DIFFERENTIALLY PRIVATE SYNTHETIC DATA FROM SLICED CHEBYSHEV MOMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Research on practical differentially private (DP) synthetic data has largely focused on methods that approximately preserve low-dimensional marginals of a target distribution, and can thus only hope to preserve higher-dimensional, global structure as a lucky by-product. At the same time, existing methods that target global accuracy guarantees pay a price exponential in the data dimension, d.

In this work, we address this "curse of dimensionality" by presenting a new DP synthetic data algorithm, `SPLICE`, that comes with a compelling global distributional guarantee, even in high-dimensions, and moreover, is practically state-of-the-art. Our algorithm is based on measuring noisy *sliced* Chebyshev polynomial moments of the target distribution and fitting a distribution to match, a twist on recent polynomial methods for 1-D synthetic data.

We prove that, given a dataset of n examples, `SPLICE` returns a private distribution with Sliced Wasserstein-1 distance $\tilde{O}(d^{1/4}/\sqrt{n})$ from the target, for constant privacy parameters. So, we only need n to be *polynomial* in the data dimension for a high-quality approximation. Along the way, we prove a moment matching result of independent interest: for a distribution on the unit ball, matching the first $O(1/\alpha)$ Chebyshev moments along $\tilde{O}(d/\alpha^2)$ random directions guarantees α Sliced Wasserstein error.

Finally, we provide a practical implementation of the algorithm that borrows tools from the tensor decomposition literature and achieves state-of-the-art performance compared to prior DP synthetic data methods on an extensive benchmark.

1 INTRODUCTION

The generation of high-utility differentially private (DP) synthetic data has become a focal point of research into DP data release (Cummings et al., 2024). In the basic setup, we are given a dataset $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and wish to privately compute a distribution q that well-approximates the empirical distribution of $\mathcal{X}$, which we denote by p (i.e., p is uniform over $x_1, \ldots, x_n$). The hope is that q, or samples from q, can be used as a drop-in replacement for $\mathcal{X}$ in downstream machine learning and data analysis tasks, without the need for any further differential privacy mechanisms.

To guarantee strong performance on downstream tasks, a natural goal is to develop algorithms that ensure q is close to p with respect to some meaningful statistical distance, like Wasserstein distance or some other integral probability metric. There has been significant theoretical work targeting such guarantees (He et al., 2023; Boedihardjo et al., 2024; Feldman et al., 2024; Musco et al., 2025; Ding et al., 2026). However, all of this work obtains bounds that suffer exponentially in the data dimension, d, so it has yielded few practically competitive algorithms.

Indeed, empirically oriented synthetic data methods give up on strong distribution guarantees, focusing on achieving practical utility on realistic downstream tasks (Zhang et al., 2017; Liu et al., 2021b; McKenna et al., 2022; Liu et al., 2023). Many of the best practical methods are based on noisily measuring only low-order marginals of p (involving at most 2 or 3 variables (McKenna et al., 2022)) and returning a synthetic distribution that matches these marginals (the "Select-Measure-Estimate" paradigm (Ponomareva et al., 2026)). While they sometimes come with workload-specific error guarantees, it is not surprising that higher-dimensional correlations and structure are missed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1.1 Our Contributions

This work seeks to address this "curse of dimensionality" in synthetic data generation. In short, we introduce a new algorithm, SPLICE, that enjoys a strong distributional error guarantee, even in high dimensions, and moreover, offers state-of-the-art practical performance on downstream tasks.

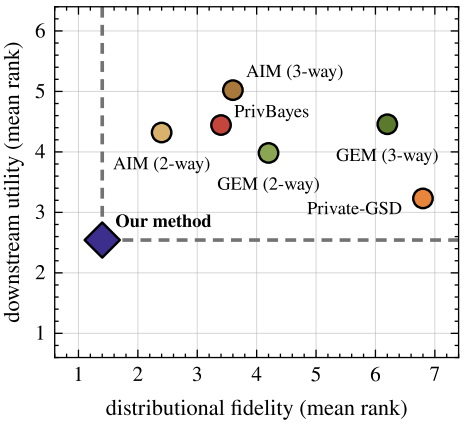


Figure 1: This plot summarizes a comparison of state-of-the-art DP synthetic data algorithms. Each method is evaluated on the 48 downstream train-on-synthetic, test-on-real tasks and ranked according to accuracy. The distribution returned by each method was also ranked using five fidelity metrics (1-way, 2-way, and 3-way marginal TVD, kernel MMD, and sliced W_1). The average rank determines the x and y coordinates. Lower is better on both axes.

SPLICE is inspired by a recent approach for low-dimensional synthetic data generation based on matching polynomial moments (specifically, noisy Chebyshev polynomial moments) (Wang et al., 2016). This approach yields near-optimal bounds in terms of Wasserstein-1 (W_1) distance: Musco et al. (2025) show that, for a dataset with n 1-d points, moment matching yields an (ϵ, δ)-DP distribution q such that $W_1(p, q) \leq \tilde{O}(\sqrt{\log(1/\delta)}/\epsilon n)$. At the same time, Chebyshev moment matching falls under the "Select-Measure-Estimate" paradigm, so is easy to implement efficiently.

Unfortunately, however, like other approaches listed above, Chebyshev moment matching does not readily scale to high dimensions. For d-dimensional data, the bound of Musco et al. (2025) degrades to $\tilde{O}(1/n^{1/d})$ for constant ϵ, δ. This is vacuous for $d \gtrsim \log n$ and weak even for small constant d. Moreover, the bound cannot be improved: Boedihardjo et al. (2024) show that no method can privately achieve Wasserstein-1 error below $O(1/n^{1/d})$.

Algorithmic Contribution. The key issue with prior work is that obtaining small Wasserstein-1 error requires matching multivariate products of Chebyshev polynomials, the number of which grows exponentially in the number of data variables. In SPLICE, we propose to instead measure *sliced* Chebyshev polynomial moments, which for any unit vector $\theta \in \mathbb{R}^d$, take the form:

$$\mathbb{E}_{x \sim \mathrm{Unif}(\mathcal{X})} \left[T_j(\theta^T x) \right],$$

where T_j is a Chebyshev polynomial of degree j. The method computes moments of this form for L different random vectors $\theta_1, \ldots, \theta_L$, where L scales polynomially in d. It then adds noise to those moments to ensure differential privacy, and finally fits a distribution q that matches the noisy moments. In other words, the method again fits into the popular "Select-Measure-Estimate" paradigm, but replaces low-degree marginal measurements with *sliced* moment measurements.

Our main theoretical result is to prove that, while Wasserstein distance is out of reach, SPLICE yields a strong bound on the *sliced Wasserstein-1 distance* between our private distribution q and target p. Letting W_1 denotes the one-dimensional Wasserstein-1 distance and $\mathcal{S}^{d-1}$ denotes the unit sphere in d dimensional Euclidean space, sliced Wasserstein-1 distance is defined as:

$$\mathrm{SW}_1(p, q) := \mathbb{E}_{\theta \sim \mathrm{Unif}(\mathcal{S}^{d-1})} \left[W_1(\bar{p}_\theta, \bar{q}_\theta) \right], \tag{1}$$

where $\bar{p}_\theta$ (analogously, $\bar{q}_\theta$) is the pushforward distribution obtained by projecting p onto $\theta \in \mathcal{S}^{d-1}$:

$$\bar{p}_\theta(A) = p(\{x : \theta^\top x \in A\}) \text{ for any measurable set } A \subseteq \mathbb{R}. \tag{2}$$

In other words, $\mathrm{SW}_1(p, q)$ is the average Wasserstein-1 distance between one-dimensional projections of p and q. Originally introduced as a tractable approximation for the standard Wasserstein-1 distance (Rabin et al., 2011; Bonneel et al., 2015), sliced Wasserstein distance has been widely adopted across machine learning, especially in the generative model literature, as a valuable measure in its own right (Deshpande et al., 2018; 2019; Kolouri et al., 2018; 2019a; Liutkus et al., 2019; Nadjahi et al., 2020; Sebag et al., 2025; Gu et al., 2025).

Concretely, we prove the following bound, which is stated more formally and proven in Section 3:

Theorem 5. *Let $\mathcal{B}^d$ denote the unit ball in $\mathbb{R}^d$. There is an (ϵ, δ)-differentially private algorithm (SPLICE) that, given a dataset $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathcal{B}^d$ with $p = \mathrm{Unif}(\mathcal{X})$, returns q such that*

$$\mathrm{SW}_1(p, q) \leq \tilde{O}\left(\frac{d^{1/4}}{\sqrt{n\epsilon}}\right), \qquad \textit{with probability at least } 99/100. \tag{9}$$

Critically, the dependence on the data dimension d in Theorem 5 is polynomial: we can obtain a meaningfully accurate synthetic data distribution with just $n = O(\sqrt{d}/\epsilon)$ data examples. In contrast, as discussed above, prior moment matching methods require $n = 2^{\Omega(d)}$ Boedihardjo et al. (2024).

Key to the proof of Theorem 5 is a moment matching result of potentially independent interest. We show that, if two high-dimensional distributions, p and q, match on their first $1/\alpha$ moments along $L = \tilde{O}(d/\alpha^2)$ random slices, then $\mathrm{SW}_1(p, q) \leq O(\alpha)$. This generalizes the well-known fact that, in one dimension, matching the first $1/\alpha$ moments of two distributions guarantees $W_1(p, q) \leq \alpha$ (Kong & Valiant, 2017; Braverman et al., 2022). The key technical challenge in establishing this result is to show that a polynomial bound on the required number of slices, L, suffices. A naive analysis suggests that L needs to scale exponentially in d to cover the d dimensional sphere.

Empirical Contribution. Theorem 5 tells us how many data examples, n suffice to build an accurate synthetic distribution, q, but does not say anything about the complexity of representing or learning that distribution from noisy moment measurements. To make our results practical, we choose to represent q as a mixture of K product distributions: in each component of the mixture, discrete random variables follow independent distributions, and continuous random variables are drawn from a single joint Gaussian distribution with low-rank-plus-diagonal covariance. Dependencies between discrete variables are introduced by mixing multiple such distributions.

The class of distributions we consider thus involves $O(d)$ parameters per component, so $O(Kd)$ in total. We fit these parameters using gradient descent, relying on the key fact that the distribution's sliced Chebyshev moments have an exact, differentiable closed form. Empirically, using datasets from the extensive TALENT tabular data benchmark (Liu et al., 2025), we evaluate SPLICE with this chosen representation for q. Our empirical results are quite striking.

Concretely, we find that the mechanism gives the best regression utility of any method among a large set of state-of-the-art benchmarks, at every privacy budget. It also offers the best error on 2-way and 3-way marginal fidelity for $\epsilon \geq 1$, despite not targeting marginals directly, and achieves the best causal structure recovery by a significant margin. These results are summarized in Figure 1 at $\epsilon = 1$, where SPLICE outperforms strong marginal-based baselines on both composite axes simultaneously, distributional fidelity and downstream utility, with detailed experiments contained in Section 5. Finally, we note that the runtime of SPLICE is competitive with the fastest baselines and dramatically faster than the slowest for typical dataset sizes.

Roadmap. Section 2 provides our key technical result, that sliced moments along a small number of random directions suffice to control SW_1. Section 3 leverages this result to obtain the DP guarantee of Theorem 5. Section 4 instantiates the algorithm with a practical learned representation, and Section 5 empirically tests it. Section A surveys related work in additional depth.

2 Sliced Chebyshev Moment Matching

We begin with basic preliminaries on the Wasserstein distance and moment matching. Throughout, our goal will be to approximate a distribution p, which in the case of differentially private synthetic data, will be taken to be the uniform distribution over some dataset $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$.

Let q be some other distribution. The Wasserstein-1 distance between p and q is defined as follows:

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

where $\|\cdot\|_2$ is the Euclidean distance and $Z(p, q)$ is the set of all couplings between p and q.

A powerful fact is that, if two 1-dimensional distributions p and q supported[1] on $[-1, 1]$ match on their first k polynomial moments (i.e., $\mathbb{E}_{x\sim p}[x^j] = \mathbb{E}_{x\sim q}[x^j]$ for all $j \in 1, \ldots, k$), then $W_1(p, q) \leq O(1/k)$ (Kong & Valiant, 2017; Braverman et al., 2022). This claim follows from combining Jackson's theorem (Jackson, 1912), which says that any 1-Lipschitz function on $[-1, 1]$ can be approximated to error $O(1/k)$ by a degree k polynomial, with the Kantorovich-Rubinstein dual view of Wasserstein distance, which states that $W_1(p, q) = \max_{1 \text{ Lipschitz } f} \mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{x\sim q}[f(x)]$.

Importantly, this moment matching claim can be made robust when using a basis of polynomial moments besides the standard basis (which is ill-conditioned). For example, let T_j denote the j^{th} Chebyshev polynomial of the first kind:

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_{j+1}(x) = 2x \cdot T_j(x) - T_{j-1}(x) \text{ for } j \geq 1.$$

Let $m_j(p) = \mathbb{E}_{x\sim p}[T_j(x)]$ denote the j^{th} Chebyshev moment. A recent result shows the following.

Theorem 1 ((Musco et al., 2025, Theorem 1))**.** *Let p, q be probability measures on $[-1, 1]$. For any positive integer k,*

$$W_1(p, q) \leq \frac{36}{k} + \Gamma_k(p, q), \qquad \textit{where} \qquad \Gamma_k(p, q) = \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(m_j(p) - m_j(q)\right)^2\right)^{1/2}. \tag{3}$$

Theorem 1 is useful when we wish to recover a distribution p from *noisy* approximations to its moments. In particular, suppose we know the first k Chebyshev moments up to cumulative error Δ_k (in the sense of equation 3) and find q matching them to error Δ_k (such a q always exists, since p is a valid choice). Because Γ_k is a norm satisfying triangle inequality, any such q satisfies $W_1(p, q) \leq 36/k + 2\Delta_k$. This argument was used by Musco et al. (2025) to obtain a near-optimal algorithm for 1-dimensional DP synthetic data based on noising Chebyshev moments with the Gaussian mechanism (Dwork & Roth, 2014). They show that, for a dataset with n 1-dimensional points, it is possible to return an (ϵ, δ)-differentially private distribution q satisfying $W_1(p, q) \leq \tilde{O}(\sqrt{\log(1/\delta)}/\epsilon n)$.

