Sharper Bounds for Chebyshev Moment Matching with Applications to Differential Privacy and Beyond

Cameron Musco
UMass Amherst
cmusco@cs.umass.edu

Christopher Musco
New York University
cmusco@nyu.edu

Lucas Rosenblatt
New York University
lucas.rosenblatt@nyu.edu

Apoorv Vikram Singh
New York University
apoorv.singh@nyu.edu

Abstract

We study the problem of approximately recovering a probability distribution given noisy measurements of its Chebyshev polynomial moments. We sharpen prior work, proving that accurate recovery in the Wasserstein distance is possible with more noise than previously known.

As a main application, our result yields a simple "linear query" algorithm for constructing a differentially private synthetic data distribution with Wasserstein-1 error $\tilde{O}(1/n)$ based on a dataset of n points in $[-1, 1]$. This matches a recent breakthrough of Boedihardjo, Strohmer, and Vershynin [Probab. Theory. Rel., 2024], which uses a more complex "superregular random walk" method to beat an $O(1/\sqrt{n})$ accuracy barrier inherent to earlier approaches. We also show how to extend our result to higher dimensions in a black-box way: given any mechanism that achieves error $\tilde{O}(1/n)$ for data on $[-1, 1]$, we can obtain a private distribution with Wasserstein-1 error $\tilde{O}(1/n^{1/d})$ for data in $[-1, 1]^d$. This compliments previous non-black box results.

We believe our new moment-based recovery bound will find applications beyond differential privacy. We illustrate one example in numerical linear algebra: by improving an approach of Braverman, Krishnan, and Musco [STOC 2022], our result yields a faster algorithm for estimating the spectral density of a symmetric matrix up to small error in the Wasserstein distance.

1 Introduction

The problem of recovering a probability distribution (or its parameters) by "matching" noisy estimates of the distribution's moments goes back over 100 years to the work of Chebyshev and Pearson [Pea94; Pea36; Fis11]. Moment matching continues to find a wide variety of applications, both in traditional statistical problems [KMV10; MV10; RSS14; WY19; WY20; FL23] and beyond. For example, moment matching is now widely used for solving eigenvalue estimation problems in numerical linear algebra and computational chemistry [WWAF06; CKSV18; CTU21].

One powerful and general result on moment matching for distributions with *bounded support* is that the method directly leads to approximations with small error in the Wasserstein-1 distance (a.k.a. earthmover's distance). Concretely, given a distribution p supported on $[-1, 1]$,[1] any distribution q for which $\mathbb{E}_{x\sim p}[x^i] = \mathbb{E}_{x\sim q}[x^i]$ for $i = 1, \ldots, k$ satisfies $W_1(p, q) \leq O(1/k)$, where W_1 denotes the Wasserstein-1 distance [KV17; CTU21]. I.e., to compute an ϵ-accurate approximation to p, it suffices to compute p's first $O(1/\epsilon)$ moments and to return any distribution q with the same moments.

Unfortunately, the above result is highly sensitive to noise, so is difficult to apply in the typical setting where, instead of p's exact moments, we only have access to *estimates* of the moments (e.g., computed from a sample). In particular, it can be shown that the accuracy of these estimates needs to be proportional to $1/2^k$ if we want to approximate p up to Wasserstein error $O(1/k)$ [JMSS23]. In other words, distribution approximation is *poorly conditioned* with respect to the standard moments.

1.1 Chebyshev moment matching

One way of avoiding the poor conditioning of moment matching is to move from the standard moments, $\mathbb{E}_{x\sim p}[x^i]$, to a better conditioned set of "generalized" moments. Specifically, significant prior work [WWAF06; WJF+16; BKM22] leverages *Chebyshev moments* of the form $\mathbb{E}_{x\sim p}[T_i(x)]$, where T_i is the i^{th} Cheybshev polynomial of the first kind, defined as:

$$T_0(x) = 1 \qquad T_1(x) = x \qquad T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x), \text{ for } i \geq 2.$$

The Cheybshev moments are known to be less noise sensitive than the standard moments: instead of exponentially small error, $\tilde{O}(1/k)$ additive error[2] in computing p's first k Chebyshev moments suffices to find a distribution that is $O(1/k)$ close to p in Wasserstein distance (see, e.g., Lemma 3.1 in [BKM22]). This fact has been leveraged to obtain efficient algorithms for distribution estimation in a variety of settings. For example, Chebyshev moment matching leads to $O(n^2/\mathsf{poly}(\epsilon))$ time algorithms for estimating the eigenvalue distribution (i.e., the spectral density) of an $n\times n$ symmetric matrix A to error $\epsilon\|A\|_2$ in the Wasserstein distance [BKM22].

Cheybshev moment matching has also been leveraged for *differentially private synthetic data generation*. In this setting, p is the uniform distribution over a dataset $x_1, \ldots, x_n$. The goal is to find some q that approximates p, but in a differentially private way, which informally means that q cannot reveal too much information about any one data point, x_j [DNRRV09; RLP+20; MMSM22]. Such a q can be used to generate private synthetic data that is representative of the original data. One approach to solving this problem is to compute p's Chebyshev moments, and then add noise,

[1] The result easily extends to p supported on any finite interval by shifting and scaling the distribution to $[-1, 1]$. For a general interval $[a, b]$, matching k moments yields error $O(|a - b|/k)$ in Wasserstein-1 distance.

[2] Throughout, we let $\tilde{O}(z)$ denote $O(z \log^c(z))$ for constant c.

which is known to ensure privacy [DR14]. Then, one can find a distribution q that matches the noised moments. It has been proven that, for a dataset of size n, this approach yields a differentially private distribution q that is $\tilde{O}(1/n^{1/3})$ close to p in Wasserstein distance [WJF+16].

1.2 Our contributions

Despite the success of Chebyshev moment matching, including for the applications discussed above, there is room for improvement. For example, for private distribution estimation, it is known that alternative methods can achieve error $\tilde{O}(1/n)$ in Wasserstein distance for a dataset of size n [BSV24], improving on the existing $\tilde{O}(1/n^{1/3})$ bound for moment matching. For eigenvalue estimation, existing moment matching methods obtain an optimal quadratic dependence on the matrix dimension n, but a suboptimal polynomial dependence on the accuracy parameter, ϵ.

The main contribution of this work is to resolve these limitations by proving a sharper bound on the accuracy with which the Chebyshev moments need to be approximated to recover a distribution to high accuracy in the Wasserstein distance. Formally, we prove the following:

Theorem 1. *Let p, q be distributions supported on $[-1, 1]$. For any positive integer k, if the distributions' first k Chebyshev moments satisfy*

$$\sum_{j=1}^{k} \frac{1}{j^2} \left(\mathop{\mathbb{E}}_{x\sim p} T_j(x) - \mathop{\mathbb{E}}_{x\sim q} T_j(x) \right)^2 \leq \Gamma^2, \tag{1}$$

then, for an absolute constant c,[3]

$$W_1(p, q) \leq c \cdot \left(\frac{1}{k} + \Gamma \right). \tag{2}$$

As a special case, (1) holds if for all $j \in \{1, \ldots, k\}$,

$$\left| \mathop{\mathbb{E}}_{x\sim p} T_j(x) - \mathop{\mathbb{E}}_{x\sim q} T_j(x) \right| \leq \Gamma \cdot \sqrt{\frac{j}{1 + \log k}}.$$[4] \tag{3}

Theorem 1 characterizes the Chebyshev moment error required for a distribution q to approximate p in Wasserstein distance. The main requirement, (1), involves a weighted ℓ_2 norm with weights $1/j^2$, which reflects the diminishing importance of higher moments on the Wasserstein distance. Referring to (3), we obtain a bound of $W_1(p, q) \leq O(1/k)$ as long as q's j^{th} moment differs from p's by $\tilde{O}(\sqrt{j}/k)$. In contrast, prior work requires error $\tilde{O}(1/k)$ *for all of the first k moments* to ensure the same Wasserstein distance bound (Lemma 3.1, [BKM22]).

As a corollary of Theorem 1, we obtain the following algorithmic result:

Corollary 2. *Given estimates $\hat{m}_1, \ldots, \hat{m}_k$ satisfying $\sum_{j=1}^{k} \frac{1}{j^2} \left(\mathbb{E}_{x\sim p} T_j(x) - \hat{m}_j \right)^2 \leq \Gamma^2$, Algorithm 1 returns a distribution q with $W_1(p, q) \leq c' \cdot \left(\frac{1}{k} + \Gamma \right)$ for a fixed constant c' in $\mathsf{poly}(k)$ time.*

Algorithm 1 simply solves a linearly-constrained least-squares regression problem to find a distribution q supported on a sufficiently fine grid whose moments are nearly as close to those of p as $\hat{m}_1, \ldots, \hat{m}_k$. We then obtain Corollary 2 by applying Theorem 1 to bound $W_1(p, q)$. The linear

[3] Concretely, we prove a bound of $\frac{36}{k} + \sqrt{\pi}\Gamma$, although we believe the constants can be improved, at least to $\frac{2\pi}{k} + \sqrt{\pi}\Gamma$, and possibly further. See Section 3 for more discussion.

[4] Throughout, we let $\log k$ denote the natural logarithm of k, i.e., the logarithm with base e.

constraints ensure that q is positive and sums to one (i.e, that it is a valid distribution). This problem is easily solved using off-the-shelf software: in our experiments, we use a solver from MOSEK [DB16; MOS19].

Like prior work, our proof of Theorem 1 (given in Section 3) relies on tools from polynomial approximation theory. In particular, we leverage a constructive version of Jackson's theorem on polynomial approximation of Lipschitz functions via "damped Chebyshev expansions" [Jac12]. Lipschitz functions are closely related to approximation in Wasserstein distance through the Kantorovich-Rubinstein duality: $W_1(p,q) = \max_{1-\text{Lip } f} \int_{-1}^{1} f(x)(p(x) - q(x))dx$. In contrast to prior work, we couple Jackson's theorem with a tight "global" characterization of the coefficient decay in the Chebyshev expansion of a Lipschitz function. In particular, we prove that any 1-Lipschitz function f with Chebyshev expansion $f = \sum_{j=0}^{\infty} c_j T_j$ has coefficients that satisfy $\sum_{j=1}^{\infty} j^2 c_j^2 \leq \pi$. Prior work only leveraged the well-known "local" decay property, that the i^{th} coefficient has magnitude bounded by $O(1/j)$ [Tre19]. This property is implied by our bound, but much weaker.

1.3 Applications

To illustrate the applicability of Theorem 1, we highlight two concrete applications.

Differentially Private Synthetic Data. Privacy-enhancing technologies seek to protect individuals' data without preventing learning from the data. For theoretical guarantees of privacy, *differential privacy* [DR14] has become the industry standard, having been used in massive data products like the US Census, and included as a core tenet of the recent Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence [Bid23; Abo18; AAS+19].

Concretely, we are interested in the ubiquitous notion of *approximate differential privacy*:
Definition 3 (Approximate Differential Privacy)**.** A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if, for all pairs of neighboring datasets X, X', and all subsets $\mathcal{B}$ of possible outputs:

$$\mathbb{P}[\mathcal{A}(X) \in \mathcal{B}] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{A}(X') \in \mathcal{B}] + \delta\,.$$

In our setting, a dataset X is a collection of n points in a bounded interval (without loss of generality, $[-1, 1]$). Two datasets of size n are considered "neighboring" if all of their data points are equal except for one. Intuitively, Definition 3 ensures that the output of $\mathcal{A}$ is statistically indistinguishable from what the output would be if any one individual's data was replaced with something arbitrary.

There exist differentially private algorithms for a wide variety of statistical tasks [JL14; LLSY17; MTV+20]. One task of primary importance is *differentially private data synthesis*. Here, the goal is to generate *synthetic data* in a differentially private way that matches the original dataset along a set of relevant statistics or distributional properties. The appeal of private data synthesis is that, once generated, the synthetic data can be used for a wide variety of downstream tasks: a separate differentially private algorithm is not required for each potential use case.

Many methods for private data synthesis have been proposed [HLM12; ZCPSX17; LVW21; AAS+19; ABK+21; RHR+23; DSB21]. Such methods offer strong empirical performance and a variety of theoretical guarantees, e.g., that the generated synthetic data can effectively answer a fixed set of data analysis queries with high accuracy [HLM12; MMSM22]. Recently, there has been interest in algorithms with more general statistical guarantees – e.g., guarantees that the synthetic data comes from a distribution close in statistical distance to the original data [WJF+16; BSV24; HVZ23]. By leveraging Theorem 1, we contribute the following result to this line of work:

Theorem 4. *Let $X = \{x_1, \ldots, x_n\}$ be a dataset with each $x_j \in [-1, 1]$. Let p be the uniform distribution on X. There is an (ϵ, δ)-differentially private algorithm based on Chebyshev moment matching that, in $\mathsf{poly}(\epsilon n)$ time, returns a distribution q satisfying for a fixed constant c_1,*

$$\mathbb{E}[W_1(p, q)] \leq c_1 \frac{\log(\epsilon n)\sqrt{\log(1/\delta)}}{\epsilon n}.$$

Moreover, for any $\beta \in (0, 1/2)$, $W_1(p, q) \leq c_1 \frac{\sqrt{\log(1/\beta)+\log(\epsilon n)}\sqrt{\log(\epsilon n)\log(1/\delta)}}{\epsilon n}$ with probability $\geq 1-\beta$.

The distribution q returned by the algorithm behind Theorem 4 is represented as a discrete distribution on $O(\epsilon n)$ points in $[-1, 1]$, so can be sampled from efficiently to produce a synthetic dataset of arbitrary size. Typically, δ is chosen to be $1/\mathsf{poly}(n)$, in which case Theorem 4 essentially matches a recent break-through result of Boedihardjo, Strohmer, and Vershynin [BSV24], who give an $(\epsilon, 0)$-differentially private method with expected Wasserstein-1 error $O(\log^{3/2}(n)/(\epsilon n))$. Like that method, we improve on a natural barrier of $\tilde{O}(1/(\epsilon\sqrt{n}))$ error that is inherent to "private histogram" methods, which offer an alternative approach to approximation in the Wasserstein-1 distance [HRMS10; XWG10; QYL13; XZX+13; DR14; ZXX16; LLSY17].

