Robust and differentially private mean estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

In statistical learning and analysis from shared data, which is increasingly widely adopted in platforms such as federated learning and meta-learning, there are two major concerns: privacy and robustness. Each participating individual should be able to contribute without the fear of leaking one's sensitive information. At the same time, the system should be robust in the presence of malicious participants inserting corrupted data. Recent algorithmic advances in learning from shared data focus on either one of these threats, leaving the system vulnerable to the other. We bridge this gap for the canonical problem of estimating the mean from i.i.d. samples. We introduce PRIME, which is the first efficient algorithm that achieves both privacy and robustness for a wide range of distributions. We further complement this result with a novel exponential time algorithm that improves the sample complexity of PRIME, achieving a near-optimal guarantee and matching a known lower bound for (non-robust) private mean estimation. This proves that there is no extra statistical cost to simultaneously guaranteeing privacy and robustness.

1 Introduction

When releasing database statistics on a collection of entries from individuals, we would ideally like to make it impossible to reverse-engineer each individual's potentially sensitive information. Privacy-preserving techniques add just enough randomness tailored to the statistical task to guarantee protection. At the same time, it is becoming increasingly common to apply such techniques to databases collected from multiple sources, not all of which can be trusted. Emerging data access frameworks, such as federated analyses across users' devices or data silos [48], make it easier to temper with such collected datasets, leaving private statistical analyses vulnerable to a malicious corruption of a fraction of the data.

Differential privacy has emerged as a widely accepted de facto measure of privacy, which is now a standard in releasing the statistics of the U.S. Census data [2] statistics and also deployed in real-world commercial systems [70, 38, 39]. A statistical analysis is said to be *differentially private* (DP) if the likelihood of the (randomized) outcome does not change significantly when a single arbitrary entry is added/removed (formally defined in §1.2). This provides a strong privacy guarantee: even a powerful adversary who knows all the other entries in the database cannot confidently identify whether a particular individual is participating in the database based on the outcome of the analysis. This ensures *plausible deniability*, central to protecting an individual's privacy.

In this paper, we focus on one of the most canonical problems in statistics: estimating the mean of a distribution from i.i.d. samples. For distributions with unbounded support, such as sub-Gaussian and heavy-tailed distributions, fundamental trade-offs between accuracy, sample size, and privacy have only recently been identified [56, 50, 52, 3] and efficient private estimators proposed. However, these approaches are brittle when a fraction of the data is corrupted, posing a real threat, referred to as *data poisoning* attacks [18, 76]. In defense of such attacks, robust (but not necessarily private) statistics has emerged as a popular setting of recent algorithmic and mathematical breakthroughs [69, 28].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

One might be misled into thinking that privacy ensures robustness since DP guarantees that a single outlier cannot change the estimation too much. This intuition is true only in a low dimension; each sample has to be an obvious outlier to significantly change the mean. However, in a high dimension, each corrupted data point can look perfectly uncorrupted but still shift the mean significant when colluding together (e.g., see Fig. 1). Focusing on the canonical problem of mean estimation, we introduce novel algorithms that achieve robustness and privacy simultaneously even when a fraction of data is corrupted arbitrarily. For such algorithms, there is a fundamental question of interest: do we need more samples to make private mean estimation also robust against adversarial corruption?

Sub-Gaussian distributions. If we can afford exponential run-time in the dimension, robustness can be achieved without extra cost in sample complexity. We introduce a novel estimator that (i) satisfies (ε, δ)-DP, (ii) achieves near-optimal robustness under α-fraction of corrupted data, achieving accuracy of $O(\alpha\sqrt{\log(1/\alpha)})$ nearly matching the fundamental lower bound of $\Omega(\alpha)$ that holds even for a (non-private) robust mean estimation with *infinite* samples, and (iii) achieves near-optimal sample complexity matching that of a fundamental lower bound for a (non-robust) private mean estimation as shown in Table 1.

Theorem 1 (Informal Theorem 7, exponential time)**.** *Algorithm 2 is (ε, δ)-DP. When α fraction of the data is arbitrarily corrupted from n samples from a d-dimensional sub-Gaussian distribution with mean μ and an identity sub-Gaussian parameter, if $n = \widetilde{\Omega}(d/\alpha^2 + (d + d^{1/2}\log(1/\delta))/(\alpha\varepsilon))$ then Algorithm 2 achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha\sqrt{\log(1/\alpha)})$ w.h.p.*

We introduce PRIME (PRIvate and robust Mean Estimation) in §2.3 with details in Algorithm 9 in Appendix E.1, to achieve computational efficiency. It requires a run-time of only $\widetilde{O}(d^3 + nd^2)$, but at the cost of requiring extra $d^{1/2}$ factor larger number of samples. This cannot be improved upon with current techniques since efficient robust estimators rely on the top PCA directions of the covariance matrix to detect outliers. [75] showed that $\widetilde{\Omega}(d^{3/2})$ samples are necessary to compute PCA directions while preserving (ε, δ)-DP when $\|x_i\|_2 = O(\sqrt{d})$. It remains an open question if this $\widetilde{\Omega}(d^{3/2}/(\alpha\varepsilon))$ bottleneck is fundamental; no matching lower bound is currently known.

Theorem 2 (Informal Theorem 6, polynomial time)**.** *PRIME is (ε, δ)-DP and under the assumption of Thm.1 if $n = \widetilde{\Omega}(d/\alpha^2 + (d^{3/2}\log(1/\delta))/(\alpha\varepsilon))$ achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha\sqrt{\log(1/\alpha)})$ w.h.p.*

	Upper bound (poly-time)	Upper bound (exp-time)	Lower bound
(ε, δ)-DP [50]	$\widetilde{O}(\frac{d}{\alpha^2} + \frac{d\log^{1/2}(1/\delta)}{\alpha\varepsilon})$	$\widetilde{O}(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon})^{\clubsuit}$	$\widetilde{\Omega}(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon})^{\spadesuit}$
α-corruption [34]	$\widetilde{O}(\frac{d}{\alpha^2})$	$\widetilde{O}(\frac{d}{\alpha^2})$	$\Omega(\frac{d}{\alpha^2})$
α-corruption and (ε, δ)-DP (this paper)	$\widetilde{O}(\frac{d}{\alpha^2} + \frac{d^{3/2}\log(1/\delta)}{\alpha\varepsilon})$ [Theorem 6]	$\widetilde{O}(\frac{d}{\alpha^2} + \frac{d+d^{1/2}\log(1/\delta)}{\alpha\varepsilon})$ [Theorem 7]	$\widetilde{\Omega}(\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon})^{\spadesuit}$ [50]

Table 1: For estimating the mean $\mu \in [-R, R]^d$ of a *sub-Gaussian* distribution with a known covariance, we list the sufficient or necessary conditions on the sample sizes to achieve an error $\|\hat{\mu} - \mu\|_2 = \widetilde{O}(\alpha)$ under (ε, δ)-DP, corruption of an α-fraction of samples, and both. $\clubsuit$ requires the distribution to be a Gaussian [12] and $\spadesuit$ requires $\delta \leq \sqrt{d}/n$.