2.1 Sliced Moment Matching

As discussed, extending Theorem 1 to d dimensions requires matching a number of moments exponential in d, resulting in Wasserstein error scaling as $O(1/n^{1/d})$ Musco et al. (2025). Our goal is to avoid this poor dependence on d by focusing on the relaxed sliced Wasserstein-1 distance defined in equation 1, noting that we always have $\text{SW}_1(p, q) \leq W_1(p, q)$ Carlier et al. (2025).

Towards this goal, we can immediately obtain the following corollary of Theorem 1 for high dimensional distributions. In particular, let p, q be probability measures on the unit ball $\mathcal{B}^d = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ and define $m_j(\theta, p) = \mathbb{E}_{x\sim p}[T_j(x^\top \theta)]$ to be the j^{th} Chebyshev moment of the one-dimensional projection of p as defined in equation 2. Define $m_j(\theta, q)$ analogously. For any probability measure q on $\mathcal{B}^d$, define its weighted moment error in direction θ by

$$\Gamma_k(\theta, p, q) := \left(\sum_{j=1}^{k} \frac{1}{j^2} \left(m_j(\theta, p) - m_j(\theta, q)\right)^2\right)^{1/2}. \tag{4}$$

Applying Theorem 1 to the one-dimensional projections and averaging over directions gives:

Corollary 1. *Let p, q be probability measures on $\mathcal{B}^d$. For any positive integer k,*

$$\text{SW}_1(p, q) \leq \frac{36}{k} + \bar{\Gamma}_k(p, q), \qquad \textit{where} \qquad \bar{\Gamma}_k(p, q) = \mathbb{E}_{\theta\sim \text{Unif}(S^{d-1})}[\Gamma_k(\theta, p, q)]. \tag{5}$$

Corollary 1 is not immediately algorithmically useful, as it requires us to control (on average) how closely q matches the first k Chebyshev moments of p in every direction $\theta \in \mathcal{S}^{d-1}$. Our main technical tool is a proof that it suffices to instead only match moments for a small random sample of directions $\theta_1, \ldots, \theta_L$ drawn uniformly from $\mathcal{S}^{d-1}$. In particular, we show:

[1] The support assumption is to simplify notation, since we can always rescale the data to lie in $[-1, 1]$. Without scaling, the error would be $O(1/k)$ times the diameter of the support. For high-dimensional distributions, we will likewise assume that p and q are supported on the unit ball $\mathcal{B}^d$ to reduce notational complexity.

Theorem 2. *Let p be a probability measure on $\mathcal{B}^d$, let L be a positive integer, let $\beta \in (0,1)$, and let $\theta_1, \ldots, \theta_L$ be independent draws from $\mathrm{Unif}(S^{d-1})$. For any positive integer k, with probability at least $1-\beta$ over the choice of $\theta_1, \ldots, \theta_L$, simultaneously for all probability measures q on $\mathcal{B}^d$,*

$$\mathrm{SW}_1(p,q) \leq \frac{36}{k} + \tilde{\Gamma}_{k,L}(p,q) + \frac{32}{\beta}\sqrt{\frac{d\log(6Lk^2)}{L}}, \quad \textit{where } \tilde{\Gamma}_{k,L}(p,q) = \frac{1}{L}\sum_{\ell=1}^{L}\Gamma_k(\theta_\ell, p, q). \tag{6}$$

A consequence of this theorem is that, if we compute the first k moments of p along $L = \tilde{O}(dk^2)$ random directions, the resulting information suffices to find a distribution that is $O(1/k)$ close to p in sliced Wasserstein distance. In particular, we return any distribution that matches all of the resulting kL moments (so $\tilde{\Gamma}_{k,L}(p,q) = 0$). Plugging into Theorem 2, we obtain $\mathrm{SW}_1(p,q) = O(1/k)$.

However, the theorem is even more flexible in that, like Theorem 1, it allows us to return a q that only approximately matches p's sliced moments. Approximation will be necessary in our application to differentially private data generation in Section 3, since we only have access to noised moments.

Theorem 2 is proven by establishing that $\bar{\Gamma}_k(p,q) = \mathbb{E}_{\theta\sim\mathrm{Unif}(S^{d-1})}[\Gamma_k(\theta,p,q)]$ is well approximated by $\tilde{\Gamma}_{k,L}(p,q) = \frac{1}{L}\sum_{\ell=1}^{L}\Gamma_k(\theta_\ell,p,q)$ for all q. In particular, we show:

Lemma 3. *Fix any positive integer k and any probability measure p on $\mathcal{B}^d$. Let L be a positive integer, let $\beta \in (0,1)$, and let $\theta_1, \ldots, \theta_L$ be independent draws from $\mathrm{Unif}(S^{d-1})$. With probability at least $1-\beta$ over the choice of $\theta_1, \ldots, \theta_L$, simultaneously for all probability measures q on $\mathcal{B}^d$,*

$$\left|\frac{1}{L}\sum_{\ell=1}^{L}\Gamma_k(\theta_\ell,p,q) - \mathbb{E}_{\theta\sim\mathrm{Unif}(S^{d-1})}[\Gamma_k(\theta,p,q)]\right| \leq \frac{32}{\beta}\sqrt{\frac{d\log(6Lk^2)}{L}}. \tag{7}$$

Proof of Theorem 2. By Corollary 1, $\mathrm{SW}_1(p,q) \leq 36/k + \bar{\Gamma}_k(p,q)$. By Lemma 3, $\bar{\Gamma}_k(p,q) \leq \tilde{\Gamma}_{k,L}(p,q) + \frac{32}{\beta}\sqrt{\frac{d\log(6Lk^2)}{L}}$ for all q, which immediately gives the theorem. □

Due to space limits, Theorem 3 is proven in Section B. A natural proof approach would be to apply a concentration bound for each fixed distribution q, then extend to all distributions using an ϵ-net argument. However, directly discretizing the space of distributions introduces an exponential dependence on d, since representing a distribution on the d-dimensional ball requires roughly $2^{O(d)}$ parameters. We avoid this issue by using symmetrization and Gaussian comparison arguments to reduce the problem to bounding a random linear combination of empirical moments. Since moments are linear in q, the resulting expression is maximized by a distribution that places all its mass at a single point. We can thus union bound over the d-dimensional ball, which has log-covering number just $O(d)$, leading to a linear dependence in the required number of direction samples, L.

3 The SPLICE Algorithm for DP Synthetic Data

Theorem 2 suggests a route to differentially private synthetic data that mirrors the one-dimensional mechanism of Musco et al. (2025): compute sliced moments of the target dataset, intentionally add noise to those moments to ensure privacy, and construct a distribution q that matches the noised moments. We present this approach formally in Algorithm 1 below, showing how to calibrate the noise, and we then analyze the resulting accuracy of the method for any given privacy parameters.

We remark that, as in Musco et al. (2025), more noise is added to higher-degree polynomial moments, since these moments play a diminishing role in capturing global structure in p. Additionally, we highlight that we intentionally do not specify the details of the final "Match" step of the algorithm. Our goal in this section is to show that, if we solve the minimization problem shown, then q will be close to p in sliced Wasserstein distance with high probability. This is our main utility result. In the subsequent section, we describe an efficient heuristic for solving the problem in practice.

We first establish the privacy of Algorithm 1. Doing so is straight-forward, as q only depends on p through the noisy moments, $\tilde{m}_{j,\ell}$. These moments, in turn, are obtained by adding $\mathcal{N}(0,\sigma^2)$ noise (i.e., applying the standard Gaussian mechanism) to the vector valued function:

$$f(\mathcal{X})_{(\ell,j)} = \frac{1}{\sqrt{j}}\cdot\frac{1}{n}\sum_{i=1}^{n}T_j(\theta_\ell^\top x_i) \qquad \text{for } \ell \in 1,\ldots,L \text{ and } j \in 1,\ldots,k. \tag{8}$$

Algorithm 1 Private Sliced Chebyshev Moment Matching

Input: Dataset $\mathcal{X} = (x_1, \ldots, x_n) \in (\mathcal{B}^d)^n$; privacy parameters $\epsilon, \delta \in (0,1)$; integers $k, L \geq 1$.
Output: (ϵ, δ)-differentially private distribution q approximating $p = \mathrm{Unif}(x_1, \ldots, x_n)$.

1: **Sample directions.** Sample $\theta_1, \ldots, \theta_L \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(S^{d-1})$.
2: **Compute noisy moments.** For all $\ell \in 1, \ldots, L$ and $j \in 1, \ldots, k$, compute $\tilde{m}_{j,\ell} = \mathbb{E}_{x \sim p}[T_j(x^\top \theta_\ell)] + \eta_{j,\ell}$, where $\eta_{j,\ell} \sim \mathcal{N}(0, j\sigma^2)$ with $\sigma^2 = 8L(1 + \log k)\log(1.25/\delta)/(n^2\epsilon^2)$.
3: **Match.** Using the sampled directions and noisy moments, return a distribution

$$q \in \operatorname*{arg\,min}_{r \text{ a probability measure on } \mathcal{B}^d} \frac{1}{L} \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{(\tilde{m}_{j,\ell} - m_j(\theta_\ell, r))^2}{j^2}.$$

Note the scaling by $1/\sqrt{j}$: this ensures that the addition of $\mathcal{N}(0, j\sigma^2)$ noise in Algorithm 1 is equivalent to adding $\mathcal{N}(0, \sigma^2)$ noise to each coordinate of $f(\mathcal{X})$.

Proving privacy thus amounts to bounding the ℓ_2 sensitivity of $f(\mathcal{X})$. We do so under the standard replacement adjacency model, meaning that neighboring datasets differ in one record. Using that $|T_j(z)| \leq 1$ for all $z \in [-1, 1]$, we can bound the ℓ_2 sensitivity of this function as:

$$\Delta_{2,f}^2 = \max_{\text{neighboring datasets } X, X'} \|f(X) - f(X')\|_2^2 \leq \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{4}{jn^2} = \frac{4LH_k}{n^2} \leq \frac{4L}{n^2}(1 + \log k),$$

where $H_k = \sum_{j=1}^{k} 1/j$ denotes the k^{th} harmonic number. Dwork & Roth (2014, Theorem 3.22) show that (ϵ, δ)-differential privacy can be achieved by adding Gaussian noise with variance $\Delta_{2,f}^2 \cdot 2\log(1.25/\delta)/\epsilon^2$, which equals σ^2. We thus conclude:

Proposition 4 (Privacy)**.** *For $\epsilon, \delta \in (0,1)$, Algorithm 1 is (ϵ, δ)-differentially private.*

Next, we leverage the results of the previous section to show that Algorithm 1 also achieves a strong utility guarantee. I.e., that q is close to p in the sliced Wasserstein-1 distance. Formally we have:

Theorem 5. *Let q be the output of Algorithm 1. For any $\beta \in (0,1)$, with probability at least $1 - \beta$,*

$$\mathrm{SW}_1(p, q) \leq \frac{36}{k} + \frac{8(1 + \log k)\sqrt{L\log(1.25/\delta)}}{\epsilon n\sqrt{\beta}} + \frac{64}{\beta}\sqrt{\frac{d\log(6Lk^2)}{L}}. \tag{9}$$

Assuming $n\epsilon > 1$, $\delta \leq 1/2$, and β to be a fixed constant (e.g. $1/100$), the bound in Theorem 5 can be optimized by choosing:

$$k = \Theta\left(\max\left\{1, \frac{\sqrt{n\epsilon}}{d^{1/4}}\right\}\right) \qquad \text{and} \qquad L = \Theta\left(\max\left\{1, \frac{n\epsilon\sqrt{d}}{\sqrt{\log(d + n\epsilon)\log(1/\delta)}}\right\}\right)$$

Plugging in, we conclude the bound stated in Section 1: for any fixed constant failure probability,

$$\mathrm{SW}_1(p, q) \leq O\left(\frac{d^{1/4}\log^{3/4}(d + n\epsilon)\log^{1/4}(1/\delta)}{\sqrt{n\epsilon}}\right). \tag{10}$$

Proof. Let Φ be the least-squares objective minimized in Algorithm 1. Since p is feasible for the minimization, $\Phi(q) \leq \Phi(p)$. Jensen's inequality and the triangle inequality give

$$\begin{aligned}
\tilde{\Gamma}_{k,L}(p, q) &\leq \left(\frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{k}\frac{(m_j(\theta_\ell, p) - m_j(\theta_\ell, q))^2}{j^2}\right)^{1/2} \\
&\leq \left(\frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{k}\frac{(m_j(\theta_\ell, p) - \tilde{m}_{j,\ell})^2}{j^2}\right)^{1/2} + \left(\frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{k}\frac{(\tilde{m}_{j,\ell} - m_j(\theta_\ell, q))^2}{j^2}\right)^{1/2} \\
&= \sqrt{\Phi(p)} + \sqrt{\Phi(q)} \leq 2\sqrt{\Phi(p)}.
\end{aligned} \tag{11}$$

We can check that:

$$\mathbb{E}[\Phi(p)] = \mathbb{E}\left[\frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{k}\frac{1}{j^2}\left(m_j(\theta_\ell, p) - \tilde{m}_{j,\ell}\right)^2\right] \leq (1+\log k)\cdot\sigma^2. \tag{12}$$

Markov's inequality therefore gives, with probability at least $1-\beta/2$, $\Phi(p) \leq \frac{2(1+\log k)\sigma^2}{\beta}$. Plugging into Equation (11) to bound $\tilde{\Gamma}_{k,L}(p,q)$ and applying Theorem 2 with failure probability $\beta/2$ and taking a union bound, we obtain, with probability at least $1-\beta$,

$$\mathrm{SW}_1(p,q) \leq \frac{36}{k} + 2\sqrt{\frac{2(1+\log k)\sigma^2}{\beta}} + \frac{64}{\beta}\sqrt{\frac{d\log(6Lk^2)}{L}}$$

Plugging in $\sigma^2 = 8L(1+\log k)\log(1.25/\delta)/(n^2\epsilon^2)$ yields the theorem. □

4 A Tractable Representation

Our main result from the previous section, Theorem 5, says that it is information theoretically possible to return a private distribution q such that $\mathrm{SW}_1(p,q) \leq \tilde{O}(d^{1/4}/\sqrt{n\epsilon})$. Indeed, q can be obtained by postprocessing a set of sliced Chebyshev moments noised using the Gaussian mechanism. However, the theorem does not say anything about how to implement the postprocessing step efficiently, i.e. how to find q that minimizes or approximately minimizes the moment error in Algorithm 1. We address that gap in this section, providing an effective heuristic for fitting a distribution q.