The result of [BSV24] avoids this barrier by introducing a "superregular random walk" to directly add noise to $x_1, \ldots, x_n$ using a correlated distribution based on a Haar basis. Our method is simpler, more computationally efficient, and falls directly into the empirically popular *Select, Measure, Project* framework for differentially private synthetic data synthetis [VAA+22; LVW21]. In particular, we simply compute the Chebyshev moments of p, add independent noise to each moment using the standard Gaussian mechanism [DKMMN06; MM09], then apply Theorem 1 to recover q . We verify the strong empirical performance of the method in Section 7. A method similar to ours was analyzed in prior work [WJF+16], although that work obtains a Wasserstein error bound of $\tilde{O}(1/\epsilon n^{1/3})$. Our tighter connection between Chebyshev moment estimation and distribution approximation proven in Theorem 1 allows us to obtain a significantly better dependence on n.

Theorem 4 is proven in Section 4. We also generalize the result to higher dimensions in Section 6, proving that we can obtain expected Wasserstein error $\tilde{O}(1/(\epsilon n)^{1/d})$, which matches prior work up to a logarithmic factor [BSV24]. Notably, our reduction is completely black-box: we show that such a result can be obtained given access to *any* differentially private algorithm that achieves error $\tilde{O}(1/\epsilon n)$ for one dimensional data. This reduction might be of independent interest.

Finally, we note that [HVZ23] also claims a faster and simpler alternative to [BSV24]. While the simplest method in that paper has error scaling with $\tilde{O}(1/\sqrt{n})$, they describe a more complex method that matches our $\tilde{O}(1/n)$ result up to a $\log(n)$ factor. While we are not aware of an implementation of that algorithm, empirically comparing alternative methods for generating synthetic data with Wasserstein distance guarantees would be a productive line of future work. Additionally, we note that, in concurrent work to ours, Feldman et al. study a stronger notion of *instance optimal* private distribution estimation in the Wasserstein distance [FMST24]. It would be interesting to explore if Cheybshev moment matching has any applications in this setting.

Matrix Spectral Density Estimation. Spectral density estimation (SDE) is a problem of central importance in numerical linear algebra. In the standard version of the problem, we are given a symmetric $n \times n$ matrix A, which has real-valued eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Letting p denote the uniform distribution over these n eigenvalues, the goal is to output q which is close to p in the Wasserstein distance. An approximate spectral density can be useful in determining a

variety of properties of A's eigenvalue spectrum – e.g., if its eigenvalues are decaying rapidly or if they follow a distribution characteristic of random matrices.

Efficient SDE algorithms were originally studied in computational physics and chemistry, and are widely used to compute the "density of states" of quantum systems [Ski89; SR94; MAP20]. More recently, the problem has found applications in network science [DBB19; CKSV18; JKMSS24], deep learning [CKS91; PSG18; MM19; YGKM20], optimization [GKX19], and beyond [LXES19; CTU22].

Many popular SDE algorithms are based on Chebyshev moment matching [WWAF06]. The i^{th} Chebyshev moment of the spectral density is equal to $\mathbb{E}_{x\sim p} T_j(x) = \frac{1}{n}\sum_{j=1}^{n} T_i(\lambda_j) = \text{tr}(\frac{1}{n}T_i(A))$. This trace can be estimated using a small number of matrix-vector products with $T_i(A)$, using stochastic trace estimation techniques like Hutchinson's estimator [Hut90; MMMW21]. Since T_i is a degree-i polynomial, each matrix-vector product with $T_i(A)$ requires just i products with A. Thus, with a small number of products with A, we can obtain approximate moments for use in estimating p. Importantly, this approach can be applied even in the common *implicit* setting, where we do not have direct accress to the entries of A, but can efficiently multiply the matrix by vectors [AT11].

Recently, [BKM22] gave a theoretical analysis of Chebyshev moment-matching for SDE, along with the closely related and practically popular Kernel Polynomial Method [WWAF06]. They show that when n is sufficiently large, specifically, $n = \tilde{\Omega}(1/\epsilon^2)$, then $\tilde{O}(1/\epsilon)$ matrix-vector products with A (and $\mathsf{poly}(1/\epsilon)$ additional runtime) suffice to output q with $W_1(p,q) \leq \epsilon \|A\|_2$, where $\|A\|_2 = \max_i |\lambda_i|$ is A's spectral norm. This improves on the naive "implicit method" of simply recovering A from n matrix-vector products and running an explicit eigenvalue method, like the QR algorithm, to compute A's spectral density. Moreover, if A is a dense matrix, the algorithm runs in $\tilde{O}(n^2/\epsilon + \mathsf{poly}(1/\epsilon))$ time, which can be much faster than the $O(n^\omega)$ time required to compute p directly via a full eigendecomposition.

While the result of [BKM22] also holds without the restriction that $n = \tilde{\Omega}(1/\epsilon^2)$, for smaller n it suffers from a polynomially worse $1/\epsilon$ dependence in the number of matrix-vector products required. By leveraging Theorem 1, we resolve this issue, showing that $\tilde{O}(1/\epsilon)$ matrix-vector products suffice for any n. Roughly, by weakening the requirements on how well we approximate A's spectral moments, Theorem 1 allows us to decrease the accuracy with which moments are estimated, and thus the number of matrix-vector products used by Hutchinson's method. Formally, we prove:

Theorem 5. *There is an algorithm that, given $\epsilon \in (0,1)$ and symmetric $A \in \mathbb{R}^{n\times n}$ with spectral density p, computes $\tilde{O}\left(\frac{1}{\epsilon}\right)$ matrix-vector products[5] with A and uses $\tilde{O}(1/\epsilon^{3.25})$ additional time to output a distribution q such that, with probability $\geq 1-\delta$, $W_1(p,q) \leq \epsilon \|A\|_2$.*

In terms of matrix-vector products, Theorem 5 cannot be improved by more than logarithmic factors. In particular, by leveraging a recent lower bound on estimating the trace of a positive definite matrix [WZZ22], we prove that $\Omega(1/\epsilon)$ matrix-vector products with A are necessary to approximate the spectral density p up to error $\epsilon\|A\|_2$. The proof is given in Appendix C. Together with Theorem 5, this lower bound resolves, up to logarithmic factors, the complexity of the SDE problem in the "matrix-vector query model" of computation, where cost is measured via matrix-vector products with A. Understanding this model has become a core topic in theoretical work on numerical

[5] Formally, we prove a bound of $\min\left\{n, O\left(\frac{1}{\epsilon}\cdot\left(1+\frac{\log^2(1/\epsilon)\log^2(1/(\epsilon\delta))}{n\epsilon}\right)\right)\right\}$ matrix-vector products, which, for constant δ, is at worst $O\left(\frac{\log^2(1/\epsilon)}{\epsilon}\right)$, but actually on the order of $O(\frac{1}{\epsilon})$ for all $\epsilon \leq cn/\log^3 n$ for constant c.

linear algebraic, as it generalizes other important models like the matrix sketching and Krylov subspace models [SWYZ21]. Our work contributes to recent progress on establishing tight upper and lower bounds for central problems like linear system solving [BHSW20], eigenvector approximation [SER18], trace estimation [JPWZ24], and more [CDLLN23; BN23; ACK+24; CKHMM24].

2 Preliminaries

Before our main analysis, we introduce notation and technical preliminaries.

Notation. We let $\mathbb{Z}_{\geq 0}$ denote the natural numbers and $\mathbb{Z}_{>0}$ denote the positive integers. For a vector $x \in \mathbb{R}^k$, we let $\|x\|_2 = \sqrt{\sum_{i=1}^k x_i^2}$ denote the Euclidean norm. We often work with functions from $[-1, 1] \to \mathbb{R}$. For two such functions, f, g, we use the convenient inner product notation:

$$\langle f, g \rangle \stackrel{\text{def}}{=} \int_{-1}^{1} f(x)g(x)\,dx.$$

We will often work with products, quotients, sums, and differences of two functions f, g, which are denoted by $f \cdot g$, f/g, $f + g$, and $f - g$, respectively. E.g., $[f \cdot g](x) = f(x)g(x)$. For a function $f : [-1, 1] \to \mathbb{R}$, we let $\|f\|_\infty$ denote $\|f\|_\infty = \max_{x\in[-1,1]} |f(x)|$ and $\|f\|_1 = \int_{-1}^{1} |f(x)|\,dx$.

Wasserstein Distance. This paper concerns the approximation of probability distributions in the Wasserstein-1 distance. We consider the standard version involving the Euclidean metric:

Definition 6 (Wasserstein-1 Distance, Euclidean Metric)**.** Let p and q be two distributions on $\mathbb{R}^d$. Let $Z(p, q)$ be the set of all couplings between p and q, i.e., the set of distributions on $\mathbb{R}^d \times \mathbb{R}^d$ whose marginals equal p and q. Then the Wasserstein-1 distance between p and q is:

$$W_1(p, q) = \inf_{z\in Z(p,q)} \left[\mathop{\mathbb{E}}_{(x,y)\sim z} \|x - y\|_2 \right].$$

The Wasserstein-1 distance measures the total cost (in terms of distance per unit mass) required to "transport" the distribution p to q. Alternatively, it has a well-known dual formulation:

Fact 7 (Kantorovich-Rubinstein Duality)**.** *Let p, q be as in Definition 6. Then:*

$$W_1(p, q) = \sup_{1\text{-}Lipschitz\ f} \mathop{\mathbb{E}}_{x\sim p} f(x) - \mathop{\mathbb{E}}_{y\sim q} f(y),$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a Lipschitz function under the Euclidean metric.

We use Fact 7 in the one dimensional setting. In this case, the supremum is over all functions $f : \mathbb{R} \to \mathbb{R}$ satisfying $|f(x) - f(y)| \leq |x - y|$. In our analysis, it will be convenient to assume that, in addition to being Lipschitz, f is smooth, i.e. that it is infinitely differentiable. Since any Lipschitz function can be arbitrarily well approximated by a smooth function, we can do so without changing the distance. In particular, for 1 dimensional distributions on $[-1, 1]$ we have:

$$W_1(p, q) = \sup_{1\text{-Lipschitz, smooth } f} \int_{-1}^{1} f(x)(p(x) - q(x))\,dx = \sup_{1\text{-Lipschitz, smooth } f} \langle f, p - q \rangle. \tag{4}$$

Above we slightly abuse notation and use p and q to denote (generalized) probability density functions[6] instead of the distributions themselves. We will do so throughout the paper.

[6] p and q might correspond to discrete distributions, in which case they will be sums of Dirac delta functions.

Chebyshev Polynomials and Chebyshev Series. Our main result analyzes the accuracy of (noisy) Chebyshev polynomial moment matching for distribution approximation. The Chebyshev polynomials are defined in Section 1.1, and can alternatively be defined on $[-1,1]$ via the trigonometric definition, $T_j(\cos\theta) = \cos(j\theta)$. We use a few basic properties about these polynomials.

Fact 8 (Boundedness and Orthogonality, see e.g. [Hal15])**.** *The Chebyshev polynomials satisfy:*

1. **Boundedness:** $\forall x \in [-1,1]$ *and* $j \in \mathbb{Z}_{\geq 0}$, $|T_j(x)| \leq 1$.
2. **Orthogonality:** *The Chebyshev polynomials are orthogonal with respect to the weight function* $w(x) = \frac{1}{\sqrt{1-x^2}}$. *In particular, for* $i, j \in \mathbb{Z}_{\geq 0}$, $i \neq j$, $\langle T_i \cdot w, T_j \rangle = 0$.

To obtain an orthonormal basis we also define the *normalized* Chebyshev polynomials as follows:

Definition 9 (Normalized Chebyshev Polynomials)**.** The j^{th} *normalized* Chebyshev polynomial, $\bar{T}_j$, is defined as $\bar{T}_j \stackrel{\text{def}}{=} T_j / \sqrt{\langle T_j \cdot w, T_j \rangle}$. Note that $\langle T_j \cdot w, T_j \rangle$ equals π for $j = 0$ and $\pi/2$ for $j \geq 1$.

We define the *Chebyshev series* of a function $f : [-1,1] \to \mathbb{R}$ as $\sum_{j=0}^{\infty} \left\langle f \cdot w, \bar{T}_j \right\rangle \bar{T}_j$. If f is Lipschitz continuous then the Chebyshev series of f converges absolutely and uniformly to f [Tre19, Theorem 3.1]. Throughout this paper, we will also write the Chebyshev series of generalized probability density functions, which could involve Dirac delta functions. This is standard in Fourier analysis, even though the Chebyshev series does not converge pointwise [Lig58]. Formally, any density p can be replaced with a Lipschitz continuous density (which has a convergent Cheybshev series) that is arbitrarily close in Wasserstein distance and the same analysis goes through.

3 Main Results

In this section, we prove our main result, Theorem 1, as well as Corollary 2. To do so, we require two main ingredients. The first is a constructive version of Jackson's theorem on polynomial approximation of Lipschitz functions [Jac30]. A modern proof can be found in [BKM22, Fact 3.2].

Fact 10 (Jackson's Theorem [Jac30])**.** *Let* $f : [-1,1] \to \mathbb{R}$ *be an* ℓ*-Lipschitz function. Then, for any* $k \in \mathbb{Z}_{>0}$, *there are* $k+1$ *constants* $1 = b_k^0 > \ldots > b_k^k \geq 0$ *such that the polynomial* $f_k = \sum_{j=0}^{k} b_k^0 \cdot \left\langle f \cdot w, \bar{T}_j \right\rangle \cdot \bar{T}_j$ *has the property that* $\|f - f_k\|_\infty \leq 18\ell/k$.