Heavy-tailed distributions. When samples are drawn from a distribution with a bounded covariance, parameters of Algorithm 2 can be modified to nearly match the optimal sample complexity of (non-robust) private mean estimation in Table 2. This algorithm also matches the fundamental limit on the accuracy of (non-private) robust estimation, which in this case is $\Omega(\alpha^{1/2})$.

Theorem 3 (Informal Theorem 8, exponential time)**.** *From a distribution with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \preceq \mathbf{I}$, n samples are drawn and α-fraction is corrupted. Algorithm 2 is (ε, δ)-DP and if $n = \widetilde{\Omega}((d + d^{1/2}\log(1/\delta))/(\alpha\varepsilon) + d^{1/2}\log^{3/2}(1/\delta)/\varepsilon)$ achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha^{1/2})$ w.h.p.*

The proposed PRIME-HT for covariance bounded distributions achieve computational efficiency at the cost of an extra factor of $d^{1/2}$ in sample size. This bottleneck is also due to DP PCA, and it remains open whether this gap can be closed by an efficient estimator.

Theorem 4 (Informal Theorem 9, polynomial time)**.** PRIME-HT *is (ε, δ)-DP and if $n = \widetilde{\Omega}((d^{3/2}\log(1/\delta))/(\alpha\varepsilon))$ achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha^{1/2})$ w.h.p. under the assumptions of Thm. 3.*

	Upper bound (poly-time)	Upper bound (exp-time)	Lower bound
(ε, δ)-DP [52]	$\widetilde{O}(\frac{d \log^{1/2}(1/\delta)}{\alpha\varepsilon})$	$\widetilde{O}(\frac{d \log^{1/2}(1/\delta)}{\alpha\varepsilon})$	$\Omega(\frac{d}{\alpha\varepsilon})$
α-corruption [34]	$\widetilde{O}(\frac{d}{\alpha})$	$\widetilde{O}(\frac{d}{\alpha})$	$\Omega(\frac{d}{\alpha})$
α-corruption and (ε, δ)-DP (this paper)	$\widetilde{O}\big(\frac{d^{3/2}\log(1/\delta)}{\alpha\varepsilon}\big)$ [Theorem 9]	$\widetilde{O}(\frac{d+d^{1/2}\log^{3/2}(1/\delta)}{\alpha\varepsilon})$ [Theorem 8]	$\Omega(\frac{d}{\alpha\varepsilon})$ ([52])

Table 2: For estimating the mean $\mu \in [-R, R]^d$ of a covariance bounded distribution, we list the sufficient or necessary conditions on the sample size to achieve an error $\|\hat{\mu} - \mu\|_2 = O(\alpha^{1/2})$ under (ε, δ)-DP, corruption of an α-fraction of samples, and both.

1.1 Technical contributions

We introduce PRIME which simultaneously achieves (ε, δ)-DP and robustness against α-fraction of corruption. A major challenge in making a standard filter-based robust estimation algorithm (e.g., [28]) private is the high sensitivity of the filtered set that we pass from one iteration to the next. We propose a new framework which makes private only the statistics of the set, hence significantly reducing the sensitivity. Our major innovation is a tight analysis of the end-to-end sensitivity of this multiple interactive accesses to the database. This is critical in achieving robustness while preserving privacy and is also of independent interest in making general iterative filtering algorithms private.

The classical filter approach (see, e.g. [28]) needs to access the database $O(d)$ times, which brings an extra $O(\sqrt{d})$ factor in the sample complexity due to DP composition. In order to reduce the iteration complexity, following the approach in [34], we propose filtering multiple directions simultaneously using a new score based on the matrix multiplicative weights (MMW). In order to privatize the MMW filter, our major innovation is a novel adaptive filtering algorithm DPTHRESHOLD$(\cdot)$ that outputs a *single private threshold* which guarantees sufficient progress at every iteration. This brings the number of database accesses from $O(d)$ to $O((\log d)^2)$.

One downside of PRIME is that it requires an extra $d^{1/2}$ factor in the sample complexity, compared to known lower bounds for (non-robust) DP mean estimation. To investigate whether this is also necessary, we propose a *sample optimal* exponential time robust mean estimation algorithm in §4 and prove that there is no extra statistical cost to jointly requiring privacy and robustness. Our major technical innovations is in using *resilience property of the dataset* to not only find robust mean (which is the typical use case of resilience) but also bound sensitivity of that robust mean.

1.2 Preliminary on differential privacy (DP)

DP is a formal metric for measuring privacy leakage when a dataset is accessed with a query [35].

Definition 1.1. *Given two datasets $S = \{x_i\}_{i=1}^n$ and $S' = \{x_i'\}_{i=1}^{n'}$, we say S and S' are* neighboring *if $d_\triangle(S, S') \le 1$ where $d_\triangle(S, S') \triangleq \max\{|S \setminus S'|, |S' \setminus S|\}$, which is denoted by $S \sim S'$. For an output of a stochastic query q on a database, we say q satisfies (ε, δ)-*differential privacy *for some $\varepsilon > 0$ and $\delta \in (0, 1)$ if $\mathbb{P}(q(S) \in A) \le e^\varepsilon \mathbb{P}(q(S') \in A) + \delta$ for all $S \sim S'$ and all subset A.*

Let $z \sim \mathrm{Lap}(b)$ be a random vector with entries i.i.d. sampled from Laplace distribution with pdf $(1/2b)e^{-|z|/b}$. Let $z \sim \mathcal{N}(\mu, \Sigma)$ denote a Gaussian random vector with mean μ and covariance Σ.

Definition 1.2. *The* sensitivity *of a query $f(S) \in \mathbb{R}^k$ is defined as $\Delta_p = \sup_{S \sim S'} \|f(S) - f(S')\|_p$ for a norm $\|x\|_p = (\sum_{i \in [k]} |x_i|^p)^{1/p}$. For $p = 1$, the Laplace mechanism outputs $f(S) + \mathrm{Lap}(\Delta_1/\varepsilon)$ and achieves $(\varepsilon, 0)$-DP [35]. For $p = 2$, the Gaussian mechanism outputs $f(S) + \mathcal{N}(0, (\Delta_2(\sqrt{2\log(1.25/\delta)})/\varepsilon)^2 \mathbf{I})$ and achieves (ε, δ)-DP [36].*

We use these output perturbation mechanisms along with the exponential mechanism [65] as building blocks. Appendix A provides detailed survey of privacy and robust estimation.