Mixture of product distributions. Importantly, our approach requires a more efficient representation for q than, for example, a naive d dimensional histogram, which would require $2^{O(d)}$ parameters to even represent. We choose to instead optimize over a parameter-efficient mixture of product distributions. I.e., we consider distributions of the form $\hat{q} = \sum_{h=1}^{K}\pi_h\hat{q}_h$, where K is an integer parameter, π is a vector in the probability simplex, Δ_K, and each $\hat{q}_h$ is a product distribution.

Concretely, for every index a corresponding to a categorical variable with r_a levels, the distribution of $\hat{q}_h$ at index a is represented by a probability vector $\phi^{(h,a)} \in \Delta_{r_a}$. Each level is associated with a point on an equispaced grid of r_a values on $[-1,1]$. On the other hand, all continuous variables are modeled as a single joint Gaussian distribution with mean $\mu_h \in \mathbb{R}^{d_N}$ and covariance $\Sigma_h \in \mathbb{R}^{d_N\times d_N}$, where d_N (respectively, d_C) counts the continuous (respectively, categorical) variables. We assume that Σ_h is low-rank-plus-diagonal for some rank R, i.e. $\Sigma_h = U_hU_h^\top + \mathrm{diag}(D_h)$ with $U_h \in \mathbb{R}^{d_N\times R}$ and $D_h \in \mathbb{R}^{d_N}_{>0}$. The product of this single continuous distribution with the d_C categorical distributions defines $\hat{q}_h$. Overall, we have:

$$\hat{q}(x_C, x_N) = \sum_{h=1}^{K}\pi_h\left(\prod_{a\in C}\phi^{(h,a)}(x_a)\right)\cdot\mathcal{N}\left(x_N;\mu_h,\Sigma_h\right)\ . \tag{13}$$

Any distribution of this form requires $O(K(d_Cr_{max} + d_NR))$ parameters to represent, where r_{max} is an upper bound on the number of levels for any categorical variable. We note that, for each component distribution, $\hat{q}_h$, categorical and continous variables are independent, but by mixing K distributions, we can capture categorical-categorical and categorical-continuous dependence.

Optimization. With the model above in place, our goal becomes to fit a distribution that minimizes the moment error in Algorithm 1, i.e., to minimize the following expression over parameters of $\hat{q}$:

$$\frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{k}\frac{(\tilde{m}_{j,\ell} - m_j(\theta_\ell,\hat{q}))^2}{j^2}. \tag{14}$$

While we cannot ensure that doing so obtains near-minimal error over all possible distributions, as will be shown in the next section, empirically we find that our mixture model is rich enough to accurately match many practical datasets from noisy sliced moments, even with $K=5$ and $R=2$.

We minimize equation 14 using gradient descent. We enforce simplex constraints on π and each $\phi^{(h,a)}$ using a softmax reparameterization and we ensure the covariance diagonal is positive through a logarithmic reparameterization. To compute gradients, we take advantage of the fact that there is an efficient, differentiable procedure for computing $m_j(\theta_\ell,\hat{q})$, which we describe in Section C.

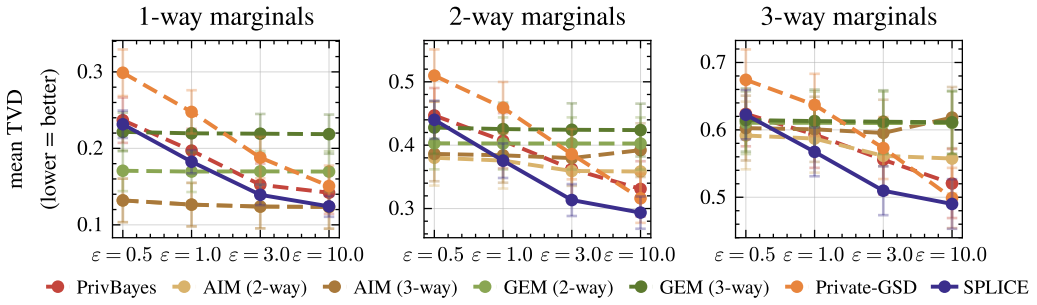


Figure 3: Mean k-way marginal total variation distance (lower is better) on a fixed set of $n = 43$ datasets where every method has a value at every budget. Left to right: 1-way, 2-way, 3-way marginals. SPLICE never targets marginals yet leads 2-way and 3-way fidelity from $\epsilon \geq 1$.

5 Experiments

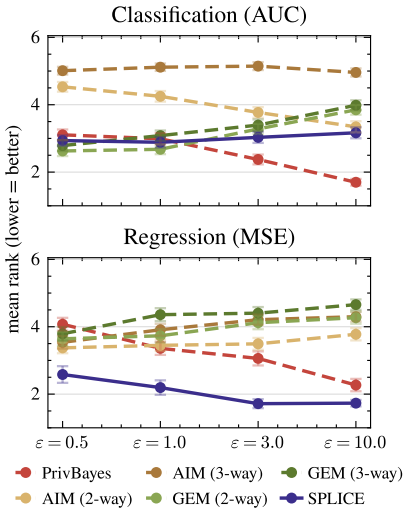


Figure 2: Mean TSTR rank across datasets (lower is better) by privacy budget, best downstream learner per dataset. Classification (AUC), $n = 94$ datasets, regression (MSE), $n = 67$.

Data and Baselines. We evaluate on 209 datasets from the TALENT benchmark (Liu et al., 2025), spanning binary classification (75), multiclass classification (47), and regression (87), at privacy budgets $\epsilon \in \{0.5, 1, 3, 10\}$ with $\delta = 10^{-9}$ and five seeds per cell. Appendix G lists every dataset with its size, feature composition, and task. We consider a number of state-of-the-art baselines: AIM (McKenna et al., 2022) (with 2-way and 3-way marginal workloads), PrivBayes (Zhang et al., 2017), GEM (Liu et al., 2021b) (with 2-way and 3-way marginal workloads), and Private-GSD (Liu et al., 2023), all under the public-schema convention described in Section E for fairness. Our downstream utility for classification/regression measurements use the train-on-synthetic, test-on-real (TSTR) protocol, scoring each generated dataset by the best of four learners (logistic regression or ridge, random forest, XGBoost, TabPFN) and aggregating by rank. Despite checking 209 TALENT datasets, we *only* report aggregate comparisons on what we call "well-formed prediction tasks" i.e. tasks where the train-on-real ceiling of AUC at least 0.7 for classification and R^2 at least 0.3 for regression. This means that "unlearnable" datasets do not contribute noisy wins; still, the vast majority of datasets are kept (184 of 209). Note that, unless otherwise specified, results for SPLICE use the mixture of categorical-Gaussian product distributions of Algorithm 2; a comparison among the SPLICE model variants is in Section E.1.

Overall comparison. Figure 1 summarizes the benchmark at $\epsilon = 1$ by two composite ranks: distributional fidelity (1-way, 2-way, and 3-way marginal TVD, kernel MMD^2, and sliced Wasserstein-1) and downstream TSTR utility, macro-averaged over the classification and regression families with equal weight. SPLICE is ranked first on *four of the five* fidelity metrics, third on 1-way TVD behind only AIM (which measures those exact marginals directly), and outperforms every baseline on both composite axes simultaneously. On utility, SPLICE dominates regression and is competitive on classification tasks. Per-metric values appear in Section F.

Downstream predictive utility. Figure 2 shows mean TSTR rank by privacy budget. On regression, SPLICE has the best mean rank at every budget (2.58, 2.19, 1.72, and 1.73 at $\epsilon = 0.5, 1, 3, 10$ over 67 datasets). On classification, SPLICE is in the top 3 performers at each privacy budget. Classification is a more difficult setting for SPLICE, as the target is discrete and the datasets have more categorical features on average; the strength of sliced Chebyshev moment projections is in natively handling continuous or ordinal values, while categorical values without order are not faith-

fully represented by the projection into continuous space. See Figure 6 in the Appendix for the data type stratum breakdown.

Marginal fidelity. Despite never explicitly measuring a marginal, SPLICE is **the best 3-way method at every budget, as well as the best 2-way method** from $\epsilon = 1$ (Figure 3); AIM, which matches noisy 1-way marginals directly, is best on that metric, but even there its margin shrinks as ϵ increases. SPLICE's advantage grows with the interaction order, which is the main benefit of the method; matching higher-order Chebyshev moment measurements across random projections leads to distributional recovery containing higher-order interactions.

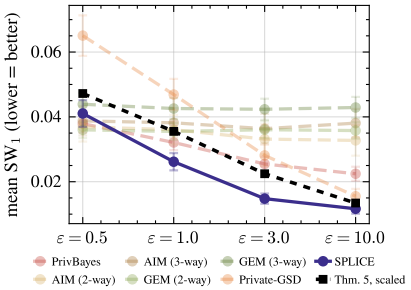


Figure 4: Sliced Wasserstein-1 distance (lower is better), 256 Monte Carlo directions on the unit ball, $n = 36$ datasets, with the bound of Theorem 5 parameterized at each dataset's (n, d), minimized over k and L, and divided through by a constant; the overlay compares rates in ϵ, and baselines are faded.

Distributional distance and causal structure. SPLICE is theoretically motivated by the sliced Wasserstein-1 metric (SW_1), which is *distributional* and sensitive to higher order relationships in data. Perhaps unsurprisingly, our method is then best at every budget for SW_1 (Figure 4), and the method using a mixture of categorical-Gaussian product distributions matches the theoretical rate almost exactly (the worst-case rate is scaled by a large *constant* factor, to match the practical SW_1 error scale, but the decreasing rate across ϵ appears faithful). Additionally, the commonly used Kernel MMD distributional distance (Gretton et al., 2012) gives the same ordering (Figure 7 in the appendix).

It has been noted that causal path analysis would be an impactful area for advances in DP synthetic data Rosenblatt et al. (2024a); Ponomareva et al. (2026). Thus, we run GRaSP, a standard causal discovery method Lam et al. (2022), on real and synthetic data and compare the recovered graphs (Figure 5). At tight and moderate budgets SPLICE roughly doubles edge recovery (measured F1) of the best marginal baseline (0.42 to 0.49 at $\epsilon \leq 3$ against 0.00 to 0.37).

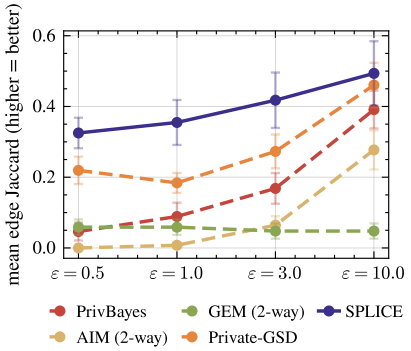


Figure 5: Jaccard similarity of undirected edge sets between the real-data and synthetic-data causal graphs (higher is better), $n = 11$ datasets. Error bars are ± 1 SEM across datasets.

Compute and reliability. Thanks to its near-independence from sample size and dimensionality by construction, SPLICE runs in almost constant time and memory across three orders of magnitude of dataset size (30 to 120 seconds over a controlled grid from 12,500 to 400,000 rows and 6 to 24 columns, while e.g. Private-GSD times out beyond 25,000 rows, AIM struggles in higher dimensions and PrivBayes scales computationally with n (Section H)). Additionally, across the full generations of the benchmark, SPLICE **completed all 16,800 of its generations with zero failures**, while every other baseline failed somewhere due to each method's structural constraints.

6 Conclusion

We introduced SPLICE, a differentially private synthetic data mechanism built on noisy sliced Chebyshev moments. Our analysis shows that matching moments along a number of random directions that is linear in the dimension controls the sliced Wasserstein-1 distance, and that the private version inherits a rate whose exponent in the sample budget is independent of the dimension. Our mixture of categorical-Gaussian product distributions makes the mechanism practical, reliable and competitive: it outperforms strong marginal-based baselines on both composite axes simultaneously, distributional fidelity and downstream utility, across a 209-dataset benchmark, and in particular leads on regression, higher-order marginal fidelity and causal edge recovery.

AI USE STATEMENT

In this work, we used generative AI tools (large language model assistants) to improve our code quality for the experiments. An initial implementation of the SPLICE mechanism and benchmark harness (with TALENT data) was written from scratch. It was originally built off of publicly available code from Musco et al. (2025), and then adapted to use torch and gradient descent with e.g. the differentiable form of the Chebyshev moments to enable gradient-based optimization. Eventually, generative AI tools were used to improve this approach (through EMA and Huber techniques), to expand the benchmark harness and experimental infrastructure, and finally to run and monitor experimental sweeps. It was also used to quickly improve visual elements of plotting code from the manual benchmark. Generative AI tools were further used to edit and debug LaTeX, and to assist in generating the exhaustive TALENT table in the appendix.

The authors did *not* use generative AI tools for research ideation in any way. The problem formulation, the algorithm design, and the theoretical analysis are the authors' own. The authors reviewed and tested all AI-assisted code and checked every empirical claim manually. We take responsibility for the final content of this work, including any artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a privacy-enhancing technology: differentially private synthetic data aims to enable data analysis while limiting the disclosure of information about individuals. All experiments use publicly available benchmark datasets, and we collected no new data involving human subjects. We caution that the privacy guarantee of `SPLICE` holds only at the stated (ϵ, δ) parameters and under the public metadata convention described in Section E, and that synthetic data does not by itself remove the need for careful deployment decisions. We foresee no other ethical concerns.

REPRODUCIBILITY STATEMENT

Section D states the full practical algorithm (Algorithm 2) together with every default hyperparameter (Table 1), and Section E documents the experimental protocol, the public metadata convention, and the baseline implementations with their sources. Section G lists all benchmark datasets, and Section F reports the per-metric values behind the figures. We include an anonymized implementation of `SPLICE`, with tests and a script that regenerates the main figures, as supplementary material.

