PLAN: Variance-Aware Private Mean Estimation

Martin Aumüller
IT University of Copenhagen
Copenhagen, Denmark
maau@itu.dk

Christian Janos Lebeda
Basic Algorithms Research Copenhagen
IT University of Copenhagen
Copenhagen, Denmark
christian.j.lebeda@gmail.com

Boel Nelson
Aarhus University
Aarhus, Denmark
boel@cs.au.dk

Rasmus Pagh
Basic Algorithms Research Copenhagen
University of Copenhagen
Copenhagen, Denmark
pagh@di.ku.dk

ABSTRACT

Differentially private mean estimation is an important building block in privacy-preserving algorithms for data analysis and machine learning. Though the trade-off between privacy and utility is well understood in the worst case, many datasets exhibit structure that could potentially be exploited to yield better algorithms. In this paper we present *Private Limit Adapted Noise (PLAN)*, a family of differentially private algorithms for mean estimation in the setting where inputs are independently sampled from a distribution $\mathcal{D}$ over $\mathbf{R}^d$, with coordinate-wise standard deviations $\boldsymbol{\sigma} \in \mathbf{R}^d$. Similar to mean estimation under Mahalanobis distance, PLAN tailors the shape of the noise to the shape of the data, but unlike previous algorithms the privacy budget is spent non-uniformly over the coordinates. Under a concentration assumption on $\mathcal{D}$, we show how to exploit skew in the vector $\boldsymbol{\sigma}$, obtaining a (zero-concentrated) differentially private mean estimate with ℓ_2 error proportional to $\|\boldsymbol{\sigma}\|_1$. Previous work has either not taken $\boldsymbol{\sigma}$ into account, or measured error in Mahalanobis distance — in both cases resulting in ℓ_2 error proportional to $\sqrt{d}\|\boldsymbol{\sigma}\|_2$, which can be up to a factor $\sqrt{d}$ larger. To verify the effectiveness of PLAN, we empirically evaluate accuracy on both synthetic and real-world data.

KEYWORDS

differential privacy, mean estimation

1 INTRODUCTION

Differentially private mean estimation is an important building block in many algorithms, notably in for example implementations of private stochastic gradient descent [1, 16, 36]. While differential privacy is an effective and popular definition for privacy-preserving data processing, privacy comes at a cost of accuracy, and striking a good trade-off between utility and privacy can be challenging. Achieving a good trade-off is especially hard for high-dimensional data, as the required *noise* that ensures privacy increases with the number of dimensions. Making matters worse, differential privacy operates on a worst-case basis. Adding noise naïvely may result in a *one-size-fits-none* noise scale that, although private, fails to give meaningful utility for many datasets.

Motivating example. Consider the following toy example: take a constant $q \in (0,1)$ and let $\lambda \ll 1/d$ be a value close to zero. We consider a set of vectors $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbf{R}^d$ where, independently,

$$\boldsymbol{x}^{(j)} = \begin{cases} (1, \lambda, \ldots, \lambda) & \text{with probability } q \\ (0, 0, \ldots, 0) & \text{with probability } 1-q \end{cases} .$$

Though the nominal dimension is d, the distribution is "essentially 1-dimensional". Thus, we should be able to release a private estimate of the mean $(q, \lambda q, \ldots, \lambda q)$ with about the same precision as a scalar value with sensitivity 1. However, previous private mean estimation techniques either:

- add the same amount of noise to all d dimensions, making the ℓ_2 norm of the error a factor $\Theta(\sqrt{d})$ larger, or
- split the privacy budget evenly across the d dimensions, making the noise added to the first coordinate a factor $\Theta(\sqrt{d})$ larger than in the scalar case.

Instead, we would like to adapt to the situation, and spend most of the privacy budget on the first coordinate while still keeping the noise on the remaining $d-1$ coordinates low. Dimension reduction using private PCA [3, 21, 22] could be used, but we aim for something simpler and more general: to spend more budget on coordinates with higher variance.

We explore the design space of variance-aware budget allocations and show that, perhaps surprisingly, the intuitive approach of splitting the budget across coordinates proportional to their spread is *not optimal* for ℓ_2 error, and ℓ_p error in more generality. This is in stark contrast to the case of measuring error in Mahalanobis distance for which this allocation is optimal [9]. Another surprise is that we achieve better bounds than the *instance-optimal* algorithm proposed by Huang et al. in [24] in the case where variances *are skewed*. Using the idea of variance-aware budgeting, we design a family of algorithms, Private Limit Adapted Noise (PLAN), that distributes the privacy budget optimally. For the analysis, we assume that an estimate on the variances is known. Since this is often unrealistic in a practical setting, we propose and experimentally evaluate different methods to obtain a differentially private estimate on the variances. Even using such rough estimates, PLAN is competitive to, or better than the state-of-the-art solutions in

Proceedings on Privacy Enhancing Technologies 2024(3), 606–625

https://doi.org/10.56553/popets-2024-0095

practice. The contributions of this work cover both theoretical and practical ground:

- We present Private Limit Adapted Noise (PLAN) (Section 3), a family of algorithms for differentially private mean estimation for ℓ_2 error, and its accompanying privacy and utility analysis (Section 4).
- We formalize, introduce, and exemplify the notion of $\boldsymbol{\sigma}$-*well concentrated* distributions (Section 4 & Section 5) that allow for the study of variance-aware algorithms for mean estimation.
- We generalize the analysis of PLAN to hold for arbitrary ℓ_p error (Section 4.5).
- We prove that PLAN matches the utility of the current state-of-the-art algorithm [24] for differentially private mean estimation up to constant factors in expectation, even if no estimate on the variances is known or if there are correlations in the data. This makes PLAN a suitable replacement for mean estimation in general settings without requiring computationally expensive random rotations (Section 6).
- We implement two instantiations of PLAN, for ℓ_1 and ℓ_2 error respectively, and empirically evaluate the algorithms on synthetic and real-world datasets (Section 7), proposing and evaluating several techniques for differentially private variance estimation (Appendix D).

Scope. We consider the setting where we have independent samples $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ from a distribution $\mathcal{D}$ over $\mathbf{R}^d$, and use $\boldsymbol{\sigma} \in \mathbf{R}^d$ to denote the vector where $\boldsymbol{\sigma}_i$ is the standard deviation of the ith coordinate of a sample from $\mathcal{D}$. Our objective is to estimate the mean $\boldsymbol{\mu}$ of $\mathcal{D}$ with small ℓ_p error. This is a natural error measure in many settings. For example, $p = 1$ is analogous to the mean absolute error, while $p = 2$ represents the mean squared error. Other interesting applications arise if input vectors:

- represent probability distributions over $\{1, \ldots, d\}$, so that ℓ_1 error corresponds to variation distance, or
- are binary and model d counting queries, so that small ℓ_1 and ℓ_2 error corresponds to small mean error and root mean square error, respectively.

Our exposition will focus primarily on the case of ℓ_2 error, which is most easily compared to previous work. We remark that in the special case of $\mathcal{D}$ being a Gaussian distribution, *Mahalanobis distance* (Section 2.2) is a natural, well-understood [9] error measure.

PLAN's privacy guarantees are expressed as ρ-zero-Concentrated Differential Privacy (ρ-zCDP), where similar results follow for other notions, such as approximate differential privacy (Lemma 2.3). We consider the number of inputs, n, fixed — the neighboring relation is changing one vector among the inputs.

In the *non-private* setting, the empirical mean $\frac{1}{n}\sum_j \boldsymbol{x}^{(j)}$ is known to yield the smallest ℓ_2 error of size $O(\|\boldsymbol{\sigma}\|_2/\sqrt{n})$, which is also called the *sampling error*. Since this sampling error is unavoidable, we are interested in the setting where the error introduced by differential privacy is larger than the sampling error.

Our results. We show, both theoretically and empirically, that a carefully chosen privacy budgeting can improve the ℓ_p error compared to existing methods when the vector $\boldsymbol{\sigma}$ is skewed. First, we provide a characterization of distributions $\mathcal{D}$ that are suitable for our budget allocation. We say that $\mathcal{D}$ is $\boldsymbol{\sigma}$-*well concentrated* if, roughly speaking, the norm of distance to the mean $\boldsymbol{\mu}$ of vectors sampled from $\mathcal{D}$ is unlikely to be much larger than vectors sampled from a multivariate exponential distribution with standard deviations given by $\boldsymbol{\sigma}$ or some root of $\boldsymbol{\sigma}$. In the case $p = 2$ our upper bound is particularly simple to state:

Theorem 1.1. *(simplified version) Suppose $\mathcal{D}$ is $\boldsymbol{\sigma}$-well concentrated and that we know $\hat{\boldsymbol{\sigma}}$ such that $\|\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}\|_\infty < \|\boldsymbol{\sigma}\|_1/d$. Then for $n = \tilde{\Omega}\left(\max\left(\sqrt{d/\rho}, \rho^{-1}\right)\right)$ there is a mean estimation algorithm that satisfies ρ-zCDP and has expected ℓ_2 error*

$$\tilde{O}(1 + \|\boldsymbol{\sigma}\|_2/\sqrt{n} + \|\boldsymbol{\sigma}\|_1/(n\sqrt{\rho})),$$

where $\tilde{O}$ suppresses polylogarithmic factors in d, n, and a bound on the ℓ_∞ norm of input vectors.

Comparing Theorem 1.1 to applying the Gaussian mechanism directly provides a useful example. Given the vectors $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ sampled from $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ for $\Sigma = \text{diag}(\boldsymbol{\sigma}^2)$ where $\|\boldsymbol{\mu}\|_2 = O(\|\boldsymbol{\sigma}\|_2)$, and clipping at $C = \Theta(\|\boldsymbol{\sigma}\|_2)$, would give an ℓ_2 error bound of

$$\tilde{O}(\|\boldsymbol{\sigma}\|_2/\sqrt{n} + \sqrt{d}\|\boldsymbol{\sigma}\|_2/(n\sqrt{\rho})) \ .$$

Crucially, Theorem 1.1 is never worse, but can be better than applying the Gaussian mechanism directly. Instance-optimal mean estimation [24] matches the bound achieved by the Gaussian mechanism in this case since the diameter of the dataset is $\Theta(\|\boldsymbol{\sigma}\|_2)$. The value of $\|\boldsymbol{\sigma}\|_1$ is in the interval $[\|\boldsymbol{\sigma}\|_2, \sqrt{d}\|\boldsymbol{\sigma}\|_2]$, where a smaller value of $\|\boldsymbol{\sigma}\|_1$ indicates larger skew. Thus, Theorem 1.1 is strongest when $\boldsymbol{\sigma}$ has a skewed distribution such that $\|\boldsymbol{\sigma}\|_1$ is close to the lower bound of $\|\boldsymbol{\sigma}\|_2$, assuming that the privacy parameter ρ is small enough that the error due to privacy dominates the sampling error.

When no estimate on the variance is known, PLAN still achieves compelling worst-case guarantees on the ℓ_2 error. Theorem 6.1 showcases that skipping the scaling step, we recover the worst-case guarantees of instance-optimal mean estimating [24], without the need for an expensive random rotation of the input data. Our experimental evaluation in Section 7 provides further support that even rough estimates on the variances suffice to achieve an advantage over state-of-the-art competitors, even if data is correlated. This makes PLAN a drop-in replacement for applications using differentially private statistical mean estimation.

2 PRELIMINARIES

We provide privacy guarantees via zero-Concentrated Differential Privacy (zCDP) in the *bounded setting*, where the dataset has a fixed size, as defined in this section. Since previous work measure error using different distance measures, we give both the definition for ℓ_p error (which we use), as well as Mahalanobis distance. Lastly, we give an overview of private quantile estimation which is a central building block of our algorithm.

2.1 Differential privacy

Differential privacy [20] is a statistical property of an algorithm that limits information leakage by introducing controlled randomness to the algorithm. Formally, differential privacy restricts how much the output distributions can differ between any neighboring datasets.

We say that a pair of datasets are neighboring, denoted $x \sim x'$, if and only if there exists an $j \in [n]$ such that $\boldsymbol{x}^{(i)} = \boldsymbol{x}'^{(i)}$ for all $i \neq j$.

Definition 2.1 ([20] (ε, δ)-Differential Privacy)**.** A randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if and only if for all pairs of neighboring datasets $x \sim x'$ and all set of outputs Z we have

$$\Pr[\mathcal{M}(x) \in Z] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in Z] + \delta$$

Definition 2.2 ([13] zero-Concentrated Differential Privacy (zCDP))**.** Let $\mathcal{M}$ denote a randomized mechanism satisfying ρ-zCDP for any $\rho > 0$. Then for all $\alpha > 1$ and all pairs of neighboring datasets $x \sim x'$ we have

$$D_\alpha\left(\mathcal{M}(x) || \mathcal{M}(x')\right) \leq \rho\alpha,$$

where $D_\alpha(X||Y)$ denotes the α-Rényi divergence between two distributions X and Y.

Lemma 2.3 ([13] zCDP to (ε, δ)-DP conversion)**.** *If $\mathcal{M}$ satisfies ρ-zCDP, then $\mathcal{M}$ is (ε, δ)-DP for any $\delta > 0$ and $\varepsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$.*

Lemma 2.4 ([13] Composition)**.** *If M_1 and M_2 satisfy ρ_1-zCDP and ρ_2-zCDP, respectively. Then $M = (M_1, M_2)$ satisfies $(\rho_1 + \rho_2)$-zCDP.*

The Gaussian Mechanism adds noise from a Gaussian distribution independently to each coordinate of a real-valued query output. The scale of the noise depends on the privacy parameter and the ℓ_2-sensitivity denoted Δ_2. A query q has sensitivity Δ_2 if for all $x \sim x'$ we have $\|q(x) - q(x')\|_2 \leq \Delta_2$.

Lemma 2.5 ([13] The Gaussian Mechanism)**.** *If $q : \mathbf{R}^{n \times d} \to \mathbf{R}^d$ is a query with sensitivity Δ_2 then releasing $\mathcal{N}(q(x), \frac{\Delta_2^2}{2\rho}\mathbf{I})$ satisfies ρ-zCDP.*

2.2 Distance measures

Here we introduce the distance measures for the error of a mean estimate. Let $\tilde{\boldsymbol{\mu}} = \mathcal{M}(x)$ denote the mean estimate of a distribution with mean $\boldsymbol{\mu}$. We measure the utility of our mechanism in terms of expected ℓ_p error as defined below. Here p is a parameter of our mechanism where the most common values for p are 1 (Manhattan distance), and 2 (Euclidean distance).

Definition 2.6 (ℓ_p error)**.** For any real value $p \geq 1$ the ℓ_p error is

$$\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_p = \left(\sum_{i=1}^{d} |\tilde{\mu}_i - \mu_i|^p\right)^{1/p}$$

Note that we sometimes present error guarantees in the form of the pth moment, i.e. $\mathbf{E}[\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_p^p]$, when it follows naturally from the analysis. Notice that the pth moment bounds the expected ℓ_p error for any $p \geq 1$ as $\mathbf{E}[\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_p] \leq (\mathbf{E}[\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_p^p])^{1/p}$.

As stated in Section 8 several previous work on mean estimates measure error in Mahalanobis distance. Although this is not the focus of our work we include the definition here for completeness. A key difference between Mahalanobis distance measure and ℓ_p error is that the directions are weighted by the uncertainty of the underlying data. We weight the error in all coordinates equally.

Definition 2.7 (Mahalanobis distance)**.** The error in Mahalanobis distance of a mean estimate for a distribution with covariance matrix Σ is defined as

$$\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_\Sigma = \|\Sigma^{-1/2}(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})\|_2$$

2.3 Private quantile estimation

Our work builds on using differentially private quantiles, whose rank error (i.e. how many elements away from the desired quantile the output is) we will define for our utility analysis. When estimating the q'th quantiles of a sequence $\boldsymbol{z}^{(1)} \dots \boldsymbol{z}^{(n)}$ with $\boldsymbol{z}^{(j)} \in \mathbf{R}^d$, we ideally return a vector $\tilde{\boldsymbol{z}} \in \mathbf{R}^d$ such that for each coordinate i we have $|\{j \in [n] : z_i^{(j)} \leq \tilde{z}_i\}|/n = q$. We use the notation $\textsc{PrivQuantile}_\rho^M(\boldsymbol{z}^{(1)} \dots \boldsymbol{z}^{(n)}, q)$ to denote a ρ-zCDP mechanism that estimates the q'th quantiles of $\boldsymbol{z}^{(1)} \dots \boldsymbol{z}^{(n)}$ where each coordinate is bounded to $[-M, M]$.