1.3 Problem formulation

We are given n samples from a sub-Gaussian distribution with a known covariance but unknown mean, and α fraction of the samples are corrupted by an adversary. Our goal is to estimate the

unknown mean. We follow the standard definition of adversary in [28], which can adaptively choose which samples to corrupt and arbitrarily replace them with any points.

Assumption 1. *An uncorrupted dataset S_{good} consists of n i.i.d. samples from a d-dimensional sub-Gaussian distribution with mean $\mu \in [-R, R]^d$ and covariance $\mathbb{E}[xx^\top] = \mathbf{I}_d$, which is 1-sub-Gaussian, i.e., $\mathbb{E}[\exp(v^\top x)] \leq \exp(\|v\|_2^2/2)$. For some $\alpha \in (0, 1/2)$, we are given a corrupted dataset $S = \{x_i \in \mathbb{R}^d\}_{i=1}^n$ where an adversary adaptively inspects all the samples in S_{good}, removes αn of them, and replaces them with S_{bad} which are αn arbitrary points in $\mathbb{R}^d$.*

Similarly, we consider the same problem for heavy-tailed distributions with a bounded covariance. We present the assumption and main results for covariance bounded distributions in Appendix B.

Notations. Let $[n] = \{1, 2, \ldots, n\}$. For $x \in \mathbb{R}^d$, we use $\|x\|_2 = (\sum_{i\in[d]}(x_i)^2)^{1/2}$ to denote the Euclidean norm. For $X \in \mathbb{R}^{d\times d}$, we use $\|X\|_2 = \max_{\|v\|_2=1} \|Xv\|_2$ to denote the spectral norm. The $d \times d$ identity matrix is $\mathbf{I}_{d\times d}$. Whenever it is clear from context, we use S to denote both a set of data points and also the set of indices of those data points. $\widetilde{O}$ and $\widetilde{\Omega}$ hide poly-logarithmic factors in $d, n, 1/\alpha$, R, and the failure probability.

Outline. We present PRIME for sub-Gaussian distribution in §2, and present theoretical analysis in §3. We then introduce an exponential time algorithm with near optimal guarantee in §4. Due to space constraints, analogous results for heavy-tailed distributions are presented in Appendix B.

2 PRIME: efficient algorithm for robust and DP mean estimation

In order to describe the proposed algorithm PRIME, we need to first describe a standard (non-private) iterative filtering algorithm for robust mean estimation.

2.1 Background on (non-private) iterative filtering for robust mean estimation

Non-private robust mean estimation approaches recursively apply the following *filter*, whose framework is first proposed in [26]. Given a dataset $S = \{x_i\}_{i=1}^n$, the current set $S_0 \subseteq [n]$ of data points is updated starting with $S_1 = [n]$. At each step, the following filter (Algorithm 1 in [61]) attempts to detect the corrupted data points and remove them.

1. Compute the top eigenvector $v_t \leftarrow \arg\max_{v:\|v\|_2=1} v^\top \mathrm{Cov}(S_{t-1})v$ of the covariance of the current data set $\{x_i\}_{i\in S_{t-1}}$;
2. Compute scores for all data points $j \in S_{t-1}$: $\tau_j \leftarrow \left(v_t^\top (x_j - \mathrm{Mean}(S_{t-1}))\right)^2$;
3. Draw a random threshold: $Z_t \leftarrow \mathrm{Unif}([0, 1])$;
4. Remove outliers from S_{t-1} defined as $\{i \in S_{t-1} : \tau_i$ is in the largest 2α-tail of $\{\tau_j\}_{j\in S_{t-1}}$ and $\tau_i \geq Z_t\, \tau_{\max}\}$, where $\tau_{\max} = \max_{j\in S_{t-1}} \tau_j$

This is repeated until the empirical covariance is sufficiently small and the empirical mean $\hat{\mu}$ is output. At a high level, the correctness of this algorithm relies on the key observation that the α-fraction of adversarial corruption can not significantly change the mean of the dataset without introducing large eigenvalues in the empirical covariance. Therefore, the algorithm finds top eigenvector of the empirical covariance in step 1, and tries to correct the empirical covariance by removing corrupted data points. Each data point is assigned a score in step 2 which indicates the "badness" of the data points, and a threshold Z_t in step 3 is carefully designed such that step 4 guarantees to remove more corrupted data points than good data points (in expectation). This guarantees the following bound achieving the near-optimal sample complexity shown in the second row of Table 1. A formal description of this algorithm is in Algorithm 4 in Appendix C.

Proposition 2.1 (Corollary of [61, Theorem 2.1])**.** *Under assumption 1, the above filtering algorithm achieves accuracy $\|\hat{\mu} - \mu\|_2 \leq O(\alpha\sqrt{\log(1/\alpha)})$ w.p. 0.9 if $n \geq \widetilde{\Omega}(d/\alpha^2)$.*

Challenges in making robust mean estimation private. To get a DP and robust mean, a naive attempt is to apply a standard output perturbation mechanism to $\hat{\mu}$. However, this is obviously challenging since the end-to-end sensitivity is intractable. The standard recipe to circumvent this is to make the current "state" S_t private at every iteration. Once S_{t-1} is private (hence, public knowledge), making the next "state" S_t private is simpler. We only need to analyze the sensitivity of a single step

and apply some output perturbation mechanism with $(\varepsilon_t, \delta_t)$. End-to-end privacy is guaranteed by accounting for all these $(\varepsilon_t, \delta_t)$'s using the advanced composition [49]. This recipe has been quite successful, for example, in training neural networks with (stochastic) gradient descent [1], where the current state can be the optimization variable $\mathbf{x}_t$. However, for the above (non-private) filtering algorithm, this standard recipe fails, since the state S_t is a set and has large sensitivity. Changing a single data point in S_t can significantly alter which (and how many) samples are filtered out.

2.2 A new framework for *private* iterative filtering

Instead of making the (highly sensitive) S_t itself private, we propose a new framework which makes private only the statistics of S_t: the mean μ_t and the top principal direction v_t. There are two versions of this algorithm, which output the exactly same $\hat{\mu}$ with the exactly same privacy guarantees, but are written from two different perspectives. We present here the *interactive* version from the perspective of an analyst accessing the dataset via DP queries ($q_{\mathrm{range}}, q_{\mathrm{size}}, q_{\mathrm{mean}}, q_{\mathrm{norm}}$ and q_{PCA}), because this version makes clear the inner operations of each private mechanisms, hence making (i) the sensitivity analysis transparent, (ii) checking the correctness of privacy guarantees easy, and (iii) tracking privacy accountant simple. In practice, one should implement the *centralized* version (Algorithm 7 in Appendix D), which is significantly more efficient.