REFERENCES

John Abowd, Robert Ashmead, Simson Garfinkel, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, and William Sexton. Census TopDown: Differentially private data, incremental schemas, and consistency with public knowledge. Draft working paper, U.S. Census Bureau, July 2019. URL `https://systems.cs.columbia.edu/private-systems-class/papers/Abowd2019Census.pdf`.

John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. The 2020 census disclosure avoidance system TopDown algorithm. *Harvard Data Science Review*, 2022. Special Issue 2.

Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A. Siva. Differentially private query release through adaptive projection. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 18–24 Jul 2021.

Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the 26th Symposium on Principles of Database Systems (PODS)*, 2007.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12:1–12:25, 2013.

March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Private measures, random walks, and synthetic data. *Probability theory and related fields*, 189(1–2):569–611, 2024.

Clément Bonet, Paul Berg, Nicolas Courty, François Septier, Lucas Drumetz, and Minh-Tan Pham. Spherical sliced-Wasserstein. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

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.

Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. Data synthesis via differentially private Markov random fields. *Proceedings of the VLDB Endowment*, 14(11), 2021.

Guillaume Carlier, Alessio Figalli, Quentin Mérigot, and Yi Wang. Sharp comparisons between sliced and standard 1-Wasserstein distances. arXiv:2510.16465, 2025.

Antoine Chatalic, Vincent Schellekens, Florimond Houssiau, Yves-Alexandre de Montjoye, Laurent Jacques, and Rémi Gribonval. Compressive learning with privacy guarantees. *Information and Inference: A Journal of the IMA*, 11(1):251–305, 2022.

Rachel Cummings, Damien Desfontaines, David Evans, Roxana Geambasu, Yangsibo Huang, Matthew Jagielski, Peter Kairouz, Gautam Kamath, Sewoong Oh, Olga Ohrimenko, et al. Advancing differential privacy: Where we are now and future directions for real-world deployment. *Harvard Data Science Review*, 6(1), 2024.

Yuval Dagan and Gil Kur. A bounded-noise mechanism for differential privacy. In *Proceedings of the 35th Annual Conference on Computational Learning Theory (COLT)*, 2022.

Ishan Deshpande, Ziyu Zhang, and Alexander G. Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G. Schwing. Max-sliced Wasserstein distance and its use for GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Rundong Ding, Yiyun He, and Yizhe Zhu. Minimax optimal differentially private synthetic data for smooth queries. In *Proceedings of the 39th Annual Conference on Computational Learning Theory (COLT)*, 2026. Full version: arXiv:2602.01607.

Konstantin Donhauser, Johan Lokna, Amartya Sanyal, March Boedihardjo, Robert Hönig, and Fanny Yang. Certified private data release for sparse Lipschitz functions. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

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

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)*, 2009.

Alessandro Epasto, Jieming Mao, Andres Muñoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. In *Proceedings of the 14th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2023.

Vitaly Feldman, Audra McMillan, Satchit Sivakumar, and Kunal Talwar. Instance-optimal private density estimation in the Wasserstein distance. *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Miguel Fuentes, Brett C. Mullins, Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Joint selection: Adaptively incorporating public information for private synthetic data. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On avoiding the union bound when answering multiple differentially private queries. In *Proceedings of the 34th Annual Conference on Computational Learning Theory (COLT)*, 2021.

Badih Ghazi, Junfeng He, Kai Kohlhoff, Ravi Kumar, Pasin Manurangsi, Vidhya Navalpakkam, and Nachiappan Valliappan. Differentially private heatmaps. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 37(6), Jun. 2023.

Badih Ghazi, Cristóbal Guzmán, Pritish Kamath, Alexander Knop, Ravi Kumar, Pasin Manurangsi, and Sushant Sachdeva. PREM: Privately answering statistical queries with relative error. In *Proceedings of the 38th Annual Conference on Computational Learning Theory (COLT)*, 2025. Full version: arXiv:2502.14809.

Kristjan Greenewald, Yuancheng Yu, Hao Wang, and Kai Xu. Privacy without noisy gradients: Slicing mechanism for generative model training. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

Anming Gu, Edward Chien, and Kristjan Greenewald. Private continuous-time synthetic trajectory generation via mean-field Langevin dynamics. arXiv:2506.12203, 2025.

Frederik Harder, Kamil Adamczewski, and Mijung Park. DP-MERF: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010.

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.

Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3(1–2), 2010.

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)*, 2023.

Yiyun He, Roman Vershynin, and Yizhe Zhu. Online differentially private synthetic data generation. *IEEE Transactions on Privacy*, 1:19–30, 2024.

Yiyun He, Thomas Strohmer, Roman Vershynin, and Yizhe Zhu. Differentially private low-dimensional synthetic data from high-dimensional datasets. *Information and Inference: A Journal of the IMA*, 14(1):iaae034, 2025.

Peter J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Dunham Jackson. On approximation by trigonometric sums and polynomials. *Transactions of the American Mathematical Society*, 13(4):491–515, 1912.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

Soheil Kolouri, Gustavo K. Rohde, and Heiko Hoffmann. Sliced Wasserstein distance for learning Gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced Wasserstein distances. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019a.

Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced Wasserstein auto-encoders. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019b.

Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45 (5):2218–2247, 2017.

Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Proceedings of the 38th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

Jaewoo Lee, Yue Wang, and Daniel Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB journal*, 24: 757–781, 2015.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. Talent: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26 (226):1–16, 2025.

Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021a.

Terrance Liu, Giuseppe Vietri, and Steven Z. Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021b.

Terrance Liu, Jingwu Tang, Giuseppe Vietri, and Steven Z. Wu. Generating private synthetic data with genetic algorithms. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment*, 11(10), 2018.

Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3), 2021a.

Ryan McKenna, Siddhant Pradhan, Daniel R. Sheldon, and Gerome Miklau. Relaxed marginal consistency for differentially private query answering. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021b.

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), 2022.

Dimitri Meunier, Massimiliano Pontil, and Carlo Ciliberto. Distribution regression with sliced Wasserstein kernels. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Brett Mullins, Miguel Fuentes, Yingtai Xiao, Daniel Kifer, Cameron Musco, and Daniel R. Sheldon. Efficient and private marginal reconstruction with local non-negativity. *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Cameron Musco, Christopher Musco, Lucas Rosenblatt, and Apoorv Vikram Singh. Sharper bounds for Chebyshev moment matching, with applications. In *Proceedings of the 38th Annual Conference on Computational Learning Theory (COLT)*, 2025.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Kimia Nadjahi, Alain Durmus, Lénaïc Chizat, Soheil Kolouri, Shahin Shahrampour, and Umut Şimşekli. Statistical and topological properties of sliced probability divergences. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

Jonathan Niles-Weed and Jacob Shkrob. Nearly sharp comparison results for sliced and max-sliced Wasserstein distances. arXiv:2608.13374, 2026.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Natalia Ponomareva, Zheng Xu, H. Brendan McMahan, Peter Kairouz, Lucas Rosenblatt, Vincent Cohen-Addad, Cristóbal Guzmán, Ryan McKenna, Galen Andrew, Alex Bie, Da Yu, Alex Kurakin, Morteza Zadimoghaddam, Sergei Vassilvitskii, and Andreas Terzis. How to DP-fy your data: A practical guide to generating synthetic data with differential privacy. *Journal of Artificial Intelligence Research*, 86:17:1–17:102, 2026.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. Springer, 2011.

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.

Lucas Rosenblatt, Bernease Herman, Anastasia Holovenko, Wonkwon Lee, Joshua Loftus, Elizabeth McKinnie, Taras Rumezhak, Andrii Stadnik, Bill Howe, and Julia Stoyanovich. Epistemic parity: Reproducibility as an evaluation metric for differential privacy. *ACM SIGMOD Record*, 53(1): 65–74, 2024a.

Lucas Rosenblatt, Bill Howe, and Julia Stoyanovich. Are data experts buying into differentially private synthetic data? gathering community perspectives. arXiv:2412.13030, 2024b.

Raif M. Rustamov and Subhabrata Majumdar. Intrinsic sliced Wasserstein distances for comparing collections of probability distributions on manifolds and graphs. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Ilana Sebag, Muni Sreenivas Pydi, Jean-Yves Franceschi, Alain Rakotomamonjy, Mike Gartrell, Jamal Atif, and Alexandre Allauzen. Differentially private gradient flow based on the sliced Wasserstein distance. *Transactions on Machine Learning Research*, 2025.

Thomas Steinke and Jonathan R. Ullman. Between pure and approximate differential privacy. *J. Priv. Confidentiality*, 7(2), 2017.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.

Jonathan R. Ullman and Salil P. Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography Conference (TCC)*, 2011.

Salil P. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pp. 347–450. Springer International Publishing, 2017.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, 1996.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008. ISBN 9783540710509.

Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. Hermite polynomial features for private data generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

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.

Andreas Winkelbauer. Moments and absolute moments of the normal distribution. arXiv:1209.4340, 2012.

Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. PrivBayes: Private data release via Bayesian networks. *ACM Trans. Database Syst.*, 42(4), October 2017.

Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. PrivSyn: Differentially private data synthesis. In *30th USENIX Security Symposium*, 2021.

A Extended Related Work

Here we expand the discussions of Section 2 in the context of the surrounding literature.

Practical DP synthetic data. Most practical mechanisms for private tabular synthesis follow the Select-Measure-Estimate paradigm (Ponomareva et al., 2026): choose a workload of low-dimensional statistics, measure them with calibrated noise, and fit a distribution to the noisy answers. Model lineage is extensive, from early approaches like MWEM (Hardt & Rothblum, 2010; Hardt et al., 2012) to more recent graphical-model estimators (McKenna et al., 2019; 2021a; 2022), Bayesian networks (Zhang et al., 2017), relaxed projection and generative estimators (Aydore et al., 2021; Liu et al., 2021b; McKenna et al., 2021b), marginal-selection refinements (Zhang et al., 2021; Cai et al., 2021; Liu et al., 2021a; Fuentes et al., 2024; Mullins et al., 2024), and genetic search over discretized domains (Liu et al., 2023). These build on a long line of work on accurate query answering under DP, including consistency and matrix mechanism techniques (Barak et al., 2007; Hay et al., 2010; Li et al., 2015; McKenna et al., 2018; Lee et al., 2015; Nikolov et al., 2013), and they have led to real deployments (Abowd et al., 2019; 2022; Ghazi et al., 2023). Because these mechanisms measure discrete marginals, they must discretize numerical features, a step the survey of Ponomareva et al. (2026) identifies as a central difficulty for tabular data; `SPLICE` sidesteps it by measuring moments of continuous projections directly. A parallel line of work trains neural generative models under DP, from GANs (Jordon et al., 2019) to recent end-to-end approaches built on foundation models (Ponomareva et al., 2026), although marginal-based methods still dominate tabular benchmarks (Rosenblatt et al., 2024b; 2020).

Closest to our release-once-then-fit structure within this line are private feature-sketch generators, which release a fixed noisy feature sketch and fit a generator to it: noisy sliced projections of records trained against smoothed-sliced f-divergences (Greenewald et al., 2024), random Fourier features for kernel MMD (Harder et al., 2021), Hermite polynomial features approximating a Gaussian kernel (Vinaroz et al., 2022), and compressive learning sketches (Chatalic et al., 2022). `SPLICE` shares the release-once-then-fit structure, but differs in two ways: its sliced Chebyshev sketch provably controls the distributional metric SW_1, whereas these methods match kernel or feature discrepancies without a guarantee on a standard distributional metric; and these feature-matching generators have not demonstrated utility competitive with the marginal-based state of the art on realistic tabular workloads, which is the comparison our experiments target.

Hardness of private synthetic data. Answering all low-order statistics accurately is possible information-theoretically (Blum et al., 2013), but generating synthetic data that does so is computationally hard in the worst case (Dwork et al., 2009; Ullman & Vadhan, 2011; Vadhan, 2017), with sharp sample complexity transitions between pure and approximate DP (Steinke & Ullman, 2017). This explains the field's split between practical heuristics and guarantee-first mechanisms, which can ignore these real-world constraints.

Distributional guarantees. A theoretical stream targets explicit statistical distance guarantees between the synthetic and true distributions. For Wasserstein distance, private measures and random walks achieve the optimal rate, which unfortunately degrades as $n^{-1/d}$ in dimension d (Boedihardjo et al., 2024; He et al., 2023), motivating work on low-dimensional structure (He et al., 2025; Donhauser et al., 2024), online generation (He et al., 2024), instance optimality (Feldman et al., 2024), and streaming moment estimation (Epasto et al., 2023). Weakening the metric helps: targeting smooth query classes improves the rates, although the sample complexity still depends exponentially on dimension (Wang et al., 2016; Ding et al., 2026). Relative error guarantees also refine worst-case bounds (Ghazi et al., 2025; 2021; Dagan & Kur, 2022). Our approach descends from moment matching, where matching k (Chebyshev) moments controls one-dimensional Wasserstein distance (Kong & Valiant, 2017; Braverman et al., 2022; Musco et al., 2025); in this paper, we escape the dimension-exponential cost of the multivariate extension by matching moments only along random slices.

Sliced Wasserstein distance. Sliced Wasserstein distance was introduced as a tractable surrogate for Wasserstein distance (Rabin et al., 2011; Bonneel et al., 2015) and is now a standard training objective and evaluation metric for generative models (Deshpande et al., 2018; 2019; Kolouri et al., 2018; 2019b;a; Liutkus et al., 2019). Its statistical and geometric properties are an active topic,

including sharp comparisons to the unsliced distance (Carlier et al., 2025; Niles-Weed & Shkrob, 2026), kernel and intrinsic variants (Meunier et al., 2022; Rustamov & Majumdar, 2023; Bonet et al., 2023), and its role as an integral probability metric (Müller, 1997; Villani, 2008). In private learning, sliced Wasserstein distance drives gradient flow generators (Sebag et al., 2025) and trajectory synthesis (Gu et al., 2025). To our knowledge, SPLICE is the first to pair a sliced Wasserstein guarantee with a practical, state-of-the-art differentially private algorithm.