There are multiple applicable choices for the instantiation of PrivQuantile from the literature, e.g. [24, 30, 38]. In this paper we will use the binary search based quantile [24, 38] for our utility analysis (Section 4), and the exponential mechanism based quantile by Kaplan et al. [30] in our empirical evaluation (Section 7). Note that the choice of instantiation of PrivQuantile has no effect on the privacy analysis of plan. The binary search based method gives us cleaner theoretical results because the error guarantee of the exponential mechanism based technique is highly data-dependent, and the error is large for worst-case input. Empirically, however, the exponential mechanism based quantile is more robust, which is why we use it in our experiments.

The binary search based quantile subroutine performs a binary search over the interval $[-M, M]$. At each step of the search, branching is based on a quantile estimate for the midpoint of the current interval. Since branching must be performed privately the privacy budget needs to be partitioned in advance, this limits the binary search to T iterations. Huang et al. [24] describes the algorithm for discrete input where T is the base 2 logarithm of the size of the input domain. Treating T as a parameter of the algorithm gives us the following guarantees for 1-dimensional input.

Lemma 2.8 (Follows from Huang et al. [24, Theorem 1])**.** *PrivQuantile satisfies ρ'-zCDP and with probability at least $1 - \beta$ it returns an interval containing at least one point with rank error bounded by $\sqrt{T \log(T/\beta)/(2\rho')}$.*

When estimating the quantiles of multiple dimensions, we split the privacy budget evenly across each dimension such that $\rho' = \rho/d$ for each invocation of PrivQuantile. By composition releasing all quantiles satisfies ρ-zCDP.

Lemma 2.9. *With probability at least $1 - \beta$, $\textsc{PrivQuantile}_\rho^M$ returns a point that for all coordinates is within a distance of $M2^{-T}$ of a point with rank error at most $\sqrt{dT \log(Td/\beta)/2\rho}$ for the desired quantile.*

Proof. Running binary search for T iterations splits up the range $[-M, M]$ in 2^T evenly sized intervals. By a union bound there is a point with claimed rank error in each of the intervals returned by PrivQuantile. Returning the midpoint of each interval ensures we are at most distance $M2^{-T}$ from said point. □

Throughout this paper, we set $T = \log_2(M)$ unless otherwise specified such that the error distance of Lemma 2.9 is 1.

3 ALGORITHM

Conceptually, Private Limit Adapted Noise (PLAN) is a data-aware family of algorithms that exploits variance in the input data to tailor noise to the specific data at hand. Since it is a family of algorithms, a PLAN needs to be instantiated for a given problem domain, i.e. for a specific ℓ_p error and data distribution. In Section 7 we showcase two such instantiations, using ℓ_1 error for binary data, and using ℓ_2 for Gaussian data. Pseudocode for PLAN is shown in Algorithm 1, and a step-by-step illustration of PLAN is provided in Figure 1.

3.1 PLAN overview

Based on the input data (Line 1, Figure 1 (a)) PLAN first computes a private, rough, approximate mean $\tilde{\boldsymbol{\mu}}$ (Line 4). PLAN then recenters the data around $\tilde{\boldsymbol{\mu}}$, and scales the data (Line 5, Figure 1 (b)) according to an estimate on the variance $\hat{\boldsymbol{\sigma}}^2$: this scaling allows the privacy budget to be spent unevenly across dimensions. Next, PLAN privately estimates a clipping threshold that contains most of the scaled data points (Line 6) and samples sufficient Gaussian noise for the privacy budget (Line 7). Using the clipping threshold and the noise vector, Line 8 in PLAN clips the scaled input centered around $\tilde{\boldsymbol{\mu}}$ (Figure 1 (c)), adds noise to the clipped data, and finally inverts the scaling and the recentering (Figure 1 (d)). Each of these techniques has been used for private mean estimation before; we show that scaling data differently can lead to smaller ℓ_p error.

Algorithm 1 PLAN

1: **Input:** $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbf{R}^d$, estimate $\hat{\boldsymbol{\sigma}}^2 \in \mathbf{R}^d$
2: **Parameters:** M (coordinate bound), p (ℓ_p error norm), ρ_1 (recentering budget), ρ_2 (threshold b.), ρ_3 (noise b.)
3: $\hat{\Sigma} \leftarrow \mathrm{diag}(\hat{\boldsymbol{\sigma}}^2)$; $k \leftarrow \sqrt{n} + \Theta\left(\sqrt{1/\rho_2}\right)$
4: $\tilde{\boldsymbol{\mu}} \leftarrow \textsc{PrivQuantile}_{\rho_1}^{M}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}, 1/2)$
5: $\boldsymbol{y}^{(j)} \leftarrow (\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-1/(p+2)}$ for $j = 1, \ldots, n$
6: $C \leftarrow \textsc{PrivQuantile}_{\rho_2}^{M\sqrt{d}}(\|\boldsymbol{y}^{(1)}\|_2, \ldots, \|\boldsymbol{y}^{(n)}\|_2, \frac{n-k}{n})$
7: sample $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \frac{2C^2}{\rho_3}\mathbf{I})$
8: **return** $\tilde{\boldsymbol{\mu}} + \frac{1}{n}\left(\sum_j \min\left\{\frac{C}{\|\boldsymbol{y}^{(j)}\|_2}, 1\right\} \boldsymbol{y}^{(j)} + \boldsymbol{\eta}\right) \hat{\Sigma}^{1/(p+2)}$

3.2 PLAN building blocks

Like INSTANCE-OPTIMAL MEAN ESTIMATION (IOME) [24], our algorithm computes a rough estimate of the mean as a private estimate $\tilde{\boldsymbol{\mu}} \in \mathbf{R}^d$ of the coordinate-wise median. This step uses the assumption that all coordinates are in the interval $[-M, M]$. It is known that a finite output domain is needed for private quantile selection to be possible [12]. Suppose ℓ_p is the error measure we are aiming to minimize. The next step is to scale coordinates by multiplying with the diagonal matrix $\hat{\Sigma}^{-1/(p+2)}$, where $\hat{\Sigma} = \mathrm{diag}(\hat{\boldsymbol{\sigma}}^2)$ is the diagonal matrix of variance estimates $\hat{\boldsymbol{\sigma}}^2$. Note that since $\hat{\Sigma}$ is a diagonal matrix, it is not necessarily close to the covariance matrix of $\mathcal{D}$ outside of the diagonal.

Note that for $p = 2$ the exponent of $\hat{\Sigma}$ is $-1/4$, which is different from the exponent of $-1/2$ that would be used in order to compute an estimate with small Mahalanobis distance (See e.g. [9]). In fact, our choice of scaling lies between two natural choices. If we removed scaling the magnitude of noise would be the same across all coordinates which corresponds to the standard Gaussian mechanism. This is problematic if many coordinates have small variance compared to others as we would still add a lot of noise to the low variance coordinates. On the other hand, scaling by $\hat{\Sigma}^{-1/2}$ results in noise at each coordinate with magnitude linear proportional to $\hat{\sigma}_i$. That approach adds excessive noise to the high variance coordinates. We strike a middle ground when scaling by $\hat{\Sigma}^{-1/4}$. We add less noise than the first approach for low variance coordinates, and less noise than the second approach for high variance coordinates. Our choice of scaling is one of the main contributions of our work. A key feature of the scaling is that we adjust to the error measure. We discuss scaling further in Appendix C where we show that our choice of scaling is optimal for mean estimation in a simpler setting.

The scaling stretches the ith coordinate by a factor $\hat{\sigma}_i^{-2/(p+2)}$. Since $\hat{\sigma}_i^2 \approx \sigma_i^2$ this changes the standard deviation on the ith coordinate from σ_i to roughly $\sigma_i^{p/(p+2)}$. Next, we compute vectors $\boldsymbol{y}^{(j)}$ that represent the (stretched) differences $\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}}$. Conceptually, we now want to estimate the mean of $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}$, which in turn implies an estimate of $(\frac{1}{n}\sum_j \boldsymbol{x}^{(j)}) - \tilde{\boldsymbol{\mu}}$. The mean of $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}$ is estimated using the Gaussian mechanism. In order to find a suitable scaling of the noise we privately find a quantile C of the lengths $\|\boldsymbol{y}^{(1)}\|_2, \ldots, \|\boldsymbol{y}^{(n)}\|_2$ (all shorter than $M\sqrt{d}$ by assumption) such that approximately k vectors have length larger than C, and clip vectors to length at most C. Clipping, private mean estimation, scaling, and adding back $\tilde{\boldsymbol{\mu}}$ are all condensed in the estimator in Line 8 of Algorithm 1.

4 ANALYSIS

We will show that PLAN returns a private mean estimate that—assuming $\mathcal{D}$ is well behaved—has small expected ℓ_p error with probability at least $1 - \beta$. All probabilities are over the joint distribution of the input samples and the randomness of the PLAN mechanism. For simplicity we focus on the case $p = 2$, but the analysis extends to any $p \geq 1$ as we will show at the end of this section.

We make the following assumptions:

- We assume that for a known parameter M, all inputs are in $\boldsymbol{x}^{(j)} \in [-M, M]^d$ for $j = 1, \ldots, n$. (If this is not the case, the algorithm will clip inputs to this cube, introducing additional clipping error.) Also, we assume that data has been scaled sufficiently such that $\sigma_i \geq 1$ for $i = 1, \ldots, d$. (We discuss what to do if this is not the case in Section 4.6).
- We are given a vector $\hat{\boldsymbol{\sigma}} \in \mathbf{R}^d$ such that
$$\sigma_i \leq \hat{\sigma}_i < \sigma_i + \|\boldsymbol{\sigma}\|_1/d \quad . \tag{1}$$
If no such vector is known we will have to compute it, spending part of the privacy budget. We consider this a separate question and evaluate possible techniques in Section 7.

Definition 4.1. Consider a distribution $\mathcal{D}$ over $\mathbf{R}^d$, denote the mean and standard deviation of the ith coordinate by μ_i and σ_i,

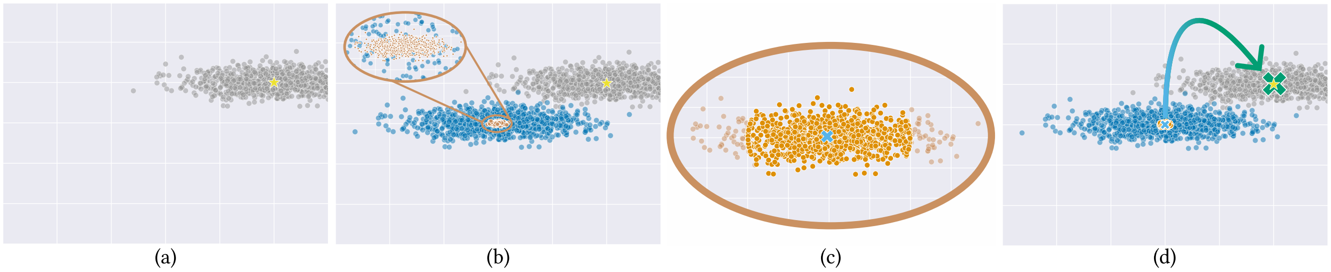


Figure 1: Step-by-step illustration of PLAN (Algorithm 1): (a) Raw data, with statistical mean (yellow star), (b) Recentering (blue) and scaling (orange) corresponding to Line 5, (c) Clipping, as determined by Line 6, (d) Private mean (green cross).

respectively. We say that $\mathcal{D}$ is $\boldsymbol{\sigma}$*-well concentrated* if for any vector $\hat{\boldsymbol{\sigma}} \in \mathbf{R}^d$ with $\sigma_i \leq \hat{\sigma}_i < \sigma_i + \|\boldsymbol{\sigma}\|_1/d$, the following holds for $t > \ln d$

$$\Pr_{X\sim\mathcal{D}}\left[\sum_{i=1}^{d}(X_i-\mu_i)^2 > t\|\boldsymbol{\sigma}\|_2^2\right] = \exp(-\Omega(t)) \tag{2}$$

$$\Pr_{X\sim\mathcal{D}}\left[\sum_{i=1}^{d}(X_i-\mu_i)^2/\hat{\sigma}_i > t\|\boldsymbol{\sigma}\|_1\right] = \exp(-\Omega(t)) \tag{3}$$

Intuitively, these assumptions require concentration of measure of the norms before and after scaling.

Analysis outline

Privacy. It is not hard to see that PLAN satisfies ρ-zCDP with $\rho = \rho_1+\rho_2+\rho_3$: The computation of $\tilde{\boldsymbol{\mu}}$ satisfies ρ_1-zCDP and the computation of C satisfies ρ_2-zCDP by definition of PRIVQUANTILE. Finally, given the values C and $\tilde{\boldsymbol{\mu}}$ the ℓ_2-sensitivity of $\sum_j \min\left\{\frac{C}{\|\boldsymbol{y}^{(j)}\|_2}, 1\right\}\boldsymbol{y}^{(j)}$ with respect to an input $\boldsymbol{x}^{(j)}$ is $2C$, so adding Gaussian noise with variance $2C^2/\rho_3$ gives a ρ_3-zCDP mean estimate. By composition, and since the returned estimator is a post-processing of these private values, the estimator is ρ-zCDP with $\rho = \rho_1 + \rho_2 + \rho_3$.

Utility. [24, 26] present multiple lower bounds that show that there exist inputs such that mean estimation in $\mathbf{R}^d$ requires $n = \Omega(\sqrt{d/\rho})$ samples to achieve meaningful utility. In our analysis, we need the same assumption and require $n = \tilde{\Omega}(\sqrt{d/\rho})$, where the $\tilde{\Omega}$ notation hides polylogarithmic factors in d, β, and M. Let $\hat{\Sigma}$ denote the scaling matrix $\mathrm{diag}(\hat{\boldsymbol{\sigma}}^2)$. In the following, we assume that ρ_1, ρ_2, ρ_3 are fixed fractions of ρ.

The utility analysis has three parts:

(1) First, we argue that $|\tilde{\mu}_i - \mu_i| < 3\sigma_i$ for all $i = 1,\ldots,d$ with high probability.
(2) Let J denote the set of indices for which $\|\boldsymbol{y}^{(j)}\|_2 > C$. We argue that with high probability,

$$\sum_{j\in J}\|\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}}\|_2 = \tilde{O}\left(\left(k+\sqrt{\tfrac{1}{\rho}}\right)\|\boldsymbol{\sigma}\|_2\right),$$

bounding the error due to clipping.
(3) Finally, we argue that with high probability

$$\left\|\boldsymbol{\eta}\,\hat{\Sigma}^{\frac{1}{p+2}}\right\|_2^2 = \tilde{O}(\|\boldsymbol{\sigma}\|_1^2/\rho),$$

bounding the error due to Gaussian noise.

4.1 Part 1: Bounding $|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}|$

In the following K is a universal constant chosen to be sufficiently large.

Lemma 4.2. *Assuming $n > K\max(\sqrt{d\log(M)\log(\log(M)d/\beta)/\rho}$, $\log(d/\beta))$ then with probability $1-\beta$, the coordinate-wise median $\tilde{\mu}_i$ satisfies $\tilde{\mu}_i \in [\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$ for all $i = 1,\ldots,d$.*

PROOF. Sample $X = (X_1,\ldots,X_d) \sim \mathcal{D}$. By Chebychev's inequality, for each i with $1 \leq i \leq d$:

$$\Pr[|X_i - \mu_i| \leq 2\sigma_i] \geq 1 - \frac{\sigma_i^2}{(2\sigma_i)^2} = 3/4 \ .$$

Fixing i, these events are independent for $j = 1,\ldots,n$ so by a Chernoff bound with probability $1-\exp(-\Omega(n))$ there are at least $\frac{2}{3}n$ indices j such that $|X_i^{(j)} - \mu_i| \leq 2\sigma_i$. Condition on the event that this is true for $i = 1,\ldots,d$.

If the rank error of the median estimate $\tilde{\mu}_i$ is smaller than $n/6$, $\tilde{\mu}_i \in [\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$, using the assumption that $\sigma_i \geq 1$. By Lemma 2.9 this is true for every i with probability at least $1-\beta$, given our assumption on n and K if K is a sufficiently large constant. □

4.2 Part 2: Bounding clipping error

Let $J = \{j \in \{1,\ldots,n\} \mid \|\boldsymbol{y}^{(j)}\|_2 > C\}$ denote the set of indices of vectors affected by clipping in PLAN. By the triangle inequality and assuming the bound of part 1 holds,

$$\begin{aligned}\sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}})\right\|_2 &\leq \sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\right\|_2 + |J|\cdot\|\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}}\|_2 \\ &\leq \sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\right\|_2 + 3|J|\cdot\|\boldsymbol{\sigma}\|_2 \ .\end{aligned}$$

By assumption (2) the probability that $\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_2^2 > t\|\boldsymbol{\sigma}\|_2^2$ is exponentially decreasing in $\Omega(t)$ for $t \geq \ln d$, so setting $t = \Omega(\max\{\ln d, \ln(n/\beta)\})$ we have $\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_2 = \tilde{O}(\|\boldsymbol{\sigma}\|_2)$ for all $j = 1,\ldots,n$ with probability at least $1-\beta$. Using the triangle inequality again we can now bound

$$\sum_{j\in J}\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_2 = \tilde{O}(|J|\cdot\|\boldsymbol{\sigma}\|_2) \ .$$

For this part of the analysis, we assume that PRIVQUANTILE is instantiated as in Lemma 2.9, but we return the maximum of the interval instead of the midpoint. This ensures that no vectors whose length fall in the returned interval are clipped. This choice increases the value of C by 1 which slightly increases the magnitude of noise in part 3 of the analysis.

