Efficient Mean Estimation with Pure Differential Privacy via a Sum-of-Squares Exponential Mechanism

Samuel B. Hopkins
UC Berkeley and MIT
Cambridge, MA, USA
samhop@mit.edu

Gautam Kamath
Cheriton School of Computer Science,
University of Waterloo
Waterloo, ON, Canada
g@csail.mit.edu

Mahbod Majid
Cheriton School of Computer Science,
University of Waterloo
Waterloo, ON, Canada
m2majid@uwaterloo.ca

ABSTRACT

We give the first polynomial-time algorithm to estimate the mean of a d-variate probability distribution with bounded covariance from $\tilde{O}(d)$ independent samples subject to *pure* differential privacy. Prior algorithms for this problem either incur exponential running time, require $\Omega(d^{1.5})$ samples, or satisfy only the weaker *concentrated* or *approximate* differential privacy conditions. In particular, all prior polynomial-time algorithms require $d^{1+\Omega(1)}$ samples to guarantee small privacy loss with "cryptographically" high probability, $1 - 2^{-d^{\Omega(1)}}$, while our algorithm retains $\tilde{O}(d)$ sample complexity even in this stringent setting.

Our main technique is a new approach to use the powerful *Sum of Squares method (SoS)* to design differentially private algorithms. *SoS proofs to algorithms* is a key theme in numerous recent works in high-dimensional algorithmic statistics – estimators which apparently require exponential running time but whose analysis can be captured by *low-degree Sum of Squares proofs* can be automatically turned into polynomial-time algorithms with the same provable guarantees. We demonstrate a similar *proofs to private algorithms* phenomenon: instances of the workhorse *exponential mechanism* which apparently require exponential time but which can be analyzed with low-degree SoS proofs can be automatically turned into polynomial-time differentially private algorithms. We prove a meta-theorem capturing this phenomenon, which we expect to be of broad use in private algorithm design.

Our techniques also draw new connections between differentially private and robust statistics in high dimensions. In particular, viewed through our proofs-to-private-algorithms lens, several well-studied SoS proofs from recent works in algorithmic robust statistics directly yield key components of our differentially private mean estimation algorithm.

CCS CONCEPTS

• **Theory of computation** → **Sample complexity and generalization bounds**; **Rounding techniques**; • **Mathematics of computing** → **Multivariate statistics**; • **Security and privacy**;

STOC '22, June 20–24, 2022, Rome, Italy
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9264-8/22/06.
https://doi.org/10.1145/3519935.3519947

KEYWORDS

differential privacy, sum-of-squares, exponential mechanism, mean estimation, robust estimation

ACM Reference Format:
Samuel B. Hopkins, Gautam Kamath, and Mahbod Majid. 2022. Efficient Mean Estimation with Pure Differential Privacy via a Sum-of-Squares Exponential Mechanism. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3519935.3519947

1 INTRODUCTION

Mean estimation is perhaps the most elementary statistical task: given samples from a probability distribution D, estimate its expected value. In this paper, we study mean estimation in d dimensions subject to *differential privacy (DP)* [27], a rigorous notion of data privacy. Privacy is of natural concern in high-dimensional statistics, where data may be sensitive and standard estimators like the empirical mean may leak information about individuals in a dataset (see, e.g., privacy attacks in [15, 24, 29, 32] and the survey [28]). On the other hand, privacy and statistical estimation seem largely compatible: in the large-sample limit, good statistical estimators will have vanishing dependence on each individual sample anyway. Indeed, even though the empirical mean (the natural benchmark for accuracy in mean estimation) is not differentially private, estimators are known which match its accuracy guarantees while also providing differential privacy [13, 43]. Namely, an accurate and private estimate of the mean can be obtained using $n = O(d)$ samples. However, known estimators achieving this sample complexity require exponential running time.

If one instead focuses on polynomial-time algorithms, existing estimators all face significant drawbacks: either they require $n \geq d^{1+\Omega(1)}$ samples, or they may leak private information with probability 2^{-d^c} for small $c > 0$ [31, 39, 43]. (In privacy language, they satisfy only *concentrated* or *approximate* differential privacy.) Since a major goal of privacy is to make strong assurances on leakage of sensitive information, best practices disallow private data leakage even with "cryptographically" small probability $2^{-\operatorname{poly}(d)}$. One way to satisfy this stringent requirement is to provide *pure* differential privacy, which disallows substantial leakage of private information with any nonzero probability. Thus, the main question in our paper is:

Is there a polynomial-time pure DP algorithm for mean estimation using $n = O(d)$ samples?

Our main result answers this question affirmatively, up to logarithmic factors. To state our main result, we define differential privacy:

Definition 1 ((Pure) differential privacy). *For $\varepsilon > 0$, a (randomized) algorithm A which takes n inputs $X_1, \ldots, X_n$ is ε-differentially private (ε-DP) if for every pair of inputs $X_1, \ldots, X_n$ and $X_1', \ldots, X_n'$ such that $X_i = X_i'$ for all except a single index $i \in [n]$ and for all possible events $S \subseteq \mathrm{Range}(A)$,*

$$\mathbb{P}(A(X_1, \ldots, X_n) \in S) \leq e^{\varepsilon} \cdot \mathbb{P}(A(X_1', \ldots, X_n') \in S) \,.$$

Theorem 2 (Main Theorem). *For every $n, d \in \mathbb{N}$ and $R, \alpha, \varepsilon, \beta > 0$ there is a polynomial-time ε-DP algorithm such that for every distribution D on $\mathbb{R}^d$ such that $\| \mathbf{E}_{X \sim D} X \| \leq R$ and $\mathrm{Cov}_{X \sim D}(X) \preceq I$, given $X_1, \ldots, X_n \sim D$, with probability $1 - \beta$ the algorithm outputs $\hat{\mu}$ such that $\|\hat{\mu} - \mathbf{E}_{X \sim D} X\| \leq \alpha$, so long as*

$$n \geq \tilde{O}\left(\frac{d + \log(1/\beta)}{\alpha^2 \varepsilon} + \frac{d \log R + \min(d, \log R) \cdot \log(1/\beta)}{\varepsilon} \right) .$$

Furthermore, if an η-fraction of the samples $X_1, \ldots, X_n$ are adversarially corrupted, the algorithm maintains the same guarantee, at the cost that now $\|\hat{\mu} - \mathbf{E} X\| \leq \alpha + O(\sqrt{\eta})$.

The sample complexity of our algorithm is nearly linear in d, thereby answering our main question up to logarithmic factors. Beyond this core goal, we highlight that our algorithm is *also* robust to adversarial contaminations and enjoys sub-Gaussian confidence intervals when $n \gg \frac{d \log R + \min(d, \log R) \log(1/\beta)}{\varepsilon}$, both sought-after features in recent non-private algorithms [17, 21, 33, 49]. The sample complexity of our algorithm is nearly optimal, with the exception of the term $(\log(1/\beta) \cdot \min(d, \log R))/\varepsilon$ – see the full version of the paper for the corresponding lower bounds. (Put differently, information-theoretically it is possible to achieve sub-Gaussian confidence intervals under the slightly milder assumption $n \gg \frac{d \log R + \log(1/\beta)}{\varepsilon}$.)

Beyond clip-and-noise. Obtaining the guarantees of our algorithm requires going beyond existing techniques for private mean estimation, which we briefly review. A common technique in differential privacy is to "just add noise." More precisely, suppose that $f(X_1, \ldots, X_n)$ is a *bounded sensitivity* function of n inputs, meaning that replacing a single input X_i with an arbitrary X_i' changes the value of f by at most Δ (in an appropriate choice of norm). Then, $f(X_1, \ldots, X_n) + Z$ will be private, where Z is an appropriate random variable whose magnitude depends on Δ.

Existing polynomial-time algorithms for private mean estimation all take this approach. First, to limit the sensitivity of the empirical mean, the algorithms clip the samples $X_1, \ldots, X_n \in \mathbb{R}^d$ to lie in an ℓ_2 ball. Considering the case $\mathrm{Cov}(X) \approx I$, we will have $\|X - \mathbf{E} X\| \approx \sqrt{d}$, so using a ball of radius at least $\sqrt{d}$ is unavoidable without introducing too much error in the clipping phase. Then, the algorithms output $\overline{\mu} + Z$, where $\overline{\mu}$ is the empirical mean of the clipped samples.

If one takes the coordinates of Z to be draws from a Laplace distribution, the resulting algorithm will satisfy ε-DP, but the relatively heavy tails of the Laplace distribution impose a cost that n must be at least $d^{1.5}/\varepsilon$ to obtain nontrivial guarantees. On the other hand, if Z is Gaussian, the resulting algorithm appears to tolerate $n \approx d/\varepsilon$ samples, but it no longer satisfies *pure* DP. Instead, to guarantee privacy loss at most ε with probability at least δ over the internal randomness in the algorithm, $n \geq d\sqrt{\log(1/\delta)}/\varepsilon$ is required (i.e. the algorithm satisfies *concentrated* DP), meaning that for "cryptographically small" δ, this algorithm, too, has super-linear sample complexity.

One might naturally suspect, therefore, that strong privacy guarantees like this simply require $\Omega(d^{1.5})$ samples. Indeed, if nastier distributions than we consider here (violating bounded covariance) are allowed, these two bounds of $\Omega(d^{1.5})$ and $\Omega(d)$ are known to be tight for pure and concentrated DP, respectively [15, 31, 39], and similar separations were previously conjectured even for bounded-covariance distributions [13].

However, [43] (building on related techniques of [13]) show that in exponential time one can go beyond the clip-and-noise approach to mean estimation. In particular, a tournament-based approach gives a pure-*DP* algorithm using $O(d/\varepsilon)$ samples. We obtain nearly-matching guarantees in polynomial time – ours is the first polynomial-time private algorithm to go beyond the clip-and-noise approach.

Sum-of-Squares Proofs to Private Algorithms. Proofs to Algorithms has become a powerful algorithm-design technique in computational high-dimensional statistics. Roughly speaking, the proofs to algorithms technique shows that statistical estimation problems which can be solved to a given accuracy by some (not necessarily polynomial-time) estimator can actually be solved in polynomial time with the same accuracy *if the* analysis *of that estimator can be captured by a certain restricted but powerful formal proof system, known as the SoS proof system.* This insight has had major consequences in robust and heavy-tailed statistics, clustering, learning latent variable models, and beyond. (For instance, [35, 47, 48, 59], and the survey [61].)