B Proof of the Moment Error Approximation Lemma

Here we provided the deferred proof of Theorem 3, which we restate below:

Lemma 3. *Fix any positive integer k and any probability measure p on $\mathcal{B}^d$. Let L be a positive integer, let $\beta \in (0,1)$, and let $\theta_1, \ldots, \theta_L$ be independent draws from $\mathrm{Unif}(S^{d-1})$. With probability at least $1-\beta$ over the choice of $\theta_1, \ldots, \theta_L$, simultaneously for all probability measures q on $\mathcal{B}^d$,*

$$\left| \frac{1}{L} \sum_{\ell=1}^{L} \Gamma_k(\theta_\ell, p, q) - \mathbb{E}_{\theta \sim \mathrm{Unif}(S^{d-1})} \left[\Gamma_k(\theta, p, q) \right] \right| \leq \frac{32}{\beta} \sqrt{\frac{d \log(6Lk^2)}{L}}. \tag{7}$$

Proof of Theorem 3. Define

$$Z := \sup_q \left| \frac{1}{L} \sum_{\ell=1}^{L} \Gamma_k(\theta_\ell, p, q) - \mathbb{E}_{\theta \sim \mathrm{Unif}(S^{d-1})} \left[\Gamma_k(\theta, p, q) \right] \right|,$$

where the supremum is over all probability measures on $\mathcal{B}^d$. We are going to bound $\mathbb{E}_{\theta_1,\ldots,\theta_L}[Z]$ and then use Markov's inequality to obtain a probabilistic upper bound.

Symmetrization and Gaussian Comparison. Let $\sigma_1, \ldots, \sigma_L$ be independent Rademacher random variables, independent of the directions. A standard symmetrization inequality (see e.g. (van der Vaart & Wellner, 1996, Lemma 2.3.1)) gives

$$\mathbb{E}_{\theta_1,\ldots,\theta_L}[Z] \leq \frac{2}{L} \mathbb{E}_{\theta_1,\ldots,\theta_L,\sigma_1,\ldots,\sigma_L} \left[\sup_q \left| \sum_{\ell=1}^{L} \sigma_\ell \Gamma_k(\theta_\ell, p, q) \right| \right]. \tag{15}$$

We will bound the above by showing that the expectation over $\sigma_1, \ldots, \sigma_L$ is bounded for any fixed directions $\theta_1, \ldots, \theta_L$. To do so, we introduce the notation

$$v_\ell(q) := \left(\frac{m_1(\theta_\ell, p) - m_1(\theta_\ell, q)}{1}, \ldots, \frac{m_k(\theta_\ell, p) - m_k(\theta_\ell, q)}{k} \right) \in \mathbb{R}^k, \tag{16}$$

so $\Gamma_k(\theta_\ell, p, q) = \|v_\ell(q)\|_2$.

We first note that we can simplify the supremum in equation 15 by removing the absolute value:

$$\mathbb{E}_{\sigma_1,\ldots,\sigma_L} \left[\sup_q \left| \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 \right| \right] \leq 2\mathbb{E}_{\sigma_1,\ldots,\sigma_L} \left[\sup_q \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 \right]. \tag{17}$$

This follows from noting that, for any realization of the variables $\sigma_1, \ldots, \sigma_L$,

$$\sup_q \left| \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 \right| \leq \sup_q \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 + \sup_q \sum_{\ell=1}^{L} (-\sigma_\ell) \|v_\ell(q)\|_2.$$

Both suprema on the right are nonnegative, since we can always choose $q = p$, in which case $v_\ell(q) = 0$. Taking expectations and using the symmetry of the random signs gives equation 17.

So, we turn our attention to bounding $\mathbb{E}_{\sigma_1,\ldots,\sigma_L} \left[\sup_q \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 \right]$. We will do so by using a Gaussian comparison lemma to remove the ℓ_2 norms. In particular, we claim that:

$$\mathbb{E}_\sigma \sup_q \sum_{\ell=1}^{L} \sigma_\ell \|v_\ell(q)\|_2 \leq \sqrt{\pi/2} \mathbb{E}_g \sup_q \sum_{\ell=1}^{L} \langle g_\ell, v_\ell(q) \rangle, \tag{18}$$

We prove this claim inductively using the following lemma:

Lemma 6. *Fix $\ell \in \{1, \dots, L\}$ and let $v_\ell(q)$ be as in equation 16. Let $a(q)$ be a function over probability measures with $\sup_q |a(q)| < \infty$. Let σ be a Rademacher random variable and let $g \sim \mathcal{N}(0, I_k)$. Then*

$$\mathbb{E}_\sigma \sup_q \{a(q) + \sigma \|v_\ell(q)\|_2\} \le \mathbb{E}_g \sup_q \{a(q) + \sqrt{\pi/2}\langle g, v_\ell(q)\rangle\},$$

where both suprema range over all probability measures q on $\mathcal{B}^d$.

Proof. Fix any two probability measures q_+ and q_- on $\mathcal{B}^d$. We restrict the Gaussian supremum to these two measures and use that, for two quantities a and b, $\max\{a, b\} = \frac{a+b}{2} + \frac{|a-b|}{2}$ to obtain

$$\begin{aligned}
&\mathbb{E}_g \sup_q \{a(q) + \sqrt{\pi/2}\langle g, v_\ell(q)\rangle\} \\
&\ge \mathbb{E}_g \max_{q \in \{q_+, q_-\}} \{a(q) + \sqrt{\pi/2}\langle g, v_\ell(q)\rangle\} \\
&= \frac{a(q_+) + a(q_-)}{2} + \frac{1}{2}\mathbb{E}_g \left| a(q_+) - a(q_-) + \sqrt{\pi/2}\langle g, v_\ell(q_+) - v_\ell(q_-)\rangle \right| \\
&\ge \frac{a(q_+) + a(q_-)}{2} + \frac{1}{2}\|v_\ell(q_+) - v_\ell(q_-)\|_2 \\
&\ge \frac{a(q_+) + a(q_-)}{2} + \frac{\|v_\ell(q_+)\|_2 - \|v_\ell(q_-)\|_2}{2}.
\end{aligned}$$

For the second inequality we use that, if X is symmetric about zero, then $\mathbb{E}|b + X| \ge \mathbb{E}|X|$. Also, $\mathbb{E}|\langle g, u\rangle| = \sqrt{2/\pi}\|u\|_2$ for every u. The last inequality follows from triangle inequality. Taking the supremum over q_+ and q_- separately on the right-hand side yields

$$\frac{1}{2}\sup_q \{a(q) + \|v_\ell(q)\|_2\} + \frac{1}{2}\sup_q \{a(q) - \|v_\ell(q)\|_2\} = \mathbb{E}_\sigma \sup_q \{a(q) + \sigma\|v_\ell(q)\|_2\},$$

which proves the claim. □

equation 18 is obtained by applying Theorem 6 inductively to each σ_ℓ, conditioning on all other random variables and treating their contributions as the offsets $a(q)$.

Concentration and Net Argument. Combining equation 17 and equation 18 we have simplified our task, obtaining:

$$\mathbb{E}_{\sigma_1,\dots,\sigma_L}\left[\sup_q \left|\sum_{\ell=1}^{L} \sigma_\ell \Gamma_k(\theta_\ell, p, q)\right|\right] \le \sqrt{2\pi}\mathbb{E}_{g_{11},\dots,g_{Lk}}\left[\sup_q \sum_{\ell=1}^{L}\sum_{j=1}^{k} g_{\ell j}\frac{m_j(\theta_\ell, p) - m_j(\theta_\ell, q)}{j}\right], \tag{19}$$

where the $g_{\ell j}$ are independent standard normal random variables.

For fixed directions and fixed $g_{\ell j}$, define

$$h(x) := \sum_{\ell=1}^{L}\sum_{j=1}^{k} \frac{g_{\ell j}}{j} T_j(\theta_\ell^\top x), \qquad x \in \mathcal{B}^d. \tag{20}$$

By linearity of expectation, the expression inside the supremum on the right-hand side of equation 19 is

$$\mathbb{E}_{x\sim p}[h(x)] - \mathbb{E}_{x\sim q}[h(x)].$$

Since q ranges over all probability measures on $\mathcal{B}^d$, we have:

$$\sup_q \{\mathbb{E}_{x\sim p}[h(x)] - \mathbb{E}_{x\sim q}[h(x)]\} = \mathbb{E}_{x\sim p}[h(x)] - \inf_{x\in\mathcal{B}^d} h(x) \le \sup_{x\in\mathcal{B}^d} h(x) - \inf_{x\in\mathcal{B}^d} h(x). \tag{21}$$

I.e., to bound equation 19, we just need to show that the range of $h(x)$ is bounded in expectation over all vectors x in the d dimensional unit ball. A union bound over a net on this ball will lead

to a bound that is linear in d instead of exponential in d, as would have been obtained from union bounding over distributions directly.

Concretely, set $\epsilon = 1/(Lk^2)$. By the standard volumetric covering bound for the Euclidean ball (see, e.g., Vershynin (2018)), there is an ϵ-net $\mathcal{N}_\epsilon \subseteq \mathcal{B}^d$ that satisfies

$$|\mathcal{N}_\epsilon| \leq (6Lk^2)^d \qquad \text{and} \qquad \forall x \in \mathcal{B}^d, \quad \exists x' \in \mathcal{N}_\epsilon \text{ such that } \|x - x'\|_2 \leq \epsilon. \tag{22}$$

For every fixed $x' \in \mathcal{N}_\epsilon$, the random variable $h(x')$ is a centered Gaussian conditional on $\theta_1, \ldots, \theta_L$. Since $|T_j(t)| \leq 1$ for $t \in [-1, 1]$, its variance satisfies

$$\mathrm{Var}_g(h(x')) = \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{T_j(\theta_\ell^\top x')^2}{j^2} \leq L \sum_{j=1}^{k} \frac{1}{j^2} \leq \frac{\pi^2 L}{6}.$$

A standard bound on the expected maximum of a collection of Gaussian variables (Vershynin, 2018) gives

$$\mathbb{E}_{g_{11},\ldots,g_{Lk}} \left[\max_{x' \in \mathcal{N}_\epsilon} h(x') \right] \leq \frac{\pi}{\sqrt{3}} \sqrt{Ld \log(6Lk^2)}. \tag{23}$$

It remains to extend this bound to the entire ball. Let $x \in \mathcal{B}^d$ be some point in the ball and let $x' \in \mathcal{N}_\epsilon$ be the nearest point in our ϵ-net. For $t \in [-1, 1]$, we have $|T_j'(t)| \leq j^2$ (Trefethen, 2013). It follows that:

$$\begin{aligned} |h(x) - h(x')| = \left| \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{g_{\ell j}}{j} \left(T_j(\theta_\ell^\top x) - T_j(\theta_\ell^\top x') \right) \right| &\leq \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{|g_{\ell j}|}{j} \left| T_j(\theta_\ell^\top x) - T_j(\theta_\ell^\top x') \right| \\ &\leq \sum_{\ell=1}^{L} \sum_{j=1}^{k} \frac{|g_{\ell j}|}{j} j^2 |\theta_\ell^\top (x - x')| \\ &\leq \epsilon \sum_{\ell=1}^{L} \sum_{j=1}^{k} j |g_{\ell j}|. \end{aligned}$$

In the last step, we used Cauchy-Schwarz and the fact that $\|\theta_\ell\|_2 = 1$ and $\|x - x'\|_2 \leq \epsilon$. It follows that:

$$\begin{aligned} \mathbb{E}_g \sup_{x \in \mathcal{B}^d} h(x) \leq \mathbb{E}_g \max_{x' \in \mathcal{N}_\epsilon} h(x') + \epsilon \mathbb{E}_g \left[\sum_{\ell=1}^{L} \sum_{j=1}^{k} j |g_{\ell j}| \right] &\leq \mathbb{E}_g \max_{x' \in \mathcal{N}_\epsilon} h(x') + \epsilon \sqrt{\frac{2}{\pi}} L \frac{k(k+1)}{2} \\ &\leq \mathbb{E}_g \max_{x' \in \mathcal{N}_\epsilon} h(x') + \epsilon L k^2. \end{aligned}$$

Recalling that $\epsilon = 1/(Lk^2)$, combining with equation 23 yields

$$\mathbb{E}_{g_{11},\ldots,g_{Lk}} \left[\sup_{x \in \mathcal{B}^d} h(x) \right] \leq \frac{\pi}{\sqrt{3}} \sqrt{Ld \log(6Lk^2)} + 1. \tag{24}$$

The distribution of the $g_{\ell j}$ variables is unchanged when all signs are negated, so $\mathbb{E}_{g_{11},\ldots,g_{Lk}} [\sup_{x \in \mathcal{B}^d} h(x)] = -\mathbb{E}_{g_{11},\ldots,g_{Lk}} [\inf_{x \in \mathcal{B}^d} h(x)]$. It follows that the right hand side of equation 21 can be bounded by twice the right-hand side of equation 24. Combining equation 15, equation 19, equation 21, and equation 24, we obtain

$$\mathbb{E}_{\theta_1,\ldots,\theta_L}[Z] \leq 8\pi \sqrt{\frac{\pi}{6}} \sqrt{\frac{d \log(6Lk^2)}{L}} + \frac{8\sqrt{\pi/2}}{L} \leq 32 \sqrt{\frac{d \log(6Lk^2)}{L}}. \tag{25}$$

The last inequality uses $d, k, L \geq 1$, so that $1/L \leq \sqrt{d \log(6Lk^2)/L}$.

We conclude the lemma by applying Markov's inequality to the positive random variable Z. $\square$

C Chebyshev Moments for a Mixture of Categorical-Gaussian Product Distributions

In this section we describe how to efficiently compute the sliced Chebyshev moment, $m_j(\theta, \hat{q}) = \mathbb{E}_{x\sim\hat{q}}[T_j(\theta^T x)]$, for any direction $\theta \in \mathcal{S}^{d-1}$ and any distribution $\hat{q}$ that fits our mixture of categorical-Gaussian product distributions from Section 4.