With probability at least $1-\beta$ (over the random choices of the private quantile selection algorithm) there are at most $\tilde{O}\left(k+\sqrt{\frac{1}{\rho}}\right)$ vectors $\boldsymbol{y}^{(j)}$ for which $\|\boldsymbol{y}^{(j)}\|_2 > C$ (assuming $T = \log_2(M\sqrt{d})$), cf. Lemma 2.9 and using that $\left\|\boldsymbol{y}^{(j)}\right\|_2 \le M\sqrt{d}$. That is, we have $|J| = \tilde{O}\left(k+\sqrt{\frac{1}{\rho}}\right)$, and using the bounds above we get

$$\sum_{j\in J}\|\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}}\|_2 = \tilde{O}(|J|\cdot\|\boldsymbol{\sigma}\|_2) = \tilde{O}\left(\left(k+\sqrt{\tfrac{1}{\rho}}\right)\|\boldsymbol{\sigma}\|_2\right) \quad .$$

4.3 Part 3: Bounding noise

By triangle inequality:

$$\begin{aligned}\|\boldsymbol{y}^{(j)}\|_2 &= \left\|(\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2 \\ &\le \left\|(\boldsymbol{x}^{(j)}-\boldsymbol{\mu})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2 + \left\|(\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2\end{aligned}$$

For $p=2$, the ith coordinate of $(\boldsymbol{x}^{(j)}-\boldsymbol{\mu})\,\hat{\Sigma}^{-1/(p+2)}$ equals $(x_i^{(j)}-\mu_i)/\sqrt{\hat{\sigma}_i}$, so assumption (3) implies that the length of the first vector is $\tilde{O}(\sqrt{\|\boldsymbol{\sigma}\|_1})$ with probability at least $1-\beta/2$. From part 1 we know (again for $p=2$) that the ith coordinate of $(\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-1/(p+2)}$ has absolute value at most

$$3\sigma_i/\sqrt{\hat{\sigma}_i} \le 3\sqrt{\sigma_i},$$

where the inequality uses the lower bound on $\hat{\sigma}_i$ in assumption (1). So $\|(\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-1/(p+2)}\|_2 \le 3\sqrt{\|\boldsymbol{\sigma}\|_1}$, and we get that $\|\boldsymbol{y}^{(j)}\|_2 = \tilde{O}(\sqrt{\|\boldsymbol{\sigma}\|_1})$ for all j with probability at least $1-\beta/2$.

The value of C is bounded by the maximum length of a vector $\|\boldsymbol{y}^{(j)}\|_2+1$ (accounting for the interval size of PrivQuantile in part 2), and thus by a union bound, with probability at least $1-\beta$, $C^2 = \tilde{O}(\|\boldsymbol{\sigma}\|_1)$. The scaled noise vector $\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}$ has distribution $\mathcal{N}(\mathbf{0}, \frac{2C^2}{\rho_3}\hat{\Sigma}^{1/(p+2)})$, so using $\hat{\sigma}_i \le \sigma_i + \|\boldsymbol{\sigma}\|_1/d$ from assumption (1), for $p=2$:

$$\begin{aligned}\mathbf{E}[\|\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}\|_2^2] &= \tfrac{2C^2}{\rho_3}\sum_{i=1}^{d}\hat{\Sigma}_{ii}^{2/(p+2)} = \tfrac{2C^2}{\rho_3}\sum_{i=1}^{d}\hat{\sigma}_i \\ &\le \tfrac{2C^2}{\rho_3}\sum_{i=1}^{d}(\sigma_i+\|\boldsymbol{\sigma}\|_1/d) = \tilde{O}(\|\boldsymbol{\sigma}\|_1^2/\rho_3) \quad .\end{aligned}$$

4.4 Proof of Theorem 1.1

The output of PLAN can be written as

$$\tfrac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}^{(j)} - \tfrac{1}{n}\sum_{j\in J}\tfrac{\|\boldsymbol{y}^{(j)}\|_2-C}{\|\boldsymbol{y}^{(j)}\|_2}\,(\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}}) + \tfrac{1}{n}\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}$$

Using that $\frac{\|\boldsymbol{y}^{(j)}\|_2-C}{\|\boldsymbol{y}^{(j)}\|_2} < 1$ for $j\in J$ and the triangle inequality, the ℓ_2 estimation error of PLAN can thus be bounded as

$$\left\|\tfrac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}^{(j)}-\boldsymbol{\mu}\right\|_2 + \tfrac{1}{n}\sum_{j\in J}\|\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}}\|_2 + \tfrac{1}{n}\|\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}\|_2$$

The first term is the sampling error, which is $\tilde{O}(\|\boldsymbol{\sigma}\|_2/\sqrt{n})$ with probability at least $1-\beta$. The sum over J was shown in part 2 to be $\tilde{O}\left(\left(k+\sqrt{\frac{1}{\rho}}\right)\|\boldsymbol{\sigma}\|_2\right)$, so for $k\le\sqrt{n}$ and using the assumption $n = \tilde{\Omega}(\max(\sqrt{d/\rho},\rho^{-1}))$, this term is bounded by $\tilde{O}(\|\boldsymbol{\sigma}\|_2/\sqrt{n})$. Finally, the noise term in part 3 is $\tilde{O}(\|\boldsymbol{\sigma}\|_1/(n\sqrt{\rho}))$.

We are now ready to state a more detailed version of Theorem 1.1.

Theorem 4.3. *For sufficiently large $n = \tilde{\Omega}(\sqrt{d/\rho})$, PLAN with parameters $k=\sqrt{n}$ and $\rho_1=\rho_2=\rho_3=\rho/3$ is ρ-zCDP. If inputs are independently sampled from a $\boldsymbol{\sigma}$-well concentrated distribution, the mean estimate has expected ℓ_2 error*

$$\tilde{O}(1+||\boldsymbol{\sigma}||_2/\sqrt{n}+||\boldsymbol{\sigma}||_1/(n\sqrt{\rho})),$$

where $\tilde{O}$ suppresses polylogarithmic dependencies on n, d, and the bound M on the ℓ_∞ norm of inputs.

Proof. The discussion above argued for an expected ℓ_2 error in the absence of "failure events." The probability that no such event happens is at least $1-\beta$. The largest possible error is the diameter of the cube from which the input vectors come, $2M\sqrt{d}$. Thus, the largest contribution to the expected error from a probability β event is $2M\sqrt{d}\beta$. Choosing $\beta = \frac{1}{M\sqrt{d}}$ guarantees that the expected contribution from these events is $O(1)$. □

4.5 General ℓ_p error

To address the general case we need a definition of well-concentrated that depends on p. For simplicity we assume that p is a positive integer constant.

Definition 4.4. For integer constant $p\ge 1$ consider a distribution $\mathcal{D}$ over $\mathbf{R}^d$, where the ith coordinate has mean μ_i and pth central moment σ_i^p. We say that $\mathcal{D}$ is $(\boldsymbol{\sigma},p)$-*well concentrated* if for any vector $\hat{\boldsymbol{\sigma}}\in\mathbf{R}^d$ with $\sigma_i\le\hat{\sigma}_i<\left(\sigma_i^{\frac{2p}{p+2}}+\|\boldsymbol{\sigma}^{\frac{2p}{p+2}}\|_1/d\right)^{\frac{p+2}{2p}}$, the following holds for $t>\ln d$:

$$\Pr_{X\sim\mathcal{D}}\left[\sum_{i=1}^{d}|X_i-\mu_i|^p > t\|\boldsymbol{\sigma}\|_p^p\right] = \exp(-\Omega(t)) \tag{4}$$

$$\Pr_{X\sim\mathcal{D}}\left[\sum_{i=1}^{d}\frac{(X_i-\mu_i)^2}{\hat{\sigma}_i^{4/(p+2)}} > t\|\boldsymbol{\sigma}^{p/(p+2)}\|_2^2\right] = \exp(-\Omega(t)) \quad . \tag{5}$$

Theorem 4.5. *For sufficiently large $n = \tilde{\Omega}(\max(\sqrt{d/\rho},\rho^{-1}))$, PLAN with parameters $k=\sqrt{n}$ and $\rho_1=\rho_2=\rho_3=\rho/3$ is ρ-zCDP. If inputs are independently sampled from a $(\boldsymbol{\sigma},p)$-well concentrated distribution, the mean estimate has expected ℓ_p error*

$$\tilde{O}\left(1+\frac{||\boldsymbol{\sigma}||_p}{\sqrt{n}}+\frac{||\boldsymbol{\sigma}||_{2p/(p+2)}}{n\sqrt{\rho}}\right),$$

where $\tilde{O}$ suppresses polylogarithmic dependencies on $1/\beta$, n, d, and the bound M on the ℓ_∞ norm of inputs.

The proof of this theorem follows along the lines of the proof in this section and can be found in Appendix B. In comparison, the standard Gaussian mechanism for ℓ_2 sensitivity $\|\boldsymbol{\sigma}\|_2$ has expected ℓ_p error $\frac{d^{1/p}\|\boldsymbol{\sigma}\|_2}{\sqrt{\rho}}$.

4.6 Approaches for small variance

Throughout our analysis in Section 4 we assume that $\sigma_i \geq 1$ for all i. If it is not known whether this assumption holds, one can enforce this by adding independent noise of variance 1 to each coordinate. If the additional error due to this noise is considered too large, we can proceed in the following ways. For simplicity, we assume that we know $\boldsymbol{\sigma}$ exactly, but similar approaches can be used if we only have an approximation $\hat{\boldsymbol{\sigma}}$.

Let $\min\{\boldsymbol{\sigma}\}$ be the smallest entry in $\boldsymbol{\sigma}$. We assume in this paragraph that all dimensions are non-constant, i.e., $\min\{\boldsymbol{\sigma}\} > 0$. A simple technique is to scale all inputs and standard deviations by $1/\min\{\boldsymbol{\sigma}\}$ to spread out the data. That is, $\bar{\sigma}_i = \sigma_i/\min\{\boldsymbol{\sigma}\}$ and $\bar{x}_i^{(j)} = x_i^{(j)}/\min\{\boldsymbol{\sigma}\}$ for all i and j. Run PLAN on input $\bar{\boldsymbol{x}}^{(1)}, \ldots, \bar{\boldsymbol{x}}^{(n)}$, and $\bar{\boldsymbol{\sigma}}^2$, and scale back the private estimate by $\min\{\boldsymbol{\sigma}\}$.

The approach above works well when $\min\{\boldsymbol{\sigma}\}$ is not too small, but is not ideal if some coordinates have tiny variance since it requires a huge scaling. Instead we can change the parameters for PrivQuantile. We use the assumption that $\min\{\boldsymbol{\sigma}\} \geq 1$ only in Part 1 of Section 4.1. There we show that the estimate returned by PrivQuantile is close to a point in $[\mu_i - 2\sigma_i, \mu_i + 2\sigma_i]$ with high probability. Specifically, the estimate is within distance 1 of such a point and as such it is in $[\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$ for any $\sigma_i \geq 1$. When $\sigma_i < 1$ we can increase the number of steps of the binary search to get closer to the interval. For example, for a small value $\tau > 0$, we can use $T = \log_2(M/\tau)$ to return an estimate that is within distance at most τ with high probability. In the case that there are coordinates $\sigma_i < \tau$ so tiny that we do not return a point in $[\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$, this might increase the clipping error and the expected error due to noise. For this setting, the analysis in Section 4.2 and Section 4.3 can only use the bound $|\boldsymbol{\mu}_i - \tilde{\boldsymbol{\mu}}_i| \leq 2\sigma_i + \tau$. This results in an additional clipping error of $\frac{1}{n}\left(k + \sqrt{1/\rho}\right)\tau\sqrt{d}$. Using $\tau \leq \min(\sqrt{n/d}, n\sqrt{\rho/d})$ bounds the additional contribution of this clipping error to be $\tilde{O}(1)$. To bound the additional error due to noise, note that $\tau\|\hat{\Sigma}^{-1/4}\|_2 = \tau/\sqrt{\|\boldsymbol{\sigma}\|_1}$, which means that the value of C in Section 4.3 is bounded by $\tilde{O}\left(\sqrt{\|\boldsymbol{\sigma}\|_1} + \tau/\sqrt{\|\boldsymbol{\sigma}\|_1}\right)$. Carrying out the computation of the expected error due to noise with this value of C yields the bound $\tilde{O}\left((\|\boldsymbol{\sigma}\|_1^2 + \tau\|\boldsymbol{\sigma}\|_1 + \tau^2)/\rho_3\right)$. Setting $\tau \leq \|\boldsymbol{\sigma}\|_1$, the additional noise error has asymptotically not influence on the expected error bounds in Theorems 4.3 and 4.5.

5 EXAMPLES OF WELL-CONCENTRATED DISTRIBUTIONS

In the following we give examples for some $(\boldsymbol{\sigma}, p)$-well concentrated distributions. We start with a general result.

Lemma 5.1. *For integer constant $p \geq 1$, consider a distribution $\mathcal{D}$ over $\mathbf{R}^d$ where the ith coordinate is independently distributed with mean μ_i and standard deviation σ_i such that for $X \sim \mathcal{D}$, the pth moment $\mathbf{E}[|X_i - \mu_i|^p] \leq K\sigma_i^p$ for some constant K. Then $\mathcal{D}$ is $(\boldsymbol{\sigma}, p)$-well concentrated.*

Proof. We first prove the bound (4) from Definition 4.4. Sample X from $\mathcal{D}$ and define $Y_i = |X_i - \mu_i|^p - \mathbf{E}[|X_i - \mu_i|^p]$ and note that $\sum Y_i \leq \sum |X_i - \mu_i|^p$. Each Y_i is zero-centered, so we may apply Bernstein's inequality (Lemma A.1)

$$\Pr\left[\sum_{i=1}^{d} Y_i > t\|\boldsymbol{\sigma}\|_p^p\right] \leq \exp\left(-\frac{t^2\|\boldsymbol{\sigma}\|_p^{2p}/2}{\sum_{i=1}^{d}\mathbf{E}[Y_i^2] + Mt\|\boldsymbol{\sigma}\|_p^p/3}\right).$$

We distinguish two cases depending on which term is dominating the denominator. In the first case,

$$\sum_{i=1}^{d}\mathbf{E}\left[Y_i^2\right] \leq \sum_{i=1}^{d}\mathbf{E}\left[(X_i - \mu_i)^{2p}\right] \leq K\|\boldsymbol{\sigma}^{2p}\|_1 \leq \|\boldsymbol{\sigma}\|_p^{2p},$$

where we applied the triangle inequality. This means that the probability of $\sum |X_i - \mu_i|^p$ to exceed $t\|\boldsymbol{\sigma}\|_p^p$ is $\exp\left(-\Omega\left(t^2\right)\right)$. In the second case,

$$\frac{t^2\|\boldsymbol{\sigma}\|_p^{2p}/2}{Mt\|\boldsymbol{\sigma}\|_p^p/3} = \frac{3t\|\boldsymbol{\sigma}\|_p^p}{2M}.$$

By Chebychev's inequality almost surely $Y_i \leq C\max_i \sigma_i^p \leq \|\sigma\|_p^p$. Thus, the probability is bounded by $\exp(-\Omega(t))$ in this case.

For the second property (5), note that $\sum (X_i - \mu_i)^2/\hat{\sigma}_i^{4/(p+2)}$ is maximized for $\hat{\sigma}_i = \sigma_i$. For this choice, the calculations are analogous to the ones above. □

We will now proceed with studying the Gaussian distribution for ℓ_2 error and a sum of Poisson trials for ℓ_1 error. These distributions are the basis of the empirical evaluation in Section 7.

5.1 Gaussian data

Lemma 5.2. *Let $p = 2$ and fix $\boldsymbol{\mu} \in \mathbf{R}^d$ and $\Sigma \in \mathbf{R}^{d\times d}$. The multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is $(\boldsymbol{\sigma}, 2)$-well concentrated.*

Before presenting the proof, we remark that the independent case with diagonal covariance matrix Σ can easily be handled by Chernoff-type bounds. Furthermore, Lemma 5.1 holds for $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ for diagonal covariance matrix. In the lemma, we handle the general case of (non-)diagonal covariance matrices, allowing for dependence among the coordinates.