Algorithm 1: Private iterative filtering (interactive version)

Input: range $[-R, R]^d$, $\alpha \in (0, 1/2)$, probability $\zeta \in (0, 1)$, # of iterations $T = \Theta(d)$, (ε, δ)

1 $(\bar{x}, B) \leftarrow q_{\mathrm{range}}(R, 0.01\varepsilon, 0.01\delta)$

2 $\varepsilon_1 \leftarrow \min\{0.99\varepsilon, 0.9\}/(4\sqrt{2T\log(2/\delta)}),\ \delta_1 \leftarrow 0.99\delta/(8T)$

3 **if** $n < (4/\varepsilon_1)\log(1/(2\delta_1))$ **then Output:** $\emptyset$

4 **for** $t = 1, \ldots, T$ **do**

5 $\quad n_t \leftarrow q_{\mathrm{size}}(\{(\mu_\ell, v_\ell, Z_\ell)\}_{\ell\in[t-1]}, \varepsilon_1, \bar{x}, B)$, **if** $n_t < 3n/4$ **then Output:** $\emptyset$

6 $\quad \mu_t \leftarrow q_{\mathrm{mean}}(\{(\mu_\ell, v_\ell, Z_\ell)\}_{\ell\in[t-1]}, \varepsilon_1, \bar{x}, B)$

7 $\quad \lambda_t \leftarrow q_{\mathrm{norm}}(\{(\mu_\ell, v_\ell, Z_\ell)\}_{\ell\in[t-1]}, \mu_t, \varepsilon_1, \bar{x}, B)$

8 $\quad$ **if** $\lambda_t \le (C - 0.01)\alpha\log 1/\alpha$ **then Output:** μ_t

9 $\quad v_t \leftarrow q_{\mathrm{PCA}}(\{(\mu_\ell, v_\ell, Z_\ell)\}_{\ell\in[t-1]}, \mu_t, \varepsilon_1, \delta_1, \bar{x}, B))$

10 $\quad Z_t \leftarrow \mathrm{Unif}([0, 1])$

Output: μ_t

We give a high-level explanation of each step of Algorithm 1 here and give the formal definitions of all the queries in Appendix D. First, q_{range} returns (the parameters of) a hypercube $\bar{x} + [-B/2, B/2]^d$ that is guaranteed to include all uncorrupted samples while preserving privacy. This is achieved by running d coordinate-wise private histograms and selecting $\bar{x}_j$ as the center of the largest bin for the j-th coordinate. Since covariance is $\mathbf{I}$, q_{range} returns a fixed $B = 8\sigma\sqrt{\log(dn/\zeta)}$. Such an adaptive estimate of the support is critical in tightly bounding the sensitivity of all subsequent queries, which operate on the clipped dataset; all data points are projected as $\mathcal{P}_{\bar{x}+[-B/2,B/2]^d}(x) = \arg\min_{y\in\bar{x}+[-B/2,B/2]^d} \|y - x\|_2$ in all the queries that follow. With clipping, a single data point can now change at most by $B\sqrt{d}$.

The subsequent steps perform the non-private filtering algorithm of §2.1, but with private statistics μ_t and v_t. As the set S_t changes over time, we lower bound its size (which we choose to be $|S_t| > n/2$) to upper bound the sensitivity of other queries $q_{\mathrm{mean}}, q_{\mathrm{norm}}$ and q_{PCA}.

At the t-th iterations, every time a query is called the data curator (i) uses $(\bar{x}, B)$ to clip the data, (ii) computes S_t by running $t-1$ steps of the non-private filtering algorithm of §2.1 but with a given *fixed* set of parameters $\{(\mu_\ell, v_\ell)\}_{\ell\in[t-1]}$ (and the given randomness $\{Z_\ell\}_{\ell\in[t-1]}$), and (iii) computes the queried private statistics of S_t. If the private spectral norm of the covariance of S_t (i.e., λ_t) is sufficiently small, we output the private and robust mean $\hat{\mu} = \mu_t$ (line 8). Otherwise, we compute the private top PCA direction v_t and draw an randomness Z_t to be used in the next step of filtering, as in the non-private filtering algorithm. We emphasize that $\{S_\ell\}$ are not private, and hence never returned to the analyst. We also note that this interactive version is redundant as every query is re-computing S_t. In our setting, the analyst has the dataset and there is no need to separate them. This leads to a

centralized version we provide in Algorithm 7 in the appendix, which avoids redundant computations and hence is significantly more efficient.

The main challenge in this framework is the privacy analysis. Because $\{S_\ell\}_{\ell\in[t-1]}$ is not private, each query runs $t-1$ steps of filtering whose end-to-end sensitivity could blow-up. Algorithmically, (i) we start with a specific choice of a non-private iterative filtering algorithm (among several variations that are equivalent in non-private setting but widely differ in its sensitivity), and (ii) make appropriate changes in the private queries (Algorithm 1) to keep the sensitivity small. Analytically, the following key technical lemma allows a sharp analysis of the end-to-end sensitivity of iterative filtering.

Lemma 2.2. *Let $S_t(\mathcal{S})$ denote the resulting subset of samples after t iterations of the filtering in the queries (q_{size}, q_{mean}, q_{norm}, and q_{PCA}) are applied to a dataset $\mathcal{S}$ using* fixed *parameters $\{(\mu_\ell, v_\ell, Z_\ell)\}_{\ell=1}^t$. Then, we have $d_\triangle(S_t(\mathcal{S}), S_t(\mathcal{S}')) \le d_\triangle(\mathcal{S},\mathcal{S}')$, where $d_\triangle(\mathcal{S},\mathcal{S}') \triangleq \max\{|\mathcal{S}\setminus\mathcal{S}'|, |\mathcal{S}'\setminus\mathcal{S}|\}$.*

Recall that two datasets are neighboring, i.e., $\mathcal{S}\sim\mathcal{S}'$, iff $d_\triangle(\mathcal{S},\mathcal{S}')\le 1$. This lemma implies that if two datasets are neighboring, then they are still neighboring after filtering with the same parameters, no matter how many times we filter them. Hence, this lemma allows us to use the standard output-perturbation mechanisms with (ε_1,δ_1)-DP. Advanced composition ensures that end-to-end guarantee of $4T$ such queries is $(0.99\varepsilon, 0.99\delta)$-DP. Together with $(0.01\varepsilon, 0.01\delta)$-DP budget used in q_{range}, this satisfied the target privacy. Analyzing the utility of this algorithm, we get the following guarantee.