The computation is first simplified by passing to the standard moments. In particular, we can write the Chebyshev polynomial in the monomial basis as $T_j(y) = \sum_{t=0}^{j} c_{j,t} y^t$ and then use that:

$$\mathbb{E}_{x\sim\hat{q}}[T_j(\theta^T x)] = \sum_{t=0}^{j} c_{j,t} \mathbb{E}_{x\sim\hat{q}}[(\theta^T x)^t].$$

We can further simplify the computation by splitting along the mixture components:

$$\mathbb{E}_{x\sim\hat{q}}[T_j(\theta^T x)] = \sum_{h=1}^{K} \pi_h \sum_{t=0}^{j} c_{j,t} \mathbb{E}_{x\sim\hat{q}_h}[(\theta^T x)^t]. \tag{26}$$

As in Section 4, let $C \subset \{1, \ldots, d\}$ contain the indices of all categorical variables in our data vectors. Let $N = \{1, \ldots, d\} \setminus C$ contain the indices of all continuous variables. Let θ_N and x_N denote the vectors θ and x restricted to the continuous indices. We have that:

$$\theta^T x = \sum_{a\in C} \theta_a x_a + \theta_N^\top x_N.$$

We can thus compute $\mathbb{E}_{x\sim\hat{q}_h}[(\theta^T x)^t]$ by iteratively applying the following standard expression, which holds for any independent random variables A and B:

$$\mathbb{E}\big[(A+B)^t\big] = \sum_{s=0}^{t} \binom{t}{s} \mathbb{E}[A^s]\mathbb{E}[B^{t-s}], \tag{27}$$

In this case, the variables $\{x_a\}_{a\in C}$ and x_N are jointly independent. We have thus reduced our task to evaluating:

$$\mathbb{E}_{x\sim\hat{q}_h}\big[(\theta_a x_a)^t\big] \quad (a \in C) \qquad \text{and} \qquad \mathbb{E}_{x\sim\hat{q}_h}\big[(\theta_N^\top x_N)^t\big] \tag{28}$$

for all t less than the maximum moment degree considered.

Since x_a is a finite categorical variable, the first expectation reduces to a simple sum over the possible values of x_a. For $\mathbb{E}_{x\sim\hat{q}_h}\big[(\theta_N^\top x_N)^t\big]$, we use that, when $x \sim \hat{q}_h$, $\theta_N^\top x_N$ is a univariate Gaussian random variable with mean and variance:

$$\nu_h = \theta_N^\top \mu_h \qquad \text{and} \qquad \sigma_h^2 = \big\|U_h^\top \theta_N\big\|_2^2 + \sum_{i\in N} D_{h,i}\theta_i^2.$$

We can then just apply standard closed form expressions for the raw moments of a univariate Gaussian (see Winkelbauer (2012)).

We implemented the procedure described above in Torch, then used Autograd to compute gradients when optimizing over the parameters of $\hat{q}$.

D THE SPLICE ALGORITHM WITH A MIXTURE OF CATEGORICAL-GAUSSIAN PRODUCT DISTRIBUTIONS

Algorithm 2 restates the mechanism of Section 4 as we actually run it. The private release is exactly that of Algorithm 1; the "Match" step is carried out by fitting the mixture of categorical-Gaussian product distributions Equation 13 using the exact, differentiable moment computation of Section C. Table 1 lists the implementation defaults used in every experiment.

Algorithm 2 SPLICE: Sliced Moment Matching with a Mixture of Categorical-Gaussian Product Distributions

Input: Dataset $x_1, \ldots, x_n$; privacy parameters $\epsilon, \delta > 0$; components K; covariance rank R; column split into blocks C, N.

Output: Synthetic distribution $\hat{q}$ of the form Equation 13.

1: **Encode.** Normalize each continuous column to $[-1, 1]$ and map each categorical column's levels to equally spaced grid values in $[-1, 1]$, giving a design matrix $X \in [-1, 1]^{n \times d}$.
2: **Hyperparameters.** Set the maximum degree k and number of directions L (Table 1).
3: **Release.** Sample public directions $\theta_1, \ldots, \theta_L \sim \mathrm{Unif}(S^{d-1})$, compute the scaled sliced Chebyshev moments of the projections $\theta_\ell^\top x_i/\sqrt{d}$, and add Gaussian noise calibrated to the ℓ_2 sensitivity of the scaled moment vector via the analytic Gaussian mechanism.
4: **Fit.** Minimize the weighted squared moment error between the noisy targets and the exact model moments by gradient descent, with the softmax and logarithmic parameterizations, warm start, restarts, averaging, and loss from main body.
5: **Sample.** Draw components $h \sim \pi$, categorical levels from $\phi^{(h,a)}$, continuous blocks from $\mathcal{N}(\mu_h, \Sigma_h)$, and de-normalize each column to its original range.

symbol	default	role
K	5	mixture components
R	2	rank of each continuous covariance factor
L	$\min(theory, 300)$	number of directions
k	$\min(theory, 30)$	Chebyshev degree per direction
optimizer	Adam, cosine schedule	2000 iterations, learning rate 0.02
robustness	Huber loss ($\delta = 0.5$), EMA, restarts	stabilizes the non-convex fit

Table 1: Implementation defaults for Algorithm 2, as used in all experiments.

More implementation details Algorithm 2 inherits (ϵ, δ)-privacy by post-processing, with the scaled-moment sensitivity $\Delta_2 = \sqrt{4LH_k/n^2}$ of Section 3 (we use the equivalent normalized basis $\bar{T}_j = \sqrt{2/\pi}T_j$, with sensitivity $\sqrt{8LH_k/(\pi n^2)}$). Continuous columns are *not* rounded to a grid; a grid defines categorical values only. We calibrate σ with the analytic Gaussian mechanism (as do all baselines, where appropriate, for fairness).

For our objective minimization we use gradient descent on the model parameters, enforcing the simplex constraints on π and on each $\phi^{(h,a)}$ by a softmax reparameterization and keeping the covariance diagonal positive through a logarithmic parameterization. We warm start the continuous-block means from the degree-one moments by least squares, since $m_1(\theta, \hat{q})$ is linear in the means, and we stabilize the non-convex fit with random restarts (our experiments use one), an exponential moving average of the parameters (Polyak & Juditsky, 1992), a Huber loss that caps the influence of the noisiest high-degree residuals (Huber, 1992), and light entropy and ridge regularization. Note that all of these are standard moves, and additionally are post-processing with respect to the release of the noisy sliced Chebyshev moments.

Sampling. To draw a synthetic record we sample a component $h \sim \pi$, draw each categorical column independently from $\phi^{(h,a)}$, draw the continuous block from $\mathcal{N}(\mu_h, \Sigma_h)$, clip it to the encoded range, and de-normalize. We note that clipping is also what squares the model with Theorem 5: the Gaussian block in Equation 13 has unbounded support, whereas the theorem is stated for distributions on $\mathcal{B}^d$, so it is the clipped samples, rather than $\hat{q}$ itself, that satisfy the support assumption.

Since fitting $\hat{q}$ and sampling from it are both post-processing of the noisy moments, privacy is unaffected either way. Finally, because the continuous block is Gaussian, samples are smooth and do not show the quantization artifacts of grid-based mechanisms.

E Additional Setup, Experiments and Ablations

Public metadata convention. For a fair comparison, all methods are given the same kind of side information: column schema metadata is read off the training set and treated as public, so estimating it does not consume privacy budget. What each method reads differs. SPLICE uses only per-column bounds (minimum and maximum) and the level set of each categorical column. Private-GSD takes an explicit dictionary of bounds; AIM treats every column as categorical and reads the categories from the data; GEM uses quantile bin edges; and PrivBayes builds its support from the observed values. Treating this metadata as public is a small idealization, but it is the standard one in the literature. We note that two of the baselines are given strictly more than bounds: GEM's quantile edges reveal the shape of each marginal, and AIM's category lists reveal exactly which values occur. SPLICE has the smallest metadata footprint of the methods we compare.

Computational limits of baseline methods. Our comprehensive evaluation attempts $210 \times 4 \times 5 = 4200$ generation tasks per algorithm (incorporating the 209 benchmark datasets along with one synthetic smoke-test dataset excluded from the final metrics). We enforce a strict two-hour timeout per task (all methods were GPU accelerated, except for PrivBayes which has a fast C implementation). PrivBayes fails on 34 tasks due to the most massive datasets (its runtime scales as $\approx n^2$, and our estimated runtime for these larger datasets was multiple days for the method). AIM with 2-way workloads fails on 34 tasks due to timeouts and GPU memory exhaustion, primarily at $\epsilon = 10$, where it allocates larger budgets across more measurement iterations; its 3-way variant drops 153 tasks across ten high-dimensional datasets, seven of which fail across all budgets regardless of row count. GEM utilizing 3-way workloads drops 50 tasks on wide datasets with high cardinality, whereas its 2-way version successfully completes all tasks. Private-GSD runs out of host memory on larger tables and is thus evaluated solely on the subsets in general (this is despite being run on an NVIDIA RTX PRO 6000 Blackwell GPU with 96 GB of VRAM). Conversely, SPLICE successfully executed all 16,800 of its generations, with no timeouts or out-of-memory errors. This reliability is a result of the method architecture: the sliced release compresses the data into fixed-size moment sketches, scaling linearly with n and avoiding the combinatorial explosion associated with width, cardinality, or privacy budget.

Baseline implementations and parameter settings. We use the maintained or author-released implementation of every baseline. AIM runs through the OpenDP Library (`https://github.com/opendp/opendp`, version 0.14.2), whose implementation builds on the graphical-model inference of private-pgm (`https://github.com/ryan112358/private-pgm`, `mbi` 1.1.0). GEM uses the authors' `dp-query-release` codebase (`https://github.com/terranceliu/dp-query-release`), consolidated into our harness without algorithmic changes. PrivBayes runs the authors' reference C++ implementation through a wrapper following SDGym (`https://github.com/sdv-dev/SDGym`). Private-GSD uses the authors' implementation (`https://github.com/giusevtr/private_gsd`). Causal discovery uses the GRaSP implementation in `causal-learn` (`https://github.com/py-why/causal-learn`). Every method runs with the settings recommended by its source implementation, and no method (including SPLICE, whose k and L follow the fixed formulas of Table 1) receives per-dataset hyperparameter tuning. The only per-baseline variation we report is the workload degree (2-way and 3-way marginals for AIM and GEM), shown as separate baselines, and all methods share the privacy accounting conventions and public metadata described above.

E.1 SPLICE Model Variants

The mixture of categorical-Gaussian product distributions from Section 4 generalizes two simpler models: a CP-style mixture of categorical product distributions, and a low-rank Gaussian mixture. (Figure legends in this appendix abbreviate the full model as "mixed block".) We also ran both of these special cases as SPLICE variants, fit to the exact same noisy moments. Figure 8 ranks the three on roughly 97 datasets, stratified by the fraction of discrete columns. The results are what the model classes suggest: the Gaussian mixture is best on continuous data, the CP model is best on discrete data (where the Gaussian mixture does poorly), and the full mixed model is best on mixed data.

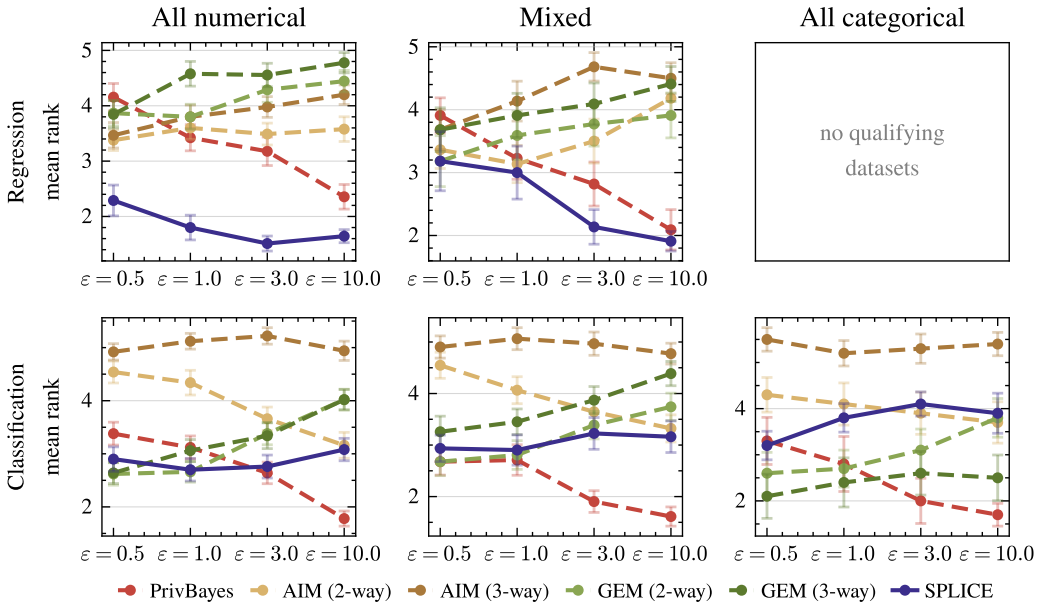


Figure 6: Mean TSTR rank (lower is better) by privacy budget, split by task family (top: regression, bottom: classification) and by feature type (columns: all numerical, mixed, all categorical). Each panel uses a fixed set of well-formed datasets on which every plotted method has a value at every budget: $n = 45$ (numerical) and 22 (mixed) for regression; $n = 50$, 31, and 10 for classification. Error bars are ± 1 SEM across datasets. The benchmark has no well-formed all-categorical regression datasets.

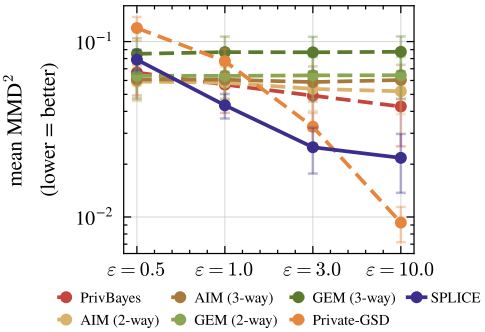


Figure 7: Unbiased RBF-kernel MMD^2 between real and synthetic data (lower is better, log scale), median-heuristic bandwidth, $n = 36$ datasets. Error bars are ± 1 SEM across datasets.

Figure 9 compares slicing to the multivariate Chebyshev mechanism of Musco et al. (2025) on 4-dimensional datasets, the only regime where the latter is computationally feasible. A sliced `SPLICE` variant is ranked first for every task type and budget. The multivariate variants are behind on regression at every ϵ; multivariate CP reaches second on classification at the looser budgets but is never first. Against the full set of baselines, the multivariate variants rank at or near the bottom. In other words, slicing helps even when d is small enough that the curse of dimensionality is not binding.