It is well-known that truncating the Chebyshev series of an ℓ-Lipschitz function f to k terms leads to error $O(\log k \cdot \frac{\ell}{k})$ in the ℓ_∞ distance [Tre19]. The above version of Jackson's theorem improves this bound by a $\log k$ factor by instead using a *damped* truncated Chebyshev series: each term in the series is multiplied by a positive scaling factor between 0 and 1. We will not need to compute these factors explicitly, but b_k^i has a simple closed form (see [BKM22, Equation 12]).

To bound the Wasserstein distance between distributions p, q, we need to upper bound $\langle f, p - q \rangle$ for every 1-Lipschitz f. The value of Fact 10 is that this inner product is closely approximated by $\langle f_k, p - q \rangle$. Since f_k is a damped Chebyshev series, this inner product can be decomposed as a difference between p and q's Chebyshev moments. Details will be shown in the proof of Theorem 1.

The second ingredient we require is a stronger bound on the decay of the Chebyshev coefficients, $\left\langle f \cdot w, \bar{T}_j \right\rangle$, which appear in Fact 10. In particular, we prove the following result:

Lemma 11 (Global Chebyshev Coefficient Decay)**.** *Let* $f : [-1,1] \to \mathbb{R}$ *be an* ℓ*-Lipschitz, smooth function, and let* $c_j = \left\langle f \cdot w, \bar{T}_j \right\rangle$ *for* $j \in \mathbb{Z}_{\geq 0}$. *Then,* $\sum_{j=1}^{\infty} (jc_j)^2 \leq \pi\ell^2$.

Lemma 11 implies the well known fact that $c_j = O(\ell/j)$ for $j \geq 1$ [Tre08]. However, it is a much stronger bound: if all we knew was that the Chebyshev coefficients are bounded by $O(\ell/j)$, then $\sum_{j=1}^{\infty}(jc_j)^2$ could be unbounded. We show that it can in fact be bounded by $O(\ell^2)$. Informally, the implication is that not all coefficients can saturate the "local" $O(\ell/j)$ constraint at the same time, but rather obey a stronger global constraint, captured by a weighted ℓ_2 norm of the coefficients.

3.1 Proof of Theorem 1

We prove Lemma 11 in Section 3.3. Before doing so, we show how it implies Theorem 1.

Proof of Theorem 1. By (4), to bound $W_1(p,q)$, it suffices to bound $\langle f, p-q\rangle$ for any 1-Lipschitz, smooth f. Let f_k be the approximation to any such f guaranteed by Fact 10. We have:

$$\begin{aligned}\langle f, p-q\rangle = \langle f_k, p-q\rangle + \langle f-f_k, p-q\rangle &\leq \langle f_k, p-q\rangle + \|f-f_k\|_\infty \|p-q\|_1 \\ &\leq \langle f_k, p-q\rangle + \frac{36}{k}. \end{aligned} \tag{5}$$

In the last step, we use that $\|f-f_k\|_\infty \leq 18/k$ by Fact 10, and that $\|p-q\|_1 \leq \|p\|_1 + \|q\|_1 = 2$. So, to bound $\langle f, p-q\rangle$ we turn our attention to bounding $\langle f_k, p-q\rangle$.

For technical reasons, we will assume from here on that p and q are supported on the interval $[-1+\delta, 1-\delta]$ for arbitrarily small $\delta \to 0$. This is to avoid an issue with the Chebyshev weight function $w(x) = 1/\sqrt{1-x^2}$ going to infinity at $x = -1, 1$. The assumption is without loss of generality, since we can rescale the support of p and q by a $(1-\delta)$ factor, and the distributions' moments and Wasserstein distance change by an arbitrarily small factor as $\delta \to 0$.

We proceed by writing the Chebyshev series of the function $(p-q)/w$:

$$\frac{p-q}{w} = \sum_{j=0}^{\infty}\left\langle \frac{p-q}{w}\cdot w, \bar{T}_j\right\rangle \bar{T}_j = \sum_{j=0}^{\infty}\langle p-q, \bar{T}_j\rangle \cdot \bar{T}_j = \sum_{j=1}^{\infty}\langle p-q, \bar{T}_j\rangle \cdot \bar{T}_j. \tag{6}$$

In the last step we use that both p and q are distributions so $\left\langle p-q, \bar{T}_0\right\rangle = 1/\pi - 1/\pi = 0$.

Next, recall from Fact 10 that $f_k = \sum_{j=0}^{k} c'_j \bar{T}_j$, where each c'_j satisfies $|c_j| \leq |\langle f\cdot w, \bar{T}_j\rangle|$. Using (6), the fact that $\langle \bar{T}_i \cdot w, \bar{T}_j\rangle = 0$ whenever $i \neq j$, and that $\langle \bar{T}_j \cdot w, \bar{T}_j\rangle = 1$ for all j, we have:

$$\langle f_k, p-q\rangle = \left\langle f_k \cdot w, \frac{p-q}{w}\right\rangle = \left\langle \sum_{j=0}^{k} c'_j \bar{T}_j \cdot w, \sum_{j=1}^{\infty}\langle p-q, \bar{T}_j\rangle \bar{T}_j\right\rangle = \sum_{j=1}^{k} c'_j \cdot \langle p-q, \bar{T}_j\rangle.$$

Via Cauchy-Schwarz inequality and our global decay bound from Lemma 11, we then have:

$$\begin{aligned}\langle f_k, p-q\rangle = \sum_{j=1}^{k} jc'_j \cdot \frac{\langle p-q, \bar{T}_j\rangle}{j} &\leq \left(\sum_{j=1}^{k} j^2 c_j^2\right)^{1/2} \cdot \left(\sum_{j=1}^{k}\frac{1}{j^2}\langle p-q, \bar{T}_j\rangle^2\right)^{1/2} \\ &\leq \left(\sum_{j=1}^{k} j^2 \langle f\cdot w, \bar{T}_j\rangle^2\right)^{1/2} \cdot \left(\sum_{j=1}^{k}\frac{1}{j^2}\langle p-q, \bar{T}_j\rangle^2\right)^{1/2} \\ &\leq \sqrt{\pi}\left(\sum_{j=1}^{k}\frac{1}{j^2}\langle p-q, \bar{T}_j\rangle^2\right)^{1/2}. \end{aligned} \tag{7}$$

Observing that $\langle p - q, \bar{T}_j \rangle$ is exactly the difference between p's j^{th} Chebyshev moment and that of q, we can apply the assumption of the theorem, (1), to upper bound (7) by $\sqrt{\pi} \cdot \Gamma$.

Plugging this bound into Equation (5), we conclude the main bound of Theorem 1:

$$W_1(p, q) = \sup_{\text{1-Lipschitz, smooth } f} \langle f, p - q \rangle \leq \sqrt{\pi}\Gamma + \frac{36}{k}.$$

We note that the constants in the above bound can likely be improved. Notably, the 36 comes from multiplying the factor of 18 in Fact 10 by 2. As discussed in [BKM22, Appendix C.2], strong numerical evidence suggests that this 18 can be improved to π, leading to a bound of $\sqrt{\pi}\Gamma + \frac{2\pi}{k}$.

Finally, we comment on the special case in (3). If $|\mathbb{E}_{x \sim p} T_j(x) - \mathbb{E}_{x \sim q} T_j(x)| = |\langle p - q, \bar{T}_j \rangle| \leq \Gamma \cdot \sqrt{\frac{j}{1 + \log k}}$ for all j then we have that $\sum_{j=1}^{k} \frac{1}{j^2} \langle p - q, \bar{T}_j \rangle^2 \leq \frac{\Gamma}{1 + \log k} \sum_{j=1}^{k} \frac{1}{j} \leq \Gamma$. □

3.2 Efficient recovery

The primary value of Theorem 1 for our applications is that, given sufficiently accurate estimates, $\hat{m}_1, \ldots, \hat{m}_k$, of p's Chebyshev moments, we can recover a distribution q that is close in Wasserstein-1 distance to p, even if there is no distribution whose moments exactly equal $\hat{m}_1, \ldots, \hat{m}_k$.

This claim is formalized in Corollary 2, whose proof is straightforward. We outline the main idea here. Recall the condition of the corollary, that $\sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \langle p, \bar{T}_j \rangle \right)^2 \leq \Gamma^2$. Now, suppose we could solve the optimization problem:

$$q^* = \operatorname*{argmin}_{\text{distributions } q \text{ on } [-1,1]} \sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \langle q, \bar{T}_j \rangle \right)^2.$$

Then by triangle inequality we have:

$$\begin{aligned} \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(\langle p, \bar{T}_j \rangle - \langle q^*, \bar{T}_j \rangle \right)^2 \right)^{1/2} &\leq \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \langle q^*, \bar{T}_j \rangle \right)^2 \right)^{1/2} + \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \langle p, \bar{T}_j \rangle \right)^2 \right)^{1/2} \\ &\leq 2 \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \langle p, \bar{T}_j \rangle \right)^2 \right)^{1/2} \leq 2\Gamma. \end{aligned} \tag{8}$$

It then follows immediately from Theorem 1 that $W_1(p, q^*) \leq O\left(\frac{1}{k} + \Gamma\right)$, as desired.

The only catch with the argument above is that we cannot efficiently optimize over the entire set of distributions on $[-1, 1]$. Instead, we have to optimize over a sufficiently fine discretization. Specifically, we consider discrete distributions on a finite grid, choosing the Chebyshev nodes (of the first kind) instead of a uniform grid because doing so yields a better approximation, and thus allows for a coarser grid. Concretely, Corollary 2 is proven by analyzing Algorithm 1. Our full analysis of the algorithm is given in Appendix A.

We note that the optimization problem solved by Algorithm 1 is a simple linearly constrained quadratic program with $g = O(k^{1.5})$ variables and $O(k^{1.5})$ constraints, so can be solved to high accuracy in $\mathsf{poly}(k)$ time using a variety of methods [YT89; KV86; ART03]. In practice, the problem can also be solved efficiently using first-order methods like projected gradient descent [WR22].

Algorithm 1 Chebyshev Moment Regression

Input: Estimates $\hat{m}_1, \ldots, \hat{m}_k$ for the first k Chebyshev polynomial moments of a distribution p.
Output: A probability distribution q approximating p.

1: For $g = \lceil k^{1.5} \rceil$, let $\mathcal{C} = \{x_1, \ldots, x_g\}$ be the degree g Cheybshev nodes. I.e., $x_i = \cos\left(\frac{2i-1}{2g}\pi\right)$.
2: Let $q_1, \ldots, q_g$ solve the following optimization problem:

$$\begin{aligned} \min_{z_1,\ldots,z_g} \quad & \sum_{j=1}^{k} \frac{1}{j^2}\left(\hat{m}_j - \sum_{i=1}^{g} z_i T_j(x_i)\right)^2 \\ \text{subject to} \quad & \sum_{i=1}^{g} z_i = 1 \text{ and } z_i \geq 0, \ \forall i \in \{1, \ldots, g\}\,. \end{aligned}$$

3: Return $q = \sum_{i=1}^{m} q_i \delta(x - x_i)$, where δ is the Dirac delta function.

3.3 Proof of Lemma 11

We conclude this section by proving our global decay bound on the Cheybshev coefficients of a smooth, Lipschitz function, which was key in the proof of Theorem 1. To do so, we use a well-known expression for the antiderivative of the (normalized) Chebyshev polynomials [Hal15].

Fact 12 (Antiderivative of $\bar{T}_i$)**.** *Let C be the constant of integration. Then,*

$$\int \bar{T}_0(x)\,dx = \frac{\bar{T}_1(x)}{\sqrt{2}} + C, \quad \int \bar{T}_1(x)\,dx = \frac{1}{2}\left(\frac{\bar{T}_2(x)}{2} + \frac{\bar{T}_0(x)}{\sqrt{2}}\right) + C,$$

and for $i \geq 2$,

$$\int \bar{T}_i(x)\,dx = \frac{1}{2}\left(\frac{\bar{T}_{i+1}(x)}{i+1} - \frac{\bar{T}_{i-1}(x)}{i-1}\right) + C\,.$$

Proof of Lemma 11. Let f be any smooth, ℓ-Lipschitz function. Let f' denote its derivative. Since f is smooth, f' is also smooth and thus has a convergent Chebyshev series. Letting $c'_j \stackrel{\text{def}}{=} \left\langle f' \cdot w, \bar{T}_j \right\rangle$, we can thus write f' as the series $f' = \sum_{j=0}^{\infty} c'_j \bar{T}_j$. Applying Fact 12 and that $f = \int f'(x)\,dx + C_1$ for a constant C_1, we have that for a constant C_2,

$$\begin{aligned} f - C_1 &= \int \sum_{j=0}^{\infty} c'_j \bar{T}_j(x)\,dx \\ &= \sum_{j=0}^{\infty} c'_j \int \bar{T}_j(x)\,dx \\ &= \frac{c'_0}{\sqrt{2}}\bar{T}_1(x) + \frac{c'_1}{2}\left(\frac{\bar{T}_2(x)}{2} + \frac{\bar{T}_0(x)}{\sqrt{2}}\right) + \sum_{j=2}^{\infty} \frac{c'_j}{2}\left(\frac{\bar{T}_{j+1}(x)}{j+1} - \frac{\bar{T}_{j-1}(x)}{j-1}\right) + C_2 \\ &= \left(\frac{\sqrt{2}c'_0 - c'_2}{2}\right)\bar{T}_1(x) + \sum_{j=2}^{\infty}\left(\frac{c'_{j-1} - c'_{j+1}}{2j}\right)\bar{T}_j(x) + C_2. \end{aligned} \tag{9}$$

Now, let $c_j \stackrel{\text{def}}{=} \left\langle f \cdot w, \bar{T}_j \right\rangle$. From (9), we have that

$$c_1 = \left(\frac{\sqrt{2}c'_0 - c'_2}{2}\right), c_j = \left(\frac{c'_{j-1} - c'_{j+1}}{2j}\right) \text{ for } j \geq 2\,. \tag{10}$$