Theorem 5. *Algorithm 1 is (ε,δ)-DP. Under Assumption 1, there exists a universal constant $c\in(0,0.1)$ such that if $\alpha\le c$ and $n=\widetilde{\Omega}\left((d/\alpha^2)+d^2(\log(1/\delta))^{3/2}/(\varepsilon\alpha)\right)$ then Algorithm 1 achieves $\|\hat{\mu}-\mu\|_2\le O(\alpha\sqrt{\log(1/\alpha)})$ with probability* 0.9.

The first term $O(d/\alpha^2)$ in the sample complexity is optimal (cf. Table 1), but there is a factor of d gap in the second term. This is due to the fact that we need to run $O(d)$ iterations in the worst-case. Such numerous accesses to the database result in large noise to be added at each iteration, requiring large sample size to combat that extra noise. We introduce PRIME to reduce the number of iterations to $O((\log d)^2)$ and significantly reduce the sample complexity.

2.3 PRIME: novel robust and private mean estimator

Algorithm 1 (specifically Filter($\cdot$) in Algorithm 1) accesses the database $O(d)$ times. This is necessary for two reasons. First, the filter checks only one direction v_t at each iteration. In the worst case, the corrupted samples can be scattered in $\Omega(d)$ orthogonal directions such that the filter needs to be repeated $O(d)$ times. Secondly, even if the corrupted samples are clustered together in one direction, the filter still needs to be repeated $O(d)$ times. This is because we had to use a large (random) threshold of $dB^2Z_t=O(d)$ to make the threshold data-independent so that we can keep the sensitivity of Filter($\cdot$) low, which results in slow progress. We propose filtering multiple directions simultaneously using a new score $\{\tau_i\}$ based on the matrix multiplicative weights. Central to this approach is a novel adaptive filtering algorithm DPTHRESHOLD($\cdot$) that guarantees sufficient decrease in the total score at every iteration.

2.3.1 Matrix Multiplicative Weight (MMW) scoring

The MMW-based approach, pioneered in [34] for non-private robust mean estimation, filters out multiple directions simultaneously. It runs over $O(\log d)$ epochs and every epoch consists of $O(\log d)$ iterations. At every epoch s and iteration t, step 2 of the iterative filtering in §2.1 is replaced by a new score $\tau_i=(x_i-\text{Mean}(S_t^{(s)}))^TU_t^{(s)}(x_i-\text{Mean}(S_t^{(s)}))$ where $U_t^{(s)}$ now accounts for all directions in $\mathbb{R}^d$ but appropriately weighted. Precisely, it is defined via the matrix multiplicative update:

$$U_t^{(s)} \;=\; \frac{\exp\left(\alpha^{(s)}\sum_{r\in[t]}(\text{Cov}(S_t^{(s)})-\mathbf{I})\right)}{\text{Tr}\left(\exp(\alpha^{(s)}\sum_{r\in[t]}(\text{Cov}(S_t^{(s)})-\mathbf{I}))\right)}\,,$$

for some choice of $\alpha^{(s)}>0$. If we set the number of iterations to one, a choice of $\alpha^{(s)}=\infty$ recovers the previous score that relied on the top singular vector from §2.1 and a choice of $\alpha^{(s)}=0$ gives a simple norm based score $\tau_i=\|x_i\|_2^2$. An appropriate choice of $\alpha^{(s)}$ smoothly interpolates between these two extremes, which ensures that $O(\log d)$ iterations are sufficient for the spectral norm of the

covariance to decrease strictly by a constant factor. This guarantees that after $O(\log d)$ epochs, we sufficiently decrease the covariance to ensure that the empirical mean is accurate enough. Critical in achieving this gain is our carefully designed filtering algorithm DPTHRESHOLD that uses the privately computed MMW-based scores using Gaussian mechanism on the covariance matrices as shown in Algorithm 11 in Appendix E.

2.3.2 Adaptive filtering with DPTHRESHOLD

Novelty. The corresponding non-private filtering of [34, Algorithm 9] for robust mean estimation takes advantage of an *adaptive threshold*, but filters out each sample independently resulting in a prohibitively large sensitivity; the coupling between each sample and the randomness used to filter it can change widely between two neighboring datasets. On the other hand, Algorithm 1 (i.e., Filter($\cdot$) in Algorithm 6) takes advantage of jointly filtering all points above a *single threshold* $B^2 d Z_t$ with a single randomness $Z_t \sim \mathrm{Unif}[0,1]$, but the non-adaptive (and hence large) choice of the range $B^2 d$ results in a large number of iterations because each filtering only decrease the score by little. To sufficiently reduce the total score while maintaining a small sensitivity, we introduce a filter with a single and adaptive threshold.

Algorithm. Our goal here is to privately find a single scalar ρ such that when a randomized filter is applied on the scores $\{\tau_i\}$ with a (random) threshold ρZ (with Z drawn uniform in $[0,1]$), we filter out enough samples to make progress in each iteration while ensuring that we do not remove too many uncorrupted samples. This is a slight generalization of the non-private algorithm in Section 2.1, which simply set $\rho = \max_{j \in S_t \tau_j}$. While this guarantees the filter removes more corrupted samples than good samples, it does not make sufficient progress in reducing the total score of the samples.

Ideally, we want the thresholding to decrease the total score by a constant multiplicative factor, which will in the end allow the algorithm to terminate within logarithmic iterations. To this end, we propose a new scheme of using the largest ρ such that the following inequality holds:

$$\sum_{\tau_i > \rho} (\tau_i - \rho) \geq 0.31 \sum_{\tau_i \in S_t} (\tau_i - 1) \ . \tag{1}$$

We use a private histogram of the scores to approximate this threshold. Similar to [53, 56], we use geometrically increasing bin sizes such that we use only $O(\log B^2 d)$ bins while achieving a preferred *multiplicative* error in our quantization. At each epoch s and iteration t, we run DPTHRESHOLD sketched in the following to approximate ρ followed by a random filter. Step 3 replaces the non-private condition in Eq. (1). A complete description is provided in Algorithm 11.

1. Privately compute scores for all data points $i \in S_t^{(s)}$: $\tau_i \leftarrow (x_i - \mu_t)^\top U_t^{(s)} (x_i - \mu_t)$;
2. Compute a private histogram $\{\tilde{h}_j\}_{j=1}^{2+\log(B^2 d)}$ of the scores over geometrically sized bins $I_1 = [1/4, 1/2), I_2 = [1/2, 1), \ldots, I_{2+\log(B^2 d)} = [2^{\log(B^2 d)-1}, 2^{\log(B^2 d)}]$;
3. Privately find the largest ℓ satisfying $\sum_{j \geq \ell} (2^j - 2^\ell)\, \tilde{h}_j \geq 0.31 \sum_{i \in S_t^{(s)}} (\tau_i - 1)$;
4. Output $\rho = 2^\ell$.