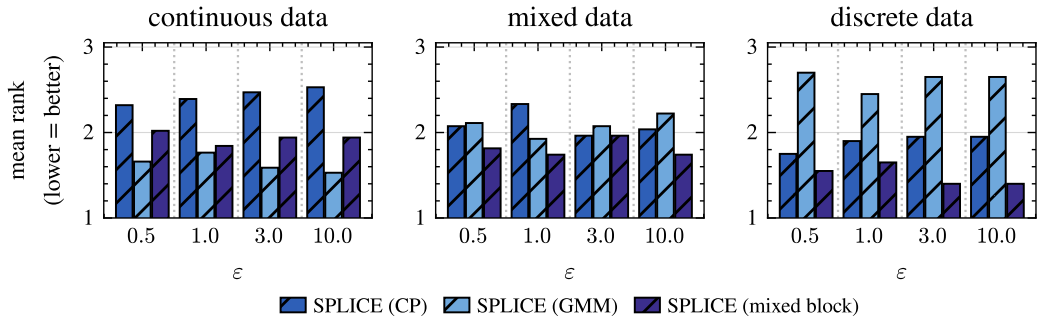


Figure 8: Mean rank (lower is better) of the three SPLICE variants (CP, GMM, and the mixture of categorical-Gaussian product distributions, labeled "mixed block"), split by the fraction of discrete columns: continuous (≤ 0.3), mixed, and discrete (≥ 0.7). Left to right, the strata contain $n = 50$ to 51, 27, and 20 datasets per budget.

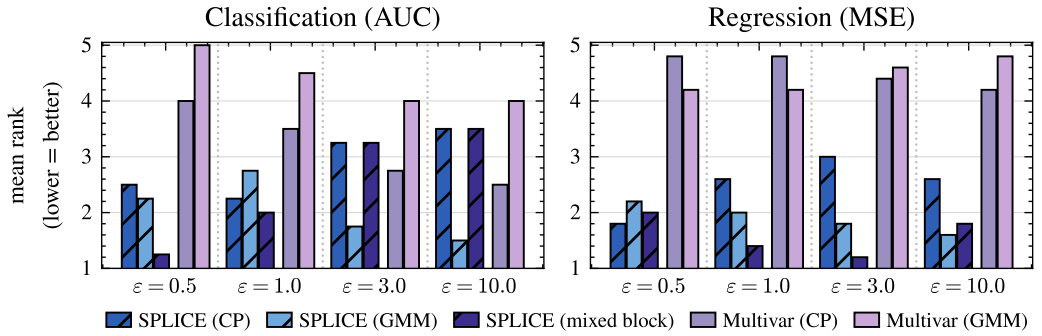


Figure 9: Mean rank (lower is better) on 4-dimensional datasets (each restricted to its top 3 features), where the multivariate Chebyshev mechanism is still computationally feasible: the two multivariate variants against the sliced SPLICE variants. Left: classification, $n = 4$ datasets. Right: regression, $n = 5$.

F RESULT TABLES

This section gives the numbers behind the figures in the main text. Table 2 reports the five fidelity metrics at $\epsilon = 1$, i.e., the components of the fidelity axis of Figure 1. Table 3 gives the mean fidelity rank at every budget. Table 4 gives mean TSTR rank by budget and task family on a fixed set of datasets, and Table 5 breaks the $\epsilon = 1$ column down by downstream learner. Every table is computed from the same saved results and the same dataset filters as the figure it supports.

Method	1-way TVD (n=58)	2-way TVD (n=58)	3-way TVD (n=58)	MMD^2 (n=40)	SW_1 (n=40)
PrivBayes	0.214 ±0.027	0.442 ±0.037	0.642 ±0.041	0.057 ±0.016	0.032 ±0.002
AIM (2-way)	**0.143** ±0.028	0.412 ±0.038	0.638 ±0.042	0.058 ±0.012	0.036 ±0.004
AIM (3-way)	**0.143** ±0.028	0.420 ±0.038	0.650 ±0.040	0.061 ±0.012	0.039 ±0.004
GEM (2-way)	0.189 ±0.026	0.438 ±0.036	0.658 ±0.039	0.065 ±0.016	0.036 ±0.002
GEM (3-way)	0.243 ±0.025	0.462 ±0.036	0.662 ±0.039	0.088 ±0.018	0.045 ±0.003
Private-GSD	0.270 ±0.027	0.486 ±0.035	0.668 ±0.038	0.077 ±0.013	0.047 ±0.004
Sliced (mixed block, ours)	0.187 ±0.013	**0.388** ±0.023	**0.586** ±0.030	**0.043** ±0.006	**0.026** ±0.003

Table 2: Distributional fidelity at $\varepsilon = 1$, the five components of the Figure 1 fidelity axis. Each entry is the mean ± 1 SEM over the datasets (n per column) where every method has a value; lower is better and the best value per column is bold.

Method	$\varepsilon = 0.5$	$\varepsilon = 1.0$	$\varepsilon = 3.0$	$\varepsilon = 10.0$
PrivBayes	5.00	3.40	3.00	3.20
AIM (2-way)	**1.40**	2.40	3.00	3.40
AIM (3-way)	2.60	3.60	4.60	5.00
GEM (2-way)	2.80	4.20	5.60	5.60
GEM (3-way)	5.40	6.20	7.00	6.80
Private-GSD	7.00	6.80	3.40	2.40
Sliced (mixed block, ours)	3.80	**1.40**	**1.40**	**1.60**

Table 3: Mean fidelity rank (over the five distributional metrics of Table 2) at every privacy budget. Rank 1 is best out of 7; the best value per column is bold.

	Classification (n=91)				Regression (n=67)			
Method	ε=0.5	ε=1.0	ε=3.0	ε=10.0	ε=0.5	ε=1.0	ε=3.0	ε=10.0
PrivBayes	3.13 ±0.16	2.95 ±0.17	**2.32** ±0.15	**1.71** ±0.11	4.07 ±0.19	3.36 ±0.19	3.06 ±0.21	2.27 ±0.18
AIM (2-way)	4.52 ±0.15	4.22 ±0.17	3.68 ±0.16	3.27 ±0.17	3.37 ±0.16	3.45 ±0.17	3.49 ±0.18	3.78 ±0.18
AIM (3-way)	4.98 ±0.12	5.11 ±0.11	5.14 ±0.12	4.93 ±0.13	3.54 ±0.19	3.91 ±0.18	4.21 ±0.15	4.30 ±0.15
GEM (2-way)	**2.64** ±0.16	**2.71** ±0.15	3.35 ±0.16	3.90 ±0.15	3.64 ±0.21	3.73 ±0.20	4.12 ±0.19	4.27 ±0.18
GEM (3-way)	2.79 ±0.17	3.12 ±0.16	3.44 ±0.17	3.98 ±0.15	3.79 ±0.20	4.36 ±0.19	4.40 ±0.18	4.66 ±0.16
Sliced (mixed block, ours)	2.95 ±0.16	2.89 ±0.16	3.07 ±0.17	3.20 ±0.17	**2.58** ±0.25	**2.19** ±0.22	**1.72** ±0.13	**1.73** ±0.10

Table 4: Downstream TSTR mean rank (± 1 SEM across datasets) by privacy budget and task family, best downstream learner per dataset. Datasets are the fixed sets of well-formed tasks covered by every shown method at every budget, so the composition is identical across columns; rank 1 is best out of 6 and the best value per column is bold. Private-GSD is excluded because it covers only 10 of the 73 well-formed regression datasets; its dedicated comparison appears in **??**.

	Classification				Regression			
Method	LogReg/Ridge	RF	XGB	TabPFN	LogReg/Ridge	RF	XGB	TabPFN
PrivBayes	3.02	**2.56**	**2.71**	2.85	3.27	3.04	3.18	3.40
AIM (2-way)	4.24	4.08	4.03	4.14	3.43	3.96	3.97	3.34
AIM (3-way)	4.92	4.81	4.80	4.92	3.70	3.96	4.03	3.76
GEM (2-way)	3.15	3.31	3.19	3.16	4.16	3.91	3.94	3.75
GEM (3-way)	3.13	3.41	3.36	3.10	4.33	4.10	4.06	4.43
Sliced (mixed block, ours)	**2.53**	2.84	2.90	**2.83**	**2.10**	**2.03**	**1.82**	**2.31**

Table 5: Downstream TSTR mean rank at $\varepsilon = 1$ broken out by downstream learner (the linear column is logistic regression on classification and ridge on regression). Same fixed dataset sets and rank convention as Table 4; the headline figures use the best learner per dataset.

G DATASET DETAILS

Table 6: The 209 TALENT datasets in the benchmark: task type, training and test rows, numerical and categorical feature counts, and whether the dataset enters the well-formed aggregate set (WF; real-data ceiling AUC ≥ 0.7 or $R^2 \geq 0.3$). A few long dataset names are truncated.

Dataset	Task	n_{train}	n_{test}	num.	cat.	WF
1000-Cameras-Dataset	regression	830	208	10	0	✓
2dplanes	regression	32,614	8,154	10	0	✓
3D_Estimation_using_RSSI_of_WLAN_dataset	regression	4,608	1,152	6	0	✓
3D_Estimation_using_RSSI_of_WLAN_dataset_c...	regression	11,520	2,880	12	0	✓
abalone	multiclass	3,341	836	7	1	✓
Abalone_reg	regression	3,341	836	7	1	✓
accelerometer	multiclass	122,403	30,601	3	1	✓
ada_prior	binary	3,649	913	6	8	✓
airfoil_self_noise	regression	1,202	301	5	0	✓
airlines_seed_0_nrows_2000_nclasses_10_nco...	binary	1,600	400	3	4	
Amazon_employee_access	binary	26,215	6,554	0	7	✓
analcatdata_supreme	regression	3,241	811	7	0	✓
Another-Dataset-on-used-Fiat-500-(1538-rows)	regression	1,230	308	6	0	✓
archive2	regression	914	229	11	1	✓
artificial-characters	multiclass	8,174	2,044	7	0	✓
auction_verification	regression	1,634	409	6	1	✓
autoUniv-au7-1100	multiclass	880	220	8	4	
avocado_sales	regression	14,599	3,650	11	2	✓
bank	binary	36,168	9,043	7	9	✓
bank8FM	regression	6,553	1,639	8	0	✓
Bank_Customer_Churn_Dataset	binary	8,000	2,000	6	4	✓
banknote_authentication	binary	1,097	275	4	0	
baseball	multiclass	1,072	268	15	1	✓
Basketball_c	binary	1,072	268	11	0	✓
Bias_correction_r	regression	6,180	1,545	21	0	✓
Bias_correction_r_2	regression	6,180	1,545	21	0	✓
BLE_RSSI_dataset_for_Indoor_localization	multiclass	7,987	1,997	3	0	✓
BNG(breast-w)	binary	31,492	7,874	9	0	✓
BNG(cmc)	multiclass	44,236	11,060	2	7	✓
BNG(echoMonths)	regression	13,996	3,500	5	3	✓
BNG(lowbwt)	regression	24,883	6,221	2	7	✓
BNG(mv)	regression	62,985	15,747	7	3	✓
BNG(stock)	regression	47,239	11,810	9	0	✓
BNG(tic-tac-toe)	binary	31,492	7,874	0	9	✓
Brazilian_houses_reproduced	regression	8,553	2,139	8	0	✓
California-Housing-Classification	binary	16,512	4,128	8	0	✓
car-evaluation	multiclass	1,382	346	0	21	✓
Cardiovascular-Disease-dataset	binary	56,000	14,000	5	6	✓
CDC_Diabetes_Health_Indicators	binary	202,944	50,736	7	14	✓
churn	binary	4,000	1,000	16	4	✓
Click_prediction_small	binary	31,958	7,990	3	0	
cmc	multiclass	1,178	295	2	7	✓
combined_cycle_power_plant	regression	7,654	1,914	4	0	✓
compass	binary	13,315	3,329	8	9	✓
compass_reg	regression	13,315	3,329	17	0	✓
concrete_compressive_strength	regression	824	206	8	0	✓
contraceptive_method_choice	multiclass	1,178	295	5	4	✓
CookbookReviews	regression	14,545	3,637	7	0	
CPMP-2015-runtime-regression	regression	1,686	422	22	1	✓
CPS1988	regression	22,524	5,631	2	4	
cpu_act	regression	6,553	1,639	21	0	✓
cpu_small	regression	6,553	1,639	12	0	✓
credit	binary	13,371	3,343	10	0	✓
Credit_c	multiclass	80,000	20,000	17	5	✓
credit_reg	regression	13,371	3,343	10	0	✓

Table 6 continued.