Accordingly,

$$\sum_{j=1}^{\infty}(2jc_j)^2 = (\sqrt{2}c'_0 - c'_2)^2 + \sum_{j=2}^{\infty}\left(c'_{j-1} - c'_{j+1}\right)^2$$
$$\leq 4c'^2_0 + 2c'^2_1 + \sum_{j=2}^{\infty} 2c'^2_{j-1} + 2c'^2_{j+1} \leq 4 \cdot \sum_{j=0}^{\infty} c'^2_j. \tag{11}$$

Finally, since the Chebyshev polynomials form an orthogonal basis under the weight function w, we have that $\sum_{i=0}^{\infty} c_i'^2 = \langle f' \cdot w, f' \rangle$, and since f is ℓ-Lipschitz, i.e. $|f'(x)| \leq \ell$,

$$\langle f' \cdot w, f' \rangle = \int_{-1}^{1} f'(x)^2 \cdot w(x) \leq \ell^2 \cdot \int_{-1}^{1} w(x)\, dx = \ell^2 \pi.$$

We conclude that $\sum_{j=1}^{\infty} c'^2_j \leq \pi\ell^2$. Combined with Equation (11), we have that $\sum_{j=1}^{\infty}(2jc_j)^2 \leq 4\pi\ell^2$. Dividing both sides by 4 gives the result. □

4 Private Synthetic Data

In this section, we present an application of our main result to differentially private synthetic data generation. We recall the setting from Section 1.3: we are given a dataset $X = x_1, \ldots, x_n$, where $x_i \in [-1, 1]$, and consider the distribution p that is uniform on X. The goal is to design an (ϵ, δ)-differentially private algorithm that returns a distribution q that is close to p in Wasserstein distance. For the purpose of defining differential privacy (see Def. 3), we consider the "bounded" notation of neighboring datasets, which applies to datasets of the same size [KM11]. Concretely, $X = x_1, \ldots, x_n$ and $X' = x'_1, \ldots, x'_n$ are *neighboring* if $x_i \neq x'_i$ for *exactly one* value of i.[7]

To solve this problem, we will compute the first n Chebyshev moments of p, then add noise to those moments using the standard *Gaussian mechanism*. Doing so ensures that the noised moments are (ϵ, δ)-differentially private. We then post-process the noised moments (which does not impact privacy) by finding a distribution q that matches the moments. The analysis of our approach follows directly from Theorem 1, although we use a slightly different method for recovering q than suggested in our general Algorithm 1: in the differential privacy setting, we are able to obtain a moderately faster algorithm that solves a regression problem involving $O(n)$ variables instead of $O(n^{1.5})$.

Before analyzing this approach, we introduce preliminaries necessary to apply the Gaussian mechanism. In particular, applying the mechanism requires bounding the ℓ_2 *sensitivity* of the function mapping a distribution p to its Chebyshev moments. This sensitivity is defined as follows:

Definition 13 (ℓ_2 Sensitivity)**.** Let $\mathcal{X}$ be some data domain (in our setting, $\mathcal{X} = [-1, 1]^n$) and let $f : \mathcal{X} \to \mathbb{R}^k$ be a vector valued function. The ℓ_2-sensitivity of f, $\Delta_{2,f}$, is defined as:

$$\Delta_{2,f} \stackrel{\text{def}}{=} \max_{\substack{\text{neighboring datasets} \\ X, X' \in \mathcal{X}}} \|f(X) - f(X')\|_2.$$

The Gaussian mechanism provides a way of privately evaluating any function f with bounded ℓ_2 sensitivity by adding a random Gaussian vector with appropriate variance. Let $\mathcal{N}(0, \sigma^2 I_k)$ denote a vector of k i.i.d. mean zero Gaussians with variance σ^2. We have the following well-known result:

[7] Although a bit tedious, our results can be extended to the "unbounded" notation of neighboring datasets, where X and X' might differ in size by one, i.e., because X' is created by adding or removing a single data point from X.

Algorithm 2 Private Chebyshev Moment Matching

Input: Dataset $x_1, \ldots, x_n \in [-1, 1]$, privacy parameters $\epsilon, \delta > 0$.

Output: A probability distribution q approximating the uniform distribution, p, on $x_1, \ldots, x_n$.

1: Let $\mathcal{G} = \{-1, -1 + \frac{1}{\lceil \epsilon n \rceil}, -1 + \frac{2}{\lceil \epsilon n \rceil}, \ldots, 1\}$. Let $r \stackrel{\text{def}}{=} |\mathcal{G}| = 2\lceil \epsilon n \rceil + 1$ and let $g_i = -1 + \frac{i-1}{\lceil \epsilon n \rceil}$ denote the i^{th} element of $\mathcal{G}$.

2: For $i = 1, \ldots, n$, let $\tilde{x}_i = \text{argmin}_{y \in \mathcal{G}} |x_i - y|$. I.e., round x_i to the nearest multiple of $1/\lceil \epsilon n \rceil$.

3: Set $\sigma^2 = \frac{\frac{16}{\pi}(1 + \log k)\ln(1.25/\delta)}{\epsilon^2 n^2}$.

4: Set $k = \lceil 2\epsilon n \rceil$.[8] For $j = 1, \ldots, k$, let $\hat{m}_j = \eta_j + \frac{1}{n}\sum_{i=1}^n \bar{T}_j(\tilde{x}_i)$, where $\eta_j \sim \mathcal{N}(0, j\sigma^2)$.

5: Let $q_0, \ldots, q_r$ be the solution to the following optimization problem:

$$\begin{aligned} \min_{z_1, \ldots, z_r} \quad & \sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j - \sum_{i=1}^{r} z_i T_j(g_i) \right)^2 \\ \text{subject to} \quad & \sum_{i=1}^{r} z_i = 1 \text{ and } z_i \geq 0, \ \forall i \in \{1, \ldots, r\}. \end{aligned}$$

6: Return $q = \sum_{i=1}^r q_i \delta(x - g_i)$, where δ is the Dirac delta function.

Fact 14 (Gaussian Mechanism [DKMMN06; DR14])**.** *Let $f : \mathcal{X} \to \mathbb{R}^k$ be a function with ℓ_2-sensitivity $\Delta_{2,f}$ and let $\sigma^2 = \Delta_{2,f}^2 \cdot 2\ln(1.25/\delta)/\epsilon^2$, where $\epsilon, \delta \in (0, 1)$ are privacy parameters. Then the mechanism $\mathcal{M} = f(X) + \eta$, where $\eta \sim \mathcal{N}(0, \sigma^2 I_k)$ is (ϵ, δ)-differentially private.*

We are now ready to prove our main result of this section, Theorem 4, which follows by analyzing Algorithm 2. Note that Algorithm 2 is very similar to Algorithm 1, but we first round our distribution to be supported on a uniform grid, $\mathcal{G}$. Doing so will allow us to solve our moment regression problem over the same grid, which is smaller than the set of Chebyshev nodes used in Algorithm 1.

Proof of Theorem 4. We analyze both the privacy and accuracy of Algorithm 2.

Privacy. For a dataset $X = \{x_1, \ldots, x_n\} \in [-1, 1]^n$, let $f(X)$ be a vector-valued function mapping to the first $k = \lceil 2\epsilon n \rceil$ (as set in Algorithm 2) *scaled* Chebyshev moments of the uniform distribution over X. I.e.,

$$f(X) = \begin{bmatrix} 1 \cdot \frac{1}{n}\sum_{i=1}^n \bar{T}_1(x_i) \\ \frac{1}{\sqrt{2}} \cdot \frac{1}{n}\sum_{i=1}^n \bar{T}_2(x_i) \\ \vdots \\ \frac{1}{\sqrt{k}} \cdot \frac{1}{n}\sum_{i=1}^n \bar{T}_k(x_i) \end{bmatrix}$$

[8] While we choose $k = \lceil 2\epsilon n \rceil$ by default, any choice of $k = \lceil c\epsilon n \rceil$ for constant c suffices to obtain the bound of Theorem 4. Similarly, the grid spacing in $\mathcal{G}$ can made finer or coarse by a multiplicative constant. A larger k or a finer grid will lead to a slightly more accurate result at the cost of a slower algorithm. We chose defauts so that any error introduced from the grid and choice of k is swamped by error incurred from the noise added in Line 4. I.e., the error cannot be improved by more than a factor of two with difference choices. See the proof of Theorem 4 for more details.

By Fact 8, $\max_{x_i \in [-1,1]} |\bar{T}_j(x_i)| \leq \sqrt{2/\pi}$ for $j \in \mathbb{Z}_{>0}$, so we have:

$$\Delta_{2,f}^2 = \max_{\substack{\text{neighboring datasets} \\ X, X' \in \mathcal{X}}} \|f(X) - f(X')\|_2^2 \leq \sum_{j=1}^{k} \frac{1}{jn^2} \cdot \frac{8}{\pi} \leq \frac{8}{\pi n^2}(1 + \log k).$$

Let $\tilde{X} = \tilde{x}_1, \ldots, \tilde{x}_n$ be as defined in Algorithm 2. It follows from Fact 14 that $\tilde{m} = f(\tilde{X}) + \eta$ is (ϵ, δ)-differentially private for $\eta \sim \mathcal{N}(0, \sigma^2 I_k)$ as long as $\sigma^2 = \frac{16}{\pi}(1 + \log k)\ln(1.25/\delta)/(n^2\epsilon^2)$. Finally, observe that $\hat{m}_j$ computed by Algorithm 2 is exactly equal to $\sqrt{j}$ times the j^{th} entry of such an $\tilde{m}$. So $\hat{m}_1, \ldots, \hat{m}_k$ are (ϵ, δ)-differentially private. Since the remainder of Algorithm 2 simply post-processes $\hat{m}_1, \ldots, \hat{m}_k$ without returning to the original data X, the output of the algorithm is also (ϵ, δ)-differentially private, as desired.

Accuracy. Algorithm 2 begins by rounding our dataset X so that every data point is a multiple of $1/\lceil \epsilon n \rceil$. Let $\tilde{p}$ be the uniform distribution over the rounded dataset $\tilde{X}$. Then it is not hard to see from the transportation definition of the Wasserstein-1 distance that:

$$W_1(p, \tilde{p}) \leq \frac{1}{2\lceil \epsilon n \rceil}. \tag{12}$$

In particular, we can transport p to $\tilde{p}$ by moving every unit of $1/n$ probability mass a distance of at most $1/2\lceil \epsilon n \rceil$. Given (12), it will suffice to show that Algorithm 2 returns a distribution q that is close in Wasserstein distance to $\tilde{p}$. We will then apply triangle inequality to bound $W_1(p, q)$.

To show that Algorithm 2 returns a distribution q that is close to $\tilde{p}$ in Wasserstein distance, we begin by bounding the moment estimation error:

$$E \stackrel{\text{def}}{=} \sum_{j=1}^{k} \frac{1}{j^2} \left(\hat{m}_j(p) - \langle \tilde{p}, \bar{T}_j \rangle \right)^2,$$

where k is as chosen in Algorithm 2 and $\langle \tilde{p}, \bar{T}_j \rangle = \frac{1}{n} \sum_{i=1}^{n} \bar{T}_j(\tilde{x}_i)$. Let σ^2 and $\eta_1, \ldots, \eta_k$ be as in Algorithm 2. Applying linearity of expectation, we have that:

$$\mathbb{E}[E] = \mathbb{E}\left[\sum_{j=1}^{k} \frac{1}{j^2} \eta_j^2 \right] = \sum_{j=1}^{k} \frac{1}{j^2} \mathbb{E}\left[\eta_j^2 \right] = \sum_{j=1}^{k} \frac{1}{j^2} \cdot j\sigma^2 \leq (1 + \log k)\sigma^2. \tag{13}$$

Now, let q be as in Algorithm 2. Using a triangle inequality argument as in Section 3.2, we have:

$$\Gamma^2 = \sum_{j=1}^{k} \frac{1}{j^2} \left(\langle q, \bar{T}_j \rangle - \langle \tilde{p}, \bar{T}_j \rangle \right)^2 \leq \sum_{j=1}^{k} \frac{1}{j^2} \left(\langle q, \bar{T}_j \rangle - \hat{m}_j \right)^2 + \sum_{j=1}^{k} \frac{1}{j^2} \left(\langle \tilde{p}, \bar{T}_j \rangle - \hat{m}_j \right)^2 \leq 2E.$$

Above we use that $\tilde{p}$ is a valid solution to the optimization problem solved in Algorithm 2 and, since q is the optimum, $\sum_{j=1}^{k} \frac{1}{j^2} \left(\langle q, \bar{T}_j \rangle - \hat{m}_j \right)^2 \leq \sum_{j=1}^{k} \frac{1}{j^2} \left(\langle \tilde{p}, \bar{T}_j \rangle - \hat{m}_j \right)^2$. It follows that $\mathbb{E}[\Gamma^2] \leq 2\,\mathbb{E}[E]$, and, via Jensen's inequality, that $\mathbb{E}[\Gamma] \leq \sqrt{2}\sqrt{\mathbb{E}[E]}$. Plugging into Theorem 1, we have:

$$\mathbb{E}[W_1(\tilde{p}, q)] \leq \sqrt{\pi}\,\mathbb{E}[\Gamma] + \frac{36}{k} \leq \sqrt{\pi}\sqrt{2(1 + \log k)\sigma^2} + \frac{36}{k} = O\left(\frac{\log(\epsilon n)\sqrt{\log(1/\delta)}}{\epsilon n} \right). \tag{14}$$

By triangle inequality and (12), $W_1(p,q) \leq W_1(\tilde{p},q) + W_1(\tilde{p},p) \leq W_1(\tilde{p},q) + \frac{1}{2\lceil \epsilon n \rceil}$. Combined with the bound above, this proves the theorem. Recall from Section 3 that the 36 in Theorem 1 can likely be replaced by 2π, in which case it can be checked that the term above involving $\frac{1}{k}$ will be dominated by the $\sqrt{2\pi(1+\log k)\sigma^2}$ term for our default of $k = \lceil 2\epsilon n \rceil$ in Algorithm 2. However, any choice $k = \Theta(\epsilon n)$ suffices to prove the theorem. We also remark that our bound on the expected value of $W_1(\tilde{p},q)$ can also be shown to hold with high probability. See Appendix B for details.