We show that this approach also applies to private algorithm design. Our techniques give a generic method to turn (potentially) exponential-time instances of the workhorse *exponential mechanism*, whose breadth of applicability is hard to overstate [58], into polynomial time algorithms, when their analyses are captured by the same SoS proof system. This gives us the following *proofs-to-algorithms* principle for private algorithm design, which we anticipate will be widely applicable:

> **Proofs to Private Algorithms** (see Theorem 11): Instances of the exponential mechanism with low-degree SoS proofs of bounded sensitivity and utility automatically yield computationally-efficient private algorithms.

(See Section 2 for a description of the exponential mechanism and more on SoS proofs.) We are able to capture this principle as a formal meta-theorem, Theorem 11 – since its statement requires a few more technical definitions than we are ready to state, we defer it for now.

Convex Programming and Private Algorithms. Like nearly all applications of the Sum of Squares method, algorithms which result from our proofs-to-private-algorithms approach ultimately use convex programs – semidefinite programs, in our case. Our techniques give a substantially novel approach to convex programming in private algorithms. Prior works largely do this in one of two ways. Some privatize the input *before* applying a convex program to it (e.g., privatizing a graph and then post-processing with convex programming [11, 30]). This can be limiting, as the convex program cannot itself be helpful in achieving privacy, only in some post-processing. Other algorithms use a private solver for the convex program itself [37, 52]. This can be quite technically challenging: few DP solvers for convex programs are known, and in particular, we are not aware of any generic DP algorithm for solving semidefinite programs.

Instead, our algorithms *hide convex programs behind the exponential mechanism.* Namely, we use convex programs as score functions. (Again, see Section 2 for an explanation of this terminology.) Our key insight is: *when convex programs are married to the exponential mechanism in the correct way, the resulting exponential mechanisms can be implemented in polynomial time using private log-concave sampling algorithms.* This builds directly on [9], who use a similar approach but for much simpler un-constrained convex optimization problems resulting from empirical risk minimization problems. The analysis of this interplay of exponential mechanism and convex programming is completely generic – meaning it has nothing to do with the particular setting of mean estimation – and is captured in the proof of our (meta)-Theorem 11.

Robust Statistics and Privacy. Robust statistics, the study of statistics in the presence of corrupted samples, has enjoyed a recent renaissance [22]. Robustness and privacy are at least spiritual cousins – both demand that a statistical estimator not change its behavior "too much" when one or several samples are replaced arbitrarily. However, the formal requirements for robustness and privacy are rather different. The output of a good robust mean estimator should not move far in Euclidean distance when 1% of the samples it is given are corrupted by a malicious adversary. Privacy, by contrast, demands that the *distribution* of outputs of an algorithm not shift too much when any single sample is replaced by another.

In spite of this formal difference, [26] was able to use robust estimators in one dimension to construct private estimators. This suggests that the flurry of recent algorithms for high-dimensional robust statistics with strong provable guarantees should yield high-dimensional private estimators.

Our work makes good on this promise. In particular, our algorithm for private mean estimation ultimately employs a well-studied Sum-of-Squares SDP from robust mean estimation, and our analysis applies several SoS proofs originally formulated to analyze that SDP in robust statistics. Thus, our proofs-to-private-algorithms approach gives a lens through which algorithms from robust statistics can yield private algorithms.

Lastly, there is an even more direct connection between our particular result and recent developments in robust statistics. The robustness renaissance was kicked off by [20, 49], which gave the first polynomial-time algorithms for robustly learning a Gaussian where corrupted samples can be tolerated with dimension-independent error. Prior polynomial-time algorithms for the same problem have guarantees no better than what can be obtained by naive sample-clipping, and as a result incur error scaling as $\eta \cdot \text{poly}(d)$ when an η-fraction of samples are corrupted. Just as [20, 49] gave the first algorithms with guarantees going beyond naive sample clipping in the robust setting, our work is the first to go beyond clip-and-noise in the private setting.

Related Work. A mature body of work focuses on private mean estimation. The most relevant results restrict the underlying distribution to satisfy a moment bound (including sub-Gaussianity), examples include [1, 4, 8, 10, 12–14, 16, 25, 38–41, 43, 46, 65]. All prior study of the multivariate case either focuses on concentrated or approximate DP, or provides computationally-inefficient algorithms for pure DP. We are the first to give an efficient $O(d)$-sample algorithm for multivariate mean estimation under pure DP. This matches the sample complexity of the best known algorithms under concentrated DP, while simultaneously strengthening the privacy guarantee.

Other prior work studies private mean estimation of arbitrary (bounded) distributions [15, 29, 63]. Interestingly, in this case, the optimal sample complexity under pure and concentrated DP are separated by a factor of $\sqrt{d}$, leading to the naive Laplace mechanism being effectively optimal. It appears that our bounded moments assumption induces a qualitatively different structure, which eliminates the benefits of relaxing to concentrated DP. Pinpointing the precise conditions under which a separation occurs remains an interesting direction for future work.

Other works study private statistical estimation in different and more general settings, including mixtures of Gaussians [2, 42], graphical models [67], discrete distributions [19], and median estimation [5, 64]. Some recent directions involve guaranteeing user-level privacy [51, 54], or a combination of local and central DP for different users [6]. See [44] for further coverage of DP statistical estimation.

Several other works employ sampling-based methods to efficiently implement the exponential mechanism [3, 9, 45, 50]. [3, 45, 50] construct sophisticated sampling algorithms by hand to sample from a non-log-concave distributions; by contrast, we use convex programming to construct our score functions, ensuring that we always stay within the realm of log-concave sampling algorithms. [9] constructs the private log-concave sampler we use in our work, but they employ it only to sample from comparatively simple log-concave distributions arising from empirical risk minimization of convex loss functions. A recent work of [57] provides a faster algorithm for private log-concave sampling.

Another recent line of work studies sub-Gaussian confidence intervals for mean estimation of heavy-tailed distributions (i.e., assuming only bounded covariance). Lugosi and Mendelson [56] proposed an inefficient algorithm, with an SoS-based algorithm coming in [33]; see also the survey [55]. Some works considered efficient algorithms for simultaneously robust and heavy-tailed mean estimation [18, 23, 60]. A recent result shows that the core solution concepts for robust and heavy-tailed mean estimation can be considered equivalent [36]. Our work demonstrates that the line of efficient estimators inspired by [33] is simultaneously effective for robust, heavy-tailed, and private mean estimation.

Relatively limited work simultaneously considers privacy and the other constraints of robustness and heavy-tailed estimation. The main result is [52] which considers robust and private mean estimation. Our result can be seen as improving on the running

Table 1: Comparing private mean estimation algorithms for distributions with bounded covariance. (Some papers contain several algorithms.) Sample complexity column ignores logarithmic factors in d. "-" indicates that we are not aware of any analysis in the literature.

Algorithm	Sample Complexity	Privacy	Sub-Gaussian Rates	Robust	Poly Time
Sample Mean	d	None	No	No	Yes
[43]-A	d	Pure	-	Yes	No
[43]-B	d	Concentrated	No	No	Yes
[43]-C	$d^{1.5}$	Pure	No	No	Yes
[52]-A	d	Approximate	-	Yes	No
[52]-B	$d^{1.5}$	Concentrated	No	Yes	Yes
Theorem 2	d	Pure	$n \gg \frac{d \log R + \min(d, \log R) \log(1/\beta)}{\varepsilon}$	Yes	Yes

time or sample complexity of their two algorithms, and the privacy guarantee of both. While other prior works focus on heavy-tailed distributions [8, 40, 43, 65], these do not address the primary goal in the non-private setting, which is to achieve sub-Gaussian rates with respect to the error probability. Instead, they generally prove guarantees with constant probability of success, which can be boosted at the cost of a multiplicative (rather than additive) logarithmic factor which is inverse in the failure probability. Note that some of the previous (inefficient) cover- and median-based approaches to private estimation [1, 12, 13, 43, 52] have varying levels of robustness and sub-Gaussian rates for heavy-tailed settings. However, none of their results give computationally efficient estimators for pure DP. A simultaneous and independent work of [53] demonstrates an interesting connection between resilience [62] and private estimation. They exploit this connection to design robust and private algorithms for a variety of settings, including mean estimation, covariance estimation, PCA, and more. However, their focus is on providing inefficient algorithms under the constraint of approximate DP, while our goal is to give a framework for efficient algorithms under pure DP. For a summarized comparison of our work with recent private algorithms, see Table 1.

2 TECHNIQUES

2.1 The SoS Exponential Mechanism

Before we turn to our algorithm for mean estimation, we offer a little more detail on the proofs-to-private-algorithms approach and its core component, which we call the SoS exponential mechanism. We begin with a review of the exponential mechanism itself.

The Exponential Mechanism. Consider the general problem of privately selecting one among a set of candidates C given a dataset X, where the quality of a candidate $x \in C$ depends on the dataset X. The candidates C could represent many different things depending on the context. In a statistical setting one may often think of C as a class of probability distributions, X as a list of samples from one of those distributions, and the goal is to select the distribution from which X came (up to small error).

To apply the exponential mechanism for this problem, one first finds a *score function* $s(X, x)$ which assigns a (real-valued) score to each dataset-candidate pair, ideally such that x's which are "good" for a given X receive high scores. Given X and a privacy parameter $\varepsilon > 0$, the exponential mechanism will sample a random x with probabilities $\mathbb{P}(x) \propto \exp(\varepsilon \cdot s(X, x))$. The output is ε-differentially private so long as the score function satisfies a *bounded sensitivity* property: for any pair of neighboring datasets X, X'[1] and for all $x \in C$, one has $|s(X, x) - s(X', x)| \leq 1$.

Beyond privacy, one also wants the resulting x to be useful. While this can also mean different things depending on the context, we will take "utility" to mean that x is close to some good $x^*(X)$. To prove that this happens for the exponential mechanism defined above, one shows:

(1) High-scoring x's are good: if $s(X, x) \geq 0$, then $\|x - x^*\| \leq \alpha$, for a small $\alpha > 0$. (The choice of 0 is without loss of generality; the mechanism is invariant under additive shifts of s.)
(2) Not too few high-scoring x's: $\frac{|\{x \in C : s(X,x) \geq t\}|}{|C|} \geq \frac{1}{r}$. (If C is an infinite set one can replace $|\cdot|$ with some measure of volume.)

Then a simple argument shows that x such that $\|x - x^*\| \leq \alpha$ is selected with probability at least $1 - r\exp(-\varepsilon t)$. In other words, the mechanism selects a good x so long as $t \gg \frac{\log r}{\varepsilon}$.