Proof. Consider property (2) of Definition 4.1 and fix any $i \in \{1, \ldots, d\}$. Let $\sigma_i^2 = \Sigma_{ii}$ and sample $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. X_i is normally distributed, so we can make use of [35, Theorem 9.3] which states that for all $t > 0$ it holds that

$$\Pr[|X - \mu_i| \geq t] \leq 2\exp(-t^2/(2\sigma_i^2)).$$

Fix a value $t > 0$ and let $t' = (t + \ln(d))\sigma_i^2$. We proceed to bound the deviation as follows:

$$\begin{aligned}\Pr[(X_i - \mu_i)^2 \geq t'] &= \Pr[|X_i - \mu_i| \geq \sqrt{t'}] \\ &\leq 2\exp\left(-\left((t + \ln(d))\sigma_i^2\right)/\left(2\sigma_i^2\right)\right) \\ &= 1/d\exp(-\Omega(t)).\end{aligned}$$

Consider the case that we sample $X_1, \ldots, X_d$ from $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. By a union bound over $X_1, \ldots, X_d$, we may assume that with probability at least $1 - \exp(-\Omega(t))$, for all $i \in \{1, \ldots, d\}$ we have $(X_i - \mu_i)^2 \leq (t + \ln(d))\sigma_i^2$, which implies that their sum is at most $(t + \ln(d))\|\boldsymbol{\sigma}\|_2^2 = \tilde{O}(t\|\boldsymbol{\sigma}\|_2^2)$.

Next, consider property (3) of Definition 4.1. Sample $X \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and consider the transformation $(X - \mu_i)^2/\sigma_i$. Setting $t' = (t +$

$\ln(d))\sigma_i$, the same calculations as above show that with probability at most $1/d\exp(-\Omega(t))$, $(X-\mu_i)^2/\sigma_i$ is larger than $(t+\ln(d))\sigma_i$.

Consider the case that we sample X from $\mathcal{N}(\boldsymbol{\mu},\Sigma)$. Again using a union bound, with probability at least $1-\exp(-\Omega(t))$, all scaled values are within $(t+\ln(d))\boldsymbol{\sigma}_i$. Conditioned on this, $\sum_{i=1}^d (X_i-\mu_i)^2/\hat{\sigma}_i \le (t+\ln(d))\|\boldsymbol{\sigma}\|_1$, which finishes the proof. □

5.2 Binary data

Consider length-d binary strings such that bit i is set independently and at random with probability q_i. Lemma 5.1 is not applicable because the pth moment is roughly equal to $\max\{q_i, 1-q_i\}$.

Lemma 5.3. *Let $d \ge 1$ be an integer, and let $\mathcal{D}_{binary}$ be the distribution over length-d binary strings such that the bit in position i is set with probability q_i. If for each $i \in \{1,\dots,d\}$, $\sigma_i^2 = q_i(1-q_i) \ge 1/d^{2/5}$, then $\mathcal{D}_{binary}$ is $(\boldsymbol{\sigma},1)$-well concentrated.*

Proof. Since we assumed in Section 4 that $\sigma_i \ge 1$, we consider the mapping $x \mapsto d^{1/5}x =: \bar{x}$. Consider Property (4) of Definition 4.4. Since $\bar{\sigma}_i^2 \ge 1$, $\|\bar{\boldsymbol{\sigma}}\|_1 \ge d$. Using a generalized Chernoff bound (Lemma A.3) we conclude that $\sum |d^{1/5}(X_i-\mu_i)|$ exceeds $t\|\bar{\boldsymbol{\sigma}}\|_1$ with probability at most

$$\exp\left(-(t\|\bar{\boldsymbol{\sigma}}\|_1)^2/\left(2d^{2/5}d\right)\right) = \exp\left(-\Omega(t^2)\right).$$

Next, consider Property (5). Define $Y_i = (d^{1/5}(X_i-q_i))^2/\sigma_i^{4/3}$. Note that Y_i takes values in an interval of length $d_i \le d^{2/5}$. $|\sum Y_i - \mathbf{E}[\sum Y_i]|$ exceeds $t\|\bar{\boldsymbol{\sigma}}^{2/3}\|_1$ with probability at most

$$\exp\left(-\frac{t^2\|\bar{\boldsymbol{\sigma}}^{2/3}\|_1^2}{2\sum_{i=1}^d d_i^2}\right) \le \exp\left(-\frac{t^2\|\bar{\boldsymbol{\sigma}}^{2/3}\|_1^2}{2d^{9/5}}\right) \le \exp\left(-t^2 d^{1/5}/2\right) = \exp\left(-\Omega(t^2)\right) \;,$$

which finishes the proof. □

6 GENERIC BOUNDS IN THE ABSENCE OF VARIANCE ESTIMATES

Algorithm 1 requires as input estimates $\hat{\boldsymbol{\sigma}}^2$ on the coordinate-wise variances. If the input is $\boldsymbol{\sigma}$-well concentrated, Theorem 4.3 provided bounds on the expected ℓ_2 error of the algorithm. In this section, we consider the case that no such estimates are known and we run the algorithm without the scaling step. This is similar in spirit to using the "shifted-clipped-mean estimator" of Huang et al. [24]. The following theorem shows that even without carrying out the random rotation (see Section 3.3 in [24]), we match their bounds up to constant terms in expectation. This result makes their algorithm useful in settings where a random rotation impacts performance negatively, e.g., when vectors are sparse.

Let $D \subseteq \mathbf{R}^d$ be the collection of n vectors $\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(n)}$. Let $w(D) = \max_{\boldsymbol{x},\boldsymbol{y}\in D}\|\boldsymbol{x}-\boldsymbol{y}\|_2$ be the diameter of the dataset. For simplicity we assume that $w(D)\ge 1$ and $M = O(w(D))$. In comparison to the distributional setting studied before, there is no sampling error involved in mean estimation and we only measure clipping error and the error due to noise.

Theorem 6.1. *Let $d \ge 1, \rho > 0$, and D be of size $n = \tilde{\Omega}\left(\sqrt{\frac{d}{\rho}}\right)$. If Algorithm 1 is run with $k = \tilde{\Theta}\left(\sqrt{\frac{d}{\rho}}\right)$, $\rho_1=\rho_2=\rho_3=\rho/3$, and $\hat{\sigma}_i^2 = 1$ for all $i \in \{1,\dots,d\}$ then the expected ℓ_2 error due to clipping and noise is*

$$\tilde{O}\left(\sqrt{\frac{d}{\rho}}\frac{w(D)}{n}\right) \;.$$

Before proceeding to the proof, we introduce some helpful notation.

Definition 6.2. Let $\boldsymbol{\mu} \in \mathbf{R}^d$ be the coordinate-wise median of D. For $0 \le \alpha < 1/2$, we say that $\tilde{\boldsymbol{\mu}} \in \mathbf{R}^d$ is α-good if each $\tilde{\mu}_i$ has rank error at most αn from μ_i.

Lemma 6.3. *Given $0 \le \alpha < 1/2$, let $\tilde{\boldsymbol{\mu}}$ be α-good. For each $\boldsymbol{x}^{(j)} \in D$, $1 \le j \le n$, $\|\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}}\|_2^2 \le \frac{w(D)^2}{1/2-\alpha}$.*

Proof. Fix $\boldsymbol{x}^{(j)}$ and compute

$$\begin{aligned}(n-1)w(D)^2 &\ge \sum_{j\ne j'}\|\boldsymbol{x}^{(j)}-\boldsymbol{x}^{(j')}\|_2^2\\ &= \sum_{i=1}^d\sum_{j\ne j'}\left(x_i^{(j)}-x_i^{(j')}\right)^2\\ &\ge \sum_{i=1}^d(1/2-\alpha)(n-1)\left(x_i^{(j)}-\tilde{\mu}_i\right)^2\\ &= (1/2-\alpha)(n-1)\|\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}}\|_2^2.\end{aligned}$$

The lemma follows by re-ordering terms. □

Proof of Theorem 6.1. We instantiate PrivQuantile as the binary search based method of [24] with $T = \log(Mn\sqrt{\rho})$. Fix some value for α, say 1/3.

By Lemma 2.9 the coordinate-wise median computed on Line 4 of Algorithm 1 is within ℓ_2 distance $\sqrt{d/\rho}/n$ of an α-good point with high probability. Let us assume that $\tilde{\boldsymbol{\mu}}$ is itself α-good as the additional error of $\sqrt{d/\rho}/n$ from the quantile search is dominated by the error from clipping and noise. The failure probability from Lemma 2.9 must be set sufficiently low such that the expected error from failure events is also dominated by the error from clipping and noise. Since the maximal possible ℓ_2 error is $\sqrt{d}2M$ this is trivially true when n is large enough such that $\beta \le \frac{w(D)}{\sqrt{\rho}Mn}$.

Since $\tilde{\boldsymbol{\mu}}$ is α-good, the maximum length of a shifted vector $\boldsymbol{x}^{(j)}-\tilde{\boldsymbol{\mu}}$ is $O(w(D))$, so $C = O(w(D))$ on Line 6 of Algorithm 1. As in Section 4.3, the expected ℓ_2 error due to noise is at most $1/n \cdot \sqrt{2C^2(\sum_{i=1}^d \hat{\sigma}_i)/\rho} = \tilde{O}\left(\sqrt{\frac{d}{\rho}}\frac{w(D)}{n}\right)$.

Analogously to the calculations carried out in Section 4.2, we can bound the error due to clipping for all vectors $\|\boldsymbol{y}^{(j)}\|_2 > C$. Setting $k = \tilde{\Theta}(\sqrt{d/\rho})$ shows that this clipping error is $\tilde{O}\left(\sqrt{d/\rho}\cdot w(D)/n\right)$. □

7 EMPIRICAL EVALUATION

To put plan's utility into context, we measure error in diverse experimental settings. We use the empirical mean, i.e., the sampling error, as a baseline, since it reflects an inevitable lower bound. Additionally we compare plan to Instance-optimal mean estimation (iome) [24], which has been shown to perform at least as well as CoinPress [8] in empirical settings [24]; hence representing the current state-of-the-art for differentially private mean estimation.

Our experimental evaluation focuses on the examples of $(\boldsymbol{\sigma}, p)$-well concentrated distributions considered in Section 5; we evaluate the utility-privacy tradeoff for Gaussian and binary data under ℓ_2 and ℓ_1 error, respectively. We run our experiments with synthetic data as input. For the binary case, we also evaluate our error on the Kosarak dataset [7] which represents user visits (or, conversely, non-visits) to webpages, as well as the Point of Sale (POS) dataset[1] which represents user purchases. The code used for our experiments is available at https://github.com/ChristianLebeda/PLAN-experiments.

Additional experiments. Due to space constraints, we provide additional experimental results in the appendices. Appendix E evaluates the performance of PLAN and IOME for correlated, Gaussian data. We demonstrate that PLAN is less affected by dependencies than IOME. Appendix F includes the variant of PLAN that does not scale the input, cf. Section 6, and compares it to PLAN and IOME. In a nutshell, this variant recovers the behavior of IOME even in empirical settings in which constant matters, as suggested by Theorem 6.1. Finally, Appendix G discusses experiments for larger input sizes, which technically is far away from our assumptions on the input size. The evaluation shows that PLAN is less robust than its competitors for largest inputs and smallest privacy budget.

7.1 Implementation

We evaluate the empirical accuracy of PLAN by instantiating Algorithm 1 in Python 3. Our implementation contains two different instantiations of PLAN: one version for binary data targeting $p = 1$, and one version for data from multivariate Gaussian distributions for $p = 2$. The pseudo code for both instantiations of PLAN is shown in Listing 1. Both instances use the PrivQuantile search by Kaplan et al. [30]. The implementation of IOME uses the original source code from Huang et al. [24].

```
def PLAN(data, n, d, M, p, rho, beta) {
 rho1, rho2, rho3 = divideBudget(rho)
 mu = center(M, rho1*0.25, data, beta/3)
 std = estimateStd(M, rho1*0.75, data, beta/3)
 scaleFactors = std**(-2/(p + 2))
 y = (data-mu) * scaleFactors #coordinate-wise
 k = sqrt(n) + rankError(M, n, d, rho2, beta/3)
 z = clip(M, y, rho2, (n-k)/n)
 return ((z+noise(p, rho3))/scaleFactors)+mu
}
```

Listing 1: Pseudo code for the PLAN instantiation for n vectors in $\mathbb{R}^d$, with each coordinate being in the range $[-M, M]$, targeting the expected ℓ_p error with privacy budget ρ and a failure probability of at most β. Note that budget is spent on estimating the standard deviation unlike in Algorithm 1 where $\hat{\sigma}^2$ is an input parameter.

Estimating $\boldsymbol{\sigma}^2$. We recall that Algorithm 1 assumes an estimate of the variances as input. In the absence of public knowledge, these parameters have to be estimated on the actual data in a differentially private way, as mentioned in Listing 1. We find estimates in the following way.

Let $X, Y \sim \mathcal{D}$ be two i.i.d. random variables. By a standard calculation, $\mathbf{E}[(X-Y)^2/2] = \mathbf{E}[X^2] - \mathbf{E}[X]^2 = \mathbf{Var}[X]$. In the Gaussian case, given $X, Y \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $(X-Y)^2/2$ follows a Chi-squared distribution with one degree of freedom. We use PrivQuantile for each coordinate to differentially privately estimate the median $\tilde{\mu}_i$ of a sequence of values $(X-Y)^2/2$. Using the Wilson-Hilferty transformation [25, (18.24)], we then estimate the mean as $\tilde{\mu}_i/(9/7)^3$ as a post-processing step. In the binary case, each coordinate is 1 with probability q_i. We estimate the variance by privately estimating the mean $\tilde{q}_i$ using the Gaussian mechanism with ℓ_2-sensitivity $1/n$, and use $\hat{\sigma}_i^2 = \tilde{q}_i(1-\tilde{q}_i)$. In both cases, we regularize the estimate $\hat{\boldsymbol{\sigma}}$ on the standard deviation by adding $\|\hat{\boldsymbol{\sigma}}\|_1/d$ to each coordinate.

In Appendix D, we generalize the estimator on Gaussian data to distributions with bounded fourth moment and provide more details on the empirical evaluation of differentially private variance estimation.

Bounding the clipping universe. Scaling the data based on the variances estimate $\hat{\sigma}^2$ gives an opportunity to trim the universe used when searching for the clipping radius (Algorithm 1, Line 6). For the ℓ_2 error, instead of using $M\sqrt{d}$ as the upper bound to cover the entire universe, we use the bound $\sqrt{\log(n)\log(1/\beta)\|\hat{\boldsymbol{\sigma}}\|_1}$ for a given failure probability β. We set this failure probability to 0.1. We can use this bound because Assumption (3) in Definition 4.1 guarantees that the length of vectors after scaling is proportional to $\sqrt{\|\hat{\boldsymbol{\sigma}}\|_1}$.

7.2 Experiment design

Parameter input space. The following parameters need to be chosen for each execution of PLAN: the universe M, the ℓ_p error norm, and the privacy budget ρ as well as the partitioning of ρ into ρ_1, ρ_2, ρ_3. We also use a failure probability β to compute exact constants for the universe after scaling and the value of k for the clipping threshold quantile, cf. Line 3 in Algorithm 1 and the discussion above. Since IOME uses a binary search for their quantile selection, the amount of steps to use also needs to be chosen. IOME sets the amount of steps to 10 by default, but an empirical investigation shows that this value is too low for many of our settings — the binary quantile search ends early which causes inaccurate results. To level the playing field, we use 20 steps to ensure that IOME does not suffer any disadvantages from the binary quantile search failing. Furthermore, we noticed that in the implementation of IOME, the quantile for the clipping threshold is set too aggressively. We used as offset a value that is twice as large as their proposal, which made their implementation much more robust without increasing the median error.

Input data. We use both synthetic and real-world datasets to evaluate PLAN. When generating synthetic data, the following parameters need to be chosen: the dataset size n, the dimensionality d, the means $\boldsymbol{\mu}$, and the variances $\boldsymbol{\sigma}^2$. Since PLAN and IOME both ignore potential correlations in data, we use covariance matrices of the form $\Sigma = \mathrm{diag}(\boldsymbol{\sigma}^2)$ for the Gaussian case. As shown in Appendix E, the trends observed for the diagonal case carry over to the non-diagonal, correlated case. In the binary case for real-world data, coordinates are correlated.