We conclude by noting that, as in our analysis of Algorithm 1 (see Section 3.2), Algorithm 2 requires solving a linearly constrained quadratic program with $r = 2\lceil \epsilon n \rceil + 1$ variables and $r+1$ constraints, which can be done to high accuracy in $\mathsf{poly}(\epsilon n)$ time. □

5 Spectral Density Estimation

In this section, we present a second application of our main result to the linear algebraic problem of Spectral Density Estimation (SDE). We recall the setting from Section 1.3: letting p be the uniform distribution over the eigenvalues given $\lambda_1 \geq \cdots \geq \lambda_n$ of a symmetric matrix $A \in \mathbb{R}^{n \times n}$, the goal is to find some distribution q that satisfies

$$W_1(p,q) \leq \epsilon \|A\|_2. \tag{15}$$

Without loss of generality, we assume from here forward that $\|A\|_2 \leq 1$. If it was not, we can scale A by $1/\|A\|_2$, compute an approximate spectral density q with error ϵ, then rescale by $\|A\|_2$ to achieve error $\epsilon\|A\|_2$. In fact, we would only need a constant factor approximation to $\|A\|_2$, which be computed using $O(\log n)$ matrix-multiplications with A [KW92].

In many settings of interest, A is implicit and can only be accessed via matrix-vector multiplications. So, we want to understand 1) how many matrix-vector multiplications with A are required to achieve (15), and 2) how efficiently can we achieve (15) in terms of standard computational complexity.

We show how to obtain improved answers to these questions by using our main result, Theorem 1, to give a tighter analysis of an approach from [BKM22]. Like other SDE methods, that approach uses *stochastic trace estimation* to estimate the Chebyshev moments of p. In particular, let $m_1, \ldots, m_k$ denote the first k Chebyshev moments. I.e., $m_j = \frac{1}{n}\sum_{i=1}^n \bar{T}_j(\lambda_i)$. Then we have for each j,

$$m_j = \frac{1}{n}\sum_{i=1}^n \bar{T}_j(\lambda_i) = \frac{1}{n}\mathrm{tr}(\bar{T}_j(A)),$$

where tr is the matrix trace. Stochastic trace estimation methods like Hutchinsons method can approximate $\mathrm{tr}(\bar{T}_j(A))$ efficiently via multiplication of $\bar{T}_j(A)$ with random vectors [Gir87; Hut90]. In particular, for any vector $g \in \mathbb{R}^n$ with mean 0, variance 1 entries, we have that:

$$\mathbb{E}[g^T \bar{T}_j(A) g] = \mathrm{tr}(\bar{T}_j(A)).$$

$\bar{T}_j(A)g$, and thus $g^T\bar{T}_j(A)g$, can be computed using j matrix-vector products with A. In fact, by using the Chebyshev polynomial recurrence, we can compute $g^T\bar{T}_j(A)g$ for all $j = 1, \ldots, k$ using k total matrix-vector products:

$$T_0(A)g = g \qquad T_1(A)g = Ag \qquad \ldots \qquad T_j(A)g = 2AT_{j-1}(A)g - T_{j-2}(A)g.$$

Optimized methods can actually get away with $\lceil k/2 \rceil$ matrix-vector products [Che23]. Using a standard analysis of Hutchinson's trace estimator (see, e.g., [RA15] or [CK22]) Braverman et al. [BKM22] prove the following:

Lemma 15 ([BKM22, Lemma 4.2]). *Let A be a matrix with $\|A\|_2 \le 1$. Let C be a fixed constant, $j \in \mathbb{Z}_{>0}$, $\alpha, \gamma \in (0,1)$, and $\ell_j = \lceil 1 + \frac{C \log^2(1/\alpha)}{nj\gamma^2} \rceil$. Let $g_1, \ldots, g_{\ell_j} \sim \text{Uniform}(\{-1,1\}^n)$ and let $\hat{m}_j = \frac{1}{\ell_j n} \sum_{i=1}^{\ell_j} g_i^\top \bar{T}_j(A) g_i$. Then, with probability $1-\alpha$, $|\hat{m}_j - m_j| \le \sqrt{j}\gamma$.*

We combine this lemma with Theorem 1 to prove the following more precise version of Theorem 5:

Theorem 16. *There is an algorithm that, given $\epsilon \in (0,1)$ and symmetric $A \in \mathbb{R}^{n \times n}$ with spectral density p and $\|A\|_2 \le 1$, computes $\min\left\{n, O\left(\frac{1}{\epsilon} \cdot \left(1 + \frac{\log^2(1/\epsilon)\log^2(1/(\epsilon\delta))}{n\epsilon}\right)\right)\right\}$ matrix-vector products with A and uses $\tilde{O}(1/\epsilon^{3.25})$ additional time to output a distribution q such that, with probability $\ge 1-\delta$, $W_1(p,q) \le \epsilon$.*

Proof. First note that, if $\epsilon \le 1/n$, the above result can be obtained by simply recovering A by multiplying by all $n \le 1/\epsilon$ standard basis vectors, then performing a full eigendecomposition to compute its spectral density, which takes $o(n^3)$ time. So we focus on the regime when $\epsilon > 1/n$.

Choose $k = c'/\epsilon$ for a sufficiently large constant c' and apply Lemma 15 for all $j = 1, \ldots, k$ with $\gamma = \frac{1}{k\sqrt{1+\log k}}$, and $\alpha = \delta/k$. By a union bound, we obtain estimates $\hat{m}_1, \ldots, \hat{m}_k$ satisfying, for all j,

$$|\hat{m}_j - m_j| \le \sqrt{j}\gamma = \sqrt{j} \cdot \frac{1}{k\sqrt{1+\log k}}. \tag{16}$$

Applying Theorem 1 (specifically, (3)) and Corollary 2, we conclude that, using these moments, Algorithm 1 can recover a distribution q satisfying:

$$W_1(p,q) \le 2c \cdot \frac{1}{k}.$$

I.e., we have $W_1(p,q) \le \epsilon$ as long as $c' \ge 2c$. This proves the accuracy bound. We are left to analyze the complexity of the method. We first bound the total number of matrix-vector multiplications with A, which we denote by T. Since $\ell_j \le \ell_{j-1}$ for all j, computing the necessary matrix-vector product to approximate m_j only costs ℓ_{j-1} additional products on top of those used to approximate m_{j-1}. So, recalling that $\ell_j = \lceil 1 + \frac{C\log^2(1/\alpha)}{nj\gamma^2} \rceil$, we have:

$$T = \left(1 + \frac{C\log^2(k/\delta)}{n\gamma^2}\right) + \left(1 + \frac{C\log^2(k/\delta)}{2n\gamma^2}\right) + \cdots + \left(1 + \frac{C\log^2(k/\delta)}{kn\gamma^2}\right).$$

Using the fact that $1 + 1/2 + \ldots + 1/k \le 1 + \log(k)$ we can upper bound T by:

$$T = O\left(k + \frac{\log^2(k/\delta)\log(k)}{n\gamma^2}\right) = O\left(k + \frac{\log^2(k/\delta)\log^2(k)}{nk^2}\right),$$

which gives the desired matrix-vector product bound since $k = O(1/\epsilon)$.

In terms of computational complexity, Corollary 2 immediately yields a bound of $\mathsf{poly}(1/\epsilon)$ time to solve the quadratic program in Algorithm 1. However, this runtime can actually be improved to $\tilde{\mathcal{O}}(1/\epsilon^3)$ by taking advantage of the fact that $\hat{m}_1, \ldots, \hat{m}_k$ obey the stronger bound of (3) instead of just (1). This allows us to solve a linear program instead of a quadratic program. In particular, let $\mathcal{C}$

be a grid of Chebyshev nodes, as used in Algorithm 1. I.e., $\mathcal{C} = \{x_1, \ldots, x_g\}$ where $x_i = \cos\left(\frac{2i-1}{2g}\pi\right)$. Let $q_1^{\text{LP}}, \ldots, q_g^{\text{LP}}$ be any solution to the following linear program with variables $z_1, \ldots, z_g$:

$$\begin{aligned} \text{minimize} \quad & 0 \\ \text{subject to} \quad & \sum_{i=1}^{g} z_i = 1 \\ & z_i \geq 0, \qquad \forall i \in \{1, \ldots, g\} \\ & \sum_{i=1}^{g} \bar{T}_j(x_i) z_i \leq \hat{m}_j + \left(\sqrt{j}\gamma + \frac{j\sqrt{2\pi}}{g}\right), \forall j \in \{1, \ldots, k\} \\ & \sum_{i=1}^{g} \bar{T}_j(x_i) z_i \geq \hat{m}_j - \left(\sqrt{j}\gamma + \frac{j\sqrt{2\pi}}{g}\right), \forall j \in \{1, \ldots, k\}. \end{aligned} \tag{17}$$

We first verify that the linear program has a solution. To do so, note that, by Equation (18) in Appendix A, there exists a distribution $\tilde{p}$ supported on $\mathcal{C} = \{x_1, \ldots, x_g\}$, such that $|m_j(p) - m_j(\tilde{p})| \leq \frac{j\sqrt{2\pi}}{g}$. By (16) and triangle inequality, it follows that $\tilde{p}$ is a valid solution to the linear program.

Next, let $q^{\text{LP}} = \sum_{i=1}^{g} q_i^{\text{LP}} \delta(x - x_i)$ be the distribution formed by any solution to the linear program. We have that, for any j,

$$\left|m_j - \langle q^{\text{LP}}, \bar{T}_j \rangle\right| \leq \left|\langle q^{\text{LP}}, \bar{T}_j \rangle - \hat{m}_j\right| + |\hat{m}_j - m_j| \leq 2\sqrt{j}\gamma + \frac{j\sqrt{2\pi}}{g}.$$

Setting $g = k^{1.5}\sqrt{1 + \log(k)}$ and plugging into Theorem 1, we conclude that

$$W_1(p, q^{\text{LP}}) \leq O(1/k).$$

The linear program in (17) has $g = \tilde{O}(k^{1.5})$ variables, $2k + 1$ constraints involving all variables, and g constraints involving a single variable. It follows that its constraint matrix has $\tilde{O}(k^{2.5})$ nonzeros, and thus the program can be solved in $\tilde{O}(k^{3.25})$ time [LS15; CLS19], which equals $\tilde{O}(1/\epsilon^{3.25})$ time. □

6 Private Synthetic Data in Higher Dimensions

When generalizing Theorem 4 to d-dimensions, we take a black-box approach that uses the following properties of differential privacy.

Theorem 17 (Basic Composition Rules [DMNS06])**.**

1. *Consider randomized algorithms $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k : \mathcal{U} \to \mathcal{B}$. Let $\mathcal{A}_j$ be ϵ_j-DP for each $j \in [k]$ and $\mathcal{A} : \mathcal{U} \to \mathcal{B}$ be defined as $\mathcal{A}(x) = (\mathcal{A}_1(x), \mathcal{A}_2(x), \cdots, \mathcal{A}_k(x))$, where each algorithm is run independently. Then $\mathcal{A}$ is ε-DP for $\varepsilon = \sum_{j=1}^{k} \varepsilon_j$.*
2. *For randomized algorithm $\mathcal{A} : \mathcal{U} \to \mathcal{B}$ that satisfies (ϵ, δ)-DP, consider disjoint datasets $\{X_1, X_2, \cdots, X_k\} = X \in \mathcal{U}$. The parallel composition mechanism $\mathcal{A}^*$, defined as $\mathcal{A}^*(X) = (\mathcal{A}(X_1), \mathcal{A}(X_2), \cdots, \mathcal{A}(X_k))$, satisfies (ϵ, δ)-DP.*

Theorem 18 (Post-processing [DMNS06])**.** *Consider an algorithm $\mathcal{A} : \mathcal{U} \to \mathcal{B}$ that is (ϵ, δ)-differentially private. Additionally, consider an arbitrary function $f : \mathcal{B} \to \mathcal{B}'$. Then $f \circ \mathcal{A} : \mathcal{U} \to \mathcal{B}'$ satisfies (ϵ, δ)-differential privacy.*

6.1 Generalizing to Higher Dimensions

Let $\mathcal{D}^{(d)}$ denote a d-dimensional distribution supported on $[-1,1]^d$. We use the superscript to denote the dimension of the distribution. Let n_d be the total number of points in the data and let $\mathcal{D}^{(d-1)}$ denote the uniform measure over the multiset of points in the data.

We assume black-box access to an algorithm $\boldsymbol{\mathcal{A}}^{(1)}$ which when given $p^{(1)}$ points in $[-1,1]$, approximates the uniform measure on these points up to an error of $\rho(1,p^{(1)}) = O\left(\frac{\log(\epsilon p^{(1)})\sqrt{\log(1/\delta)}}{\epsilon p^{(1)}}\right)$ in the Wasserstein-1 distance. This guarantee is satisfied by Theorem 4 for $d=1$.