The exponential mechanism can lead to computationally inefficient algorithms for two basic reasons. First, in high-dimensional statistics it is often natural to use score functions which seem hard to compute – the Tukey depth, for just one example [12, 52]. Second, as we will face when we turn to mean estimation, even with score functions that are easy to compute, sampling x with $\mathbb{P}(x) \propto \exp(\varepsilon \cdot s(X, x))$ may not be computationally tractable.
Convex programs as score functions The SoS exponential mechanism can address both of these sources of intractability for instances of the exponential mechanism where the *proofs* of bounded sensitivity are captured in a powerful formal proof system, the *Sum of Squares proof system* (SoS). To de-mystify this a little without yet delving into the details of SoS proofs, we can see a high-level picture of the algorithms which use the SoS exponential mechanism. Ultimately, these algorithms fit into the exponential mechanism framework, but with special score functions.

To wit, suppose that we can arrange for the score function $s(X, x)$ to take the form of a linear optimization problem over a convex set

[1] We say X, X' are neighboring if they differ on the presence/absence of just one individual.

$\mathcal{K}(X)$ with an additional linear constraint involving x:

$$s(X,x) = \max_{y \in \mathcal{K}(X)} \langle c, y \rangle \text{ such that } Ay = x$$

for some matrix A and vector c. As long as $\mathcal{K}(X)$ admits a computationally efficient separation oracle, $s(X,x)$ will be polynomial-time computable – this already addresses the first source of potential intractability in the exponential mechanism. A simple argument shows even more: for each X, $s(X,x)$ is actually concave in x. Thus, the distribution $\mathbb{P}(x) \propto \exp(\varepsilon \cdot s(X,x))$ is log-concave, so we can (usually) sample from it in polynomial time! This sampling step itself has to be done privately: luckily for us, log-concave sampling is so well understood that private methods for polynomial-time log-concave sampling are known [9].

The restriction that the score function take the form of an optimization problem is not a major one: many useful score functions in high-dimensional statistics, such as the Tukey depth, are naturally expressible in this way. But how to find a score function which is simultaneously a *convex* optimization problem and maintains both bounded sensitivity and utility? The SoS method gives us a way to construct such a score function automatically, starting from any score function which is expressible as a (potentially non-convex) optimization problem, and for which the *proofs* of bounded sensitivity and utility are expressible in the SoS proof system.

2.2 Private Mean Estimation

We proceed with a high-level description of our algorithm for private mean estimation and its analysis. One could obtain our algorithm as an instance of proofs-to-private-algorithms, by:

- Constructing a certain simple-to-analyze exponential mechanism-based mean estimator.
- Writing the simple analysis as a series of SoS proofs and applying (meta)-Theorem 11.

However, so that our algorithm can be understood without tackling the SoS exponential mechanism in full generality, we will now give a more concrete description of the algorithm and its analysis. We give a high level version of this description here, and in the main body of our paper we actually provide a full analysis of the algorithm without appealing to Theorem 11.

Let us recall the setup for our problem. There is a random variable X on $\mathbb{R}^d$ with $\mathrm{Cov}(X) \preceq I$ and $\|\mathbf{E}\, X\| \leq R$. The goal is to estimate $\mathbf{E}\, X$ from i.i.d. copies $X_1, \ldots, X_n$, subject to ε-DP. The promise $\|\mathbf{E}\, X\| \leq R$ is information-theoretically necessary for pure differential privacy [13, 46].

Our algorithm has strong guarantees in the presence of adversarially-corrupted samples, and obtains sub-Gaussian confidence intervals given heavy-tailed samples. In fact, this is a side-effect of the fact that our algorithm uses well-studied SDPs from the robust statistics setting, viewing them through the SoS exponential mechanism lens to obtain privacy. In particular, folklore adaptations of the analyses in [17, 48] directly show that our algorithm is robust, so to avoid a proliferation of notation we give the proof in the non-robust setting, and remark on the relevant robustness properties.

Like prior algorithms for private mean estimation, our algorithm has two phases.

(1) Reduce R to $\mathrm{poly}(d)$, roughly by finding a large ball containing a large number of samples.
(2) Estimate $\mathbf{E}\, X$ under the assumption $\|\mathbf{E}\, X\| \leq \mathrm{poly}(d)$.

We start with the second step, which captures much of our conceptual contribution – even under the assumption $R \leq \sqrt{d}$, prior algorithms could not achieve pure differential privacy. Then we discuss the first step, where prior algorithms require $\Omega(d(\log R + \log(1/\beta)))$ samples. We give an algorithm with sample complexity $\log R(d + \log(1/\beta))$. When $\log R \ll d$, our algorithm improves over prior work. (Information-theoretically, $d \log R + \log(1/\beta)$ is possible.)

2.2.1 Private mean estimation when $\|\mathbf{E}\, X\| \leq \mathrm{poly}(d)$. Let us write $\mu = \mathbf{E}\, X$. For simplicity, for now we focus on the case that our goal is to find $\hat{\mu}$ such that $\|\hat{\mu} - \mu\| \leq O(1)$ using $O(d)$ samples. (We can reduce from the α-error case to this one by placing the samples in buckets containing around $1/\alpha^2$ samples and taking sample means within each bucket.) We will also think of ε as a small constant. A merit of our approach is that it allows powerful techniques from robust statistics to be used for private algorithm design: in this case, our algorithm draws heavily on a robust mean estimation algorithm due to [17], using SoS exponential mechanism to privatize its use of convex programming.

Iterative refinement/gradient descent. Our algorithm will iteratively refine an initial estimate of the mean (without loss of generality, the origin), producing a series of estimates $\hat{\mu}_0 = 0, \hat{\mu}_1, \ldots, \hat{\mu}_T$. In each step t, we will privately find a unit vector v such that $\langle v, \mu - \hat{\mu}_t \rangle \geq \Omega(1) \cdot \|\mu - \hat{\mu}_t\|$. Then we can replace $\hat{\mu}_t$ with $\hat{\mu}_{t+1} = \hat{\mu}_t + r \cdot v$ for some appropriate step size $r > 0$. By standard reasoning this means that $O(\log d)$ steps suffice to obtain $\|\hat{\mu}_T - \mu\| \leq O(1)$.

This gradient-descent approach introduces only logarithmic overheads into our sample complexity and running time, so now we turn to the heart of our algorithm: privately finding v.

Finding private gradients. Given samples $X_1, \ldots, X_n \in \mathbb{R}^d$ and $\hat{\mu}_t$, we would like to privately select a unit vector v such that $\langle v, \hat{\mu}_t - \mu \rangle \geq \Omega(1) \cdot \|\hat{\mu}_t - \hat{\mu}\|$. We will use the exponential mechanism, for which we need to define a score function. For this, we are inspired by recent work in robust and heavy-tailed statistics, where the following has become a standard fact [56]:

Fact 3 (Directions with many outliers are good, informal). *Suppose that $\|\hat{\mu}_t - \mu\| \gg 1$. Then, with high probability over $X_1, \ldots, X_n$, so long as $n \gg d$, there are at least $0.9n$ samples X_i such that $\langle X_i - \hat{\mu}_t, \frac{\mu - \hat{\mu}_t}{\|\mu - \hat{\mu}_t\|} \rangle \geq \Omega(1) \cdot \|\hat{\mu}_t - \mu\|$. Furthermore, for every unit vector v such that*

$$|\{i : \langle X_i - \hat{\mu}_t, v \rangle \geq \Omega(1) \cdot \|\hat{\mu}_t - \mu\|\}| \geq 0.8n\,,$$

we have $\langle v, \mu - \hat{\mu}_t \rangle \geq \Omega(1) \cdot \|\mu - \hat{\mu}_t\|$.

Fact 3 makes a good choice of score function clear: we should use

$$s(X, v) = |\{i : \langle X_i - \hat{\mu}_t, v \rangle \geq \Omega(1) \cdot \|\hat{\mu}_t - \mu\|\}|\,.$$

Utility of this score function is captured by Fact 3, and bounded sensitivity is clear by construction. (In fact, it is exactly this bounded sensitivity property which has already made Fact 3 so important in robust and heavy-tailed statistics.) We do not necessarily know the value of $\|\hat{\mu}_t - \mu\|$, but getting a private estimate of this quantity is not too difficult – we privatize a procedure due to [17].

A straightforward analysis shows that, since the volume of the d-dimensional unit ball is roughly $\exp(d)$, exponential mechanism with this score function will ε-privately select a good v so long as $n \gg d/\varepsilon$, which is exactly the sample complexity we expect for estimating μ to error $O(1)$.

Finding private gradients in polynomial time. Of course, the key problem is that sampling with $\mathbb{P}(v) \propto \exp(\varepsilon \cdot s(X,v))$ may not be possible in polynomial time. Indeed, a seemingly-easier problem, finding the highest-scoring v, seems closely related to computing Tukey depth, which is NP-hard.

However, some hope comes again from robust/heavy-tailed statistics, where approximation algorithms (often based on semidefinite programming) for the problem of finding the highest scoring v have become an invaluable tool. Our starting point is the by-now standard construction of such a semidefinite relaxation, which we briefly review. First, we write this optimization problem as a degree-2 polynomial optimization problem in variables $v_1, \ldots, v_d$ and $b_1, \ldots, b_n$, where the latter are constrained so as to be $0/1$ indicators of $\langle X_i - \hat{\mu}_t, v\rangle \geq r$ for some threshold value r:

$$\max \sum_{b=1}^{n} b_i \text{ such that } \|v\|^2 \leq 1,$$

$$b_i^2 = b_i,$$

$$\text{and } b_i \langle X_i - \hat{\mu}_t, v\rangle \geq b_i \cdot r \text{ for all } i\,.$$

The degree-2 SoS relaxation of this optimization problem optimizes over (degree 2) *pseudoexpectations*, which are linear functionals $\tilde{\mathbf{E}} : \mathbb{R}[b,v]_{\leq 2} \to \mathbb{R}$ defined on degree at most 2 polynomials in b, v which are normalized and positive: $\tilde{\mathbf{E}}\, 1 = 1$ and $\tilde{\mathbf{E}}\, p(b,v)^2 \geq 0$ for all linear p. Concretely:

$$\max_{\tilde{\mathbf{E}}} \tilde{\mathbf{E}} \sum_{i \leq n} b_i \text{ s.t. } \tilde{\mathbf{E}}\, \|v\|^2 \leq 1,$$

$$\tilde{\mathbf{E}}\, b_i^2 = \tilde{\mathbf{E}}\, b_i,$$

$$\text{and } \tilde{\mathbf{E}}\, b_i \langle X_i - \hat{\mu}_t, v\rangle \geq r \cdot \tilde{\mathbf{E}}\, b_i \text{ for all } i. \tag{1}$$

This optimization problem can be solved via semidefinite programming – the resulting (equivalent) SDP optimizes over $(1+n+d) \times (1+n+d)$ block matrices

$$\max \operatorname{Tr} B \text{ such that } \begin{pmatrix} 1 & \tilde{b}^\top & \tilde{v}^\top \\ \tilde{b} & B & W^\top \\ \tilde{v} & W & V \end{pmatrix} \succeq 0,$$

$$\operatorname{Tr} V \leq 1, B_{ii} = \tilde{b}_i,$$

$$\text{and } \langle X_i - \hat{\mu}_t, W_i\rangle \geq r \cdot B_{ii}$$

Here, V is a proxy for the rank-one matrix $vv^\top$ and similarly for B and $bb^\top$. This SDP is known to be a good approximation to the problem of finding the maximum-score v.