7.2.1 Gaussian data. To show the effectiveness of PLAN on Gaussian data, we design three diverse experiments. The first experiment (Gaussian A) reflects the parameter settings used in previous work

[1] https://github.com/cpearce/HARM/blob/master/datasets/BMS-POS.csv

Name	n	ρ	M	d	$\boldsymbol{\sigma}^2$	$\boldsymbol{\mu}$
Gaussian A	4000	$\{1, 0.5, 0.125\}$	$50\sqrt{d}$	$\{16, 32, \ldots, 1024\}$	$(1, \ldots, 1)$	0^d
Gaussian B	10 000	$\{1, 0.125\}$	$100\sqrt{d} \max\{\boldsymbol{\sigma}\}$	512	$\left(\left(\frac{d}{d}\right)^{\alpha}, \left(\frac{d}{d-1}\right)^{\alpha}, \ldots, \left(\frac{d}{1}\right)^{\alpha}\right)$	10^d
Gaussian C	10 000	$\{1, 0.5, 0.125\}$	$100\sqrt{d} \max\{\boldsymbol{\sigma}\}$	$\{16, 32, \ldots, 1024\}$	$\left(\left(\frac{d}{d}\right)^{2}, \left(\frac{d}{d-1}\right)^{2}, \ldots, \left(\frac{d}{1}\right)^{2}\right)$	10^d
Binary	4096	$\{1, 0.5, 0.125\}$	$\Vert\hat{\boldsymbol{\sigma}}^{2/3}\Vert_2$	$\{256, 512, \ldots, 2048\}$	$\sigma_i^2 = \mu_i(1-\mu_i)$	$\mu_i = \begin{cases} \frac{1}{2} & i \leq \alpha d \\ \frac{1}{100} & \text{else} \end{cases}$
Kosarak	75 462	$\{1, 0.5, 0.25, 0.125, 0.0625\}$	$\Vert\hat{\boldsymbol{\sigma}}^{2/3}\Vert_2$	27 983	*Fixed dataset*	
POS	515 596	$\{1, 0.5, 0.25, 0.125, 0.0625\}$	$\Vert\hat{\boldsymbol{\sigma}}^{2/3}\Vert_2$	1 657	*Fixed dataset*	
Global setting: $\rho_1 = 0.25\rho/d, \rho_2 = 0.25(\rho - \rho_1), \rho_3 = \rho - \rho_1 - \rho_2$.						

Table 1: Settings for the experiments. All experiment settings run 50 times for each algorithm. IOME is run with 20 steps for the binary quantile selection. Note that ρ_1 is split between recentering and variance prediction for PLAN.

by Huang et al. [24] where data has no skew, which is the worst case for our proposed method. The second experiment (Gaussian B) simulates data ranging from no to significant skew across dimensions, showcasing how PLAN improves with increasing skew. Finally, the third experiment (Gaussian C) highlights how PLAN scales as dimensionality increases for data with a skew. For each experiment, we vary ρ between 1 and 0.01 to show how accuracy scales in higher and lower privacy regimes. We summarize the settings used in the experiments in Table 1. Note that the settings $\rho = 0.01$ corresponds to ($\varepsilon \approx 0.75, \delta \approx 10^{-6}$)-approximate differential privacy.

Budget division. The values we report for ρ in Table 1 correspond to the total privacy budget for the entire experiment, also including variance estimation. Our algorithm needs to perform two preprocessing steps: estimating $\boldsymbol{\mu}$ for re-centering, and estimating $\boldsymbol{\sigma}^2$ for scaling the noise. We fix the initial estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ to use 25% of the total privacy budget — the same proportion used for preprocessing as in Huang et al. [24]. In the same spirit, we set the budget to determine the clipping threshold (ρ_2) to 25% of the remaining budget, and use the larger part (ρ_3) for the Gaussian noise.

Choosing valid settings. Just like for IOME, M needs to be set such that $\boldsymbol{\mu}$ is within the universe. We will use two different approaches to set M: the approach from [24] ($M = 50\sqrt{d}$), and a more pessimistic approach where we assume all values have the worst-case standard deviation across all dimensions, and create more leeway by scaling with a constant ($M = 100\sqrt{d}\max\{\boldsymbol{\sigma}\}$).

Additionally, the rank error needs to be tuned such that PRIVQUANTILE search can be expected to return a quantile close to the requested one. Since PLAN calls PRIVQUANTILE multiple times, PLAN needs to tolerate the worst-case rank error for all calls. We set n such that the rank error is at most $0.1n$ for each value of ρ.

Gaussian A: *no skew.* To show that PLAN performs comparatively to IOME we run it on data where variance is the same across all dimensions. In this setting we expect PLAN to perform similarly to IOME. We reuse the experiment settings used by Huang et al. [24] for a fair comparison.

Gaussian B: *varying skewness.* To show how PLAN improves as the input data's skew increases, we vary the skewness of the variance. We introduce a parameter α, and simulate a Zipfian like skew to the data, and set the variances $\boldsymbol{\sigma}^2 = ((d/d)^{\alpha}, (d/(d-1))^{\alpha}, \ldots, (d/1)^{\alpha})$ for $\alpha \in \{0, 0.5, \ldots, 2\}$. In this setting we expect PLAN to outperform IOME for $\alpha > 0$.

Gaussian C: *varying dimensionality.* To show how PLAN's advantage scales compared to IOME, we vary dimensionality as we expect an improvement up to a factor $\sqrt{d}$. Since PLAN's advantage is based on data having a skew, we set $\boldsymbol{\sigma}^2 = \left((d/d)^2, (d/(d-1))^2, \ldots, (d/1)^2\right)$. Note that PLAN's improvement is in *noise error*, as sampling error is unavoidable — we compute the error relative to the empirical mean in this experiment to showcase the difference in noise error.

7.2.2 Binary data.

To diversify our experimental scope, we consider binary data represented by n bitvectors of length d in which each bit i, $1 \leq i \leq d$, is set independently to 1 with probability q_i. We usually think about these bitvectors as sets representing a selection of items from $\{1, \ldots, d\}$. To vary the error measure, we focus on the ℓ_1 error. This is akin to computing the total variation distance (TVD) $1/2\Vert x - y\Vert_1$, but we avoid the normalization of x and y to have unit ℓ_1 norm.

We design three experiments: the first (Binary) varies the skewness in the probabilities to make controlled experiments on the accuracy of PLAN. The two remaining experiments (Kosarak, POS) use real-world datasets that naturally exhibit skew between coordinates.

Binary: *Varying skewness.* This experiment follows the same design principle as Gaussian B: to show how skewness affects the performance of PLAN. Given n, d, and ρ, we choose two probabilities $q_1 = 0.5$ (high variance) and $q_2 = 0.01$ (low variance). Given $\alpha \in [0, 1]$, we sample the first $\lceil \alpha d \rceil$ bits with probability q_1 each, and the remaining positions with probability q_2. The low variance setting is slightly below the minimum threshold of $1/d^{2/5}$ discussed in Lemma 5.3 to test the robustness of our implementation. We clip all estimated variances to $1/d^{2/5}$ from below.

Kosarak: *Website visits.* The *Kosarak dataset*[2] represents clickstream data of a Hungarian news portal. There are $n = 75\,462$ users and a collection of $d = 27\,983$ websites. In total, users clicked on

[2]http://fimi.uantwerpen.be/data/

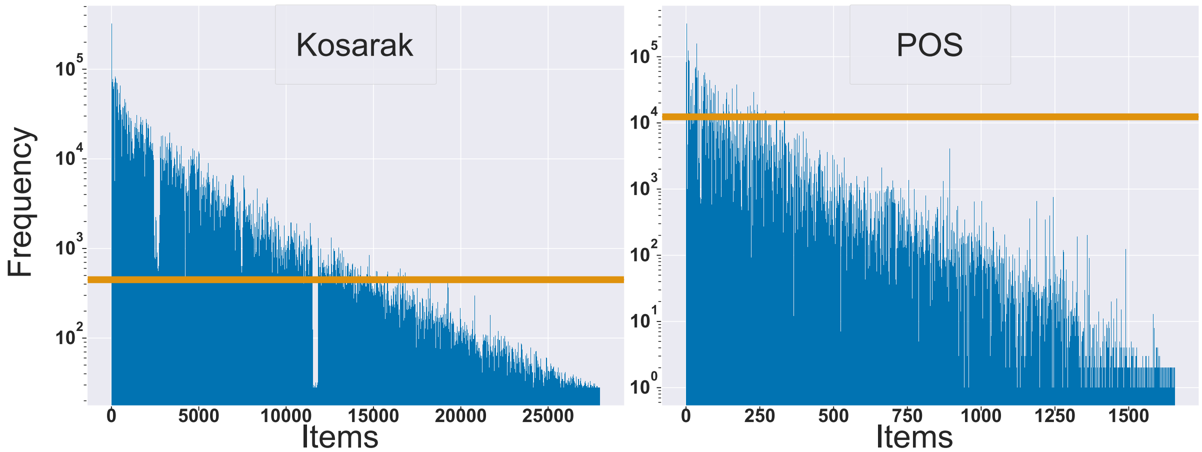


Figure 2: Histogram for Kosarak (left) and POS (right). The orange line is the smallest allowed variance according to Lemma 5.3 which we clip to.

4 194 414 websites (each user clicked on 55.6 websites on average), and there is a large skew between the websites, see Figure 2.

POS: *Shopping baskets*. The *POS dataset* contains merchant transactions on $d = 1\,657$ categories from $n = 515\,596$ users. In total, there are 3 367 019 transactions (around 6.5 on average per user). Again, there is large skew in the different categories, see Figure 2. The dataset is particularly challenging because the minimum variance $d^{-2/5}$ (cf. Lemma 5.3) has to be clipped on many coordinates.

7.3 Results

For the Gaussian case, we ran our experiments using Python 3.11.3 on a MacBook Pro with 24GB RAM, and the Apple M2 chip (8-core CPU). For the binary case, we had to run the experiments on a more powerful machine to support IOME on the real-world datasets. While Kosarak and POS are sparse datasets, IOME requires that the entire dataset (not just the sparse representation) is loaded into memory to perform a random rotation the algorithm uses as a preprocessing step. As a consequence, we ran the binary experiments using Python 3.6.9 on a machine with 512GB RAM, on 2x Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz (2x 14-core CPU). Even on this more powerful machine with multiprocessing using all cores, a single run of IOME took at least 26 minutes on Kosarak, and at least 9 minutes on POS. In comparison, PLAN spent an average of 12 and 9 seconds running on Kosarak and POS, respectively.

7.3.1 *Gaussian data.*

All results are shown in Figure 3 and Figure 4. Additionally, in Appendix G we also include $d = 2048$ for the experiments. Figure 3 (a) shows Gaussian A, where PLAN and IOME have comparable accuracy. The same behavior is observed for smaller ρ, as shown in Figure 4 (a). Both plots show expected behavior that aligns with our theoretical results.

Figure 3 (b) shows Gaussian B, where PLAN performs better than IOME for $\alpha > 1$. For $\alpha \leq 1$ the error between PLAN and IOME is similar. This is expected behavior, as $\alpha = 0$ represents the same input data as in Gaussian A. Notice how PLAN approaches the empirical mean as α grows for both $\rho = 0.5$ and $\rho = 1$. For smaller values of ρ, IOME is less robust, as can be seen by the error bars in Figure 4 (b), whereas PLAN retains robust error for all values of α using the same settings.

Figure 3 (c) shows Gaussian C, where we compare against the empirical mean since sampling error is larger than the noise error for PLAN in this case. As expected, PLAN increases its advantage over IOME as d grows. Since we have subtracted the empirical mean the error scale is smaller, which is why we observe larger error bars in this setting. The difference between IOME and PLAN's error is further showcased in Figure 4 (c), where PLAN has a lower error, but the variance increases with d.

7.3.2 *Binary data.*

All results are shown in Figure 5. Figure 5 (a) shows Binary, where PLAN has an advantage over IOME for $\alpha < 1$ which increases as α decreases. This is the expected behavior, as PLAN is able to exploit the skew in variance whereas IOME treats every dimension the same.

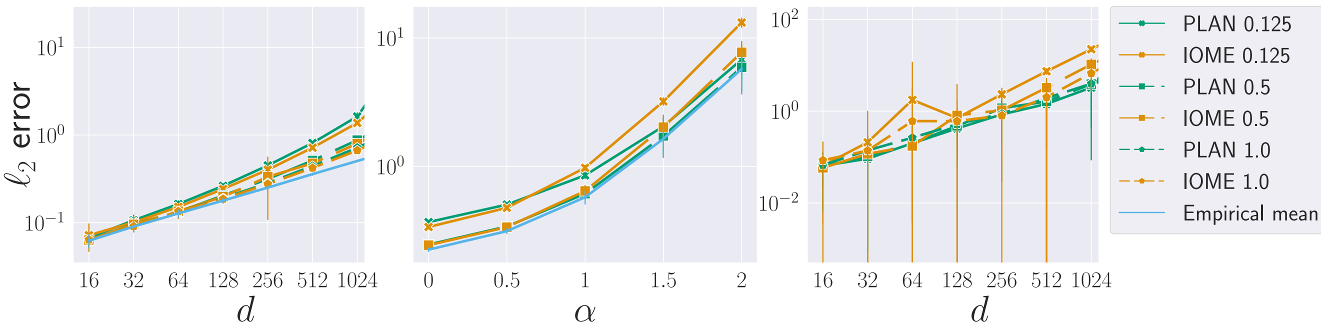
Figure 5 (b) shows Kosarak. As we can see, PLAN outperforms IOME for sufficiently large values of ρ. For small ρ ($\rho \leq 0.125$), PLAN is running in an invalid setting — our assumptions on rank error are not fulfilled in these cases.

Figure 5 (c) shows POS. PLAN has a slight advantage compared to INSTANCE-OPTIMAL MEAN ESTIMATION in this case, which decreases as ρ grows.

8 RELATED WORK

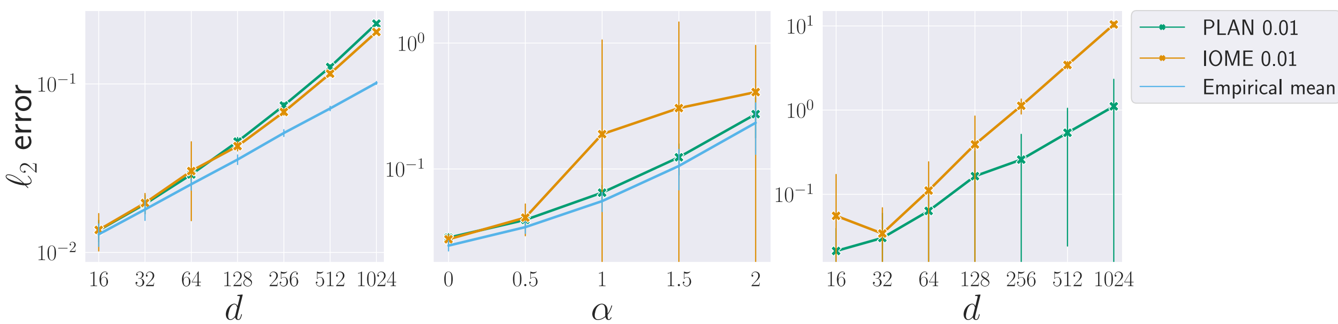
Our work builds on concepts from multiple areas within the literature on differential privacy. We provide an overview of the most closely related work.

Statistical private mean estimation. There is a large, recent literature on statistical estimation for d-dimensional distributions under differential privacy, mainly focusing on the case of Gaussian or subgaussian distributions [2, 5, 8–10, 17, 19, 23, 26, 28, 31, 32]. The error on mean estimates is generally expressed in terms of Mahalanobis distance, which is natural if we want the error to be preserved under affine transformations. Some of these efficient estimators are even *robust* against adversarial changes to the input



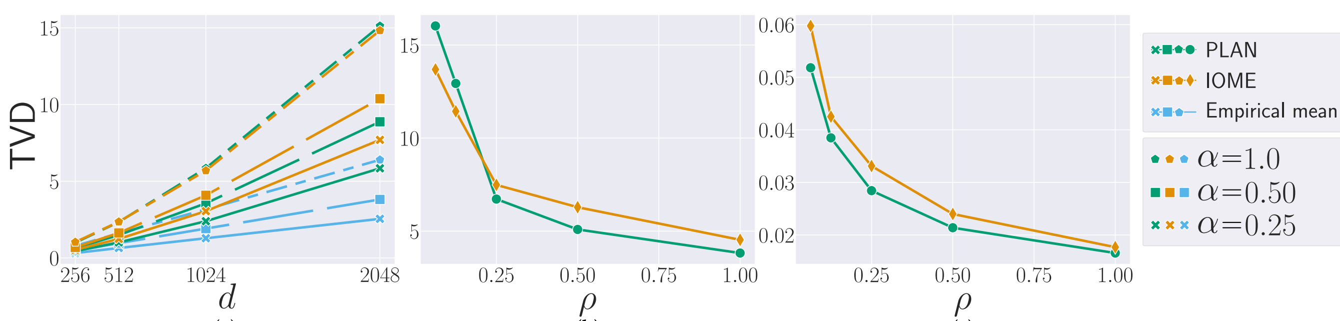
(a) (b) (c)

Figure 3: ℓ_2 error for synthetic Gaussian data when varying (a) dimensions with data without a skew, (b) skewness of the variances, and (c) dimensions for skewed data — note that we compute error relative to the empirical mean rather than the statistical mean in this experiment as sampling error dominates in this setting. Also notice the different scales on the y-axis.