Algorithm 3 outputs an approximate distribution $\hat{\mathcal{D}}^{(d)}$ to the original distribution $\mathcal{D}^{(d)}$ as follows. Let s_d denote the number of equal-sized *slices* of the space $[-1,1]^d$ along the d-th dimension (or any arbitrary dimension). For each slice $i \in [s_d]$, we shift the points in the i-th slice, so that they lie in a $(d-1)$ dimensional hyperplane inside the i-th slice, and denote the uniform measure on the shifted points in the slice by $\tilde{\mathcal{D}}_i^{(d-1)}$. For $i \in [s_d]$, let $p_i^{(d-1)}$ denote the number of data-points in the i-th slice, and therefore, the "mass" along the i-th slice is $p_i^{(d-1)}/n_d$ (where the superscript helps us keep a track of the dimension of the distribution after slicing and shifting the mass). Our algorithm $\texttt{HDAD}^{(d-1)}$ does this recursively for all the $(d-1)$-dimensional distribution $\tilde{\mathcal{D}}_i^{(d-1)}$, where it uses $s_{(d-1)}$ slices for the $(d-1)$-dimensional distribution and approximates the one-dimensional distribution using the black-box algorithm $\boldsymbol{\mathcal{A}}^{(1)}$. $\texttt{HDAD}^{(d)}$ constructs the approximate distribution

Algorithm 3 HDAD (High-Dimensional Approximate Distributions): Generalizing to d-Dimensions

Input: $\mathcal{D}^{(d)}$, a d-dimensional point-mass distribution supported on $[-1,1]^d$; black-box algorithm $\boldsymbol{\mathcal{A}}^{(1)}$, which takes $p^{(1)}$ points in $[-1,1]$ and approximates the point distribution supported on these points.

Output: Approximate distribution $\hat{\mathcal{D}}^{(d)}$ supported on $[-1,1]^d$.

function $\texttt{HDAD}^{(d)}(\mathcal{D}^{(d)})$
  if $d = 1$ **then**
    $\hat{\mathcal{D}}^{(1)} \leftarrow \boldsymbol{\mathcal{A}}^{(1)}(\mathcal{D}^{(1)})$ ▷ Apply base case approximation
    return $\hat{\mathcal{D}}^{(1)}$
  else
    Let s_d be the number of equal-sized slices of $[-1,1]^d$ along the d-th dimension
    Let n_d be the total number of points in $\mathcal{D}^{(d)}$
    for $i \in [s_d]$ **do** ▷ Iterate over each slice
      Let $p_i^{(d-1)}$ be the number of points in the i-th slice.
      Shift the points in the i-th slice so that they lie in a $(d-1)$-dimensional hyperplane inside the i-th slice.
      Let $\tilde{\mathcal{D}}_i^{(d-1)}$ denote the uniform measure over the shifted points in the i-th slice.
      $\hat{\mathcal{D}}_i^{(d-1)} \leftarrow \texttt{HDAD}^{(d-1)}(\tilde{\mathcal{D}}_i^{(d-1)})$ ▷ Recursively approximate the shifted distribution
      Accumulate $\hat{\mathcal{D}}_i^{(d-1)}$ into $\hat{\mathcal{D}}^{(d)}$ i.e. $\hat{\mathcal{D}}^{(d)} = \sum_{i\in[s_d]} \frac{p_i^{(d-1)}}{n_d} \hat{\mathcal{D}}_i^{(d-1)}$
    end for
    return $\hat{\mathcal{D}}^{(d)}$
  end if
end function
return $\texttt{HDAD}^{(d)}(\mathcal{D}^{(d)})$

$\hat{\mathcal{D}}^{(d)}$ as follows

$$\texttt{HDAD}^{(d)}\left(\mathcal{D}^{(d)}\right) = \hat{\mathcal{D}}^{(d)} = \sum_{i\in[s_d]} \frac{p_i^{(d-1)}}{n_d} \texttt{HDAD}^{(d-1)}\left(\tilde{\mathcal{D}}_i^{(d-1)}\right),$$
$$\texttt{HDAD}^{(1)}\left(\tilde{D}_j^{(2)}\right) = \boldsymbol{\mathcal{A}}^{(1)}\left(\tilde{D}_j^{(2)}\right), \text{ for } j \in [s_2].$$

Claim 19. *Let* $\rho(1, n_d) = O\left(\frac{\log(\epsilon n_d)\sqrt{\log(1/\delta)}}{\epsilon n_d}\right)$. *Then,*

$$W_1(\hat{\mathcal{D}}^{(d)}, \mathcal{D}^{(d)}) \le \sum_{j=2}^{d} \frac{1}{s_j} + \rho(1, n_d) \prod_{i=2}^{d} s_i .$$

Proof. We use the fact that the Wasserstein distance between the two distributions is less than the transportation scheme of moving a mass in the sliced dimension to make it a $(d-1)$-dimensional distribution $\tilde{\mathcal{D}}_i^{(d-1)}, i \in [s_d]$ plus the Wasserstein distance between $\tilde{\mathcal{D}}_i^{(d-1)}$ and $\texttt{HDAD}^{(d-1)}\left(\tilde{\mathcal{D}}_i^{(d-1)}\right)$, for $i \in [s_d]$.

The proof proceeds via induction. In the base case for $d = 2$, let $\mathcal{D}^{(2)}$ be the distribution supported on $[-1, 1]^2$ with n_2 points. The algorithm $\texttt{HDAD}^{(2)}$ makes s_2 slices along the second-dimension, and forms distribution $\tilde{D}_i^{(1)}$, for $i \in [s_2]$. The mass in each slice move a distance of $\frac{1}{s_2}$. Then for each $i \in [s_2]$, the algorithm uses $\boldsymbol{\mathcal{A}}^{(1)}$ to approximate $\tilde{D}_i^{(1)}$. By triangle inequality, we get that the Wasserstein-1 distance is

$$W_1\left(\mathcal{D}^{(2)}, \texttt{HDAD}^{(2)}\left(\mathcal{D}^{(2)}\right)\right) \le \sum_{i\in[s_2]} \frac{p_i^{(1)}}{n_2}\frac{1}{s_2} + \frac{p_i^{(1)}}{n_2} \cdot \rho(1, p_i^{(1)}) \le \frac{1}{s_2} + s_2\rho(1, n_2) .$$

By the inductive hypothesis, assume that the claim holds for $k \in \mathbb{Z}_{\ge 0}$-dimensions. We now show the claim holds for $k+1$ dimensions as well. Let $\mathcal{D}^{(k+1)}$ be a $(k+1)$-dimensional distribution on $[-1, 1]^{(k+1)}$, with $n_{(k+1)}$ points. We get by triangle inequality and the inductive hypothesis that

$$\begin{aligned} W_1\left(\mathcal{D}^{(k+1)}, \texttt{HDAD}^{(k+1)}\left(\mathcal{D}^{(k+1)}\right)\right) &\le \sum_{i\in[s_{k+1}]} \frac{p_i^{(k)}}{n_{k+1}}\frac{1}{s_{k+1}} + \frac{p_i^{(k)}}{n_{k+1}} \cdot W_1\left(\tilde{\mathcal{D}}_i^{(k)}, \texttt{HDAD}^{(k)}\left(\tilde{\mathcal{D}}_i^{(k)}\right)\right) \\ &\le \frac{1}{s_{k+1}} + \frac{p_i^{(k)}}{n_{k+1}}\left(\sum_{j=2}^{k}\frac{1}{s_j} + \rho(1, p_i^{(k)})\prod_{i=2}^{k} s_i\right) \\ &\le \sum_{j=2}^{k+1}\frac{1}{s_j} + \rho(1, n_{k+1})\prod_{i=2}^{k+1} s_i . \end{aligned}$$

□

Theorem 20. *Let* $\hat{\mathcal{D}}^{(d)}$ *be the distribution returned by the Algorithm 3, for input distribution* $\mathcal{D}^{(d)}$. *Then,*

$$W_1(\hat{\mathcal{D}}^{(d)}, \mathcal{D}^{(d)}) \le 2(d-1)^{(1-1/d)}\left(\frac{c\log(\epsilon n)\sqrt{\log(1/\delta)}}{\epsilon n}\right)^{1/d} .$$

Proof. The proof follows from Claim 19 and Theorem 4 (the black-box one-dimensional algorithm that our algorithm uses). Plugging in the value of $\rho(1,n) = c\frac{\log(\epsilon n)\sqrt{\log(1/\delta)}}{\epsilon n}$ defined in Theorem 4 and setting $s_2 = \cdots = s_d = \left(\frac{(d-1)}{\rho(1,n)}\right)^{1/d}$, gives us the statement of the theorem. □

Claim 21. *Initializing Algorithm 3 with an (ϵ, δ)-differentially private $\boldsymbol{\mathcal{A}}^{(1)}$ yields an (ϵ, δ)- differentially private distribution $\hat{\mathcal{D}}^{(d)}$.*

Proof. In Algorithm 3, we recursively approximate slices of the distribution. The base case, given in line 3, applies the 1-dimensional, black box (ϵ, δ)-differentially private algorithm $\boldsymbol{\mathcal{A}}^{(1)}$ to a 1-dimensional slice. In other words, for $d = 1$ at the bottom of the recursive stack, for each of s_1 slices, we have an (ϵ, δ)-differentially private approximation $\hat{\mathcal{D}}^{(1)}$. Each slice is disjoint, so by composition rules given in Theorem 17, estimating s_1 disjoint slices is (ϵ, δ)-differentially private. Line 13 then accumulates and re-scales each $\hat{\mathcal{D}}_i^{(d-1)}$ into $\hat{\mathcal{D}}^{(d)}$; each accumulation and rescaling of the (ϵ, δ)-differentially private slices from the recursive base-case maintains privacy due to post-processing rules given in Theorem 18. □

7 Empirical Evaluation of Algorithm 2

We validate Theorem 4 by empirically evaluating Algorithm 2. In the privacy setting, we approximate k moments, $\hat{m}_1, \ldots, \hat{m}_k$, which each satisfy $|\hat{m}_j - \langle p, \bar{T}_j \rangle| \leq O\left(\sqrt{j}/k\right)$. I think it's probably In our experiments, k is set to the number of data points n. To generate the plots, we vary n and take an n sized subsample of the data from which we construct p, apply Algorithm 2 to p to construct (ϵ, δ)-differentially private distribution q, and report $W_1(p, q)$ on the y-axis. For each n we parameterize the private noise parameter σ^2 setting $\epsilon = 0.5$ and $\delta = 1/n^2$, which are standard settings for private synthetic data [MMSM22; RHR+23].

Data. . We verify Algorithm 2 on both real world data and data generated directly from known pdfs. In terms of real world data, we first include the American Community Survey (ACS) data from the Folktables repository [DHMS21]. We use the 2018 ACS 1-Year data for the state of New York; Figure 1 shows our algorithm run on `PINCP` (personal income) column from this data. We also include the California Housing dataset [PB97]; Figure 1 gives results on the `HouseAge` column, which represents the median house age in a district. Finally, we include the CDC Diabetes Health Indicators dataset [Teb21; KLN24]; Figure 1 gives results on `PhysHlth` (number of physically unhealthy days) from this data.

Remark 22. The pdfs for the distributions tested are as follows: `Gaussian`, $f(x) = \frac{1}{\sqrt{2\pi}} e^{-0.5x^2}$. `Sine`, $f(x) = \sin(\pi x) + 1$. `Power Law`, $f(x) = (x + 1.1)^{-2}$ We sample from each f on a uniform grid on $[-1, 1]$ to generate data for the experiments.

- Make plots bigger by stretching vertically and making sure they take up full page width. There is a slight alignment issue with top and bottom plots.
- Don't repeat y axis label. only have on left most plots. Make that label more concrete: $W_1(p, q)$ is preferable over W_1 distance.
- Make font size a bit bigger.
- *log* doesn't seem to be in the right mode. It should be log (no italics.
- Don't abreviate Algorithm since doing so isn't necessary here.

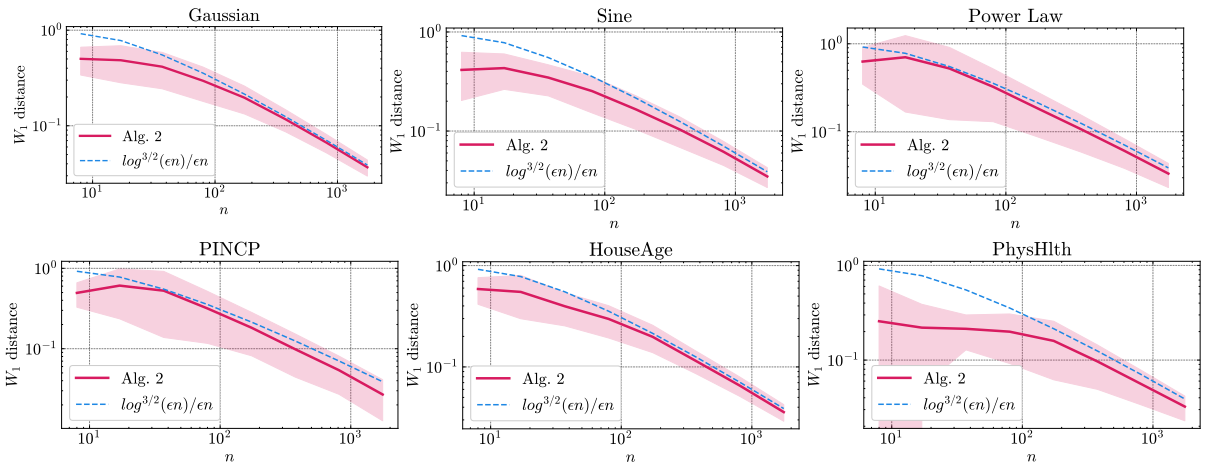

Figure 1: Experimental validation of the application of Theorem 1 to Algorithm 2 e.g. for private synthetic data. We plot the W_1 distance between the true distribution p (from real data or generated (see Remark 22), and the distribution q constructed from k Chebyshev moments corrupted with Gaussian noise whose variance scales as j/k^2 for the j^{th} moment. As predicted by Theorem 1, the Wasserstein error scales roughly as $\tilde{O}(1/n)$.

- Make the y axis range and x-axis range exactly the same for all plots. Will look better. Also tighten the x range so there aren't parts of the plot with nothing plotted.