3 Analyses of PRIME

Building on the framework of Algorithm 1, PRIME (Algorithm 9) replaces the score with the MMW-based score presented in §2.3.1 and the filter with the adaptive DPTHRESHOLD. This reduces the number of iterations to $T = O((\log d)^2)$ achieving the following bound.

Theorem 6. *PRIME is (ε, δ)-differentially private. Under Assumption 1 there exists a universal constant $c \in (0, 0.1)$ such that if $\alpha \leq c$ and $n = \widetilde{\Omega}((d/\alpha^2) + (d^{3/2}/(\varepsilon\alpha)) \log(1/\delta))$, then PRIME achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha\sqrt{\log(1/\alpha)})$ with probability* 0.9.

A proof is provided in Appendix F. The notation $\widetilde{\Omega}(\cdot)$ hides logarithmic terms in d, R, and $1/\alpha$. To achieve an error of $O(\alpha\sqrt{\log(1/\alpha)})$, the first term $\widetilde{\Omega}(d/\alpha^2 \log(1/\alpha))$ is necessary even if there is no corruption. The accuracy of $O(\alpha\sqrt{\log(1/\alpha)})$ matches the lower bound shown in [31] for any polynomial time statistical query algorithm, and it nearly matches the information theoretical lower bound on robust estimation of $\Omega(\alpha)$. On the other hand, the second term of $\widetilde{\Omega}(d^{3/2}/(\varepsilon\alpha \log(1/\alpha)))$

has an extra factor of $d^{1/2}$ compared to the optimal one achieved by exponential time Algorithm 2. It is an open question if this gap can be closed by a polynomial time algorithm.

The bottleneck is the private matrix multiplicative weights. Such spectral analyses are crucial in filter-based robust estimators. Even for a special case of privately computing the top principal component, the best polynomial time algorithm requires $O(d^{3/2})$ samples [37, 17, 75], and this sample complexity is also necessary as shown in [37, Corollary 25].

To boost the success probability to $1-\zeta$ for some small $\zeta > 0$, we need an extra $\log(1/\zeta)$ factor in the sample complexity to make sure the dataset satisfies the regularity condition with probability $\zeta/2$. Then we can run PRIME $\log(1/\zeta)$ times and choose the output of a run that satisfies $n^{(s)} > n(1-10\alpha)$ and $\lambda^{(s)} \leq C\alpha\log(1/\alpha)$ at termination.

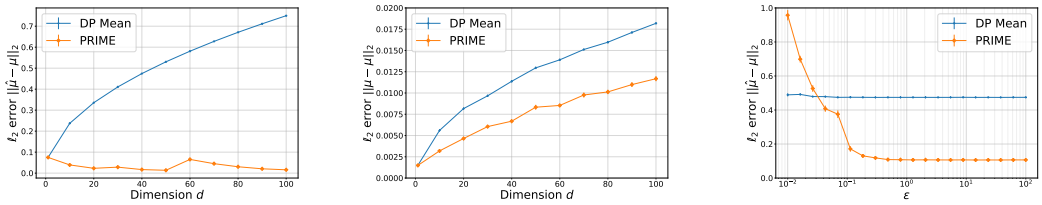

Figure 1: Private mean estimators (e.g., DP mean [50]) are vulnerable to adversarial corruption especially in high dimensions, while the proposed PRIME achieves robustness (and privacy) regardless of the dimension of the samples.

Numerical experiments support our theoretical claims. The left figure with $(\alpha, \varepsilon, \delta, n) = (0.05, 10, 0.01, 10^6)$ is in the large α regime where the DP Mean error is dominates by $\alpha\sqrt{d}$ and PRIME error by $\alpha\sqrt{\log(1/\alpha)}$. Hence, PRIME error is constant whereas DP Mean error increases with the dimension d. The second figure with $(\alpha, \varepsilon, \delta, n) = (0.001, 20, 0.01, 10^6)$ is in the small α regime when DP Mean error consists of $\alpha\sqrt{d} + \sqrt{d/n}$ and PRIME is dominated by $\sqrt{d/n}$. Both increase with the dimension d, and the gap can be made large by increasing α. The right figure with $(\alpha, \delta, d, n) = (0.1, 0.01, 10, 10^6)$ is when DP Mean error is dominated by $\alpha\sqrt{d}$ and PRIME by $\alpha\sqrt{\log(1/\alpha)}$ when $\varepsilon > cd^{1.5}/(\alpha n)$. Below this threshold, which happens in this example around $\varepsilon = 0.05$, the added noise in the private mechanism starts to dominate with decreasing ε. Both algorithms have respective thresholds below which the error increases with decreasing ε. This threshold is larger for PRIME because it uses the privacy budget to perform multiple operations and hence the noise added to the final output is larger compared to DP Mean. Below this threshold, which can be easily determined based on the known parameters $(\varepsilon, \delta, n, \alpha)$, we should either collect more data (which will decrease the threshold) or give up filtering and spend all privacy budget on q_{range} and the empirical mean (which will reduce the error). Details of the experiments are in Appendix J.

4 Exponential time algorithm with near-optimal sample complexity

Novelty. An existing exponential time algorithm for robust and private mean estimation in [12] strictly requires the uncorrupted samples to be drawn from a Gaussian distribution. We introduce a novel estimator that achieves near-optimal guarantees for more general sub-Gaussian distributions (and also covariance bounded distributions) but takes an *exponential* run-time. Its innovation is in leveraging on the *resilience* property of well-behaved distributions not only to estimate the mean robustly (which is the standard use of the property) but also to adaptively bound the sensitivity of the estimator, thus achieving optimal privacy-accuracy tradeoff.

Definition 4.1 (Resilience from Definition 1 in [69]). *A set of points $\{x_i\}_{i\in S}$ lying in $\mathbb{R}^d$ is (σ, α)-resilient around a point μ if $\|(1/|T|)\sum_{i\in T}(x_i - \mu)\|_2 \leq \sigma$ for all subsets $T \subset S$ of size $(1-\alpha)|S|$.*

Algorithm. As data is corrupted, we define $R(S)$ as a surrogate for resilience of the uncorrupted part of the set. If S indeed consists of a $1-\alpha$ fraction of independent samples from the promised class of distributions, the goodness score $R(S)$ will be close to the resilience property of the good data.