(a) (b) (c)

Figure 4: ℓ_2 error for Gaussian A, Gaussian B, and Gaussian C with $\rho = 0.01$ which implies (ε, δ)-approximate DP with $\varepsilon < 1$ for $\delta \approx 10^{-6}$. To tolerate the rank error, we reuse the parameters from the original experiments and set $n = 100\,000$, and $d = 64$ for (b).



(a) (b) (c)

Figure 5: (a) Synthetic binary data, varying the ratio of 0s to 1s (b) Kosarak dataset (c) POS dataset

data [2, 32]. Other estimators work even for rather heavy-tailed distributions [29]. What all these estimators have in common is that the nominal dimension d influences the privacy-utility trade-off such that higher-dimensional vectors have a worse trade-off. To our best knowledge, the algorithm among these that has been shown to work best in practical (non-adversarial) settings is the CoinPress algorithm of Biswas et al. [8].

Adapting to the data. The best private mean estimation algorithms are near-optimal for worst-case d-dimensional distributions in view of known lower bounds [14]. However, it is natural to consider ways of improving the privacy-utility trade-off whenever the input distribution has some structure. One way of going beyond the worst case is by privately identifying low-dimensional structure [3, 21, 22, 37]. Such methods effectively reduce the mean estimation problem to an equivalent problem with a dimension smaller than d. However, we are not aware of any work showing this approach to be practically relevant for mean estimation.

Another approach for adapting to the data is *instance optimality*, introduced by Asi and Duchi [6] and studied in the context of mean estimation by Huang et al. [24] who use ℓ_2 error (or mean square error) as the utility metric. The goal is optimality, i.e. matching lower bounds, for a class of inputs with a given diameter but no further structure. Huang et al. [24] found that their private mean estimation algorithm often has smaller error than CoinPress in practice. Because of this, and since they also aim to minimize an ℓ_p error, this algorithm was chosen as our main point of comparison.

Neither of the mentioned approaches takes *skew* in the data distribution into account, so we believe this is a novel aspect of our work in the context of mean estimation. However, we mention that privacy budgeting in skewed settings has recently been studied in the context of multi-task learning Krichene et al. [33]. Also, the related setting of mean estimation with heterogeneous data (where the sensitivity with respect to different clients' data can differ) was recently studied by Cummings et al. [15].

Clipping. An important aspect of private mean estimation for unbounded distributions, in theory and practice, is how to perform *clipping* to reduce the sensitivity. This has in particular been studied in the context of differentially private stochastic gradient descent [4, 11, 34, 36]. Though clipping introduces bias, Kamath et al. [27] have shown that this is unavoidable without additional assumptions.

Huang et al. [24] used a clipping method designed to cut off a carefully chosen, small fraction of the data points. The clipping done in PLAN follows the same pattern, though it is applied only after carefully scaling data according to the coordinate variances. Thus, it corresponds to clipping to an axis-aligned ellipsoid.

To formally bound clipping error one can either express the error in terms of the diameter of the dataset or analyze the error under some assumption on the data distribution. Both approaches are explored in Huang et al. [24], but in this paper we have chosen to focus on the latter.

9 CONCLUSION AND FUTURE WORK

We introduce Private Limit Adapted Noise (PLAN), a family of algorithms for differentially private mean estimation of d-dimensional data. PLAN exploits skew in data's variance to achieve better ℓ_p error. In the case of ℓ_2 error we achieve a particularly clean bound, namely error proportional to $\|\boldsymbol{\sigma}\|_1$. This is never worse than the error of $\sqrt{d}\|\boldsymbol{\sigma}\|_2$ obtained by previous methods and gives an improvement up to a factor of up to $\sqrt{d}$ when $\boldsymbol{\sigma}$ is skewed. While the privacy guarantees hold for any input, the error bounds hold for independently sampled data from distributions that follow a well-defined assumption on concentration.

Finally, we implement two PLAN instantiations and empirically evaluate their utility. Practice follows theory — PLAN outperforms the current state-of-the-art for skewed datasets, and is able to perform competitively for datasets without skewed variance. To aid practitioners in implementing their own PLAN, we summarize some practical advice based on our lessons learned.

Advice for practitioners. When implementing a PLAN instantiation, practitioners should pose the following questions:

(1) Is there a suitable estimator for the variances σ_i^2 of the data distribution?
(2) Can a tighter bound on the clipping universe ($M\sqrt{d}$) be used?
(3) How robust is PLAN for my given settings, i.e., will the rank error be too high and cause PLAN to fail?

We give examples of how to answer these questions in our evaluation. To answer Question 1, we derive *private* variance estimators tuned to the data distribution. As for Question 2, when the distribution is $(\boldsymbol{\sigma}, p)$-well concentrated, much better bounds on the universe size can be derived by using Assumption (5) in Definition 4.4. Finally, answering Question 3, robustness must carefully be evaluated using the assumption on minimum ρ values and maximum dimensionality d. Both parameters in conjunction give minimum requirements to the required sample size n.

Future work. We conclude with some possible future directions.

- An interesting avenue to explore would be to capture skew in the input vector that is not necessarily visible in the standard basis. For example, one could use private PCA to rotate the space into a basis in which coordinates are nearly independent, and then apply PLAN.
- Though our algorithm chooses optimal parameters within a class of mechanisms, we have not ruled out that an entirely different approach could have better performance. We conjecture that for any choice of $\boldsymbol{\sigma}$ there exists an input distribution for which our mechanism achieves an optimal trade-off up to logarithmic factors.
- Finally, we rely on having access to reasonable estimates of coordinate variances (the diagonal of the covariance matrix). It would be interesting to study this problem in its own right. Covariance estimation is a common problem in differential privacy (see e.g. [26]), but it is likely that estimating the entire covariance matrix is strictly harder than estimating the diagonal.

ACKNOWLEDGMENTS

Lebeda, Nelson, and Pagh carried out this work at Basic Algorithms Research Copenhagen (BARC), supported by the VILLUM Foundation grant 16582. Nelson and Pagh were also supported by Providentia, a Data Science Distinguished Investigator grant from Novo Nordisk Fonden.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Daniel Alabi, Pravesh K. Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. 2022. Privately Estimating a Gaussian: Efficient, Robust and Optimal. arXiv:2212.08018 [cs, math, stat]

[3] Kareem Amin, Travis Dick, Alex Kulesza, Andres Munoz, and Sergei Vassilvitskii. 2019. Differentially Private Covariance Estimation. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[4] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially Private Learning with Adaptive Clipping. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 17455–17466.

[5] Hassan Ashtiani and Christopher Liaw. 2022. Private and Polynomial Time Algorithms for Learning Gaussians and Beyond. In *Proceedings of Thirty Fifth Conference on Learning Theory*. PMLR, 1075–1076.

[6] Hilal Asi and John C Duchi. 2020. Instance-Optimality in Differential Privacy via Approximate Inverse Sensitivity Mechanisms. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 14106–14117.

[7] Austin R. Benson, Ravi Kumar, and Andrew Tomkins. 2018. A Discrete Choice Model for Subset Selection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, Marina Del Rey CA USA, 37–45. https://doi.org/10.1145/3159652.3159702

[8] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. 2020. Coin-Press: Practical Private Mean and Covariance Estimation. *Advances in Neural Information Processing Systems* 33 (2020), 14475–14485.

[9] Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakynthinou. 2021. Covariance-Aware Private Mean Estimation Without Private Covariance Estimation. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 7950–7964. https://proceedings.neurips.cc/paper/2021/hash/42778ef0b5805a96f9511e20b5611fce-Abstract.html

[10] Gavin Brown, Samuel B. Hopkins, and Adam Smith. 2023. Fast, Sample-Efficient, Affine-Invariant Private Mean and Covariance Estimation for Subgaussian Distributions. https://doi.org/10.48550/arXiv.2301.12250 arXiv:2301.12250 [cs]

[11] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2022. Automatic Clipping: Differentially Private Deep Learning Made Easier and Stronger. arXiv:2206.07136 [cs]

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

[13] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography (Lecture Notes in Computer Science)*, Martin Hirt and Adam Smith (Eds.). Springer, Berlin, Heidelberg, 635–658. https://doi.org/10.1007/978-3-662-53641-4_24

[14] T. Tony Cai, Yichen Wang, and Linjun Zhang. 2021. The Cost of Privacy: Optimal Rates of Convergence for Parameter Estimation with Differential Privacy. *The Annals of Statistics* 49, 5 (Oct. 2021), 2825–2850. https://doi.org/10.1214/21-AOS2058

[15] Rachel Cummings, Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Mean Estimation with User-level Privacy under Data Heterogeneity. *Advances in Neural Information Processing Systems* 35 (Dec. 2022), 29139–29151.

[16] Sergey Denisov, H Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Thakurta. 2022. Improved Differential Privacy for SGD via Optimal Private Linear Operators on Adaptive Streams. *Advances in Neural Information Processing Systems* (2022).

[17] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. 2020. Differentially Private Confidence Intervals. https://doi.org/10.48550/arXiv.2001.02285 arXiv:2001.02285 [cs, stat]

[18] Devdatt P. Dubhashi and Alessandro Panconesi. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press.

[19] John Duchi, Saminul Haque, and Rohith Kuditipudi. 2023. A Fast Algorithm for Adaptive Private Mean Estimation. arXiv:2301.07078 [cs, stat]

[20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography (Lecture Notes in Computer Science)*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, 265–284.

[21] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. 2014. Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing (STOC '14)*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/2591796.2591883

[22] Moritz Hardt and Eric Price. 2014. The Noisy Power Method: A Meta Algorithm with Applications. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.

[23] 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 2022)*. Association for Computing Machinery, New York, NY, USA, 1406–1417. https://doi.org/10.1145/3519935.3519947

[24] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. *Advances in Neural Information Processing Systems* 34 (2021), 25993–26004. https://proceedings.neurips.cc/paper/2021/file/da54dd5a0398011cdfa50d559c2c0ef8-Paper.pdf

[25] NL Johnson, S Kotz, and N Balakrishnan. 1994. Chi-square distributions including Chi and Rayleigh. *Continuous univariate distributions* 1 (1994), 415–493.

[26] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. 2019. Privately Learning High-Dimensional Distributions. In *Proceedings of the Thirty-Second Conference on Learning Theory*. PMLR, 1853–1902.

[27] Gautam Kamath, Argyris Mouzakis, Matthew Regehr, Vikrant Singhal, Thomas Steinke, and Jonathan Ullman. 2023. A Bias-Variance-Privacy Trilemma for Statistical Estimation. https://doi.org/10.48550/arXiv.2301.13334 arXiv:2301.13334 [cs, math, stat]

[28] Gautam Kamath, Argyris Mouzakis, Vikrant Singhal, Thomas Steinke, and Jonathan Ullman. 2022. A Private and Computationally-Efficient Estimator for Unbounded Gaussians. In *Proceedings of Thirty Fifth Conference on Learning Theory*. PMLR, 544–572.

[29] Gautam Kamath, Vikrant Singhal, and Jonathan Ullman. 2020. Private Mean Estimation of Heavy-Tailed Distributions. In *Proceedings of Thirty Third Conference on Learning Theory*. PMLR, 2204–2235.

[30] Haim Kaplan, Shachar Schnapp, and Uri Stemmer. 2022. Differentially Private Approximate Quantiles. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 10751–10761. https://proceedings.mlr.press/v162/kaplan22a.html ISSN: 2640-3498.

[31] Vishesh Karwa and Salil Vadhan. 2018. Finite Sample Differentially Private Confidence Intervals. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 9 pages. https://doi.org/10.4230/LIPICS.ITCS.2018.44

[32] Pravesh Kothari, Pasin Manurangsi, and Ameya Velingker. 2022. Private Robust Estimation by Stabilizing Convex Relaxations. In *Proceedings of Thirty Fifth Conference on Learning Theory*. PMLR, 723–777.

[33] Walid Krichene, Prateek Jain, Shuang Song, Mukund Sundararajan, Abhradeep Thakurta, and Li Zhang. 2023. Multi-Task Differential Privacy Under Distribution Skew. arXiv:2302.07975 [cs, stat]

[34] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. https://doi.org/10.48550/arXiv.1710.06963 arXiv:1710.06963 [cs]

[35] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press.

[36] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X. Yu, Sashank J. Reddi, and Sanjiv Kumar. 2019. AdaCliP: Adaptive Clipping for Private SGD. https://doi.org/10.48550/arXiv.1908.07643 arXiv:1908.07643 [cs, stat]

[37] Vikrant Singhal and Thomas Steinke. 2021. Privately Learning Subspaces. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 1312–1324.

[38] Adam Smith. 2011. Privacy-Preserving Statistical Estimation with Optimal Convergence Rates. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (STOC '11)*. Association for Computing Machinery, New York, NY, USA, 813–822. https://doi.org/10.1145/1993636.1993743

[39] Andreas Winkelbauer. 2012. Moments and absolute moments of the normal distribution. *arXiv preprint arXiv:1209.4340* (2012).

A USEFUL STATEMENTS FROM PROBABILITY THEORY

Lemma A.1 (Bernstein's inequality). *Let $X_1, \dots, X_n$ be independent zero-mean random variables. Suppose that $|X_i| \leq M$ almost surely for all i. Then for all $t > 0$,*

$$\Pr\left[\sum_{i=1}^{n} X_i \geq t\right] \leq \exp\left(-\frac{t^2/2}{\sum_{i=1}^{n} \mathbf{E}[X_i^2] + Mt/3}\right).$$

Lemma A.2 ([39, Equation (18)]). *The pth absolute moment of a zero-centered Gaussian distribution for any $p > 0$ is*

$$\mathbf{E}\left[\left|\mathcal{N}\left(0, \sigma^2\right)\right|^p\right] = \sigma^p \cdot \frac{2^{p/2}\Gamma\left(\frac{p+1}{2}\right)}{\sqrt{\pi}}.$$

Lemma A.3 (Generalized Chernoff-Hoeffding Bound [18]). *Let $X := \sum_{1 \leq i \leq n} X_i$ where $X_i, 1 \leq i \leq n$ are independently distributed in $[a_i, b_i]$ for $a_i, b_i \in \mathbf{R}$. Then for all $t > 0$*

$$\Pr[|X - \mathbf{E}[X]| \geq t] \leq 2\exp\left(\frac{-t^2/2}{\sum_i (a_i - b_i)^2}\right).$$

B PROOF OF THEOREM 4.5

Theorem B.1. *For sufficiently large $n = \tilde{\Omega}(\max(\sqrt{d/\rho}, \rho^{-1}))$, PLAN with parameters $k = \sqrt{n}$ and $\rho_1 = \rho_2 = \rho_3 = \rho/3$ is ρ-zCDP. If* *inputs are independently sampled from a $(\boldsymbol{\sigma}, p)$-well concentrated distribution, the mean estimate has expected ℓ_p error*

$$\tilde{O}\left(1 + \frac{\|\boldsymbol{\sigma}\|_p}{\sqrt{n}} + \frac{\|\boldsymbol{\sigma}\|_{2p/(p+2)}}{n\sqrt{\rho}}\right),$$

where $\tilde{O}$ suppresses polylogarithmic dependencies on $1/\beta$, n, d, and the bound M on the ℓ_∞ norm of inputs.

Proof. We proceed in the same three steps as in the proof of Theorem 4.3 in Section 4.

By Lemma 4.2, with probability at least $1-\beta$, all $|\mu_i - \tilde{\mu}_i| \le \sigma_i$. In this case, $\|\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}}\|_p = O(\|\boldsymbol{\sigma}\|_p)$.

