSherry, F.: A simple and practical algorithm for differentially private data release. In: NIPS (2012)
28. Johnson, N., Near, J.P., Song, D.: Towards practical differential privacy for SQL queries. In: PVLDB (2018)
29. Kotsogiannis, I., Machanavajjhala, A., Hay, M., Miklau, G.: Pythia: data dependent differentially private algorithm selection. In: SIGMOD (2017)
30. Lehmann, E., Casella, G.: Theory of Point Estimation. Springer, Berlin (1998)
31. Ligett, K., Neel, S., Roth, A., Waggoner, B., Wu, S.Z.: Accuracy first: selecting a differential privacy level for accuracy constrained ERM. In: NIPS (2017)
32. Liu, J., Talwar, K.: Private selection from private candidates (2018). arXiv preprint arXiv:1811.07971
33. Lyu, M., Su, D., Li, N.: Understanding the sparse vector technique for differential privacy. In: PVLDB (2017)
34. Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., Vilhuber, L.: Privacy: from theory to practice on the map. In: ICDE (2008)
35. Maddison, C.J., Tarlow, D., Minka, T.: A∗ sampling. In: NIPS (2014)
36. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS (2007)
37. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: SIGMOD (2009)
38. Mironov, I.: Rényi differential privacy. In: 30th IEEE Computer Security Foundations Symposium. CSF (2017)
39. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)

40. Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Úlfar Erlingsson: scalable private learning with pate. In: ICLR (2018)
41. Raskhodnikova, S., Smith, A.D.: Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In: FOCS (2016)
42. Tang, J., Korolova, A., Bai, X., Wang, X., Wang, X.: Privacy loss in apple's implementation of differential privacy. In: 3rd Workshop on the Theory and Practice of Differential Privacy at CCS (2017)
43. Team, A.D.P., Team: Learning with privacy at scale. Appl. Mach. Learn. J. **1**(8), 1–25 (2017)
44. Thakurta, A.G., Smith, A.: Differentially private feature selection via stability arguments, and the robustness of the lasso. In: COLT (2013)
45. Wang, Y., Ding, Z., Wang, G., Kifer, D., Zhang, D.: Proving differential privacy with shadow execution. In: PLDI (2019)
46. Zhang, D., Kifer, D.: Lightdp: towards automating differential privacy proofs. In: POPL (2017)
47. Zhang, D., McKenna, R., Kotsogiannis, I., Hay, M., Machanavajjhala, A., Miklau, G.: Ektelo: A framework for defining differentially-private computations. In: SIGMOD (2018)