PrAda-GAN: A Private Adaptive Generative Adversarial Network with Bayes Network Structure

Ke Jia[1, 2*], **Yuheng Ma**[1, 2*], **Yang Li**[1, 2], **Feifei Wang** [1, 2†]

[1]Center for Applied Statistics, Renmin University of China
[2]School of Statistics, Renmin University of China
{jiake1999, yma, yang.li, feifei.wang}@ruc.edu.cn

Abstract

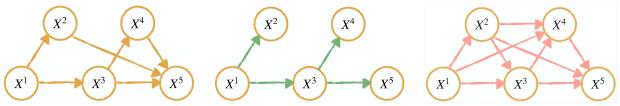
We revisit the problem of generating synthetic data under differential privacy. To address the core limitations of marginal-based methods, we propose the Private Adaptive Generative Adversarial Network with Bayes Network Structure (`PrAda-GAN`), which integrates the strengths of both GAN-based and marginal-based approaches. Our method adopts a sequential generator architecture to capture complex dependencies among variables, while adaptively regularizing the learned structure to promote sparsity in the underlying Bayes network. Theoretically, we establish diminishing bounds on the parameter distance, variable selection error, and Wasserstein distance. Our analysis shows that leveraging dependency sparsity leads to significant improvements in convergence rates. Empirically, experiments on both synthetic and real-world datasets demonstrate that `PrAda-GAN` outperforms existing tabular data synthesis methods in terms of the privacy–utility trade-off.

1 Introduction

Synthetic data is the new fossil fuel of modern AI, driving the success of multiple domains as models grow larger and demand unprecedented amounts of data for training (Wang et al. 2022; Gadre et al. 2023; Lu et al. 2023). However, generative models are not immune to privacy risks—particularly membership inference attacks (MIAs), in which adversaries attempt to determine whether specific records are part of the training data. These vulnerabilities arise when generative models inadvertently memorize training samples, causing the synthetic outputs to closely resemble the originals or reveal exploitable statistical patterns (Sun et al. 2021; Andrey, Bars, and Tommasi 2025).

To address these risks, differential privacy (DP, Dwork et al. 2006) is commonly applied during the training of generative models (Jordon, Yoon, and Van Der Schaar 2018; Xie et al. 2018), ensuring that the model's outputs remain indistinguishable regardless of whether any individual data point is included in the training set. Numerous studies demonstrated DP generation of tabular data (Tao et al. 2021; Yang et al. 2024; Chen et al. 2025), which remains the most prevalent data type in data science (Hollmann et al. 2025; Zhang et al. 2025). Among these, marginal-based methods, such as `PrivBayes` (Zhang et al. 2017) and `AIM` (McKenna et al. 2022), often achieve superior utility (NIST 2019).

However, marginal-based approaches have notable limitations. First, they rely on low-dimensional structural assumptions for effective performance. For instance, `PrivBayes` assumes a Bayes network structure, while `AIM` requires that the underlying distribution be well-approximated by a low-dimensional marginal structure in terms of workload error. Although such assumptions are often reasonable, they are difficult to verify and to adaptively match to the true, unknown degree of low-dimensionality. For example, if a few nodes have significantly more parent nodes than others, it becomes challenging to infer the overall network structure using the single hyperparameter in `PrivBayes` (Zhang et al. 2017). This issue is illustrated in Figure 1, where it may either underfit (by ignoring meaningful dependencies) or overfit (by including redundant ones). Moreover, these marginal-based methods are primarily designed for categorical variables, requiring continuous variables to be discretized through binning. In addition to the extra tuning cost involved in selecting appropriate bin sizes, this process also hinders the generation of heavy-tailed distributions.



(a) True network (b) Underfitting (c) Overfitting

Figure 1: Drawback illustration of marginal-based methods.

In this paper, we address these challenges by proposing the Private Adaptive Generative Adversarial Network with Bayes Network Structure (`PrAda-GAN`), a novel approach for DP tabular data generation that integrates generative adversarial networks (GANs) with low-dimensional structural modeling. Our method employs a sequential generator architecture to capture complex dependencies among variables, while adaptively regularizing the learned structure to promote sparsity in the underlying Bayes network. Compared to existing methods, `PrAda-GAN` offers two key advantages:

*These authors contributed equally.
†Corresponding author.

(1) it adapts to unknown low-dimensional structures without the need to tune sensitive hyperparameters; and (2) it naturally supports unbounded continuous domains without requiring discretization. Our contributions are:

- We revisit differentially private tabular data generation and identify key limitations of prior marginal-based approaches. To address these challenges, we propose `PrAda-GAN`, a novel method that combines the strengths of both marginal-based and GAN-based models. By introducing adaptive regularization, our approach implicitly recovers the underlying low-dimensional Bayes network structure during GAN training.
- We provide a theoretical analysis of `PrAda-GAN`. First, we establish a bound on the distance between the trained generator and the optimal candidate set. Then, by analyzing the recovery of low-dimensional structures, we derive the first generalization bound for the Wasserstein distance between the generated and true data distributions. Notably, our results demonstrate that adaptive regularization leads to a significantly improved convergence rate.
- We conduct extensive experiments on both synthetic and real-world datasets. Through a detailed analysis of parameter influence, we show that `PrAda-GAN` is robust to hyperparameter choices and supports our theoretical findings. We further benchmark `PrAda-GAN` against state-of-the-art baselines using measures of distributional similarity and downstream utility. The results demonstrate the empirical superiority of our approach.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed `PrAda-GAN` framework. Section 4 presents theoretical guarantees. Section 5 presents numerical results. Finally, Section 6 concludes the paper.