Definition 4.2 (Goodness of a set). *For $\mu(S) = (1/|S|)\sum_{i\in S} x_i$, let us define*

$$R(S) \quad \triangleq \quad \min_{S'\subset S, |S'|=(1-2\alpha)|S|.} \quad \max_{T\subset S', |T|=(1-\alpha)|S'|.} \|\mu(T) - \mu(S')\|_2 \ .$$

Algorithm 2 first checks if the resilience matches that of the promised distribution. The data is pre-processed with q_{range} to ensure we can check $R(S)$ privately. Once resilience is cleared, we can safely use the exponential mechanism based on the score function $d(\hat{\mu}, S)$ in Definition 4.3 to select an approximate robust mean $\hat{\mu}$ privately. The choice of the sensitivity critically relies on the fact that resilient datasets have small sensitivity of $O((1/n)\sqrt{\log(1/\alpha)})$. Without the resilience check, the sensitivity is $O(d^{1/2}/n)$ resulting in an extra factor of $\sqrt{d}$ in the sample complexity.

Algorithm 2: Exponential-time private and robust mean estimation

Input: $S = \{x_i\}_{i\in[n]}$, $\alpha \in (0, 1/2)$, R, (ε, δ)

if $n < cd^{1/2}\log(1/\delta)/\left(\varepsilon\alpha\sqrt{\log(1/\alpha)}\right)$ **then Output:** $\emptyset$

$(\bar{x}, B) \leftarrow q_{\text{range}}(R, (1/3)\varepsilon, (1/3)\delta)$ [$q_{\text{range-ht}}(\cdot)$ for hevay-tail]

Project the data points onto the ball: $x_i \leftarrow \mathcal{P}_{\mathcal{B}_{\sqrt{d}B/2}(\bar{x})}(x_i)$, for all $i \in [n]$

$\widehat{R}(S) \leftarrow R(S) + \text{Lap}(3Bd^{1/2}/(n\varepsilon))$

if $\widehat{R}(S) > 2\,\alpha\sqrt{\log(1/\alpha)}$ **then Output:** $\emptyset$ [$\widehat{R}(S) > 2c_\zeta\sqrt{\alpha}$ for hevay-tail]

else Output: a randomly drawn point $\hat{\mu} \in [-2R, 2R]^d$ sampled from a density

$r(\hat{\mu}) \propto e^{-(1/(24\sqrt{\log(1/\alpha)}))\varepsilon\, n\, d(\hat{\mu},S)}$ [$e^{-(\varepsilon n\sqrt{\alpha}/(24c_\zeta))d(\hat{\mu},S)}$ for heavy-tail]

We propose the score function $d(\hat{\mu}, S)$ in the following definition, which is a robust estimator of the distance between the mean and the candidate $\hat{\mu}$.

Definition 4.3. *For a set of data $\{x_i\}_{i\in S}$ lying in $\mathbb{R}^d$, for any $v \in \mathbb{S}^{d-1}$, define $\mathcal{T}^v$ to be the $3\alpha|S|$ points with the largest $v^\top x_i$ value, $\mathcal{B}^v$ to be the $3\alpha|S|$ points with the smallest $v^\top x_i$ value, and $\mathcal{M}^v = S \setminus (\mathcal{T}^v \cup \mathcal{B}^v)$. Define $d(\hat{\mu}, S) \triangleq \max_{v\in\mathbb{S}^{d-1}} \left|v^\top\left(\mu(\mathcal{M}^v) - \hat{\mu}\right)\right|$.*

Analysis. For any direction v, the truncated mean estimator $\mu(\mathcal{M}^v)$ provides a robust estimation of the true mean along the direction v, thus the distance can be simply defined by taking the maximum over all directions v. We show the sensitivity of this simple estimator is bounded by the resilience property σ divided by n, which is $O((1/n)\sqrt{\log(1/\alpha)})$ once the resilience check is passed. This leads to the following near-optimal sample complexity. We provide a proof in §H.2.

Theorem 7 (Exponential time algorithm for sub-Gaussian distributions)**.** *Algorithm 2 is (ε, δ)-DP. Under Assumption 1, this algorithm achieves $\|\hat{\mu} - \mu\|_2 = O(\alpha\sqrt{\log(1/\alpha)})$ with probability $1 - \zeta$ if*

$$n = \widetilde{\Omega}\Big(\frac{d + \log\frac{1}{\zeta}}{\alpha^2\log\frac{1}{\alpha}} + \frac{d\log\frac{dR}{\alpha} + d^{1/2}\log\frac{1}{\delta} + \log\frac{1}{\zeta}}{\varepsilon\alpha} + \frac{\sqrt{d\log\frac{1}{\delta}}\min\{\log\frac{dR}{\zeta}, \log\frac{d}{\zeta\delta}\}}{\varepsilon}\Big)\,.$$

Run-time. Computing $R(S)$ exactly can take $O(de^{\Theta(n)})$ operations. The exponential mechanism implemented with α-covering for $\hat{\mu}$ and a constant covering for v can take $O(nd(R/\alpha)^d)$ operations.

5 Conclusion

Differentially private mean estimation is brittle against a small fraction of the samples being corrupted by an adversary. We show that robustness can be achieves without any increase in the sample complexity by introducing a novel DP mean estimator, which requires run-time exponential in the dimension of the samples. The technical contribution is in leveraging the resilience property of well-behaved distributions in an innovative way to not only find robust mean (which is the typical use case of resilience) but also bound sensitivity for optimal privacy guarantee. To cope with the computational challenge, we propose an efficient algorithm, which we call PRIME, that achieves the optimal target accuracy at the cost of an increased sample complexity. The technical contributions are (i) a novel framework for private iterative filtering and its tight analysis of the end-to-end sensitivity and (ii) novel filtering algorithm of DPTHRESHOLD which is critical in privately running matrix multiplicative weights and hence significantly reducing the number of accesses to the database. With appropriately chosen parameters, we show that our exponential time approach achieves near-optimal guarantees for both sub-Gaussian and covariance bounded distributions and PRIME achieves the same accuracy efficiently but at the cost of an increased sample complexity by a $d^{1/2}$ factor.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

[3] Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. On the sample complexity of privately learning unbounded high-dimensional gaussians. *arXiv preprint arXiv:2010.09929*, 2020.

[4] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 237–245, 2015.

[5] Frank J Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960.

[6] Ainesh Bakshi and Pravesh Kothari. List-decodable subspace recovery via sum-of-squares. *arXiv preprint arXiv:2002.05139*, 2020.

[7] S. Balakrishnan, S. S. Du, J. Li, and A. Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*, pages 169–212, 2017.

[8] K. Bhatia, P. Jain, P. Kamalaruban, and P. Kar. Consistent robust regression. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 2107–2116, 2017.

[9] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems*, pages 721–729, 2015.

[10] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. *arXiv preprint arXiv:2006.06618*, 2020.

[11] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

[12] Mark Bun, Gautam Kamath, Thomas Steinke, and Steven Z Wu. Private hypothesis selection. In *Advances in Neural Information Processing Systems*, pages 156–167, 2019.