We prove the following fact. (As an aside, this is where SoS proofs enter the picture: establishing facts such as the below for SoS SDP relaxations in general requires constructing SoS proofs.)

Fact 4 (Bounded Sensitivity and Utility for (1), informal). ***Bounded Sensitivity:*** *Changing a single sample in optimization problem* (1) *can change the objective value by at most* 1. ***Utility:*** *With high probability over* $X_1, \ldots, X_n$, *if* $\|\hat{\mu}_t - \mu\| \gg 1$, *then any feasible* $\tilde{\mathbf{E}}$ *in* (1) *with objective value at least* $0.8n$ *satisfies* $\langle \tilde{\mathbf{E}}\, v, \mu - \hat{\mu}_t\rangle \geq \Omega(1) \cdot \|\mu - \hat{\mu}_t\|$.

While this proof largely adapts similar arguments in the robust statistics literature, there is a key technical innovation. Prior algorithms employing the SDP described above use a nontrivial rounding step to extract a good vector v from the $d \times d$ PSD matrix V. However, for reasons we discuss below, to use the SoS exponential mechanism, it is important that v can be read directly off of the SDP (more precisely, that the rounding algorithm used is linear). This means we need a stronger rounding procedure than that used in prior works – we are able to show that $\tilde{v} = \tilde{\mathbf{E}}\, v$, a simple linear function of $\tilde{\mathbf{E}}$, is a good choice.

Sampling with convex programs The "utility" part Fact 4 solves the problem of finding a high-scoring v in polynomial time, via semidefinite programming. But we want to find such a high-scoring v privately.

The key idea is to use the SDP to construct a score function: *convexity of the set of feasible solutions and linearity of the objective function now imply log-concavity of the resulting sampling problem*! This observation, while simple, is remarkably powerful: it allows us to employ a well-studied SDP from robust statistics nearly out-of-the-box to obtain strong privacy guarantees.

Ultimately, we employ the score function:

$$s(X, v_0) = \max_{\tilde{\mathbf{E}}} \tilde{\mathbf{E}} \sum_{i \leq n} b_i \text{ s.t. } \tilde{\mathbf{E}}\, \|v\|^2 \leq 1,$$

$$\tilde{\mathbf{E}}\, b_i^2 = \tilde{\mathbf{E}}\, b_i,$$

$$\text{and } \tilde{\mathbf{E}}\, b_i \langle X_i - \hat{\mu}_t, v\rangle \geq r \cdot \tilde{\mathbf{E}}\, b_i \text{ for all } i$$

$$\text{and } \tilde{\mathbf{E}}\, v = v_0\,,$$

together with the private sampling algorithm of [9], to sample from $\mathbb{P}(v_0) \propto \exp(\varepsilon \cdot s(X, v_0))$.

We make a few remarks on the precise way that $s(X, v_0)$ depends on v_0; that is, via the constraint $\tilde{\mathbf{E}}\, v = v_0$, since this choice is not accidental. First of all, it is important that the constraints of the optimization problem defining $s(X, v_0)$ depend *linearly* on v_0; otherwise we might not retain log-concavity of the resulting sampling problem. We can do this only because the rounding algorithm described in Fact 4, which proves that $\tilde{\mathbf{E}}\, v$ itself is useful, is a simple linear function of $\tilde{\mathbf{E}}$.

We also note an important interplay between the "lifted" nature of the SDP and the utility analysis of the exponential mechanism. On the one hand, the power of the SDP comes from lifting from the $d{+}n$ variables v, b to $(d{+}n{+}1)^2$ variables, and solving a convex problem in the lifted space. On the other hand, for the exponential mechanism to satisfy utility with just $O(d)$ samples, it is important that we use it to sample in d dimensions (where, in particular, there is a $2^{O(d)}$-sized cover) rather than, say, d^2 dimensions. This tension is resolved by hiding the additional variables inside the optimization problem which defines s, so that exponential mechanism still samples from a d-dimensional distribution, but we can still use the power of SoS and semidefinite programming.

Lipschitzness The sampling algorithm of [9] (like other algorithms for sampling from log-concave probability distributions) runs in polynomial time in the ambient dimension, *so long as the distribution has Lipschitz log-probabilities.* Natural approaches to force

$s(X, v_0)$ to be Lipschitz, which generally take the form of randomized smoothing, risk violating pure DP, because an algorithm computing a randomized smoothing of s will have some small probability of quietly failing, at which point privacy is at risk.

To ensure that the resulting algorithm runs in polynomial time, therefore, we have to show that $s(X, v_0)$ is Lipschitz with respect to v_0. To establish Lipschitzness, we make a two-step argument:

(1) We show that dual solutions to the SDP defining $s(X, v_0)$ have optimal dual certificates which are not too large in norm. (At most, say, $\text{poly}(d, n)$.)
(2) We show that any such dual solution to the SDP for $s(X, v_0)$ which certifies an upper bound of c can be adapted to a dual solution for $s(X, v_0 + \Delta)$ which certifies an upper bound of $c + \|\Delta\| \text{poly}(d, n)$, for any small perturbation vector Δ.

Together, these imply that $s(X, v_0)$ is $\text{poly}(d, n)$-Lipschitz with respect to v_0. Since the Lipschitz constant only arises in our algorithm's running time, this suffices for our purposes. This concludes our overview of the second phase of our private mean estimation algorithm.

2.2.2 Coarse estimation: from R to $\text{poly}(d)$. We now describe our algorithm to privately localize $\mu = \mathbf{E}\, X$ to a ball of radius $\text{poly}(d)$, beginning only with the promise that $\| \mathbf{E}\, X\| \leq R$ for some large number R. The goal is to do so using as few samples as possible. Existing efficient algorithms [39] can perform this task with probability $1 - \beta$ using $O(d(\log R + \log(1/\beta))/\varepsilon)$ samples (we present this analysis in our paper to capture the dependence on $\log(1/\beta)$). If we used this algorithm, we would obtain sub-Gaussian confidence intervals only when $n \gg \frac{d \log R + d \log(1/\beta)}{\varepsilon}$. However, if we do not worry about running time, the same task can be accomplished using the exponential mechanism using only $O(d \log R/\varepsilon + \log(1/\beta)/\varepsilon)$ samples – much fewer for $\beta \ll 1$, which is important for constructing confidence intervals.

While we do not quite obtain optimal complexity, we are able to improve on existing algorithms in the regime $\log R \ll d$; our algorithm requires $\tilde{O}(d \log R/\varepsilon + \log R \log(1/\beta)/\varepsilon)$ samples.

Our algorithm again follows the proofs-to-private-algorithms approach. We start with the following basic instantiation of the exponential mechanism. We are given samples $X_1, \ldots, X_n$. With probability at least $1 - \beta$ over the choice of $X_1, \ldots, X_n$, as $n \gg \log(1/\beta)$, any ball of radius $\text{poly}(d)$ containing $0.9n$ of the samples will have center which has distance at most $\text{poly}(d)$ to μ. So, we would like to use the exponential mechanism with score function $s(X, x) = |\{i : \|X_i - x\| \leq \text{poly}(d)\}|$ to select a point from the ball of radius R centered at the origin.

As before, it is not clear how to perform the sampling task that this would require in polynomial time. So, we replace the score function with a convex relaxation, in this case built from the "degree-4" SoS relaxation of of the following polynomial optimization problem:

$$\max_{b,x} \sum_{i=1}^{n} b_i \text{ s.t. } \|x\|^2 \leq R^2, b_i^2 = b_i, \\ \text{and } b_i \|X_i - x\|^2 \leq \text{poly}(d) \cdot b_i \text{ for all } i\,.$$

(The degree-4 SoS relaxation shows up here because the optimization problem involves polynomials of degree 3, and SoS relaxations are defined only for even degrees.) That is, we use the score function

$$s(X, x_0) = \max_{\tilde{\mathbf{E}}} \tilde{\mathbf{E}} \sum_{i=1}^{n} b_i \text{ s.t. } \tilde{\mathbf{E}} \text{ satisfies } \|x\|^2 \leq R^2, \\ b_i^2 = b_i, \\ b_i \|X_i - x\|^2 \leq b_i (R/10)^2, \\ \text{and } \tilde{\mathbf{E}}\, x = x_0$$

(for the definition of "satisfies", see Section 3).

Once we have decided to use this particular SoS relaxation, the outlines of the algorithm and its analysis are largely similar to the second phase of the algorithm, but with different SoS proofs plugged in. In particular, we prove an analogue of Fact 4 for this setting, and establish Lipschitzness of the SDP, so that we can use the same strategy as in second phase.

The key step is to show that a high-scoring x_0 has distance at most $R/2$ to any x' which has distance $\text{poly}(d)$ to $0.8n$ samples. This statement turns out to have a relatively simple SoS proof. This shows that SoS exponential mechanism manages to reduce R to $R/2$. Iterating this $\log R$ times completes the algorithm.

3 PRELIMINARIES

3.1 SoS Proofs and Pseudoexpectations

We give a brief overview of SoS; for details see [7].