2 Related Work

GAN Based Methods. Given the remarkable success of GANs (Goodfellow et al. 2014), a growing body of works have explored their applications to differentially private data synthesis (Jordon, Yoon, and Van Der Schaar 2018; Liu et al. 2019; Chen, Orekondy, and Fritz 2020; Long et al. 2021; Bie, Kamath, and Zhang 2023; Ma et al. 2023). A common approach to privatizing GANs is to apply differentially private stochastic gradient descent (DPSGD) (Abadi et al. 2016) when updating the discriminator, a method known as DPGAN (Xie et al. 2018; Zhang, Ji, and Wang 2018; Torkzadehmahani, Kairouz, and Paten 2019; Zhao et al. 2024). This technique is shown to be effective for generating private synthetic data across various domains.

Marginal-based Methods. Low-order marginals are widely adopted in tabular data synthesis due to their ability to capture essential low-dimensional structures while exhibiting low sensitivity under DP (Hu et al. 2024). Marginal-based approaches typically select a set of marginals, inject calibrated noise, and reconstruct the joint distribution to generate synthetic data. A promising line of work employs Bayes networks to model conditional dependencies through a directed acyclic graph (DAG). Representative examples include `PrivBayes` (Zhang et al. 2017), which learns the network structure from data and perturbs the conditional distributions, and AIM (McKenna et al. 2022), which enhances utility by tailoring the network structure to the available private marginals. Recent work established statistical foundations for marginal-based private synthesis (Li, Wang, and Cheng 2023).

An alternative line of research models data distributions using Markov random fields (MRFs), which capture symmetric relationships among variables. These methods estimate noisy low-order marginals and reconstruct the global distribution using inference techniques such as Gibbs sampling (Chen et al. 2015; McKenna, Sheldon, and Miklau 2019; Cai et al. 2021). By leveraging local Markov properties, MRF-based approaches can represent complex dependencies while maintaining scalability and privacy.

3 Proposed Method

3.1 Problem Definition

We formalize the problem of synthesizing tabular data with DP. Suppose we have a random variable $X \in \mathcal{X} = \mathbb{R}^d$, whose distribution is P. We have n observations $\mathcal{D} = \{X_i\}_{i=1}^n$ from P. Our target is to learn a generator $g_{\boldsymbol{\nu}}$, possibly parameterized by $\boldsymbol{\nu}$, such that for some easy-to-generate random variables Z, such as Gaussian or uniform random variables, the distribution of $g_\nu(Z)$ is close to P. The learned generator should preserve the privacy of training data, in the sense of differential privacy defined as follows.

Definition 1 (Differential privacy (Dwork et al. 2006))**.** *A randomized algorithm $M : \mathcal{X}^n \to \mathcal{S}$ is (ε, δ)-differentially private ((ε, δ)-DP) if for every pair of adjacent data sets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ that differ by one datum and every $S \subseteq \mathcal{S}$, $\mathbb{P}(M(\mathbf{X}) \in S) \leq e^\varepsilon \cdot \mathbb{P}(M(\mathbf{X}') \in S) + \delta$, where the probability measure $\mathbb{P}$ is induced by the randomness of M only.*

In this work, we consider the parameter space of $\boldsymbol{\nu}$ to be $\mathcal{S}$. Last, we define some notations. For any vector x, let x^i denote the i-th element of x. We use the notation $a_n \lesssim b_n$ and $a_n \gtrsim b_n$ to denote that there exist positive constant n_1, c and c' such that $a_n \leq c b_n$ and $a_n \geq c' b_n$, for all $n \geq n_1$. In addition, we denote $a_n \asymp b_n$ if $a_n \lesssim b_n$ and $b_n \lesssim a_n$. Let $[n] = \{1, \ldots, n\}$. Let $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. Besides, for any set $A \subset \mathbb{R}^d$, the diameter of A is defined by $\mathrm{diam}(A) := \sup_{x,x' \in A} \|x - x'\|_2$. Let $\mathcal{W}(\mathrm{P}, \mathrm{Q})$ be the Wasserstein distance between distribution P and Q.