With this setting of parameters, we expect to recover a distribution q such that $W_1(q,p) \leq \tilde{O}\left(\frac{\log n}{n}\right)$. This error bound is verified in Figure 1 for both the real data and the generated data. The plots were obtained by solving the linearly constrained least squares problem from Algorithm 1 using an interior-point method from MOSEK [DB16; MOS19; ART03].

Acknowledgements

We thank Raphael Meyer for suggesting the lower bound on the number of matrix-vector multiplications required for spectral density estimation. This work was partially supported by NSF Grants 2046235 and 2045590.

References

[Abo18] John Abowd. The US census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 2867–2867, 2018 (cited on page 3).

[AAS+19] John Abowd, Robert Ashmead, Garfinkel Simson, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, and William Sexton. Census topdown: differentially private data, incremental schemas, and consistency with public knowledge. *US Census Bureau*, 2019 (cited on page 3).

[ACK+24] Noah Amsel, Tyler Chen, Feyza Duman Keles, Diana Halikias, Cameron Musco, and Christopher Musco. Fixed-sparsity matrix approximation from matrix-vector products. *arXiv:2402.09379*, 2024 (cited on page 6).

[ART03] Erling D. Andersen, Cornelis Roos, and Tamás Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003 (cited on pages 9, 20).

[AT11] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2), 2011 (cited on page 5).

[ABK+21] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 457–467, 2021 (cited on page 3).

[BN23] Ainesh Bakshi and Shyam Narayanan. Krylov methods are (nearly) optimal for low-rank approximation. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2023 (cited on page 6).

[Bid23] Joseph R. Biden. Executive order on the safe, secure, and trustworthy development and use of artificial intelligence, 2023 (cited on page 3).

[BSV24] March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Private measures, random walks, and synthetic data. *Probability Theory and Related Fields*, 189(1):569–611, 2024 (cited on pages 2–4).

[BHSW20] Mark Braverman, Elad Hazan, Max Simchowitz, and Blake Woodworth. The gradient complexity of linear regression. In *Proceedings of the 33rd Annual Conference on Computational Learning Theory (COLT)*, pages 627–647, 2020 (cited on page 6).

[BKM22] Vladimir Braverman, Aditya Krishnan, and Christopher Musco. Sublinear time spectral density estimation. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*, 2022 (cited on pages 1, 2, 5, 7, 9, 14, 15).

[Che23] Tyler Chen. A spectrum adaptive kernel polynomial method. *The Journal of Chemical Physics*, 159(11), 2023 (cited on page 14).

[CKHMM24] Tyler Chen, Feyza Duman Keles, Diana Halikias, Cameron Musco, and Christopher Musco. Near-optimal hierarchical matrix approximation from matrix-vector products. *arXiv:2407.04686*, 2024 (cited on page 6).

[CTU21] Tyler Chen, Thomas Trogdon, and Shashanka Ubaru. Analysis of stochastic lanczos quadrature for spectrum approximation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021 (cited on page 1).

[CTU22] Tyler Chen, Thomas Trogdon, and Shashanka Ubaru. Randomized matrix-free quadrature for spectrum and spectral sum approximation. *arXiv:2204.01941*, 2022 (cited on page 5).

[CDLLN23] Sinho Chewi, Jaume De Dios Pont, Jerry Li, Chen Lu, and Shyam Narayanan. Query lower bounds for log-concave sampling. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2023 (cited on page 6).

[CLS19] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, pages 938–942, 2019 (cited on page 16).

[CKSV18] David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1263–1271, 2018 (cited on pages 1, 5).

[CK22] Alice Cortinovis and Daniel Kressner. On randomized trace estimates for indefinite matrices with an application to determinants. *Foundations of Computational Mathematics*, 22(3):875–903, 2022 (cited on page 14).

[CKS91] Yann Le Cun, Ido Kanter, and Sara A. Solla. Eigenvalues of covariance matrices: application to neural-network learning. *Phys. Rev. Lett.*, 66:2396–2399, May 1991 (cited on page 5).

[DB16] Steven Diamond and Stephen Boyd. CVXPY: a Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016 (cited on pages 3, 20).

[DHMS21] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: new datasets for fair machine learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021 (cited on page 19).

[DSB21] Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. The limits of differential privacy (and its misuse in data release and machine learning). *Communications of the ACM*, 64(7):33–35, 2021 (cited on page 3).

[DBB19] Kun Dong, Austin R. Benson, and David Bindel. Network density of states. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1152–1161, 2019 (cited on page 5).

[DKMMN06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT*, pages 486–503, 2006 (cited on pages 4, 12).

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006 (cited on page 16).

[DNRRV09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 381–390, 2009 (cited on page 1).

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014 (cited on pages 2–4, 12).

[FL23] Zhiyuan Fan and Jian Li. Efficient algorithms for sparse moment problems without separation. In *Proceedings of the 36th Annual Conference on Computational Learning Theory (COLT)*, pages 3510–3565, 2023 (cited on page 1).

[FMST24] Vitaly Feldman, Audra McMillan, Satchit Sivakumar, and Kunal Talwar. Instance-optimal private density estimation in the wasserstein distance. *arXiv:2406.19566*, 2024 (cited on page 4).

[Fis11] Hans Fischer. *Chebyshev's and Markov's contributions.* In *A History of the Central Limit Theorem: From Classical to Modern Probability Theory.* Springer New York, 2011, pages 139–189 (cited on page 1).

[GKX19] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2232–2241, 2019 (cited on page 5).

[Gir87] Didier Girard. Un algorithme simple et rapide pour la validation croisee géénéralisée sur des problémes de grande taille. Technical report, École nationale supérieure d'informatique et de mathématiques appliquées de Grenoble, 1987 (cited on page 14).

[Hal15] Nicholas Hale. *Chebyshev polynomials*. In *Encyclopedia of Applied and Computational Mathematics*. Springer Berlin Heidelberg, 2015, pages 203–205 (cited on pages 7, 10).

[HLM12] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25 (NeurIPS)*, 2012 (cited on page 3).

[HRMS10] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3(12):1021–1032, 2010 (cited on page 4).

[HVZ23] Yiyun He, Roman Vershynin, and Yizhe Zhu. Algorithmically effective differentially private synthetic data. In *Proceedings of the 36th Annual Conference on Computational Learning Theory (COLT)*, pages 3941–3968, 2023 (cited on pages 3, 4).

[Hut90] Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990 (cited on pages 5, 14).

[Jac12] Dunham Jackson. On approximation by trigonometric sums and polynomials. *Transactions of the American Mathematical Society*, 13(4):491–515, 1912 (cited on page 3).

[Jac30] Dunham Jackson. *The Theory of Approximation*, volume 11 of *Colloquium Publications*. American Mathematical Society, 1930 (cited on page 7).

[JL14] Zhanglong Ji and Charles Lipton Zachary C.and Elkan. Differential privacy and machine learning: a survey and review. *arXiv:1412.7584*, 2014 (cited on page 3).

[JPWZ24] Shuli Jiang, Hai Pham, David P. Woodruff, and Qiuyi Zhang. Optimal sketching for trace estimation. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024 (cited on page 6).

[JKMSS24] Yujia Jin, Ishani Karmarkar, Christopher Musco, Apoorv Singh, and Aaron Sidford. Faster spectral density estimation and sparsification in the nuclear norm. In *Proceedings of the 37th Annual Conference on Computational Learning Theory (COLT)*, 2024 (cited on page 5).

[JMSS23] Yujia Jin, Christopher Musco, Aaron Sidford, and Apoorv Vikram Singh. Moments, random walks, and limits for spectrum approximation. In *Proceedings of the 36th Annual Conference on Computational Learning Theory (COLT)*, 2023 (cited on page 1).

[KMV10] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaussians. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 553–562, 2010 (cited on page 1).

[KV86] Sanjeev Kapoor and Pravin M. Vaidya. Fast algorithms for convex quadratic programming and multicommodity flows. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 147–159, 1986 (cited on page 9).

[KLN24] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. UCI Machine Learning Repository, Diabetes Health Indicators Dataset. https://www.archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators, 2024. [Accessed 11-07-2024] (cited on page 19).

[KM11] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011 (cited on page 11).

[KV17] Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45(5):2218–2247, October 2017 (cited on page 1).

[KW92] J. Kuczyski and H. Woniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992 (cited on page 14).

[LS15] Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 230–249, 2015 (cited on page 16).

[LLSY17] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. *Differential Privacy: From Theory to Practice*. Synthesis Lectures on Information Security, Privacy, and Trust. Springer, 2017 (cited on pages 3, 4).

[LXES19] Ruipeng Li, Yuanzhe Xi, Lucas Erlandson, and Yousef Saad. The eigenvalues slicing library (EVSL): algorithms, implementation, and software. *SIAM Journal on Scientific Computing*, 41(4):C393–C415, 2019 (cited on page 5).

[Lig58] Michael J. Lighthill. *An introduction to Fourier analysis and generalised functions*. Cambridge University Press, 1958 (cited on page 7).

[LVW21] Terrance Liu, Giuseppe Vietri, and Steven Z. Wu. Iterative methods for private synthetic data: unifying framework and new methods. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021 (cited on pages 3, 4).

[MM19] Michael Mahoney and Charles Martin. Traditional and heavy tailed self regularization in neural network models. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 4284–4293, 2019 (cited on page 5).

[MMSM22] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: an adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 15(11):2599–2612, 2022 (cited on pages 1, 3, 19).

[MM09] Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the Netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 627–636, 2009 (cited on page 4).

[MMMW21] Raphael A. Meyer, Cameron Musco, Christopher Musco, and David Woodruff. Hutch++: optimal stochastic trace estimation. *Proceedings of the 4th Symposium on Simplicity in Algorithms (SOSA)*, 2021 (cited on page 5).

[MTV+20] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. Privacy in deep learning: a survey. *arXiv:2004.12254*, 2020 (cited on page 3).

[MV10] Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 93–102, 2010 (cited on page 1).

[MAP20] Dean Moldovan, Misa Andelkovic, and Francois Peeters. pybinding v0.9.5: a Python package for tight-binding calculations, 2020 (cited on page 5).

[MOS19] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.* 2019 (cited on pages 3, 20).

[PB97] R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997 (cited on page 19).

[Pea94] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. (A.)*, 185:71–110, 1894 (cited on page 1).

[Pea36] Karl Pearson. Method of moments and method of maximum likelihood. *Biometrika*, 28(1/2):34–59, 1936 (cited on page 1).

[PSG18] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1924–1932, 2018 (cited on page 5).

[QYL13] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment*, 6(14):1954–1965, 2013 (cited on page 4).

[RSS14] Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. Learning mixtures of arbitrary distributions over large discrete domains. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 207–224, 2014 (cited on page 1).

[RA15] Farbod Roosta-Khorasani and Uri M. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015 (cited on page 14).

[RHR+23] Lucas Rosenblatt, Anastasia Holovenko, Taras Rumezhak, Andrii Stadnik, Bernease Herman, Julia Stoyanovich, and Bill Howe. Epistemic parity: reproducibility as an evaluation metric for differential privacy. *Proceedings of the VLDB Endowment*, 2023 (cited on pages 3, 19).

[RLP+20] Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: applied evaluations and enhancements. *arXiv:2011.05537*, 2020 (cited on page 1).

[SR94] Richard N. Silver and H. Röder. Densities of states of mega-dimensional hamiltonian matrices. *International Journal of Modern Physics C*, 5(4):735–753, 1994 (cited on page 5).

[SER18] Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. Tight query complexity lower bounds for PCA via finite sample deformed wigner law. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*, 2018 (cited on page 6).

[Ski89] John Skilling. *The eigenvalues of mega-dimensional matrices*. In *Maximum Entropy and Bayesian Methods*. Springer Netherlands, 1989 (cited on page 5).

[SWYZ21] Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. Querying a matrix through matrix-vector products. *ACM Trans. Algorithms*, 17(4), 2021 (cited on page 6).

[Teb21] Alex Teboul. Diabetes Health Indicators Dataset. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset, 2021. [Accessed 11-07-2024] (cited on page 19).

[Tre08] Lloyd N. Trefethen. Is Gauss Quadrature Better than Clenshaw–Curtis? *SIAM Review*, 50(1):67–87, 2008 (cited on page 8).

[Tre19] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM-Society for Industrial and Applied Mathematics, 2019 (cited on pages 3, 7).

[VAA+22] Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Z Wu. Private synthetic data for multitask learning and marginal queries. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022 (cited on page 4).

[Wai19] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019 (cited on page 28).

[WJF+16] Ziteng Wang, Chi Jin, Kai Fan, Jiaqi Zhang, Junliang Huang, Yiqiao Zhong, and Liwei Wang. Differentially private data releasing for smooth queries. *Journal of Machine Learning Research*, 17(51):1–42, 2016 (cited on pages 1–4).

[WWAF06] Alexander WeiSSe, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. The kernel polynomial method. *Rev. Mod. Phys.*, 78:275–306, 2006 (cited on pages 1, 5).

[WZZ22] David Woodruff, Fred Zhang, and Richard Zhang. Optimal query complexities for dynamic trace estimation. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022 (cited on pages 5, 29).

[WR22] Stephen J. Wright and Benjamin Recht. *Optimization for Data Analysis*. Cambridge University Press, 2022 (cited on page 9).

[WY19] Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, 47(2):857–883, 2019 (cited on page 1).

[WY20] Yihong Wu and Pengkun Yang. Optimal estimation of Gaussian mixtures via denoised method of moments. *The Annals of Statistics*, 48(4):1981–2007, 2020 (cited on page 1).

[XWG10] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering*, 23(8):1200–1214, 2010 (cited on page 4).

[XZX+13] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22:797–822, 2013 (cited on page 4).

[YGKM20] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. PyHessian: neural networks through the lens of the Hessian. In *2020 IEEE International Conference on Big Data*, pages 581–590, 2020 (cited on page 5).

[YT89] Yinyu Ye and Edison Tse. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, 44(1):157–179, 1989 (cited on page 9).