SoS Proofs. We first informally review the SoS proof system. Let $p(x), p_1(x), \ldots, p_m(x)$ be multivariate polynomials in indeterminates $x_1, \ldots, x_n$. The SoS proof system can prove statements of the form:

$$\text{For all } x \in \mathbb{R}^n, \text{ if } p_1(x) \geq 0, \ldots, p_m(x) \geq 0, \text{ then } p(x) \geq 0.$$

Polynomials are highly expressive, so a wide range of mathematical statements can be encoded in the above form. An SoS proof of such a statement is a family of polynomials $\{q_S(x) : S \subseteq [m]\}$ such that each $q_S(x) = \sum_{i=1}^{r} (q_S^{(i)}(x))^2$ is a sum of squares, and $p(x) = \sum_{S \subseteq [m]} q_S(x) \cdot \prod_{i \in S} p_i(x)$. The proof has degree $D \in \mathbb{N}$ if each term in this sum is a polynomial of degree at most D. We write:

$$p_1 \geq 0, \ldots, p_m \geq 0 \vdash_D p \geq 0\,.$$

Additionally, we will need one non-standard definition: for any given $j \in [m]$, we say that an SoS proof is degree D_j *with respect to* p_j if every term $q_S(x) \cdot \prod_{i \in S} p_i(x)$ in the proof such that $j \in S$ has degree at most D'. We will sometimes write $p_1 \geq 0, \ldots p_m \geq_0 \vdash_{D, \deg_{p_j} = D_j} p \geq 0$.

SoS proofs are dual solutions to semidefinite programs arising from the Sum of Squares method, where pseudoexpectations are primal solutions. As in many applications of convex programming, to prove facts about primal solutions, the main technique is to construct duals. In particular, to show that the SoS SDPs we use for exponential mechanism score functions satisfy bounded sensitivity and privacy, it suffices to construct SoS proofs witnessing these facts.

Pseudoexpectations. For even $d \in \mathbb{N}$, a degree-d pseudoexpectation in indeterminates $x = x_1, \ldots, x_n$ is a linear operator $\tilde{\mathbf{E}}$:

$\mathbb{R}[x_1, \dots, x_n]_{\leq d} \to \mathbb{R}$ which satisfies $\tilde{\mathbf{E}}\, 1 = 1$ an $\tilde{\mathbf{E}}\, p^2 \geq 0$ for every degree-$d/2$ polynomial p. We say that $\tilde{\mathbf{E}}$ *satisfies* an inequality $p(x) \geq 0$ if for every q such that $\deg(p \cdot q^2) \leq d$ we have $\tilde{\mathbf{E}}\, pq^2 \geq 0$.

Archimedean systems and duality. We say that a system of polynomial inequalities $p_1(x) \geq 0, \dots, p_m \geq 0$ is Archimedean if for some real $M > 0$ it contains the inequality $\|x\|^2 \leq M$. SoS proofs and pseudoexpectations satisfy a natural duality for Archimedean systems, which we use often. Namely: for every Archimedian system $p_1, \dots, p_m$ and every polynomial f and every degree d, exactly one of the following holds.

(1) For every $\varepsilon > 0$ there is an SoS proof $p_1 \geq 0, \dots, p_m \geq 0 \vdash_d f \geq -\varepsilon$
(2) There is a degree-d pseudoexpectation satisfying $p_1 \geq 0, \dots, p_m \geq 0$ but $\tilde{\mathbf{E}}\, f < 0$.

3.2 Privacy

We have already seen the definition of (pure) differential privacy. Our approach relies heavily upon the exponential mechanism of [58], we employ a volume-based version which appears in [45]. The proof is standard but we include it for completeness.

Theorem 5 (volume-based exponential mechanism [45, 58]). *The exponential mechanism M_E on inputs $X, \mathcal{H} \subset \mathbb{R}^d$, s, selects and outputs some object $h \in \mathcal{H}$, where the probability a particular h is selected is proportional to $\exp(\frac{\varepsilon s(X,h)}{2\Delta})$. Let $\mathcal{H}^* \subseteq \mathcal{H}$ be a set such that, $\mathrm{OPT}(X) \leq \inf_{h \in \mathcal{H}^*} s(X, h)$ be a lower bound for the score attained by the objects in $\mathcal{H}^*$ with respect to the dataset X. Moreover, let $\mathrm{vol}(S)$ denote the Lebesgue measure of S in $\mathbb{R}^d$. Then*

$$\mathbb{P}\left[s(M_E(X)) \leq \mathrm{OPT}(X) - \frac{2\Delta}{\varepsilon}\left(\ln\left(\frac{\mathrm{vol}(\mathcal{H})}{\mathrm{vol}(\mathcal{H}^*)} + t\right)\right)\right] \leq \exp(-t).$$

Proof. We follow the same argument as the standard exponential mechanism analysis.

$$\begin{aligned}\mathbb{P}[s(M_E(X)) \leq c] &= \frac{\int_{h: s(X,h) \leq c, h \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h)}{2\Delta}\right) \mathrm{d}h}{\int_{h': h' \in \mathcal{H}} \exp\left(\frac{\varepsilon s(X,h')}{2\Delta}\right) \mathrm{d}h'} \\ &\leq \frac{\mathrm{vol}(\mathcal{H}) \exp\left(\frac{\varepsilon c}{2\Delta}\right)}{\mathrm{vol}(\mathcal{H}^*) \exp\left(\frac{\varepsilon\, \mathrm{OPT}(X)}{2\Delta}\right)} \\ &= \frac{\mathrm{vol}(\mathcal{H})}{\mathrm{vol}(\mathcal{H}^*)} \exp\left(\frac{\varepsilon(c - \mathrm{OPT}(X))}{2\Delta}\right).\end{aligned}$$

From this inequality, the theorem statement can be obtained by substituting in the prescribed value for c. It remains to explain the first inequality. The numerator can be upper bounded since we are taking the integral at most over $\mathcal{H}$, and the value of the integral at each point is less than $\exp(\varepsilon c/2\Delta)$. Similarly, the denominator can be lower bounded by considering only the points in $\mathcal{H}^*$, all of which have a lower bound of $\exp(\varepsilon\, \mathrm{OPT}(X)/2\Delta)$. □

We will also extensively use the following result of [9].

Theorem 6 (Lemma 6.5 of [9]). *For every ε and $d \in \mathbb{N}$, there is an ε-DP algorithm A with the following guarantees. Given access to an evaluation oracle for a concave, L-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$ and to membership and projection oracles for a convex set $C \subseteq \mathbb{R}^d$, the algorithm produces a sample from a distribution D such that for every (measurable) $S \subseteq C$,*

$$e^{-\varepsilon}\, \mathbb{P}(A \in S) \leq \frac{\int_S \exp(f)}{\int_C \exp(f)} \leq e^{\varepsilon}\, \mathbb{P}(A \in S)\,.$$

The algorithm runs in time $\mathrm{poly}(d, L\, \mathrm{diam}(C), 1/\varepsilon, \log \mathrm{diam}(C))$, *making at most that many queries to the oracles.*

[9] actually provides the running time bound $\mathrm{poly}(d, L, \mathrm{diam}(C), 1/\varepsilon)$, but we can deduce the more precise bound in the theorem statement by a simple scaling argument. We note that recent work of [57] provides an algorithm which improves the running time's dependence on $1/\varepsilon$ from polynomial to polylogarithmic, as well as reducing the polynomial dependence on d for convex sets C that contain a ball of radius r, at an additional cost of $\mathrm{poly}\log(1/r)$ (Remark 2.5 of [57]). As our main focus is on providing polynomial time algorithms and not on designing the fastest algorithms, we do not further employ their improved methods.

4 META-THEOREM ON SOS EXPONENTIAL MECHANISM

To describe the SoS exponential mechanism more completely, we prove a meta-theorem on its performance. This meta-theorem could be extended in various ways, but the version we give here captures the mechanism as it is used in our paper. The reader interested solely in our result on mean estimation may comfortably skip this section as we do not rely on it elsewhere, but the exposition and abstraction here may make it easier to follow.

Meta-theorem Setup. Consider the following template to capture the exponential mechanism in the language of polynomials. Let $\mathcal{X}$ be a universe of possible datasets and $C \subseteq \mathbb{R}^n$ be a set of candidates. Consider a score function of the following form. For each dataset X, suppose there is a family of polynomials $p^X, p_1^X, \dots, p_m^X$ in variables $x_1, \dots, x_n, y_1, \dots, y_N$, and the score function $s(X, x)$ is given by

$$s(X, x) = \max_y p^X(x, y) \text{ such that } p_1^X(x, y) \geq 0, \dots, p_m^X(x, y) \geq 0\,.$$

Example 7 (Tukey depth with margin). *To make things a bit less abstract, let us see that a score function closely related to the Tukey depth is expressible in this form. Suppose $X = X_1, \dots, X_N \in \mathbb{R}^n$ is a dataset. Recall that the Tukey depth of a point $z \in \mathbb{R}^n$ is given by the maximum number of X_i's which lie on one side of a hyperplane through z.*

Later on, for a fixed $z \in \mathbb{R}^n$ with $\|z\| = 1$, we will want to run the exponential mechanism to select a direction v such that many X_i's lie above a hyperplane through z in direction v, at distance at least distance $r > 0$ from that hyperplane, so our score function for v is the number of such X_i's. We can express this as follows:

$$s(X, v) = \max_{b_1, \dots, b_N} \sum_{i=1}^N b_i \text{ s.t. for all } i,\ b_i^2 = b_i \text{ and } b_i \langle X_i - z, v\rangle \geq b_i \cdot r\,. \tag{2}$$

Note that for fixed z and v it is easy to compute the number of X_i's lying above the resulting hyperplane, but finding the maximizing v already appears to be a hard problem. (Of course, we want to accomplish only a related task: sampling *from a distribution on high-scoring v's.)*

Indeed, approximation algorithms for finding such v play a key role in recent algorithmic advances in robust and heavy-tailed statistics statistics.

Utility and Bounded Sensitivity in the Language of Polynomials Utility in this framework takes exactly the same form as in the usual exponential mechanism – to demonstrate utility, one would show that high-scoring x's are good, and there are not too many low-scoring x's. The first of these statements is naturally expressible in the SoS proof system, and it turns out that the latter will not need to be expressed as an SoS proof. There is just one technical subtlety: the requirement that $x \in C$, if it is used in the proof of utility, must be captured by the polynomials $p_1(x, y) \geq 0, \ldots, p_m(x, y) \geq 0$, as the SoS proof system has no other way of natively using the hypothesis that $x \in C$. (We illustrate this in the example below.)

To formulate bounded sensitivity within SoS will take a little additional work – SoS will require for there to be a certain kind of witness to bounded sensitivity. This is not a major restriction, as natural proofs of bounded sensitivity for optimization-based score functions typically yield such witnesses anyway.