3.2 Generators and Bayes Network

We use an autoregressive approach to model the distribution. Specifically, let $\mathrm{P}[X]$ denote the joint distribution. Then the joint distribution can be decomposed into

$$\mathrm{P}[X] = \mathrm{P}[X^1] \cdot \prod_{j=2}^{d} \mathrm{P}[X^j | X^1, \ldots, X^{j-1}]. \tag{1}$$

To model (1), we use d sub-generators to model the condition distributions of $X^j, j = 1, \ldots, d$, respectively. Specifically, let $Z^j, j = 1, \ldots, d$ be some easy-to-general random variables, generated from Q. Then, the j-th generator

g^j wants to model

$$g^j(X^1, \ldots, X^{j-1}, Z^j)|X^{1:(j-1)} \sim X^j|X^{1:(j-1)}. \quad (2)$$

Then the integrated generator is

$$g(Z) = \left(g^1(Z^1), g^2(g^1(Z^1), Z^2), \ldots\right)^\top.$$

We adopt a similar assumption of Bayes network dependence as in (Zhang et al. 2017), which significantly improves the efficiency of data generation. A Bayes network over $\mathcal{X}$ provides a compact representation of the distribution by specifying conditional independencies among attributes in $\mathcal{X}$. Specifically, a Bayes network is a DAG that represents each attribute in $\mathcal{X}$ as a node and uses directed edges to model the conditional dependencies between attributes. The assumption is formally specified as follows.

Assumption 1. Assume that there exists a Bayes network $\mathcal{N} = \{(X^j, \Pi_j), j = 1, \ldots, d\}$, such that : (i) Π_j contains a subset of $[d]$, (ii) X^j is only dependent on Π_j, and (iii) $j \notin \Pi_i$ for $i < j$.

An example of a Bayes network is provided in Example 1 in the appendix of the extended version (Jia et al. 2025). Under Assumption 1, the autoregressive modeling (1) can be further simplified as

$$\mathrm{P}[X] = \mathrm{P}[X^1] \cdot \prod_{j=2}^{d} \mathrm{P}[X^j|\Pi_j] = \prod_{j=1}^{d} \mathrm{P}[X^j|\Pi_j], \quad (3)$$

where we let $\Pi_1 = \emptyset$. Under (3), we can reduce the estimation in (2) into estimating the conditional relationship $X^j|\Pi_j$, which would reduce the intrinsic dimensionality.Denote the parameter of each generator g^j by θ_j.

One may argue that Assumption (1) is too strong in practice. However, the marvelous performance of (Zhang et al. 2017) shows that the assumption is amenable since there exists a satisfiable $\mathcal{N}$ that captures most of the useful information in the conditional independence relationships at most of the times. Thus, an approximated $\mathcal{N}$ could be a nice surrogate to the true conditional relationship. Moreover, Rojas-Carulla et al. (2018); Zheng et al. (2018); Wang and Song (2025) yield that the relationship can be well approximated. If there exist additional public datasets, whether in distribution or out of distribution, that share the same network structure, one can approximate the network and shift the order of the variable to let it satisfies Assumption 1. The existence of such a similar public dataset is a common assumption in privacy-preserving machine learning (Yu et al. 2021; Ma and Yang 2024; Hod, Rosenblatt, and Stoyanovich 2025).

3.3 Private Generative Adversarial Network

To generate high-dimensional, complex data (e.g., images), recent work has explored privatizing generative adversarial networks (Goodfellow et al. 2014) to produce DP synthetic data via DPSGD (Abadi et al. 2016), a line of research known as DPGAN (Xie et al. 2018). The common framework for GAN is the mini-max optimization problem

$$\min_{g\in\mathcal{G}} \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{X\sim \mathrm{P}}\left[f(X)\right] - \mathbb{E}_{Z\sim \mathrm{Q}}\left[f(g(Z))\right] \right), \quad (4)$$

where $\mathcal{F}$ and $\mathcal{G}$ are the class of possible functions of discriminators and generators, and Z is sampled from Q. The solution of (4) is obtained approximately through the minimization of an empirical objective

$$\min_{g\in\mathcal{G}} \sup_{f\in\mathcal{F}} \left(\frac{1}{n}\sum_{i=1}^{n} f(X_i) - \frac{1}{n_g}\sum_{i=1}^{n_g} f(g(Z_i)) \right), \quad (5)$$

where an observation of n real samples $\{X_i\}_{i=1}^n$ and n_g easy-to-sample samples $\{Z_i\}_{i=1}^{n_g}$ are available. Denote the parameter of g and f by $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$, respectively. We denote the objective

$$\Delta(\boldsymbol{\theta}, \boldsymbol{\nu}, D) = \frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{\nu}}(X_i) - \frac{1}{n_g}\sum_{i=1}^{n_g} f_{\boldsymbol{\nu}}\left(g_{\boldsymbol{\theta}}(Z_i)\right). \quad (6)$$

Finding the exact solution of (5) is usually infeasible if the function class $\mathcal{F}$ and $\mathcal{G}$ are complex enough, as the optimization is usually non-convex. Thus, the optimization of (5) is conducted via iteratively minimizing w.r.t. g and maximizing w.r.t. f over the objective function. The update of $g_{\boldsymbol{\theta}}$ is done by stochastic gradient descent $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_\theta \nabla_{\boldsymbol{\theta}} \Delta(\boldsymbol{\theta}^t, \boldsymbol{\nu}^t, D)$. The maximization over f should be conducted under the constraint of DP. Specifically, the update of $f_{\boldsymbol{\nu}}$ is conducted with DPSGD (Abadi et al. 2016), denoted as $\boldsymbol{\nu}^{t+1} = \boldsymbol{\nu}^t + \eta_\nu \, \texttt{PrivGrad}(\nabla_{\boldsymbol{\nu}} \Delta(\boldsymbol{\theta}^t, \boldsymbol{\nu}^t, D), \sigma)$, where σ is the privacy noise level. Note that the minimization over g is independent of the real data and is thus free of privacy concerns. The optimization of $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$ conducted iteratively, meaning that one should update $\boldsymbol{\theta}^{t+1}$, use $\boldsymbol{\theta}^{t+1}$ to update $\boldsymbol{\nu}^{t+1}$, and continue this process. In practice, however, due to the performance drop brought by DPSGD during optimizing $f_{\boldsymbol{\nu}}$, the update of discriminator $f_{\boldsymbol{\nu}}$ should proceed multiple times before one update of the generator $g_{\boldsymbol{\theta}}$, as argued by Bie, Kamath, and Singhal (2022). Thus, we introduce an additional parameter t_g to account for this relative number of iterations.