Let $J = \{j \in \{1, \dots, n\} \mid \|\boldsymbol{y}^{(j)}\|_2 > C\}$ denote the set of indices of vectors affected by clipping in PLAN. By the triangle inequality and assuming the bound of part 1 holds,

$$\begin{aligned}\sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}})\right\|_p &\le \sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\right\|_p + |J|\cdot\|\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}}\|_p \\ &\le \sum_{j\in J}\left\|(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\right\|_p + 2|J|\cdot\|\boldsymbol{\sigma}\|_p \ .\end{aligned}$$

By assumption (4) the probability that $\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_p^p > t\|\boldsymbol{\sigma}\|_p^p$ is exponentially decreasing in t, so setting $t \ge \max\{\ln d, \log(n/\beta)\}$ we have $\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_p = \tilde{O}(\|\boldsymbol{\sigma}\|_p)$ for all $j = 1, \dots, n$ with probability at least $1-\beta$. Using the triangle inequality again we can now bound

$$\sum_{j\in J}\|\boldsymbol{x}^{(j)} - \boldsymbol{\mu}\|_p = \tilde{O}(|J|\cdot\|\boldsymbol{\sigma}\|_p) \ .$$

The same line of argument as in Section 4.2 shows that $|J| = \tilde{O}(k + \sqrt{1/\rho})$. Thus, the clipping error can be bounded by $O((k + \sqrt{1/\rho})\|\boldsymbol{\sigma}\|_p)$, and setting $k = n^{-1/2}$ balances the clipping error with the sampling error $\frac{\|\boldsymbol{\sigma}\|_p}{\sqrt{n}}$.

Lastly, we consider the error due to noise. First, we find a bound on the clipping threshold C. By the triangle inequality, we may bound

$$\begin{aligned}\|\boldsymbol{y}^{(j)}\|_2 &= \left\|(\boldsymbol{x}^{(j)} - \tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2 \\ &\le \left\|(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2 + \left\|(\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-\frac{1}{p+2}}\right\|_2\end{aligned}$$

The ith coordinate of the first vector $(\boldsymbol{x}^{(j)} - \boldsymbol{\mu})\,\hat{\Sigma}^{-1/(p+2)}$ equals $(x_i^{(j)} - \mu_i)/\hat{\sigma}_i^{2/(p+2)}$, so assumption (5) implies that the length of the first vector is $\tilde{O}(\sqrt{\|\boldsymbol{\sigma}^{2p/(p+2)}\|_1})$ with high probability. By using that the ith coordinate of $|\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}}| \le \boldsymbol{\sigma}$, $(\boldsymbol{\mu} - \tilde{\boldsymbol{\mu}})\,\hat{\Sigma}^{-1/(p+2)}$ has absolute value $\tilde{O}(\sqrt{\|\boldsymbol{\sigma}^{2p/(p+2)}\|_1})$ as well. The clipping value of C is bounded by the maximum length of a vector $\|\boldsymbol{y}^{(j)}\|_2$, and thus $C^2 = \tilde{O}(\|\boldsymbol{\sigma}^{2p/(p+2)}\|_1)$, with probability at least $1-\beta$.

The scaled noise vector $\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}$ has distribution $\mathcal{N}(\mathbf{0}, \frac{2C^2}{\rho_3}\hat{\Sigma}^{2/(p+2)})$. Using $\hat{\sigma}_i < \left(\sigma_i^{\frac{2p}{p+2}} + \|\boldsymbol{\sigma}^{\frac{2p}{p+2}}\|_1/d\right)^{\frac{p+2}{2p}}$ and Lemma A.2, we conclude

$$\begin{aligned}\mathbf{E}[\|\boldsymbol{\eta}\,\hat{\Sigma}^{1/(p+2)}\|_p^p] &= O\left(\frac{2^p C^p}{\rho_3^{p/2}}\sum_{i=1}^{d}\hat{\Sigma}_{ii}^{p/(p+2)}\right) \\ &= O\left(\frac{2^p C^p}{\rho_3^{p/2}}\sum_{i=1}^{d}\hat{\sigma}_i^{2p/(p+2)}\right) \\ &= \tilde{O}\left(\frac{2^p\|\boldsymbol{\sigma}\|_{2p/(p+2)}^p}{\rho_3^{p/2}}\right) \ .\end{aligned}$$

The result of Theorem 4.5 is achieved by putting together the different error terms as in the proof of Theorem 4.3. Finally, with probability at most β a failure event will happen. In that case, the largest ℓ_p error is bounded by the diameter of $[-M, M]^d$, which is $2Md^{1/p}$. Thus, the contribution to the expected ℓ_p error is $2Md^{1/p}\beta$, and can be made $O(1)$ by setting $\beta = \frac{1}{Md^{1/p}}$. □

C CHOOSING THE SCALING VALUE

Here we discuss the choice of scaling of PLAN. For parameters p and $\hat{\boldsymbol{\sigma}} \in \mathbf{R}^d$ we scale each coordinate i by $\boldsymbol{\sigma}^{-2/(p+2)}$ on line 5 of Algorithm 1. When returning the mean estimate on line 8 each coordinate i is scaled back by $\boldsymbol{\sigma}^{2/(p+2)}$. The choice of scaling affects how noise is distributed across coordinates. In this section we consider a simpler setting with independent queries with known sensitivities where we restrict ourselves to adding noise such that there is no clipping error. We choose the scaling parameters that minimizes the pth moment of the noise.

Consider the problem of privately answering d independent real-valued queries with different sensitivities. We denote the sensitivity of the ith query as Δ_i and the vector of all sensitivities as $\boldsymbol{\Delta}$. We want to minimize the pth moment of the noise. That is, let $\boldsymbol{\eta} \in \mathbf{R}^d$ denote the vector where $\boldsymbol{\eta}_i$ is the noise added to the ith query. We want to minimize $\mathbf{E}[\|\boldsymbol{\eta}\|_p^p]$. As discussed in section 1, standard approaches either adds a lot of noise for queries with low or high sensitivity as we show here. We can think of the queries as a d-dimensional query which we can release by adding noise from $\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}\mathbf{I})$ where the sensitivity is $\bar{\Delta} = \sqrt{\sum_{i\in[d]}\Delta_i^2} = \|\Delta\|_2$. This approach adds noise of the same magnitude to each coordinate which might be excessive for queries with very low sensitivity. The privacy budget is effectively split up between queries weighted by $\boldsymbol{\Delta}^2$. Alternatively we could split the privacy budget evenly and answer each query independently by adding noise from $\mathcal{N}(\mathbf{0}, d\cdot\frac{\Delta_i^2}{2\rho})$. This approach is equivalent to first scaling each coordinate by Δ_i^{-1} and scaling back after adding noise from $\mathcal{N}(\mathbf{0}, \frac{d}{2\rho}\mathbf{I})$. However, this approach adds too much noise to coordinates with high sensitivity. We want to balance the two approaches and as such PLAN uses an in-between scaling value. We can think of our approach as allocating more privacy budget for queries with high sensitivity, but not so much that it creates large errors for queries with low sensitivity.

Let $\boldsymbol{x} \in \mathbf{R}^d$ be the answer to the d queries and let $\boldsymbol{s} \in \mathbf{R}^d$ be a scaling vector. Let $\boldsymbol{y} = \boldsymbol{x}S^{-1}$ denote the scaled down answers where

S is a diagonal matrix with $\boldsymbol{s}$ on the diagonal. We can privately release $\boldsymbol{y}$ by adding noise from $\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}\mathbf{I})$ where $\bar{\Delta}^2 = \sum_{i\in[d]} \Delta_i^2/s_i^2$. When scaling back we multiply the noisy answers with S. Our goal is to minimize the pth moment. Specifically, we want to choose $\boldsymbol{s}$ to minimize

$$\mathbf{E}\left[\|\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}\mathbf{I})S\|_p^p\right] ,$$

where $\bar{\Delta}$ and S are defined as above.

Lemma C.1. *The pth moment of the algorithm above is*

$$\mathbf{E}\left[\|\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}S^2)\|_p^p\right] = \frac{\Gamma\left(\frac{p+1}{2}\right)}{\rho^{p/2}\sqrt{\pi}} \sum_{i\in[d]} (\bar{\Delta}\cdot s_i)^p$$

Proof. It follows from linearity of expectation by summing over each coordinate. By Lemma A.2 we have

$$\mathbf{E}\left[|\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}s_i)|^p\right] = \left(\frac{\bar{\Delta}^2}{2\rho}s_i^2\right)^{p/2} \frac{2^{p/2}\Gamma\left(\frac{p+1}{2}\right)}{\sqrt{\pi}} = \frac{\Gamma\left(\frac{p+1}{2}\right)}{\rho^{p/2}\sqrt{\pi}}(\bar{\Delta}\cdot s_i)^p$$

□

Since the fraction outside the sum is not affected by our choice of $\boldsymbol{s}$ we just have to find $\boldsymbol{s}$ that minimizes $\sum_{i\in[d]}(\bar{\Delta}\cdot s_i)^p$.

Lemma C.2. *The pth moment of the noise is minimized for*

$$s_i \propto \Delta_i^{2/(p+2)}.$$

Proof. First notice that multiplying all entries in $\boldsymbol{s}$ by the same scalar does not change the result since the magnitude of noise added to y would be scaled by the inverse of the scalar. As such, we can restrict our search to $\boldsymbol{s}$ such that $\bar{\Delta}^2 = \sum_{i\in[d]}(\Delta_i/s_i)^2 = 1$. We also restrict our search to $\boldsymbol{s}$ with no negative values since negative values would simply mirror the input and has no impact on the sensitivity. As such, any $\boldsymbol{s}$ that satisfies those constraints and minimizes $\sum_{i\in[d]} s_i^p$ also minimizes the error.

We use the method of Lagrange multipliers to find $\boldsymbol{s}$. The method is used to find local maxima or minima of a function subject to equality constraints. In our case, it turns our that there is only one minimum which implies that it is the global minimum. We want to minimize $\sum_{i\in[d]} s_i^p$ subject to the restriction $\sum_{i\in[d]}(\Delta_i/s_i)^2 - 1 = 0$. The minimum is at the stationary point of the Lagrangian function

$$\mathcal{L}(\boldsymbol{s},\lambda) = \sum_{i\in[d]} s_i^p + \lambda\left(\sum_{i\in[d]}(\Delta_i/s_i)^2 - 1\right),$$

where $\lambda \in \mathbf{R}$ is the Lagrange multiplier.

We start by finding the partial derivative of the Lagrangian function with respect to s_i.

$$\frac{\partial}{\partial s_i}\mathcal{L}(\boldsymbol{s},\lambda) = \frac{\partial}{\partial s_i} s_i^p + \lambda(\Delta_i/s_i)^2 = ps_i^{p-1} - 2\lambda\Delta_i^2/s_i^3 .$$

We then find the root of this function with respect to s_i, that is,

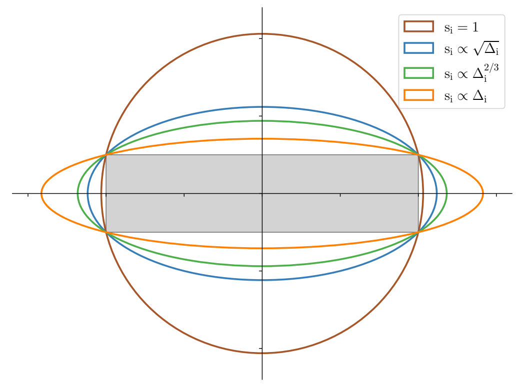


Figure 6: Shape of noise for different scaling parameters.

$$\begin{aligned} ps_i^{p-1} - 2\lambda\Delta_i^2/s_i^3 &= 0 \\ s_i^{p+2} &= 2\lambda\Delta_i^2/p \\ s_i &= \Delta_i^{2/(p+2)}\cdot\gamma , \end{aligned}$$

where $\gamma = (2\lambda/p)^{1/(p+2)}$ is a scalar. This finishes the proof. □

Lemma C.3. *Let $\boldsymbol{s} \in \mathbf{R}^d$ be defined as in Lemma C.2. Then the pth moment of the mechanism described in this section is*

$$\mathbf{E}\left[\|\mathcal{N}(\mathbf{0}, \frac{\bar{\Delta}^2}{2\rho}\mathbf{I})S\|_p^p\right] = \frac{\Gamma\left(\frac{p+1}{2}\right)}{\rho^{p/2}\sqrt{\pi}}\|\Delta\|_{2p/(p+2)}^p$$

Proof. From Lemma C.1 we know that the above equality holds if $\sum_{i\in[d]}(\bar{\Delta}/s_i)^p = \|\Delta\|_{2p/(p+2)}^p$. Continuing the calculations from the proof of Lemma C.2 we see that for $\bar{\Delta} = 1$ we have

$$\begin{aligned} \sum_{i\in[d]}(\Delta_i/s_i)^2 &= \bar{\Delta} \\ \frac{1}{\gamma^2}\sum_{i\in[d]}\Delta_i^{2p/(p+2)} &= 1 \\ \gamma &= \sqrt{\sum_{i\in[d]}\Delta_i^{2p/(p+2)}} \end{aligned}$$

Finally, we find that the pth moment is proportional to

$$\begin{aligned} \sum_{i\in[d]} s_i^p &= \sum_{i\in[d]}\left(\Delta_i^{2/(p+2)}\sqrt{\sum_{i\in[d]}\Delta_i^{2p/(p+2)}}\right)^p \\ &= \left(\sum_{i\in[d]}\Delta_i^{2p/(p+2)}\right)^{p/2}\sum_{i\in[d]}\Delta_i^{2p/(p+2)} \\ &= \left(\sum_{i\in[d]}\Delta_i^{2p/(p+2)}\right)^{1+p/2} = \|\Delta\|_{2p/(p+2)}^p . \end{aligned}$$

□

Figure 6 shows an example of this scaling in two dimensions, where the sensitivity of one query is 4 times as high as the other. The green and blue lines show the shape of noise when optimizing for the first and second moment, respectively. In both cases we use most of the privacy budget for the horizontal axis which allows us to add less noise than the orange line where the budget is split evenly. But we are still able to add less noise to the vertical axis as opposed to the brown line without scaling.

We use the scaling as described above for plan. However, in our setting we do not use sensitivities of queries. Instead we scale each coordinate based on the estimate of the standard deviation. The intuition is the same: we spend more privacy budget on coordinates with high standard deviation but we still add less noise to coordinates with low standard deviation.

D ALGORITHMS FOR VARIANCE ESTIMATION

While plan (Algorithm 1) assumes that estimates on the standard deviations are known, such estimates have to be computed in a differentially private manner. Two such ways were described in Section 7 and we will provide more details and empirical results in this section.

We remark that the standard attempt to estimate the variance from a mean estimate $\tilde{\mu}$ is

$$\left(1/n \sum_{i=1}^{n} x_i^2\right) - \tilde{\mu}^2$$

If $x_i \in [-M, M]$, the sensitivity of this function is M^2/n, which, depending on the application, means that too much noise must be added.

D.1 A Generic Variance Estimation Algorithm

Algorithm 2 VarianceEstimate

1. **Input:** Samples $x^{(1)}, \ldots, x^{(n)} \in \mathbf{R}$ from $\mathcal{D}$
2. **Parameters:** M, ρ, k
3. Split $x^{(1)}, \ldots, x^{(n)}$ into $n' = \lfloor \frac{n}{2k} \rfloor$ groups $G_1, \ldots, G_{n'}$.
4. For each $i \in \{1, \ldots, n'\}$: For $G_i = (\hat{x}_i^{(1)}, \ldots, \hat{x}_i^{(2k)})$, let $\hat{y}^{(i)} = \sum_{j=1}^{k} \left(\hat{x}_i^{(2j)} - \hat{x}_i^{(2j+1)}\right)^2 /2$.
5. **return** $\tilde{\sigma}^2 \leftarrow \frac{\text{PrivQuantile}_\rho^{M^2}(\hat{y}^{(1)}, \ldots, \hat{y}^{(n')}, 1/2)}{k}$.

Given a distribution $\mathcal{D}$ with mean μ and variance σ^2, we showed in Section 7 that for $X, Y \sim \mathcal{D}$, $\mathbf{E}\left[(X-Y)^2/2\right] = \sigma^2$. Algorithm 2 is a generalization of the approach used for Gaussian in Section 7. For Gaussian data, we made use of the fact that $(X-Y)^2/2$ is χ_2 distributed and it is well-known how to translate an approximate median to an approximate mean.

Lemma D.1. *Let $\beta > 0, \rho > 0$, and $n = \tilde{\Omega}(\rho^{-1/2})$. Let $\mathcal{D}$ be a distribution over $\mathbf{R}$ with mean μ and variance $\sigma^2 \geq 1$. For a constant κ, assume that $\mathbf{E}[(X-Y)^4] \leq \kappa\sigma^4$. With probability at least $1-\beta$, Algorithm 2 using $k = 16\kappa$ returns an estimate $\hat{\sigma}^2$ such that $\sigma^2/2 \leq \hat{\sigma}^2 \leq 3\sigma^2/2$.*

Proof. Fix a group G_i, $1 \leq i \leq n'$. Since $\mathbf{E}[(X-Y)^2/2] = \sigma^2$, we know that $\mathbf{E}\left(\hat{y}^{(i)}\right) = k\sigma^2$. By Chebychev's inequality,

$$\Pr(|\hat{y}^{(i)} - \mathbf{E}[\hat{y}^{(i)}]| > k\sigma^2/2) \leq \frac{\mathrm{Var}(\hat{y}^{(i)})}{(k\sigma^2/2)^2} \leq \frac{k \cdot \mathbf{E}[(X-Y)^4]}{(k\sigma^2/2)^2} \leq \frac{1}{4},$$

using our assumption on $\mathbf{E}[(X-Y)^4]$ and our choice of k.