Making this concrete, a natural way to show that a score function like the above satisfies bounded sensitivity is to relate feasible solutions to the optimization problem for X to those for a neighboring dataset X', without losing too much in the objective value. Let us suppose that for each neighboring pair X, X' there is a transformation $y'(y)$ such that for all x, if y is feasible for X (i.e. $p_1^X(x, y) \geq 0, \ldots, p_m^X(x, y) \geq 0$) then $y'(y)$ is feasible for X' (i.e. $p_1^{X'}(x, y'(y)) \geq 0, \ldots, p_m^{X'}(x, y'(y)) \geq 0$). If additionally $p^X(x, y) - p^{X'}(x, y'(y)) \leq 1$, this transformation (together with the corresponding one mapping y's to ys) witnesses bounded sensitivity for s. For technical reasons, our meta-theorem imposes the restriction that y' is a linear function of y, but this still suffices for our algorithms.

Example 8 (Continuation of Example 7). *First addressing utility: it turns out that the proof of utility for v having high score according to (2) will rely on $\|v\| \leq 1$. So we will have to strengthen our system of polynomials to include this constraint. As a technicality, we will also shift our objective function so that having positive score is good enough for utility.*

$$\max_{b,v} \sum_{i=1}^{N} b_i - 0.9N \text{ such that } b_i^2 = b_i, b_i \langle X_i - z, v \rangle \geq b_i \cdot r, \|v\|^2 \leq 1 \,. \tag{3}$$

Now, to establish bounded sensitivity, consider two neighboring datasets $X = X_1, \ldots, X_N$ and $X' = X_1', X_2, \ldots, X_N$, where X and X' differ on the first vector. If (b, v) is a feasible solution to (3) for X with objective value t, then we can replace it with $(0, b_2, \ldots, b_N, v)$ to get a feasible to solution for X' with objective value at least $t - 1$. The meta-theorem requires there to be an SoS proof of this fact; this (easy) SoS proof was first established in [33].

Robustly Satisfiable Polynomials Before stating our meta-theorem, we need one more technical definition, capturing a certain well-conditioned-ness property of a polynomial optimization problem. Ultimately, this condition will imply a Lipschitz property of semidefinite relaxations of that optimization problem. This Lipschitz property will be used, in turn, to bound the running time of MCMC-based samplers for probability densities using those semidefinite relaxations as log-probabilities.

The details of the following definition (Definition 9) may appear opaque, but they are not too important – a good intuitive interpretation is that the polynomial optimization problem

$$\max_{x,y} p(x, y) \text{ s.t. } p_1(x, y) \geq 0, \ldots, p_m(x, y) \geq 0$$

has a robust space of feasible solutions. Roughly, this means that for any x there is a small ball B around x such that for any $x' \in B$ there is a feasible solution (x', y) whose objective value isn't too large. (The actual condition we use is slightly weaker than this.) This type of condition is common in meta-theorems involving the SoS proof system, to rule out the use of pathological $p, p_1, \ldots, p_m$, and it is typically not too difficult to establish – see, e.g. [34, 66].

Definition 9 (Robustly satisfiable polynomial systems). *Let $C \subseteq \mathbb{R}^n$ and let $p, p_1, \ldots, p_m$ be polynomials in $x = x_1, \ldots, x_n$ and $y = y_1, \ldots, y_N$. Let $\eta > 0$. Consider a family of optimization problems, one for each $x \in C$, given by*

$$\max_{y} p(x, y) \text{ s.t. } p_1(x, y) \geq 0, \ldots, p_m(x, y) \geq 0 \,.$$

We say this family is η-robustly satisfiable if, for each $x \in C$, each x' in the ball of radius η around x can satisfy the constraints. That is, for each x' such that $\|x' - x\| \leq \eta$, there exists y such that $p_1(x', y) \geq 0, \ldots, p_m(x', y) \geq 0$.

Note that in our algorithms, η will factor only into running times and not sample complexity or accuracy guarantees, so rather coarse bounds on η, perhaps loose by polynomial factors, suffice for our purposes.

Example 10 (Continuation of Examples 7, 8). *Let us imagine now that C is a ball of radius 0.9 centered at the origin, to see a proof sketch of η-robust satisfiability for (3). For each $v \in C$ and every vector Δ with $\|\Delta\| \leq 0.1$, the constraints of (3) are satisfied by $(v + \Delta, 0)$. So, (3) is η-well-conditioned for $\eta = 0.1$, with respect to C.*

With this setup in hand, we can state our meta-theorem.

Theorem 11 (Meta-Theorem on SoS Exponential Mechanism). *Let $C \subseteq \mathbb{R}^n$ be a compact, convex set and $\mathcal{X}$ a universe of possible datasets, equipped with a "neighbors" relation. Suppose that for every dataset X there exists an Archimedean and η-robustly satisfiable system of polynomial inequalities $\mathcal{P}^X(x, y) = \{p_1^X(x, y) \geq 0, \ldots, p_N^X(x, y) \geq 0\}$ and a polynomial $p^X(x, y)$, all of degree at most D, in indeterminates $x_1, \ldots, x_n, y_1, \ldots, y_N$ such that for every neighboring dataset X' there is a linear function $y'(y)$ such that bounded sensitivity has an SoS proof:*

$$\forall j,\ \mathcal{P}^X(x, y) \vdash_{\deg(p_j^{X'})} p_j^{X'}(x, y'(y)) \geq 0$$

$$\text{and } \mathcal{P}^X(x, y) \vdash_D p^X(x, y) - p^{X'}(x, y') \leq 1 \,.$$

Suppose also that for every X, there are SoS proofs $\mathcal{P}^X(x, y) \vdash_D p(x, y) \leq 1/\eta$ and $\mathcal{P}^X(x, y) \vdash_D -p(x, y) \leq 1/\eta$. Furthermore, suppose that the polynomials $\mathcal{P}^X$ and p^X, and the polynomials used in the above SoS proofs, all have coefficients expressible in at most B bits.

Then for every $\varepsilon > 0$ and $D \in \mathbb{N}$ there exists an ε-differentially private algorithm which takes as input the polynomials $p^X, p_1^X, \ldots, p_m^X$ and $B, \eta > 0$, with the following guarantees:

Utility: *For every X, if there is an SoS proof of utility for X which is degree-$\deg(p)$ with respect to p, i.e.,*

$$\mathcal{P}^X(x,y) \cup \{p(x,y) \geq 0\} \vdash_{D,\deg_p=\deg(p)} \|x - x^*(X)\|^2 \leq \alpha^2$$

for some vector $x^*(X) \in \mathbb{R}^n$ and $\alpha > 0$, where the coefficients of all polynomials involved in the proof are expressible with B bits, and if

$$\frac{\mathrm{vol}(C)}{\mathrm{vol}(\{x \in C \,:\, \exists y \text{ s.t. } \mathcal{P}^X(x,y) \text{ and } p^X(x,y) \geq t\})} \leq r\,,$$

then the algorithm outputs x such that $\|x - x^*(X)\| \leq \alpha + 2^{-B}$ with probability at least $1 - r\exp(-\Omega(\varepsilon t))$.

Running time: *The algorithm runs in time*

$$\mathrm{poly}\left(n^D, N^D, m^D, \frac{1}{\varepsilon}, \frac{1}{\eta}, \mathrm{diam}(C), B\right),$$

making at most this many calls to membership and projection oracles for C.

4.1 Proof of Theorem 11

We describe the algorithm we use to prove Theorem 11; then we assemble the lemmas we need for the analysis.

SoSExponentialMechanism. Input: polynomials $p^X, p_1^X, \ldots, p_m^X$, $D \in \mathbb{N}$, $\eta > 0$, $B \in \mathbb{N}$.

(1) For $x_0 \in C$, let

$$\begin{aligned} s(X, x_0) = \max_{\tilde{\mathbb{E}}} \tilde{\mathbb{E}}\, p^X(x,y) \text{ such that } & \deg \tilde{\mathbb{E}} = D, \\ & \tilde{\mathbb{E}} \text{ satisfies } \mathcal{P}^X, \\ \text{and } & \tilde{\mathbb{E}}\, x = x_0 \end{aligned}$$

where the optimization is over $\tilde{\mathbb{E}}$ in indeterminates x, y.

(2) Let $B' = B + T(\mathrm{diam}\, C, 1/\eta, 1/\varepsilon, d)$, where T is a sufficiently-large polynomial in the running time of the log-concave private sampler of [9], when run with finite-precision arithmetic.

(3) Run the log-concave private sampling algorithm of [9] (Lemma 6.5) with score function s, Lipschitz parameter $\mathrm{poly}(1/\eta)$, and privacy parameter $\varepsilon/4$. Whenever the sampling algorithm makes a call to $s(X, x_0)$, solve the underlying SDP to $\mathrm{poly}(B')$ bits of precision.

Theorem 11 is immediate from the following lemmas, all of which we establish in the next section.

Lemma 12 (High-scoring x_0 is found in polynomial time). *Given the setup of Theorem 11, for all X, if*

$$\frac{\mathrm{vol}(C)}{\mathrm{vol}(\{x \in C \,:\, \exists y \text{ s.t. } \mathcal{P}^X(x,y) \text{ and } p^X(x,y) \geq t\})} \leq r\,,$$

then with probability at least $1 - r\exp(-\varepsilon(t/2 - 1))$, the x_0 output in step (2) of SoSExponentialMechanism has $s(X, x_0) \geq 0$.

Lemma 13 (High-scoring x_0 is useful). *Under the assumptions of Theorem 11, for all X, if there is an SoS proof of utility for X as described in Theorem 11, then for all x_0 such that $s(X, x_0) \geq 0$, $\|x^*(X) - x_0\| \leq \alpha + 2^{-B}$.*

Lemma 14 (Privacy). *Under the assumptions of Theorem 11, SoSExponentialMechanism satisfies ε-DP.*

4.2 Proofs of Lemmas

We will prove Lemmas 12, 13, and 14.

Remark on numerical issues: Because of the choice of B', the guarantees of the log-concave sampling algorithm will apply equally well if it receives $s(X, x_0) \pm 2^{-B'}$ as if it receives $s(X, x_0)$ when making oracle calls to s. (Given its running time, it cannot even read enough bits to tell the difference.) So, we will henceforth ignore the difference and presume that the log-concave sampler observes the values $s(X, x_0)$ exactly.

The first step is to establish that the target probability distribution is actually log-concave and Lipschitz, so that we can use the guarantees of [9].