3.4 Adaptive Feature Selection

The determination of Π_j is tricky. Zhang et al. (2017) utilized a private version of a surrogate of mutual information to determine the network. This approach is, however, highly restrictive to the choice of the degree of the networks, as illustrated in Section 1. We propose a data-driven feature selection rule that leverages a sparsity-inducing penalty. Specifically, we consider $\boldsymbol{\theta}_j = (\boldsymbol{\xi}_j, \mathbf{W}_j)$, where $\mathbf{W}_j$ represents the linear feature map from j dimensional space onto an arbitrary dimensional, say L_j, space, and $\boldsymbol{\nu}_j$ represents the parameters governing the map from this feature to the output. This is also known as single index models in the statistics community (Xia 2008). This leads to

$$g^j(X^1, \ldots, X^{j-1}, Z^j) = \tilde{g}^j_{\boldsymbol{\nu}_j}\left(\boldsymbol{W}_j^\top \left[X^{1:j-1}, Z_j\right]\right). \quad (7)$$

Here, $\boldsymbol{W}_j$ is a $\mathbb{R}^{j\times L_j}$ matrix. Model (7) includes a large class of functions, including neural networks. To induce sparsity, we penalize the sum of L_2 norm of weights associated to each feature following (Feng and Simon 2017; Dinh and Ho

2020; Wang, Huang, and Ma 2024), formally

$$L(\boldsymbol{W}_j) = \sum_{k=1}^{j-1} \|\boldsymbol{W}_j^{k,:}\|_2.$$

We refer to this penalty as a group lasso penalty, in the sense that it penalizes the entire set of weights corresponding to a feature (Yuan and Lin 2006; Friedman, Hastie, and Tibshirani 2010; Simon et al. 2013). Unlike a standard L_2 penalty, it has a composite structure: an L_2 norm is applied within each group (a row of $\boldsymbol{W}_j$), followed by an L_1-type aggregation across groups (the collection of row norms). This structure enables all weights associated with a single feature to be shrunk to zero simultaneously, thereby inducing sparsity at the feature level. Thus, we optimize a penalized version of objective function (6)

$$\begin{aligned}&\tilde{\Delta}(\boldsymbol{\theta}, \boldsymbol{\nu}, D) \qquad (8)\\ =&\frac{1}{n}\sum_{i=1}^{n} f(X_i) - \frac{1}{n_g}\sum_{i=1}^{n_g} f(g(Z_i)) + \sum_{j=1}^{d} \lambda_j L(\boldsymbol{W}_j).\end{aligned}$$

Here, $\lambda_1, \ldots, \lambda_d$ are pre-determined tuning parameters. The overall optimization process is illustrated in Algorithm 1.

Algorithm 1: PrAda-GAN

Input: Private data $D = \{X_i\}_{i=1}^n$.
Parameters: Learning rate $\eta_{\boldsymbol{\theta}}, \eta_{\boldsymbol{\nu}}$, iteration number T, relative number of iterations t_g, threshold τ, regularizations $\lambda_1, \ldots, \lambda_d$, noise level σ.
Initialization: Initial parameters $\boldsymbol{\theta}^1$, $\boldsymbol{\nu}^1$.
for t *in* $[T]$ **do**
  # Update ν every step.
  $\boldsymbol{\nu}^{t+1} = \boldsymbol{\nu}^t + \eta_\nu \cdot$ PrivGrad$(\nabla_{\boldsymbol{\nu}} \tilde{\Delta}(\boldsymbol{\theta}^t, \boldsymbol{\nu}^t, D), \sigma)$
  # Update θ every t_g steps.
  $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_\theta \cdot \mathbf{1}\,(t \mod t_g \equiv 0) \cdot \nabla_{\boldsymbol{\theta}} \tilde{\Delta}(\boldsymbol{\theta}^t, \boldsymbol{\nu}^{t+1}, D)$.
Output: Generator $g_{\boldsymbol{\theta}^{T+1}}$ and discriminator $f_{\boldsymbol{\nu}^{T+1}}$.

4 Theoretical Guarantees

4.1 Assumptions

Denote the assumed function classes for generators and discriminators as $\mathcal{G}$ and $\mathcal{F}$.