[ZCPSX17] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: private data release via Bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017 (cited on page 3).

[ZXX16] Jun Zhang, Xiaokui Xiao, and Xing Xie. PrivTree: a differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*, pages 155–170, 2016 (cited on page 4).

A Proof of Corollary 2

In this section, we give the full proof of Corollary 2. We require the following basic property about the Chebyshev nodes:

Lemma 23 (Chebyshev Node Approximation)**.** *Let $\mathcal{C} = \{x_1, \ldots, x_g\}$ be the degree g Chebyshev nodes. I.e., $x_i = \cos\left(\frac{2i-1}{2g}\pi\right)$. Let $r_{\mathcal{C}} : [-1,1] \to \mathcal{C}$ be a function that maps a point $x \in [-1,1]$ to the point $y \in \mathcal{C}$ that minimizes $\left|\cos^{-1}(x) - \cos^{-1}(y)\right|$, breaking ties arbitrarily. For any $x \in [-1,1]$, $\left|\cos^{-1}(x) - \cos^{-1}(r_{\mathcal{C}}(x))\right| \leq \frac{\pi}{2g}$.*

Proof. For any two consecutive points x_i, x_{i+1} in the $\mathcal{C}$,

$$\left|\cos^{-1}(x_i) - \cos^{-1}(x_{i+1})\right| = \frac{\pi}{g}.$$

Since $\cos^{-1}(x)$ is non-increasing, for any $x \in [x_{i+1}, x_i]$, $\cos^{-1}(x) \in [\cos^{-1}(x_i), \cos^{-1}(x_{i+1})]$. So, $\cos^{-1}(x)$ has distance at most $\frac{\pi}{2g}$ from either $\cos^{-1}(x_i)$ or $\cos^{-1}(x_{i+1})$. Additionally, we can check that $\left|\cos^{-1}(x) - \cos^{-1}(x_1)\right| \leq \frac{\pi}{2g}$ for any $x < x_1$ and $\left|\cos^{-1}(x) - \cos^{-1}(x_g)\right| \leq \frac{\pi}{2g}$ for any $x > x_g$. □

With Lemma 23 in place, we are ready to prove Corollary 2.

Proof of Corollary 2. Let $\mathcal{C}$ and $r_{\mathcal{C}} : [-1, 1] \to \mathcal{C}$ be as in Lemma 23. For $i \in \{1, \ldots, g\}$, let Y_i be the set of points in $[-1, 1]$ that are closest to $x_i \in \mathcal{C}$, i.e., $Y_i = \{x \in [-1, 1] : r_{\mathcal{C}}(x) = x_i\}$. Let $\tilde{p}$ be a distribution supported on the set $\mathcal{C}$ with mass $\int_{Y_i} p(x)\,dx$ on $x_i \in \mathcal{C}$. For all $j \in 1, \ldots, k$ we have:

$$\begin{aligned}
\left|\langle p, \bar{T}_j\rangle - \langle \tilde{p}, \bar{T}_j\rangle\right| &= \left|\sum_{i=1}^{g} \int_{Y_i} \bar{T}_j(x)p(x)\,dx - \left(\int_{Y_i} p(x)\,dx\right)\bar{T}_j(x_i)\right| \\
&= \left|\sum_{i=1}^{g} \left(\int_{Y_i} p(x)\,dx\right)\bar{T}_j(y_i) - \left(\int_{Y_i} p(x)\,dx\right)\bar{T}_j(x_i)\right| \qquad \text{(for some } y_i \in Y_i\text{)} \\
&\leq \sum_{i=1}^{g} \left(\int_{Y_i} p(x)\,dx\right)\left|\bar{T}_j(y_i) - \bar{T}_j(x_i)\right| \\
&= \sum_{i=1}^{g} \left(\int_{Y_i} p(x)\,dx\right) \cdot \sqrt{\frac{2}{\pi}} \cdot \left|\cos(j\cos^{-1}(y_i)) - \cos(j\cos^{-1}(x_i))\right| \\
&\leq \sum_{i=1}^{g} \left(\int_{Y_i} p(x)\,dx\right) \cdot \sqrt{\frac{2}{\pi}} \cdot \frac{j\pi}{2g} = \frac{j\sqrt{\pi/2}}{g} \qquad (18)
\end{aligned}$$

The second equality follows from the intermediate value theorem. The first inequality follows by triangle inequality. The third equality follows by the trigonometric definition of the (normalized) Chebyshev polynomials. The second inequality follows from Lemma 23 and the fact that the derivative of $\cos(jx)$ is bounded by j. The bound in (18) then yields:

$$\left(\sum_{j=1}^{k} \frac{1}{j^2}\left(\langle p, \bar{T}_j\rangle - \langle \tilde{p}, \bar{T}_j\rangle\right)^2\right)^{1/2} \leq \frac{\sqrt{\pi k/2}}{g}. \qquad (19)$$

Observe also that, since $\tilde{p}$ is supported on $\mathcal{C}$, it is a valid solution to the optimization problem solved by Algorithm 1. Accordingly, we have that:

$$\left(\sum_{j=1}^{k} \frac{1}{j^2}\left(\hat{m}_j - \langle q, \bar{T}_j\rangle\right)^2\right)^{1/2} \leq \left(\sum_{j=1}^{k} \frac{1}{j^2}\left(\hat{m}_j - \langle \tilde{p}, \bar{T}_j\rangle\right)^2\right)^{1/2} \qquad (20)$$

Applying triangle inequality, followed by (20), triangle inequality again, and finally (19), we have:

$$\begin{aligned}
\left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\langle p,\bar{T}_j\rangle-\langle q,\bar{T}_j\rangle\right)^2\right)^{1/2} &\le \left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\langle p,\bar{T}_j\rangle-\hat{m}_j\right)^2\right)^{1/2}+\left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\hat{m}_j-\langle q,\bar{T}_j\rangle\right)^2\right)^{1/2}\\
&\le \left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\langle p,\bar{T}_j\rangle-\hat{m}_j\right)^2\right)^{1/2}+\left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\hat{m}_j-\langle \tilde{p},\bar{T}_j\rangle\right)^2\right)^{1/2}\\
&\le 2\left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\langle p,\bar{T}_j\rangle-\hat{m}_j\right)^2\right)^{1/2}+\left(\sum_{j=1}^{k}\frac{1}{j^2}\left(\langle p,\bar{T}_j\rangle-\langle \tilde{p},\bar{T}_j\rangle\right)^2\right)^{1/2}\\
&\le 2\Gamma+\frac{\sqrt{2\pi k}}{g}.
\end{aligned}$$

Setting $g=\lceil k^{1.5}\rceil$, we can apply Theorem 1 to conclude that, for a fixed constant c',

$$W_1(p,q)\le c\cdot\left(\frac{1}{k}+2\Gamma+\frac{\sqrt{\pi/2}}{k}\right)\le c'\cdot\left(\frac{1}{k}+\Gamma\right). \qquad \square$$

B Theorem 4 High Probability Bound

In this section, we prove the high probability bound on Wasserstein distance stated in Theorem 4, which follows from a standard concentration bound for sub-exponential random variables [Wai19]. We recall that a random variable X is subexponential with parameters (ν,α) if:

$$\mathbb{E}[e^{\lambda(X-\mathbb{E}[X])}]\le e^{\nu^2\lambda^2/2} \qquad \text{for all} \qquad |\lambda|\le\frac{1}{\alpha}.$$

We require the following well-known fact that a chi-square random variable with one degree of freedom is subexponential:

Fact 24 (Sub-Exponential Parameters [Wai19, Example 2.8])**.** *Let $\eta\sim\mathcal{N}(0,\sigma^2)$. Then, η^2 is sub-exponential random variable with parameters $(2\sigma^2,4\sigma^2)$.*

We also require the following concentration inequality for a sum of sub-exponential random variable:

Fact 25 ([Wai19, Equation 2.18])**.** *Consider independent random variables $\gamma_1,\dots,\gamma_k$, where, $\forall j\in 1,\dots,k$, γ_j is sub-exponential with parameters (ν_j,α_j). Let $\nu_*=\sqrt{\sum_{j=1}^k\nu_j^2}$ and $\alpha_*=\max\{\alpha_1,\dots,\alpha_k\}$. Then we have:*

$$\mathbb{P}\left[\sum_{j=1}^{k}(\gamma_j-\mathbb{E}[\gamma_j])\ge t\right]\le\begin{cases}\exp\left(\frac{-t^2}{2\nu_*^2}\right) & \text{for } 0\le t\le\frac{\nu_*^2}{\alpha_*},\\ \exp\left(\frac{-t}{2\alpha_*}\right) & \text{for } t>\frac{\nu_*^2}{\alpha_*}.\end{cases}$$

Proof of high-probability bound of Theorem 4. Recalling the proof of the expectation bound of Theorem 4 from Section 4, it suffices to bound $E=\sum_{j=1}^k\frac{1}{j^2}\left(\hat{m}_j(p)-\langle\tilde{p},\bar{T}_j\rangle\right)^2$ with high probability. Let $\gamma_j=\eta_j^2/j^2$, where $\eta_j\sim\mathcal{N}(0,j\sigma^2)$ is as in Algorithm 2. Then recall that $E=\sum_{j=1}^k\gamma_j$.

From Fact 24, γ_j is a sub-exponential random variable with parameter $(2\sigma^2/j,4\sigma^2/j)$. We can then apply Fact 25, for which we have $\nu_*=\sqrt{\sum_{j=1}^k 4\sigma^4/j^2}\le 2\pi\sigma^2/\sqrt{6}$ and $\alpha_*=4\sigma^2$. For any failure probability $\beta\in(0,1/2)$, setting $t=8\log(1/\beta)\sigma^2$, we conclude that:

$$\mathbb{P}\left[E-\mathbb{E}[E]\ge 8\log(1/\beta)\,\sigma^2\right]\le\beta\,.$$

Recalling from Equation (13) that $\mathbb{E}[E] \le (1+\log k)\sigma^2$, we conclude that $E \le 8\log(1/\beta)\,\sigma^2 + (1+\log k)\sigma^2$ with probability at least $1-\beta$.

The rest of the details follow as before. In particular, as in Equation (14), we can bound:

$$W_1(p,q) \le \sqrt{\pi}\Gamma + \frac{36}{k} + \frac{1}{2\lceil \epsilon n \rceil},$$

where $\Gamma \le \sqrt{2E}$. Plugging in $k = \lceil 2\epsilon n \rceil$ (as chosen in Algorithm 2) and recalling that $\sigma^2 = \frac{16}{\pi}(1+\log k)\ln(1.25/\delta)/(\epsilon^2 n^2)$, we conclude that with probability $\ge 1-\beta$, for a fixed constant c,

$$W_1(p,q) \le c\left(\frac{\sqrt{\log(\epsilon n)+\log(1/\beta)}\sqrt{\log(\epsilon n)\log(1/\delta)}}{\epsilon n}\right). \qquad \square$$

C Spectral Density Estimation Lower Bound

In this section, we provide a lower bound on the number of matrix-vector multiplications required for spectral density estimation. We first need the following theorem from Woodruff et al. [WZZ22], which shows that estimating the trace of a positive semi-definite matrix A to within a multiplicative error of $(1\pm\epsilon)$ requires $\Omega(1/\epsilon)$ number of matrix-vector multiplications with A.

Theorem 26 (Restated [WZZ22, Theorem 4.2])**.** *For all $\delta > 0$ and $\epsilon = O\left(1/\sqrt{\log(1/\delta)}\right)$, any algorithm that is given matrix-vector multiplication access to a positive semi-definite (PSD) input matrix $A \in \mathbb{R}^{n\times n}$ with $\|A\|_2 \le 1$, $n/4 \le \mathrm{tr}(A) \le n$ and succeeds with probability at least $1-\delta$ in outputting an estimate $\tilde{t}$ such that $|\tilde{t} - \mathrm{tr}(A)| \le \epsilon \cdot \mathrm{tr}(A)$ requires $\Omega\left(\frac{\log(1/\delta)}{\epsilon}\right)$ matrix-vector multiplications with A.*

As a corollary of this result, we obtain the following lower bound, which shows that Theorem 5 is tight up to $\log(1/\epsilon)$ factors:

Corollary 27. *Any algorithm that is given matrix-vector multiplication access to a symmetric matrix A with spectral density p and $\|A\|_2 \le 1$ requires $\Omega\left(\frac{\log(1/\delta)}{\epsilon}\right)$ matrix-vector multiplications with A to output a distribution q such that $W_1(p,q) \le \epsilon$.*

Proof. The proof is via a direct reduction. Consider a PSD matrix A with $\|A\|_2 \le 1$, $n/4 \le \mathrm{tr}(A) \le n$, and spectral density p. Suppose we have a spectral density estimate q of p such that $W_1(p,q) \le \epsilon/4$. We claim that $\tilde{t} = n\cdot\int_{-1}^{1} xq(x)\,dx$ yields a relative error approximate to A's trace, implying that computing such a q requires $\Omega(\log(1/\delta)/\epsilon)$ by Theorem 26.

In particular, applying Kantorovich-Rubinstein duality (Fact 7) with the 1-Lipschitz functions $f(x) = x$ and $f(x) = -x$, we have that:

$$\int_{-1}^{1} xp(x)\,dx - \int_{-1}^{1} xq(x)\,dx \le \epsilon/4 \qquad \text{and} \qquad \int_{-1}^{1} xq(x)\,dx - \int_{-1}^{1} xp(x)\,dx \le \epsilon/4. \tag{21}$$

We have that $\int_{-1}^{1} xp(x)\,dx = \frac{1}{n}\mathrm{tr}(A)$. So (21) implies that $\tilde{t} = n\cdot\int_{-1}^{1} xq(x)\,dx$ satisfies:

$$|\tilde{t} - \mathrm{tr}(A)| \le n\cdot\epsilon/4 \le \epsilon\cdot\mathrm{tr}(A). \qquad \square$$