[13] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *ITCS*, pages 369–380, 2016.

[14] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

[15] Clément L Canonne, Gautam Kamath, Audra McMillan, Jonathan Ullman, and Lydia Zakynthinou. Private identity testing for high-dimensional distributions. *arXiv preprint arXiv:1905.11947*, 2019.

[16] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60, 2017.

[17] Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *The Journal of Machine Learning Research*, 14(1):2905–2943, 2013.

[18] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[19] Yu Cheng, Ilias Diakonikolas, and Rong Ge. High-dimensional robust mean estimation in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2755–2771. SIAM, 2019.

[20] Yu Cheng, Ilias Diakonikolas, Rong Ge, and David P Woodruff. Faster algorithms for high-dimensional robust covariance estimation. In *Conference on Learning Theory*, pages 727–757. PMLR, 2019.

[21] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. *arXiv preprint arXiv:2005.09796*, 2020.

[22] Arnak Dalalyan and Philip Thompson. Outlier-robust estimation of a sparse linear model using ℓ_1-penalized huber's m-estimator. In *Advances in Neural Information Processing Systems*, pages 13188–13198, 2019.

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

[24] Aditya Dhar and Jason Huang. Designing differentially private estimators in high dimensions. *arXiv preprint arXiv:2006.01944*, 2020.

[25] Ilias Diakonikolas, Samuel B Hopkins, Daniel Kane, and Sushrut Karmalkar. Robustly learning any clusterable mixture of gaussians. *arXiv preprint arXiv:2005.06417*, 2020.

[26] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

[27] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606, 2019.

[28] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being Robust (in High Dimensions) Can Be Practical. *arXiv e-prints*, page arXiv:1703.00893, March 2017.

[29] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2683–2702. SIAM, 2018.

[30] Ilias Diakonikolas and Daniel M Kane. Recent advances in algorithmic high-dimensional robust statistics. *arXiv preprint arXiv:1911.05911*, 2019.

[31] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017.

[32] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1060, 2018.

[33] Ilias Diakonikolas, Weihao Kong, and Alistair Stewart. Efficient algorithms and lower bounds for robust linear regression. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2745–2754. SIAM, 2019.

[34] Yihe Dong, Samuel Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. In *Advances in Neural Information Processing Systems*, pages 6067–6077, 2019.

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

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

[37] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014.

[38] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[39] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3):41–61, 2016.

[40] Chao Gao et al. Robust regression via mutivariate regression depth. *Bernoulli*, 26(2):1139–1170, 2020.

[41] Sam Hopkins, Jerry Li, and Fred Zhang. Robust and heavy-tailed mean estimation made simple, via regret minimization. *Advances in Neural Information Processing Systems*, 33, 2020.

[42] Samuel B Hopkins. Mean estimation with sub-gaussian rates in polynomial time. *Annals of Statistics*, 48(2):1193–1213, 2020.

[43] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034, 2018.

[44] Samuel B Hopkins and Jerry Li. How hard is robust mean estimation? In *Conference on Learning Theory*, pages 1649–1682. PMLR, 2019.

[45] Peter J Huber. Robust estimation of a location parameter: Annals mathematics statistics, 35. 1964.

[46] Arun Jambulapati, Jerry Li, and Kevin Tian. Robust sub-gaussian principal component analysis and width-independent schatten packing. *Advances in Neural Information Processing Systems*, 33, 2020.

[47] He Jia and Santosh Vempala. Robustly clustering a mixture of gaussians. *arXiv preprint arXiv:1911.11838*, 2019.

[48] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[49] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385, 2015.

[50] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory*, pages 1853–1902, 2019.

[51] Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan Ullman. Differentially private algorithms for learning mixtures of separated gaussians. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–62. IEEE, 2020.

[52] Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. Private mean estimation of heavy-tailed distributions. *arXiv preprint arXiv:2002.09464*, 2020.

[53] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *Conference on Learning Theory*, pages 2263–2285. PMLR, 2020.

[54] Sushrut Karmalkar, Adam Klivans, and Pravesh Kothari. List-decodable linear regression. In *Advances in Neural Information Processing Systems*, pages 7423–7432, 2019.

[55] Sushrut Karmalkar and Eric Price. Compressed sensing with adversarial sparse noise via l1 regression. In *2nd Symposium on Simplicity in Algorithms*, 2019.

[56] Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.

[57] Adam Klivans, Pravesh K Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*, pages 1420–1430, 2018.

[58] Weihao Kong, Raghav Somani, Sham Kakade, and Sewoong Oh. Robust meta-learning for mixed linear regression with small batches. *Advances in Neural Information Processing Systems*, 33, 2020.

[59] Pravesh K Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046, 2018.

[60] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.

[61] Jerry Li. CSE 599-M, Lecture Notes: Robustness in Machine Learning , 2019. URL: `https://jerryzli.github.io/robust-ml-fall19/lec7.pdf`.

[62] Jerry Li and Guanghao Ye. Robust gaussian covariance estimation in nearly-matrix multiplication time. *Advances in Neural Information Processing Systems*, 33, 2020.

[63] Liu Liu, Yanyao Shen, Tianyang Li, and Constantine Caramanis. High dimensional robust sparse regression. *arXiv preprint arXiv:1805.11643*, 2018.

[64] Gábor Lugosi, Shahar Mendelson, et al. Sub-gaussian estimators of the mean of a random vector. *Annals of Statistics*, 47(2):783–794, 2019.

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

[66] Bhaskar Mukhoty, Govind Gopakumar, Prateek Jain, and Purushottam Kar. Globally-convergent iteratively reweighted least squares for robust regression problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 313–322, 2019.

[67] A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[68] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180. SIAM, 2020.

[69] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[70] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.

[71] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

[72] John W Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

[73] Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

[74] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[75] Lu Wei, Anand D Sarwate, Jukka Corander, Alfred Hero, and Vahid Tarokh. Analysis of a privacy-preserving pca algorithm using random matrix theory. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1335–1339. IEEE, 2016.

[76] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pages 1689–1698. PMLR, 2015.

[77] Huanyu Zhang, Gautam Kamath, Janardhan Kulkarni, and Zhiwei Steven Wu. Privately learning markov random fields. *arXiv preprint arXiv:2002.09463*, 2020.

[78] Banghua Zhu, Jiantao Jiao, and Jacob Steinhardt. Generalized resilience and robust statistics. *arXiv preprint arXiv:1909.08755*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Figure 1 about the small ε regime where our analysis do not apply because we don't have enough samples. Also see Table 1 and Table 2 for the $d^{0.5}$ gap in the sample complexity for efficient algorithms.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work is theoretical and does not have a direct negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] We don't have training.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix J.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] We didn't use existing assets.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]