Assumption 2 (Effective Private Optimization). The output of Algorithm 1 achieves an optimization result of

$$\tilde{\Delta}(\boldsymbol{\theta}^{T+1}, \boldsymbol{\nu}^{T+1}, D) \le \inf_{\boldsymbol{\theta}\in\mathcal{G}} \sup_{\boldsymbol{\nu}\in\mathcal{F}} \tilde{\Delta}(\boldsymbol{\theta}, \boldsymbol{\nu}, D) + \mathcal{E} \qquad (9)$$

for some $\mathcal{E} > 0$. Suppose $\mathcal{E}$ is diminishing as $n \to \infty$.

Assumption 2 is present to control the performance drop brought by the inexact optimization of (8) using DPSGD. There are mainly two error sources in $\mathcal{E}$, guaranteed respectively by

$$\tilde{\Delta}(\boldsymbol{\theta}^{T+1}, \boldsymbol{\nu}^{T+1}, D) \le \tilde{\Delta}(\boldsymbol{\theta}_{np}, \boldsymbol{\nu}_{np}, D) + \mathcal{E}_{priv} \qquad (10)$$

and

$$\tilde{\Delta}(\boldsymbol{\theta}_{np}, \boldsymbol{\nu}_{np}, D) \le \inf_{\boldsymbol{\theta}} \sup_{\boldsymbol{\nu}} \tilde{\Delta}(\boldsymbol{\theta}, \boldsymbol{\nu}, D) + \mathcal{E}_{opt}, \qquad (11)$$

and thus $\mathcal{E} = \mathcal{E}_{priv} + \mathcal{E}_{opt}$ satisfies (2). Specifically, the presence of (10) is due to the weakened optimization performance of DPSGD compared to SGD. Namely, each $\boldsymbol{\nu}^t$ achieves (in expectation) a larger $\tilde{\Delta}$ than $\boldsymbol{\nu}_{np}^t$, and thus has a smaller discriminative capability. Thus, perceiving information from a weaker discriminator, the generator $g_{\boldsymbol{\theta}}$ is, in general, less effective. Other approaches to mitigate this issue include a more powerful remedy of DPSGD (Bu et al. 2023; Liu and Bu 2025), longer discriminator training (Bie, Kamath, and Zhang 2023), or additional information assistance (Yu et al. 2021, 2022). In general, $\mathcal{E}_{priv}$ should be smaller when ε is large. For (11), error $\mathcal{E}_{opt}$ represents the gap between optimization SGD and the exact solution, and is often referred to as optimization error. Further verification of the assumption and establishment of bounds on $\mathcal{E}$ should refer to the literature in training dynamics of GANs (Biau et al. 2020; Xu et al. 2020; Huang et al. 2022) and differentially private stochastic optimization (Bassily, Guzmán, and Menart 2021; Su, Hu, and Wang 2024).

Assumption 3 (Finite Moment). Let $X \sim \mathrm{P}$. For some $c_m > 0$, X satisfies the first moment tail condition $\mathbb{E}\left[\|X\|\mathbf{1}\left(\|X\| > \log t\right)\right] = O\left(t^{-(\log t)^{c_1}/d}\right)$, for any $t \ge 1$.

Assumption 4 (Analytic Generator). The function $g_{\boldsymbol{\theta}}$ is analytic with respect to $\boldsymbol{\theta}$.

Assumption 3 requires P is concentrated, and is commonly satisfied, for instance by sub-Gaussian variables. Assumption 4 is easily satisfied with several choices of activation functions, including the classic ones such as the linear function, tanh function, and sigmoid function, as well as the newly developed ReLU-type activation functions such as GeLU, ELU, and PELU.

4.2 Parameter Estimation and Feature Selection

In this section, we demonstrate that under the assumptions outlined in the previous section, the generator obtained from Algorithm 1 exhibits superiority in both parameter estimation and feature selection. We first define the relevant criteria for evaluation. For parameter estimation, we consider the quantity of $d\left(\boldsymbol{\theta}^{T+1}, \Theta\right)$ defined as follows. Let

$$\Theta = \left\{\boldsymbol{\theta} \in \mathcal{G} \mid g_{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}} \mathcal{W}\left(\mathrm{P}_{g_{\boldsymbol{\theta}}(Z)}, \mathrm{P}_X\right)\right\}, \qquad (12)$$

denote the set of feasible parameters that achieve the minimum Wasserstein distance within $\mathcal{G}$. Then,

$$d\left(\boldsymbol{\theta}^{T+1}, \Theta\right) = \min_{\boldsymbol{\theta}\in\Theta} d(\boldsymbol{\theta}^{T+1}, \boldsymbol{\theta}),$$

represents the closest distance between $\boldsymbol{\theta}^{T+1}$ and any element in Θ. For feature selection, we evaluate the sum of norms associated with unimportant features (i.e., those outside Π_j): $\sum_{k\in\Pi_j^c} \left\|\left(\boldsymbol{W}_j^{T+1}\right)^k\right\|_2$. Both quantities are expected to be small. Define $\mathcal{F}_{Lip,1}$ be the 1-Lipschitz class on $\mathbb{R}^d$. Given these definitions, we have the following theorem.