Lemma 15. *For all X, the function $s(X, x_0)$ is concave in x_0.*

Proof. Consider x_0 and x_0' and let $\tilde{\mathbb{E}}$ be the optimal solution to the optimization problem defining $s(X, x_0)$ and similarly for $\tilde{\mathbb{E}}'$ and $s(X, x_0')$. Then $\frac{1}{2}\tilde{\mathbb{E}} + \frac{1}{2}\tilde{\mathbb{E}}'$ is feasible for $s(X, \frac{1}{2}x_0 + \frac{1}{2}x_0')$. So $s(X, \frac{1}{2}x_0 + \frac{1}{2}x_0') \geq \frac{1}{2}s(X, x_0) + \frac{1}{2}s(X, x_0')$. □

Lemma 16 (Robustly satisfiable systems yield Lipschitz SDPs). *Given the setup in Theorem 11, for all $x_0, x_0' \in C$ and all X, we have $|s(X, x_0) - s(X, x_0')| \leq \mathrm{poly}(1/\eta) \cdot \|x_0 - x_0'\|$.*

Proof. As shorthand, let us write $s = s(X, x_0)$. Since $\mathcal{P}^X$ is Archimedean, we can apply standard pseudoexpectation/SoS proof duality to conclude that for every $\varepsilon > 0$ there is a polynomial identity in variables x, y:

$$s + \varepsilon - p^X(x,y) = \sum_{S \subseteq [m]} q_S(x,y) \prod_{i \in S} p_i^X(x,y) + \langle \lambda, x - x_0 \rangle\,,$$

where q_S are SoS polynomials, all the terms above have degree at most D, and $\lambda \in \mathbb{R}^n$ is a vector.

Our first goal is to bound $\|\lambda\|$. By robust satisfiability, if we let $x' = x_0 + \eta \cdot \frac{\lambda}{\|\lambda\|}$, there exists y' such that $p_i^X(x', y') \geq 0$ for all i. Hence,

$$s + \varepsilon - p^X(x', y') \geq \eta\|\lambda\|\,.$$

Since there are SoS proofs that $p^X(x,y) \leq 1/\eta$ and $-p^X(x,y) \leq 1/\eta$, and we can take $|\varepsilon| \leq 1/\eta$, the left-hand side is $O(1/\eta)$, so we find $\|\lambda\| \leq O(1/\eta^2)$.

We claim that $s(X, x_0') \leq s + O(1/\eta^2) \cdot \|x_0 - x_0'\|$. To see this, note that for each $\varepsilon > 0$, we can write

$$s + \varepsilon + \langle \lambda, x_0 - x_0' \rangle - p^X(x,y) = \sum_{S \subseteq [m]} q_S(x,y) \prod_{i \in S} p_i^X(x,y) + \langle \lambda, x - x_0' \rangle\,,$$

which, after Cauchy-Schwarz, certifies the upper bound $s + \varepsilon + \|\lambda\|\|x_0 - x_0\|$ on $s(X, x_0')$. Since this works for all $\varepsilon > 0$, we find $s(X, x_0') \leq s + O(1/\eta^2)\|x_0 - x_0'\|$. □

As a corollary of Lemmas 15 and 16, combined with Lemma 6.5 of [9], we obtain:

Corollary 17. *Given the setup of Theorem 11, for every X, the output of step (3) of SoSExponentialMechanism is a sample from a distribution D_X supported on C such that for every event A*

$$e^{-\varepsilon/2} \mathop{\mathbb{P}}_{D_X^{target}}(A) \leq \mathop{\mathbb{P}}_{D_X}(A) \leq e^{\varepsilon/2} \mathop{\mathbb{P}}_{D_X^{target}}(A)$$

where D_X^{target} is the distribution with density proportional to $\exp((\varepsilon/2)s(X,x_0))$ Furthermore, the sampler runs in time at most $\mathrm{poly}(d, \frac{1}{\varepsilon}, \frac{1}{\eta}, \mathrm{diam}(C))$, making at most that many calls to a membership oracle for C and to an evaluation oracle for $s(X,\cdot)$.

4.2.1 *Privacy: proof of Lemma 14.*

Lemma 18. *Given the conditions of Theorem 11, the score function s has sensitivity at most* 1.

Proof. Let X, X' be neighboring datasets, and let $\tilde{\mathbb{E}}$ be an optimal solution to the optimization problem defining $s(X,x_0)$. We will construct a feasible solution $\tilde{\mathbb{E}}'$ for $s(X',x_0)$ whose objective value is at most $s(X,x_0)+1$. Since we could swap X and X', this will prove that $|s(X,x_0)-s(X',x_0)| \le 1$.

For any degree D polynomial f, we define $\tilde{\mathbb{E}}' f(x,y) = \tilde{\mathbb{E}} f(x,y'(y))$. Note that the degree of $\tilde{\mathbb{E}}'$ as a pseudoexpectation is the same as that of $\tilde{\mathbb{E}}$, because $y'(y)$ is linear.

We claim that $\tilde{\mathbb{E}}'$ is feasible for $s(X',x_0)$. Clearly $\tilde{\mathbb{E}}' x = x_0$, so we just need to check that $\tilde{\mathbb{E}}'$ satisfies $\mathcal{P}^{X'}$. For each j, we check that $\tilde{\mathbb{E}}'$ satisfies $p_j^{X'}(x,y) \ge 0$. Consider any square polynomial q such that $\deg(q \cdot p_j^{X'}) \le D$. We need to show $\tilde{\mathbb{E}}' q \cdot p_j^{X'} \ge 0$.

Using the SoS proof $\mathcal{P}^X(x,y) \vdash_{\deg(p_j^{X'})} p_j^{X'}(x,y'(y))$, we can write

$$q(x,y') \cdot p_j^{X'}(x,y') = q(x,y') \cdot \sum_{S \subseteq [m]} q_S(x,y) \prod_{i \in S} p_i^X(x,y)$$

where every term in the sum on the right-hand side has degree at most $\deg p_j^{X'}$. Therefore, for every S, we have

$$q(x,y'(y)) \cdot q_S(x,y) \prod_{i \in S} p_i^X(x,y)$$

has degree at most D. So, applying $\tilde{\mathbb{E}}'$ to both sides, we find that $\tilde{\mathbb{E}}' q p_j^{X'} \ge 0$, using that $\tilde{\mathbb{E}}$ satisfies $\mathcal{P}^X$.

Finally, we have to check that $\tilde{\mathbb{E}} p^X(x,y) - \tilde{\mathbb{E}}' p^{X'}(x,y) \le 1$. We expand the definitions and use our SoS proof of bounded sensitivity.

$$\tilde{\mathbb{E}} p^X(x,y) - \tilde{\mathbb{E}}' p^{X'}(x,y) = \tilde{\mathbb{E}} p^X(x,y) - \tilde{\mathbb{E}} p^{X'}(x,y') \le 1\,.$$

□

Proof of Lemma 14. By Lemma 18 and the usual analysis of the exponential mechanism, an output from the target distribution D_X^{target} from Corollary 17 would satisfy $\varepsilon/2$-DP. Since, according to Corollary 17, the actual distribution output by step (3) of the algorithm differs only by multiplicative $e^{\varepsilon/2}$, the output of the log-concave sampler satisfies ε-DP. □

4.2.2 *Utility: proofs of Lemmas 12 and 13.*

Proof of Lemma 12. By the standard analysis of the exponential mechanism, a sample from D_X^{target} (as described in Corollary 17) would output x_0 with $s(X,x_0)$ with probability at least $1 - r\exp(-\varepsilon t/2)$. Since the actual output distribution of step (3) is $e^{\varepsilon/2}$ multiplicatively close to this, we are done. □

Proof of Lemma 13. Since x_0 has $s(X,x_0) \ge 0$, there exists $\tilde{\mathbb{E}}$ of degree D satisfying $p_1^X(x,y) \ge 0, \ldots, p_m^X(x,y) \ge 0$ and having $\tilde{\mathbb{E}} x = x_0$ and $\tilde{\mathbb{E}} p^X(x,y) \ge 2^{-\mathrm{poly}(B)}$. Then applying $\tilde{\mathbb{E}}$ to either side of the SoS proof of utility, we obtain $\tilde{\mathbb{E}} \|x - x^*(X)\|^2 \le \alpha^2 + 2^{-\mathrm{poly}(B)}$. By convexity, $\|\tilde{\mathbb{E}} x - x^*(X)\| \le \alpha + 2^{-\mathrm{poly}(B)}$, so we are done. □

ACKNOWLEDGMENTS

The authors would like to thank Adam Smith for his role in the conception and contributions in the early stages of this project.

REFERENCES

[1] Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. 2021. On the Sample Complexity of Privately Learning Unbounded High-Dimensional Gaussians. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory (ALT '21)*. JMLR, Inc., 185–216.

[2] Ishaq Aden-Ali, Hassan Ashtiani, and Christopher Liaw. 2021. Privately Learning Mixtures of Axis-Aligned Gaussians. In *Advances in Neural Information Processing Systems 34 (NeurIPS '21)*. Curran Associates, Inc.

[3] Kareem Amin, Travis Dick, Alex Kulesza, Andres Munoz, and Sergei Vassilvitskii. 2019. Differentially Private Covariance Estimation. In *Advances in Neural Information Processing Systems 32 (NeurIPS '19)*. Curran Associates, Inc., 14190–14199.

[4] Hassan Ashtiani and Christopher Liaw. 2021. Private and polynomial time algorithms for learning Gaussians and beyond. *arXiv preprint arXiv:2111.11320* (2021).

[5] Marco Avella-Medina and Victor-Emmanuel Brunel. 2019. Differentially Private Sub-Gaussian Location Estimators. *arXiv preprint arXiv:1906.11923* (2019).

[6] Brendan Avent, Yatharth Dubey, and Aleksandra Korolova. 2019. The Power of the Hybrid Model for Mean Estimation. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (2019), 48–68.

[7] Boaz Barak and David Steurer. 2016. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *Course notes: http://www. sumofsquares. org/public/index. html* (2016).

[8] Rina Foygel Barber and John C Duchi. 2014. Privacy and Statistical Risk: Formalisms and Minimax Bounds. *arXiv preprint arXiv:1412.4451* (2014).

[9] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS '14)*. IEEE Computer Society, Washington, DC, USA, 464–473.

[10] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. 2020. CoinPress: Practical Private Mean and Covariance Estimation. In *Advances in Neural Information Processing Systems 33 (NeurIPS '20)*. Curran Associates, Inc., 14475–14485.

[11] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*. IEEE Computer Society, Washington, DC, USA, 410–419.

[12] Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakynthinou. 2021. Covariance-Aware Private Mean Estimation Without Private Covariance Estimation. *arXiv preprint arXiv:2106.13329* (2021).