As in the proof of Lemma 4.2, with probability at least $1 - \exp(-\Omega(n'))$, there are more than $2/3n'$ groups i for which

$$\frac{|\hat{y}^{(i)} - \mathbf{E}[\hat{y}^{(i)}]|}{k} \leq \sigma^2/2.$$

As long as the private quantile selection returns an element $\hat{\sigma}^2$ with rank error at most $n'/6$, $\sigma^2/2 \leq \hat{\sigma}^2 \leq 3\sigma^2/2$. By Lemma 2.9 assuming $n > K\sqrt{\log(M^2)\log(\log(M^2)/\beta)/(2\rho)} = \tilde{\Omega}(\rho^{-1/2})$, this is true with probability at least $1-\beta$. □

In contrast to the naïve estimator mentioned above, this estimator has only a logarithmic dependency on the input universe. In contrast to it, it does not improve from increased sample size above a *minimum sample size* in relation to the parameter k that is necessary to guarantee that the rank error is at most $n/(6 \cdot 2k)$.

By running Algorithm 2 on each coordinate independently with target probability $1 - \beta/d$, and using a union bound, we may summarize:

Corollary D.2. *Let $\beta > 0, \rho > 0$, and $n = \tilde{\Omega}(\sqrt{d/\rho})$. Let $\mathcal{D}$ be a distribution over $\mathbf{R}^d$ with mean μ_i and variance $\sigma_i^2 \geq 1$ on each coordinate $i \in \{1, \ldots, d\}$. For a constant κ, assume that $\mathbf{E}[(X-Y)^4] \leq \kappa\sigma^4$. For each i, with probability at least $1-\beta$, Algorithm 2 on each coordinate using $k = 16\kappa$ returns an estimate $(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_d^2)$ such that $\sigma_i^2/2 \leq \hat{\sigma_i}^2 \leq 3\sigma_i^2/2$.*

The result from the corollary can be used to find an estimate that *almost* satisfies the conditions in (1). Given $\mathcal{D}$, use the estimator to compute individual estimates $\hat{\sigma_i}^2$. Next, set $\bar{\sigma}_i \leftarrow 2\hat{\sigma}_i + \|\hat{\boldsymbol{\sigma}}\|$. With probability at least $1-\beta$, $\sigma_i \leq \bar{\sigma}_i \leq 2(\sigma_i + \|\hat{\boldsymbol{\sigma}}\|_1)$. Using such an estimate in the analysis carried out in Section 4.3, the expected error due to noise is increased by a factor 2.

While hidden in the $\tilde{\Omega}(.)$ notation, only having $n' = n/(2k)$ samples to choose a private quantile might be incompatible with the rank error of the input domain and the dimensionality. In this case, we can use more groups to "boost" n', but we need to adjust ρ because each element is potentially present multiple times. To cover variances that are smaller than 1, let $\sigma^2_{\min}$ be a minimum bound on the variance. Then, use PrivQuantile with $T = \Theta(\log(M/\sigma^2_{\min}))$ to quantize the input space in steps of $\sigma^2_{\min}$, which gives a logarithmic depends on $1/\sigma^2_{\min}$.

D.2 Variance estimation for Gaussian Data

In the case that we know that the data is distributed as $\mathcal{N}(\mu, \sigma^2)$, we can tune the variance estimation more towards the distribution as follows. Given an estimate $\tilde{\mu}$ on μ, we can estimate the variance as follows:

$$\tilde{\sigma} \leftarrow \text{PrivQuantile}_\rho^M(x^{(1)}, \ldots, x^{(n)}, .841) - \tilde{\mu}$$

It is a well-known property of the Gaussian distribution that the .841 quantile is approximately the value $\mu + \sigma$. However, when

ρ	σ^2	Method	Relative error
0.001	0.001	direct-075	0.332
	0.001	direct-0841	0.033
	0.001	direct-09	0.279
	0.001	general (k=1)	0.027
	0.001	general (k=4)	**0.017**
	1	direct-075	0.308
	1	direct-0841	0.042
	1	direct-09	0.314
	1	general (k=1)	0.025
	1	general (k=4)	**0.012**
0.01	0.001	direct-075	0.313
	0.001	direct-0841	0.024
	0.001	direct-09	0.305
	0.001	general (k=1)	0.011
	0.001	general (k=4)	**0.007**
	1	direct-075	0.322
	1	direct-0841	0.011
	1	direct-09	0.265
	1	general (k=1)	0.020
	1	general (k=4)	**0.006**

Table 2: Comparison of variance estimation algorithms. Variants named direct directly use a quantile of the input to estimate the standard deviation. Variants named general use Algorithm 2 with $k = 1$ and $k = 4$, and use the median to mean conversion for χ_2 distributed random variables.

estimating the quantile privately we have to adjust for the rank error of the quantile selection, so aiming for this exact quantile may be unwise.

We run the following experiment: we sample $n = 10000$ from $\mathcal{N}(10, \sigma^2)$ with $\sigma^2 \in \{0.001, 1\}$. For $\rho \in \{10^{-3}, 10^{-2}\}$ we compare (i) three different methods that use different quantiles of the input data (.75, .841, and .9) to (ii) two different instantiations of Algorithm 2 for $k = 1$ and $k = 4$. Since we know that $(X - Y)^2$ is χ_2 distributed, we use the mean to median transformation and divide the approximate median by $(1 - (2/(9k)))^3$. Each parameter setting is run 100 times and we report on the average relative error $\frac{|\hat{\sigma}^2 - \sigma^2|}{\sigma^2}$. Table 2 reports on empirical results for the variance estimation. We summarize that Algorithm 2 is more accurate than direct estimation for both values of k, and guarantees very small relative error even for small σ^2.

E EXPERIMENTAL EVALUATION ON CORRELATED DATA

The experiments for Gaussian data in Section 7 focused on diagonal matrices, i.e., uncorrelated vectors. To validate the trends of our results for correlated data, we design the following additional synthetic test cases:

- Gaussian A': Given $d \geq 1$, let $\Lambda = (\Lambda_{i,j})_{1 \leq i,j \leq d}$ with

$$\Lambda_{i,j} \sim \textbf{Uniform}(0, 1/\sqrt{d})$$

for $1 \leq i < j \leq d$. Let $\Sigma = \Lambda \cdot \Lambda^T + I_d$, where I_d is the $d \times d$ identity matrix. $\Sigma_{i,j}$ is uniformly distributed in $[0, 1)$ for $i \neq j$, and uniform in $[1, 2)$ on the diagonal. Vectors are drawn from $\mathcal{N}(0^d, \Sigma)$, and the setup is identical to Gaussian A in Table 1.
- Gaussian C': Let $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$ be defined as for the case Gaussian C in Table 1. We set $\Sigma = (\Sigma_{i,j})_{1 \leq i,j \leq d}$ with $\Sigma_{i,j} = 0.5\sigma_i\sigma_j$ for $i \neq j$, and $\Sigma_{i,i} = \sigma_i^2$, so we have a correlation coefficient of .5 between the i-th and the j-th coordinate. The remaining parameters are chosen as for Gaussian C in Table 1.

Figure 7 shows the results of the experiments.[3] For non-skewed data with "random" correlations (Gaussian A'), we observe the same trends as for uncorrelated data in Figure 3(a); PLAN and IOME provide similar average ℓ_2 error for a given d, with INSTANCE-OPTIMAL MEAN ESTIMATION having a slightly larger error than in the uncorrelated setting, moving the competitors slightly further away from each than in the uncorrelated setting.

In the setting of skewed data with a constant correlation coefficient, see Figure 7(b), PLAN behaves very similarly to the uncorrelated case, depicted in Figure 3(c). In contrast, IOME has a much larger average ℓ_2 error compared to its results on uncorrelated data. Closer inspection of the results showed that this is likely due to the re-centering step for the individual coordinates. Ignoring these outliers for $d = 1024$, the median error of IOME (subtracting the empirical mean) is 22.61, 45.19, 66.5 for $\rho = 1, .5, .125$, respectively. PLAN has median error 3.41, 4.76, 9.40 in comparison.

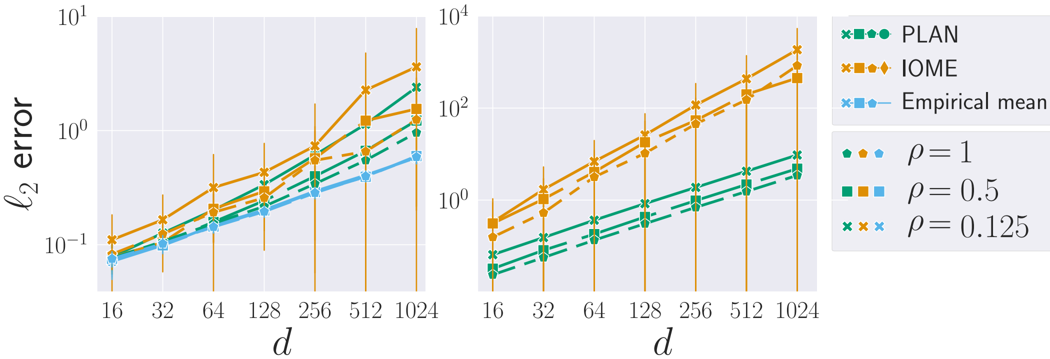
F EXPERIMENTAL EVALUATION OF PLAN WITHOUT SCALING

Section 6 focused on the setting in which we run PLAN without variance estimates, i.e., avoid the scaling step. In this section, we carry out the experiments from Section 7 with this instantiation of PLAN as competitor. From the result of Theorem 6.1, we expect this variant of PLAN to behave asymptotically equivalent to IOME.

Figure 8 reports on the results on Gaussian data, comparable with Figure 3. As we can see from the experiments, PLAN without scaling recovers the behavior from IOME, presenting a slight improvement over IOME's error. Unsurprisingly, it improves over PLAN in the non-skewed case (which unnecessarily spends privacy budget on an inaccurate variance estimation). However, as soon as data has skew, the coordinate-wise scaling of PLAN improves the error.

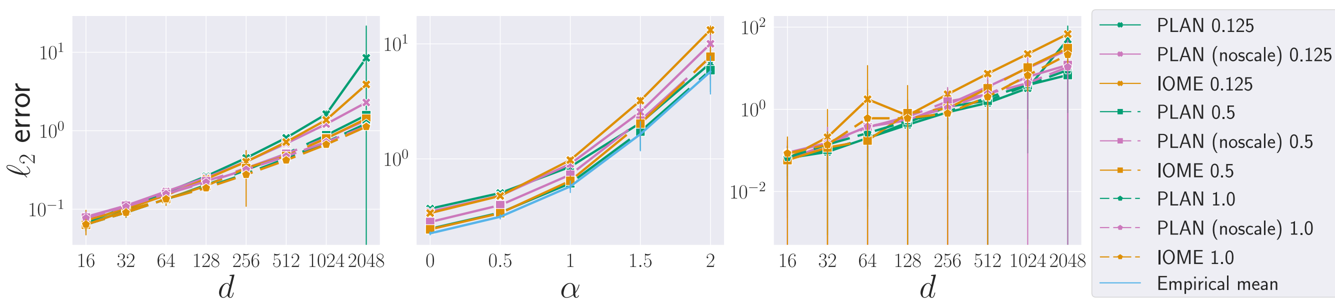
Figure 9 summarizes the empirical utility of PLAN without scaling for synthetic, and the real-world binary datasets. As in the Gaussian case, PLAN without scaling recovers the behavior from IOME. It again improves the average error slightly. For binary data, their is only a slight gap to the proposed variance-aware scaling.

[3] We did not evaluate an analogous setting to Gaussian B in Section 7, because (i) the case $\alpha = 0$ and $\alpha = 2$ is covered by our two experiments, and (ii) the way of generating different Σ in the two cases does not allow for a smooth transition from one setting to the other. For example, Gaussian A' cannot use Σ from Gaussian C' since Σ would not have full rank.



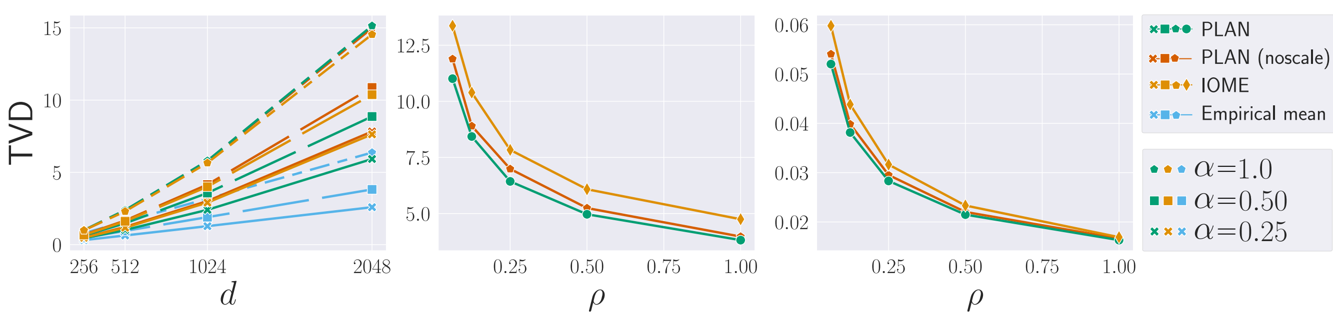
(a) (b)

Figure 7: ℓ_2 error for correlated, synthetic data: (a) Gaussian A' (random correlation, no skew) and (b) Gaussian C' (correlation coefficient .5, skewed data).



(a) (b) (c)

Figure 8: ℓ_2 error for synthetic Gaussian data using PLAN without scaling: We vary (a) dimensions with data without a skew, (b) skewness of the variances, and (c) dimensions for skewed data, cf. Figure 3.

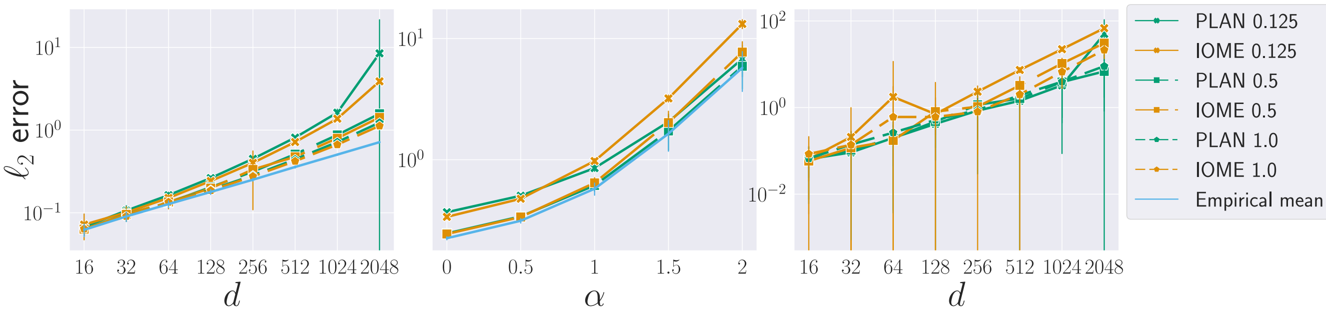


(a) (b) (c)

Figure 9: Experiments on synthetic, and real-world binary data.

G EMPIRICAL EVALUATION OF PLAN WITH EXTENDED DIMENSIONS

In Figure 10 we present the plots from the Gaussian case in Section 7.3, but here including $d = 2048$. For symmetry we also include Gaussian B in Figure 10 (b), with no changes from the original plot in Figure 3. In Figure 10 (a) we see PLAN and IOME have comparable accuracy until $d = 2048$, when PLAN performs worse (for $\rho = 0.125$). This is because for such small ρ, the assumptions on the rank error no longer hold. This is expected behavior as PLAN spends some additional budget estimating $\boldsymbol{\sigma}^2$ in comparison to IOME. Similarly in Figure 10 (c) we observe that the error reported by PLAN has a high variance due to the rank error increasing. Again, this only happens for the smallest ρ value.



(a) (b) (c)

Figure 10: ℓ_2 error for synthetic Gaussian data when varying (a) dimensions with data without a skew, (b) skewness of the variances, and (c) dimensions for skewed data — note that we compute error relative to the empirical mean rather than the statistical mean in this experiment as sampling error dominates in this setting. Also notice the different scales on the y-axis.