Theorem 1. *Suppose that Assumptions 1, 2, 3, and 4 hold. Suppose that $\lambda_j = d^{-1}\psi_{n,d}^{1/2}, j = 1, \ldots, d$, where we define*

$$\psi_{n,d} = (\sqrt{d} + \log n)n^{-\frac{1}{d}} + \mathcal{E} + \sup_{f \in \mathcal{F}_{Lip,1}} \inf_{f' \in \mathcal{F}} \|f - f'\|_\infty . \quad (13)$$

Then, the fitted parameters $\boldsymbol{\theta}^{T+1}$ and $\boldsymbol{W}^T \in \boldsymbol{\theta}^{T+1}$ from Algorithm 1 (without comment) satisfy

$$\mathbb{E}[\sum_{k \in \Pi_j^c} \|(\boldsymbol{W}_j^{T+1})^k\|_2] \lesssim d \cdot \psi_{n,d}^{\frac{1}{2(a-1)}}, \quad j \in [d], \quad (14)$$

as well as

$$\mathbb{E}\left[d\left(\boldsymbol{\theta}^{T+1}, \Theta\right)\right] \lesssim \psi_{n,d}^{\frac{1}{2(a-1)}}. \quad (15)$$

Here, $a > 2$ is a positive constant. The expectation $\mathbb{E}$ is taken w.r.t. training samples $\{X_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^n$.

The theorem are interpreted as follows. Both bounds in (15) and (14) depend on $\psi_{n,d}$, which is expected to diminish. This quantity consists of three components: the estimation error, the approximation error, and the private optimization error. The estimation error arises due to the discrepancy between the empirical and population objective function, which decreases with larger n and smaller d. The approximation error measures how well the discriminator function class $\mathcal{F}$ approximates the 1-Lipschitz class, which should be smaller for $\mathcal{F}$ that is more expressive. The private optimization error reflects the gap between privately optimized and globally optimal objective functions. See also Assumption 2 and comments below. Using developed tools (Abadi et al. 2016; Bu et al. 2023; Liu and Bu 2025), one should expect this term to diminish with larger n, larger ε, and smaller d, yet increase when the function classes $\mathcal{G}$ and $\mathcal{F}$ are most complex, i.e., harder to optimize. Thus, when n, ε, and d are fixed, the choice of function classes and optimizers should balance the approximation and optimization errors.

Combining the three terms together, Theorem 1 suggests that when $n \to \infty$, and d, ε both remain moderately small, $\psi_{n,d}$ would diminish. Consequently, the obtained $\boldsymbol{\theta}^{T+1}$ not only approaches the optimal parameter set Θ, but also converges to a well-behaved class with zero dependence on redundant variables. Theorem 1 applies to general function classes $\mathcal{G}$ and $\mathcal{F}$. Specific convergence rates for neural networks can be derived as in Dinh and Ho (2020); Wang, Huang, and Ma (2024). Also, for the rate $n^{-\frac{1}{d}}$ to diminish, we require $d = o(\log n)$. Here, we implicitly assume $d = O(\log^{c_1} n)$ for $0 < c_d < 1$.

4.3 Convergence Leveraging Sparsity

One should note that the closeness of $d\left(\boldsymbol{\theta}^{T+1}, \Theta\right)$ does not imply any convergence result for $\mathcal{W}\left(\mathrm{P}_{g_{\boldsymbol{\theta}}(Z)}, \mathrm{P}_X\right)$. Moreover, the convergence in (15) does not reflect the improvement afforded by sparsity (i.e., Assumption 1). In this section, we propose a variant of Algorithm 1 by incorporating weight thresholding. Specifically, we exclude variables with small parameter norms by setting their associated weights to zero. See the commented code in lines 9–18 of Algorithm 2 in the appendix of the extended version (Jia et al. 2025). By leveraging this approach, we establish a convergence rate that depends on the underlying sparsity structure $|\Pi_j|, j \in [d]$, rather than the full dimensionality.

For the theoretical analysis, we introduce an additional assumption regarding the second-phase optimization, analogous to Assumption 2. Let $\overline{\mathcal{G}}$ denote the class of functions where weights $\boldsymbol{W}$ are set to zero if $\left\|(\boldsymbol{W}_j^{T+1})^{k,:}\right\|_2 \leq \tau$. Note that this class is data-dependent.

Assumption 5 (Constraint Effectiveness). The commented output of Algorithm 1, i.e. $\overline{\boldsymbol{\nu}}^{T+1}$ and $\overline{\boldsymbol{\theta}}^{T+1}$ achieves an optimization result of

$$\Delta(\overline{\boldsymbol{\theta}}^{T+1}, \overline{\boldsymbol{\nu}}^{T+1}, D) \leq \inf_{\boldsymbol{\theta} \in \overline{\mathcal{G}}} \sup_{\boldsymbol{\nu} \in \mathcal{F}} \Delta(\boldsymbol{\theta}, \boldsymbol{\nu}, D) + \overline{\mathcal{E}} \quad (16)$$

for some $\overline{\mathcal{E}} > 0$. Suppose $\overline{\mathcal{E}}$ is diminishing as $n \to \infty$.