[13] Mark Bun, Gautam Kamath, Thomas Steinke, and Zhiwei Steven Wu. 2019. Private Hypothesis Selection. In *Advances in Neural Information Processing Systems 32 (NeurIPS '19)*. Curran Associates, Inc., 156–167.

[14] Mark Bun and Thomas Steinke. 2019. Average-Case Averages: Private Algorithms for Smooth Sensitivity and Mean Estimation. In *Advances in Neural Information Processing Systems 32 (NeurIPS '19)*. Curran Associates, Inc., 181–191.

[15] Mark Bun, Jonathan Ullman, and Salil Vadhan. 2014. Fingerprinting Codes and the Price of Approximate Differential Privacy. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing (STOC '14)*. ACM, New York, NY, USA, 1–10.

[16] T. Tony Cai, Yichen Wang, and Linjun Zhang. 2019. The Cost of Privacy: Optimal Rates of Convergence for Parameter Estimation with Differential Privacy. *arXiv preprint arXiv:1902.04495* (2019).

[17] Yeshwanth Cherapanamjeri, Nicolas Flammarion, and Peter L. Bartlett. 2019. Fast Mean Estimation with Sub-Gaussian Rates. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT '19)*. 786–806.

[18] Jules Depersin and Guillaume Lecué. 2019. Robust subgaussian estimation of a mean vector in nearly linear time. *arXiv preprint arXiv:1906.03058* (2019).

[19] Ilias Diakonikolas, Moritz Hardt, and Ludwig Schmidt. 2015. Differentially Private Learning of Structured Discrete Distributions. In *Advances in Neural Information Processing Systems 28 (NIPS '15)*. Curran Associates, Inc., 2566–2574.

[20] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. 2016. Robust Estimators in High Dimensions without the Computational Intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS '16)*. IEEE Computer Society, Washington, DC, USA, 655–664.

[21] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. 2019. Robust Estimators in High-Dimensions Without the

Computational Intractability. *SIAM J. Comput.* 48, 2 (2019), 742–864.

[22] Ilias Diakonikolas and Daniel M. Kane. 2019. Recent Advances in Algorithmic High-Dimensional Robust Statistics. *arXiv preprint arXiv:1911.05911* (2019).

[23] Ilias Diakonikolas, Daniel M Kane, and Ankit Pensia. 2020. Outlier Robust Mean Estimation with Subgaussian Rates via Stability. In *Advances in Neural Information Processing Systems 33 (NeurIPS '20)*. Curran Associates, Inc., 1830–1840.

[24] Irit Dinur and Kobbi Nissim. 2003. Revealing Information while Preserving Privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '03)*. ACM, New York, NY, USA, 202–210.

[25] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. 2020. Differentially Private Confidence Intervals. *arXiv preprint arXiv:2001.02285* (2020).

[26] Cynthia Dwork and Jing Lei. 2009. Differential Privacy and Robust Statistics. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing (STOC '09)*. ACM, New York, NY, USA, 371–380.

[27] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography (TCC '06)*. Springer, Berlin, Heidelberg, 265–284.

[28] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. 2017. Exposed! A Survey of Attacks on Private Data. *Annual Review of Statistics and Its Application* 4, 1 (2017), 61–84.

[29] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. 2015. Robust Traceability from Trace Amounts. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '15)*. IEEE Computer Society, Washington, DC, USA, 650–669.

[30] Marek Eliáš, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. 2020. Differentially Private Release of Synthetic Graphs. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '20)*. SIAM, Philadelphia, PA, USA, 560–578.

[31] Moritz Hardt and Kunal Talwar. 2010. On the Geometry of Differential Privacy. In *Proceedings of the 42nd Annual ACM Symposium on the Theory of Computing (STOC '10)*. ACM, New York, NY, USA, 705–714.

[32] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. 2008. Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures using High-Density SNP Genotyping Microarrays. *PLoS Genetics* 4, 8 (2008), 1–9.

[33] Samuel B Hopkins. 2020. Mean Estimation with Sub-Gaussian Rates in Polynomial Time. *The Annals of Statistics* 48, 2 (2020), 1193–1213.

[34] Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. 2017. The power of sum-of-squares for detecting hidden structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 720–731.

[35] Samuel B. Hopkins and Jerry Li. 2018. Mixture Models, Robustness, and Sum of Squares Proofs. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing (STOC '18)*. ACM, New York, NY, USA, 1021–1034.

[36] Samuel B Hopkins, Jerry Li, and Fred Zhang. 2020. Robust and Heavy-Tailed Mean Estimation Made Simple, via Regret Minimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS '20)*. Curran Associates, Inc., 11902–11912.

[37] Justin Hsu, Aaron Roth, Tim Roughgarden, and Jonathan Ullman. 2014. Privately solving linear programs. In *International Colloquium on Automata, Languages, and Programming*. Springer, 612–624.

[38] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. In *Advances in Neural Information Processing Systems 34 (NeurIPS '21)*. Curran Associates, Inc.

[39] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. 2019. Privately Learning High-Dimensional Distributions. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT '19)*. 1853–1902.

[40] Gautam Kamath, Xingtu Liu, and Huanyu Zhang. 2021. Improved Rates for Differentially Private Stochastic Convex Optimization with Heavy-Tailed Data. *arXiv preprint arXiv:2106.01336* (2021).

[41] Gautam Kamath, Argyris Mouzakis, Vikrant Singhal, Thomas Steinke, and Jonathan Ullman. 2021. A Private and Computationally-Efficient Estimator for Unbounded Gaussians. *arXiv preprint arXiv:2111.04609* (2021).

[42] Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan Ullman. 2019. Differentially Private Algorithms for Learning Mixtures of Separated Gaussians. In *Advances in Neural Information Processing Systems 32 (NeurIPS '19)*. Curran Associates, Inc., 168–180.

[43] Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. 2020. Private Mean Estimation of Heavy-Tailed Distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory (COLT '20)*. 2204–2235.

[44] Gautam Kamath and Jonathan Ullman. 2020. A Primer on Private Statistics. *arXiv preprint arXiv:2005.00010* (2020).

[45] Michael Kapralov and Kunal Talwar. 2013. On Differentially Private Low Rank Approximation. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '13)*. SIAM, Philadelphia, PA, USA, 1395–1414.

[46] Vishesh Karwa and Salil Vadhan. 2018. Finite Sample Differentially Private Confidence Intervals. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS '18)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 44:1–44:9.

[47] Pravesh K Kothari and Jacob Steinhardt. 2017. Better agnostic clustering via relaxed tensor norms. *arXiv preprint arXiv:1711.07465* (2017).

[48] Pravesh K Kothari and David Steurer. 2017. Outlier-robust moment-estimation via sum-of-squares. *arXiv preprint arXiv:1711.11581* (2017).

[49] Kevin A. Lai, Anup B. Rao, and Santosh Vempala. 2016. Agnostic Estimation of Mean and Covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS '16)*. IEEE Computer Society, Washington, DC, USA, 665–674.

[50] Jonathan Leake, Colin McSwiggen, and Nisheeth K Vishnoi. 2021. Sampling matrices from Harish-Chandra–Itzykson–Zuber densities with applications to Quantum inference and differential privacy. In *Proceedings of the 53nd Annual ACM Symposium on the Theory of Computing (STOC '21)*. ACM, New York, NY, USA, 1384–1397.

[51] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. 2021. Learning with User-Level Privacy. In *Advances in Neural Information Processing Systems 34 (NeurIPS '21)*. Curran Associates, Inc.

[52] Xiyang Liu, Weihao Kong, Sham Kakade, and Sewoong Oh. 2021. Robust and Differentially Private Mean Estimation. *arXiv preprint arXiv:2102.09159* (2021).

[53] Xiyang Liu, Weihao Kong, and Sewoong Oh. 2021. Differential privacy and robust statistics in high dimensions. *arXiv preprint arXiv:2111.06578* (2021).

[54] Yuhan Liu, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and Michael Riley. 2020. Learning Discrete Distributions: User vs Item-level Privacy. In *Advances in Neural Information Processing Systems 33 (NeurIPS '20)*. Curran Associates, Inc.

[55] Gábor Lugosi and Shahar Mendelson. 2019. Mean Estimation and Regression under Heavy-Tailed Distributions: A Survey. *Foundations of Computational Mathematics* 19, 5 (2019), 1145–1190.

[56] Gábor Lugosi and Shahar Mendelson. 2019. Sub-Gaussian Estimators of the Mean of a Random Vector. *The Annals of Statistics* 47, 2 (2019), 783–794.

[57] Oren Mangoubi and Nisheeth K. Vishnoi. 2021. Sampling from Log-Concave Distributions with Infinity-Distance Guarantees and Applications to Differentially Private Optimization. *arXiv preprint arXiv:2111.04089* (2021).

[58] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*. IEEE Computer Society, Washington, DC, USA, 94–103.

[59] Aaron Potechin and David Steurer. 2017. Exact tensor completion with sum-of-squares. In *Conference on Learning Theory*. PMLR, 1619–1673.

[60] Adarsh Prasad, Sivaraman Balakrishnan, and Pradeep Ravikumar. 2019. A Unified Approach to Robust Mean Estimation. *arXiv preprint arXiv:1907.00927* (2019).

[61] Prasad Raghavendra, Tselil Schramm, and David Steurer. 2018. High dimensional estimation via sum-of-squares proofs. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*. World Scientific, 3389–3423.

[62] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. 2018. Resilience: A Criterion for Learning in the Presence of Arbitrary Outliers. In *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science (ITCS '18)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 45:1–45:21.

[63] Thomas Steinke and Jonathan Ullman. 2015. Interactive Fingerprinting Codes and the Hardness of Preventing False Discovery. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT '15)*. 1588–1628.

[64] Christos Tzamos, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Ilias Zadik. 2020. Optimal Private Median Estimation under Minimal Distributional Assumptions. In *Advances in Neural Information Processing Systems 33 (NeurIPS '20)*. Curran Associates, Inc., 3301–3311.

[65] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. 2020. On Differentially Private Stochastic Convex Optimization with Heavy-tailed Data. In *Proceedings of the 37th International Conference on Machine Learning (ICML '20)*. JMLR, Inc., 10081–10091.

[66] Benjamin Weitz. 2017. *Polynomial proof systems, effective derivations, and their applications in the sum-of-squares hierarchy*. University of California, Berkeley.

[67] Huanyu Zhang, Gautam Kamath, Janardhan Kulkarni, and Zhiwei Steven Wu. 2020. Privately Learning Markov Random Fields. In *Proceedings of the 37th International Conference on Machine Learning (ICML '20)*. JMLR, Inc., 11129–11140.