Dataset	Task	n_{train}	n_{test}	num.	cat.	WF
Customer_Personality_Analysis	binary	1,792	448	22	2	✓
customer_satisfaction_in_airline	binary	103,904	25,976	4	17	✓
dabetes_130-us_hospitals	binary	81,412	20,354	8	12	
Data_Science_for_Good_Kiva_Crowdfunding	multiclass	536,964	134,241	7	4	✓
Data_Science_Salaries	regression	3,004	751	1	4	
dataset_sales	regression	8,590	2,148	10	0	✓
debutanizer	regression	1,915	479	7	0	✓
default_of_credit_card_clients	binary	24,000	6,000	14	9	✓
delta_ailerons	binary	5,703	1,426	5	0	✓
delta_elevators	regression	7,613	1,904	6	0	✓
Diabetic_Retinopathy_Debrecen	binary	920	231	16	3	✓
Diamonds	regression	43,152	10,788	6	3	✓
drug_consumption	multiclass	1,507	377	12	0	
dry_bean_dataset	multiclass	10,888	2,723	16	0	✓
E-CommereShippingData	binary	8,799	2,200	6	4	✓
eeg-eye-state	binary	11,984	2,996	14	0	✓
electricity	binary	36,249	9,063	7	1	✓
elevators	regression	13,279	3,320	18	0	✓
Employee	binary	3,722	931	3	5	✓
estimation_of_obesity_levels	multiclass	1,688	423	8	8	✓
eye_movements_bin	binary	6,086	1,522	20	0	✓
FICO-HELOC-cleaned	binary	7,896	1,975	21	2	✓
fifa	regression	14,450	3,613	5	0	✓
Firm-Teacher_Clave-Direction_Classification	multiclass	8,640	2,160	0	16	✓
Fitness_Club_c	binary	1,200	300	3	3	✓
Food_Delivery_Time	regression	36,474	9,119	6	2	✓
FOREX_audcad-day-High	binary	1,467	367	10	0	✓
FOREX_audcad-hour-High	binary	35,060	8,765	10	0	✓
FOREX_audchf-day-High	binary	1,466	367	10	0	✓
FOREX_audjpy-day-High	binary	1,465	367	10	0	✓
FOREX_audjpy-hour-High	binary	35,060	8,765	10	0	✓
FOREX_audsgd-hour-High	binary	35,060	8,765	10	0	✓
FOREX_audusd-hour-High	binary	35,060	8,765	10	0	✓
FOREX_cadjpy-day-High	binary	1,467	367	10	0	✓
FOREX_cadjpy-hour-High	binary	35,060	8,765	10	0	✓
fried	regression	32,614	8,154	10	0	✓
GAMETES_Epistasis_2-Way_20atts_0.1H_EDM-1_1	binary	1,280	320	0	20	
GAMETES_Heterogeneity_20atts_1600_Het_0.4_...	binary	1,280	320	0	20	✓
garments_worker_productivity	regression	957	240	8	5	✓
gas_turbine_CO_and_NOx_emission	regression	29,386	7,347	10	0	✓
Gender_Gap_in_Spanish_WP	multiclass	3,796	950	13	0	
golf_play_dataset_extended	binary	876	219	3	6	✓
Goodreads-Computer-Books	regression	987	247	5	0	
healthcare_insurance_expenses	regression	1,070	268	3	3	✓
Heart-Disease-Dataset-(Comprehensive)	regression	952	238	11	0	✓
heloc	binary	8,000	2,000	22	0	✓
house_16H	binary	10,790	2,698	16	0	✓
house_16H_reg	regression	18,227	4,557	16	0	✓
house_8L	regression	18,227	4,557	8	0	✓
house_sales_reduced	regression	17,290	4,323	18	0	✓
houses	regression	16,512	4,128	8	0	✓
housing_price_prediction	regression	436	109	5	7	✓
HR_Analytics_Job_Change_of_Data_Scientists	binary	15,326	3,832	3	10	✓
htru	binary	14,318	3,580	8	0	✓
in_vehicle_coupon_recommendation	binary	10,147	2,537	0	21	✓
INNHotelsGroup	binary	29,020	7,255	11	6	✓
Insurance	regression	18,838	4,710	3	7	
internet_firewall	multiclass	52,425	13,107	7	0	✓
Intersectional-Bias-Assessment	regression	8,800	2,200	14	5	✓
Is-this-a-good-customer	regression	1,378	345	9	4	
JapaneseVowels	multiclass	7,968	1,993	14	0	✓
jm1	binary	8,708	2,177	21	0	✓

Table 6 continued.

Dataset	Task	n_{train}	n_{test}	num.	cat.	WF
jungle_chess_2pcs_raw_endgame_complete	multiclass	35,855	8,964	6	0	✓
Kaggle_bike_sharing_demand_challange	regression	8,708	2,178	3	6	✓
kc1	binary	1,687	422	21	0	✓
kdd_ipums_la_97-small	binary	4,150	1,038	20	0	✓
kin8nm	regression	6,553	1,639	8	0	✓
kr-vs-k	multiclass	22,444	5,612	3	3	✓
kropt	multiclass	22,444	5,612	0	6	✓
Laptop_Prices_Dataset	regression	3,552	889	8	0	✓
law-school-admission-bianry	regression	16,640	4,160	7	4	✓
led24	multiclass	2,560	640	0	24	✓
led7	multiclass	2,560	640	0	7	✓
letter	multiclass	16,000	4,000	15	0	✓
Long	regression	3,581	896	19	0	✓
MagicTelescope	binary	15,216	3,804	9	0	✓
mammography	binary	8,946	2,237	6	0	✓
maternal_health_risk	multiclass	811	203	6	0	✓
mauna-loa-atmospheric-co2	regression	1,780	445	5	1	✓
mfeat-morphological	multiclass	1,600	400	6	0	✓
MiamiHousing2016	regression	11,145	2,787	16	0	✓
microaggregation2	multiclass	16,000	4,000	20	0	✓
mobile_c36_oversampling	binary	41,408	10,352	6	0	✓
Mobile_Phone_Market_in_Ghana	regression	2,880	720	4	10	✓
Mobile_Price_Classification	multiclass	1,600	400	14	6	✓
mozilla4	binary	12,436	3,109	4	0	✓
mv	regression	32,614	8,154	7	3	✓
NASA_PHM2008	regression	36,734	9,184	21	0	✓
national-longitudinal-survey-binary	binary	3,926	982	9	7	✓
National_Health_and_Nutrition_Health_Survey	binary	1,822	456	7	0	✓
NHANES_age_prediction	regression	1,821	456	4	3	✓
okcupid_stem	multiclass	21,341	5,336	2	11	✓
online_shoppers	binary	9,864	2,466	5	9	✓
page-blocks	multiclass	4,378	1,095	10	0	✓
Parkinsons_Telemonitoring	regression	4,700	1,175	18	1	✓
pc1	binary	887	222	21	0	✓
pendigits	multiclass	8,793	2,199	16	0	✓
Performance-Prediction	regression	1,072	268	19	0	
phoneme	binary	4,323	1,081	5	0	✓
Physicochemical_r	regression	36,584	9,146	9	0	✓
Pima_Indians_Diabetes_Database	binary	614	154	8	0	✓
puma8NH	regression	6,553	1,639	8	0	✓
Pumpkin_Seeds	binary	2,000	500	12	0	✓
qsar_aquatic_toxicity	regression	436	110	8	0	✓
qsar_fish_toxicity	regression	726	182	4	2	✓
Rain_in_Australia	multiclass	116,368	29,092	12	6	✓
rice_cammeo_and_osmancik	binary	3,048	762	7	0	✓
ringnorm	binary	5,920	1,480	20	0	✓
rl	binary	3,976	994	5	7	✓
SDSS17	multiclass	80,000	20,000	8	4	✓
segment	multiclass	1,848	462	17	0	✓
seismic+bumps	binary	2,067	517	14	4	
sensory	regression	460	116	0	11	
shill-bidding	regression	5,056	1,265	3	0	
Shipping	regression	8,799	2,200	5	4	
Shop_Customer_Data	regression	1,600	400	4	2	
shrutime	regression	8,000	2,000	4	6	✓
shuttle	multiclass	46,400	11,600	9	0	✓
Smoking_and_Drinking_Dataset_with_body_signal	binary	793,076	198,270	22	1	✓
socmob	regression	924	232	1	4	✓
space_ga	regression	2,485	622	6	0	✓
statlog	binary	800	200	7	13	✓
steel_industry_energy_consumption	regression	28,032	7,008	7	3	✓
stock	regression	760	190	9	0	✓

Table 6 continued.

Dataset	Task	n_{train}	n_{test}	num.	cat.	WF
stock_fardamento02	regression	5,021	1,256	5	1	
sulfur	regression	8,064	2,017	6	0	✓
svmguide3	regression	994	249	22	0	✓
sylvine	binary	4,099	1,025	20	0	✓
telco-customer-churn	binary	5,634	1,409	3	15	✓
Telecom_Churn_Dataset	binary	2,666	667	14	3	✓
thyroid	multiclass	5,760	1,440	21	0	✓
thyroid-ann	multiclass	3,017	755	21	0	✓
treasury	regression	839	210	15	0	✓
twonorm	binary	5,920	1,480	20	0	✓
vehicle	multiclass	676	170	18	0	✓
VulNoneVul	regression	4,553	1,139	16	0	
walking-activity	multiclass	119,465	29,867	4	0	✓
wall-robot-navigation	multiclass	4,364	1,092	24	0	✓
water_quality	binary	6,396	1,600	20	0	✓
Water_Quality_and_Potability	binary	2,620	656	8	0	
Waterstress	regression	950	238	22	0	
waveform_database_generator	multiclass	3,999	1,000	21	0	
waveform_database_generator_version_1	multiclass	4,000	1,000	21	0	✓
weather_izmir	regression	1,168	293	9	0	✓
website_phishing	multiclass	1,082	271	0	9	✓
Wilt	binary	3,856	965	5	0	✓
wind	regression	5,259	1,315	14	0	✓
wine	binary	2,043	511	4	0	✓
wine+quality	regression	5,197	1,300	11	0	✓
wine-quality-red	multiclass	1,279	320	4	0	✓
wine-quality-white	multiclass	3,918	980	11	0	✓
Wine_Quality_red	regression	1,279	320	11	0	✓
Wine_Quality_white	regression	3,918	980	11	0	✓
yeast	multiclass	1,187	297	8	0	✓

H Resource Scaling and Compute

The real-world viability of a synthetic data algorithm hinges heavily on its resource consumption alongside its statistical utility, so we document the measured wall-clock time and memory usage for each method using a standardized protocol. All benchmarks process a single task per GPU at $\epsilon = 1$, timing the fitting and sampling phases together, with peak resident memory recorded from the operating system and peak GPU memory tracked via NVML.

Cross-method comparison. On benchmark datasets ranging from 760 to 63,000 training rows, `SPLICE`'s cost is nearly flat: fit plus sampling takes 20 to 60 seconds, about 2 GB of resident memory, and under 1 GB of GPU memory. The reason is simple. The release is a single $O(n)$ pass over the data, and the fit only ever touches the Lk released moments. On this range only PrivBayes (2 to 12 seconds, CPU only) and GEM are faster. Private-GSD goes from 52 seconds to a timeout at 63,000 rows, and 3-way AIM spikes on wide datasets. All numbers are averages over three seeds.

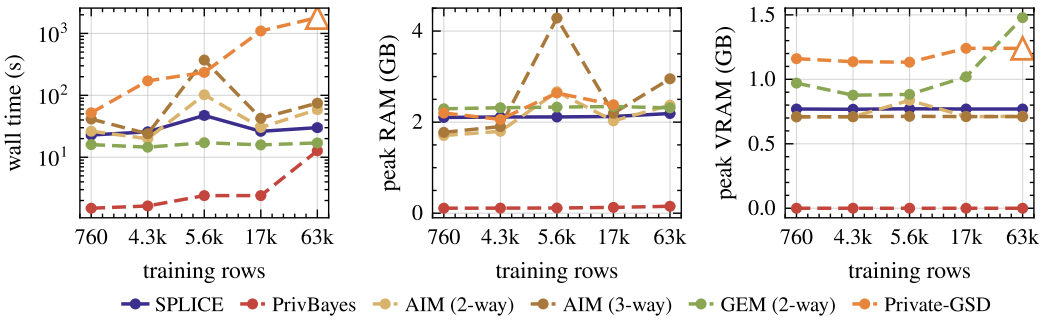


Figure 10: Cost of generation at $\epsilon = 1$: total wall time (log scale; open triangles are timeouts), peak resident memory, and peak GPU memory. Datasets, left to right: stock, phoneme, telco-customer-churn, California-Housing-Classification, BNG(mv). Markers are means over three seeds with ± 1 SEM error bars; a timed-out cell is a single run stopped at the limit.

Per-method scaling laws. We repeat the measurement on a controlled grid built from one large mixed dataset: a row sweep from 12,500 to 400,000 rows at $d = 12$, and a column sweep from 6 to 24 columns at $n = 50,000$. Fitting $t \propto n^{\alpha} c^{d}$, where α is the row exponent and c the per-column multiplier, the `SPLICE` variants are essentially flat in both ($\alpha \leq 0.09$, $c \leq 1.05$) and finish in 30 to 120 seconds everywhere on the grid. GEM is similarly flat because it runs a fixed number of epochs. 2-way AIM grows moderately with n ($\alpha \approx 0.55$); 3-way AIM does not depend on n but pays a factor of about 1.16 per column, going from 61 seconds at 6 columns to 960 seconds at 24. PrivBayes depends on both through its parent-set search, and scales as $n^{2.2}$ on wide tables. Private-GSD is the most expensive by far: 659 to 1,470 seconds across seeds at only 12,500 rows, the 2,400-second limit on two of three seeds at 25,000 rows, and a timeout beyond that even at 6 columns. So, among the methods we test, only the `SPLICE` variants are both accurate on continuous features and insensitive to growth in either n or d.

Compute. All experiments ran on a single server with two AMD EPYC 9354 processors (64 cores, 128 threads), 512 GB of RAM, and six NVIDIA RTX PRO 6000 Blackwell GPUs with 96 GB of memory each. Every generation task fits comfortably on one GPU; the resource measurements of Section H run one task at a time per GPU, while the sweeps schedule independent tasks across GPUs in parallel. `SPLICE` itself is light: a single fit-plus-sample cell takes 20 to 120 seconds and under 1 GB of GPU memory, so its full 16,800-cell sweep totals roughly 200 GPU-hours and would run on one consumer-grade GPU in about a week. The overall benchmark cost is dominated by the baselines.

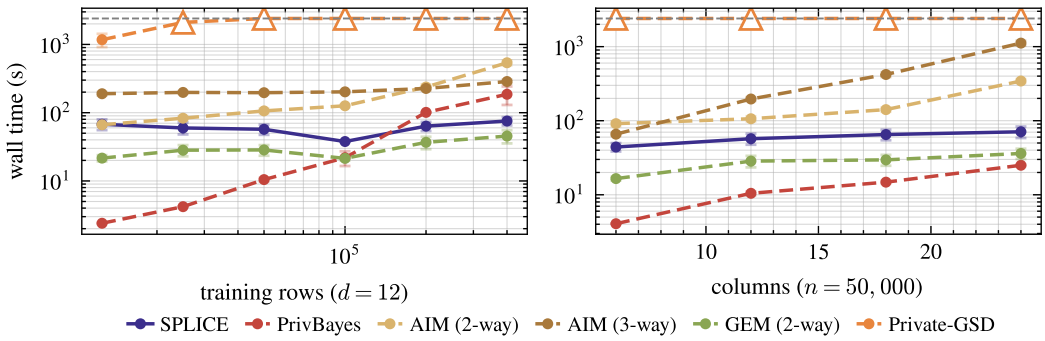


Figure 11: Controlled scaling at $\epsilon = 1$: wall time against number of training rows at $d = 12$ (left, log-log) and against number of columns at $n = 50,000$ (right). Markers are means over three seeds with ± 1 SEM error bars; open triangles are runs that hit the time limit.