This assumption ensures satisfactory optimization performance over the post-feature-selection function class $\overline{\mathcal{G}}$ and is no more restrictive than Assumption 2. In practice, however, one may prefer to run Algorithm 1 without explicitly applying the thresholding step (as commented in the code), provided the influence of unimportant variables remains negligible. However, if the true dependence structure is highly sparse, the improvements brought by this feature selection could be significant, e.g., (Ma, Jia, and Yang 2024; Kent, Berrett, and Yu 2024). Notably, this guarantee holds even if we simply clip all weights $\left\|(\boldsymbol{W}_j^{T+1})^{k,:}\right\|_2 \leq \tau$ to zero without re-optimization. Let $g_{\boldsymbol{\theta}_{clip}^{T+1}}$ denote the resulting clipped generator . Then $g_{\boldsymbol{\theta}_{clip}^{T+1}} \in \overline{\mathcal{G}}$ and under sufficiently small penalization parameters λ_j, $g_{\boldsymbol{\theta}_{clip}^{T+1}}$ satisfies Assumption 5.

The following theorem establishes a distance convergence guarantee for the commented output $\overline{\boldsymbol{\theta}}^{T+1}$ of Algorithm 1. Define the ancestor set J_∞^j of feature j by $J_0^j = \{j\}$, $J_{k+1}^j = \bigcup_{j \in J_k^j} \Pi_j$, and $J_\infty^j = \lim_{k \to \infty} J_k^j$.

Theorem 2. *Suppose that the assumptions in Theorem 1 and Assumption 5 hold. Suppose we set $\lambda_j = d^{-1}\psi_{n,s}^{1/2}, j = 1, \ldots, d$, where we define*

$$\psi_{n,s} = (\sqrt{d} + \log n)\, n^{-\frac{1}{s}} + \overline{\mathcal{E}} + \sup_{f \in \mathcal{F}_{Lip,1}} \inf_{f' \in \mathcal{F}} \|f - f'\|_\infty . \quad (17)$$

Here we have $s = \max_{j \in [d]} J_\infty^j$. Then, with probability 1, there exists a suitable choice of τ such that the fitted parameters $\overline{\boldsymbol{\theta}}^{T+1}$ and $\overline{\boldsymbol{W}}^{T+1} \in \overline{\boldsymbol{\theta}}^{T+1}$ from Algorithm 1 (with comment) satisfy

$$\mathbb{E}\left[\mathcal{W}\left(\mathrm{P}_{g_{\overline{\boldsymbol{\theta}}^{T+1}}(Z)}, \mathrm{P}_X\right)\right] \lesssim \psi_{n,s}. \quad (18)$$

The expectation $\mathbb{E}$ is taken w.r.t. $\{X_i\}_{i=1}^n$ and $\{Z_i\}_{i=1}^n$.

The theorem illustrates the generative performance of the obtained parameter $\overline{\boldsymbol{\theta}}^{T+1}$. Note that $\psi_{n,s}$ in (17) consists of three parts analogous to those in (13), but the estimation error is significantly improved from $-1/d$ to $-1/s$. In

many cases, features can be divided into several independent groups, allowing s to be much smaller than d, illustrating the improvement brought by the Bayes network structure.

The relationship between $\overline{\mathcal{E}}$ and $\mathcal{E}$ is obvious - intuitively, the expectation of $\overline{\mathcal{E}}$ is always no larger than $\mathcal{E}$. This is because $\mathcal{E}$ and $\overline{\mathcal{E}}$ only account for optimization error, and the constraint optimization problem $\inf_{\boldsymbol{\theta}\in\overline{\mathcal{G}}}\sup_{\boldsymbol{\nu}\in\mathcal{F}}\Delta(\boldsymbol{\theta},\boldsymbol{\nu},D)$ is easier than $\inf_{\boldsymbol{\theta}\in\mathcal{G}}\sup_{\boldsymbol{\nu}\in\mathcal{F}}\Delta(\boldsymbol{\theta},\boldsymbol{\nu},D)$. However, explicitly depicting such improvement again yields an involved analysis of the dynamics of GAN.

5 Experiment

In the experiments, we perform both synthetic and real-world experiments in Section 5.1 and 5.2. More details on datasets and settings can be found in Appendix D of the extended version (Jia et al. 2025).

Privacy Analysis We adopt DP-SGD as optimizer and evaluate its privacy guarantees through the Rényi differential privacy (RDP) accountant (Mironov 2017) implemented in Opacus. For each experiment, we specify a target privacy budget ε and select the noise multiplier σ and the total number of training rounds T so that the privacy loss computed by the RDP accountant matches the desired ε (with δ fixed as described in Appendix D of Jia et al. (2025)). The RDP framework provides tight privacy tracking for iterative gradient updates, which makes it well suited for DP-SGD. Through this procedure, all reported results are (ε,δ)-DP.

Evaluation Metrics (1) **Distribution Similarity**: We evaluate the generator's ability to capture distribution similarity by computing the total variation distance (TVD), computed via 2-way marginals, and the Wasserstein distance (WD). See more details in Appendix D of the extended version (Jia et al. 2025). (2) **Machine Learning Efficacy**: For real datasets, we assess the quality of synthetic data through downstream machine learning performance. We employ five robust machine learning models known for their strong generalization: MLP, CatBoost (Prokhorenkova et al. 2018), XGBoost (Chen and Guestrin 2016), Random Forest, and SVM. Performance is evaluated using the average R^2 score and root mean square error (RMSE) on held-out test data, with respect to a model trained on synthetic data.

5.1 Synthetic Experiments

Experiment Setup We generate synthetic data according to the structural equation model $X_j = f_j(\Pi_j) + z_j, j \in [d]$, where $z_j \sim \mathcal{N}(0,1)$, are independent noise terms and recall that Π_j denotes the parent set of X_j in the underlying Bayes network. Each function f_j is modeled using a multiple index model, represented as a sum of nonlinear transformations applied to one-dimensional linear projections of the parent variables Π_j. This flexible form ensures expressiveness while maintaining identifiability under mild regularity conditions. The ground truth Bayes networks are generated from the Erdős-Rényi graph model (Zheng et al. 2018), with a fixed number of nodes $d = 10$. For each generated network, we simulate $n = 2000$ training samples. Parallel sets of experiments using linear functions f_j and alternative evaluation metrics are presented in Appendix E of the extended version (Jia et al. 2025).

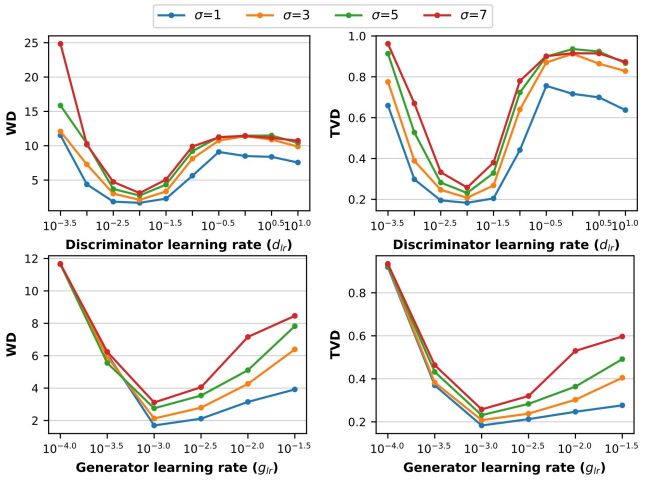


Figure 2: Average WD and TVD. Top: varying discriminator learning rates d_{lr} across 10^h $g_{\text{lr}} = 10^{-2}$. Bottom: varying generator learning rates g_{lr} across 10^h with $d_{\text{lr}} = 10^{-1}$.

Parameter Analysis of Learning Rate We investigate the impact of the discriminator and the generator learning rates, denoted by d_{lr} and g_{lr}. We use the nonlinear function $f_j(\Pi_j) = \tanh(w_{1j}^\top\Pi_j)+\cos(w_{2j}^\top\Pi_j)+\sin(w_{3j}^\top\Pi_j)$, where $w_{kj}^\top\Pi_j = \sum_{j'\in\Pi_j} w_{k,j,j'}X^{j'}$. Each weight $w_{k,j,j'}$ is randomly initialized by sampling from a uniform distribution over $[0.5, 2.0]$, and independently negated with probability 0.5. We vary d_{lr} while fixing g_{lr}, and vice versa, repeating each setting 20 times. Performance is evaluated under privacy noise levels $\sigma \in \{1,3,5,7\}$. As shown in Figure 2, for each σ, as d_{lr} increases, both WD and TVD initially decrease, reaching a minimum at a certain value of d_{lr}, and then begin to increase. A similar trend is observed when varying g_{lr}. These observations indicate the existence of an optimal learning rate that primarily governs the optimization error $\mathcal{E}$.

Parameter Analysis of Penalty Parameters Using the same functions f_j, we investigate the effects of the penalty parameters λ_j for $j \in [d]$. To reduce the complexity of the search, we parameterize the penalties as $\lambda_j = \lambda \cdot j^\gamma$ for $j = 1,\ldots,d$, and focus on varying the global parameters λ and γ. As shown in the top panel of Figure 3, there exists an optimal value of λ for each noise level σ, and this optimal value increases as σ becomes larger. This observation aligns with the theory, where the optimal choice of λ_j in equations (13) and (17) increases with the optimization error, which itself grows with larger noise levels σ. In the bottom panel of Figure 3, we observe that in most cases, setting $\gamma = 0$ yields a sufficiently low error, indicating that the magnitude of the penalty need not depend on the number of preconditioned features. This phenomenon is also consistent with Theorems 1 and 2.

Method	ε	California					House-16H					Cpu-act				
		Cat	MLP	RF	XGB	SVM	Cat	MLP	RF	XGB	SVM	Cat	MLP	RF	XGB	SVM
PrAda-GAN	0.2	**0.312***	**0.247***	**0.285***	**0.272***	**0.226***	**0.139***	**0.088***	**0.086**	**0.083***	**0.105***	**0.133***	**-0.010**	**0.098***	**0.095**	0.017
AIM		0.015	-0.004	-0.190	-0.012	-0.001	-0.039	-0.045	-0.071	-0.186	-0.002	0.003	-0.166	0.039	-0.037	-0.078
PrivMRF		0.003	0.010	-0.238	-0.019	-0.015	-0.009	-0.009	-0.020	-0.066	-0.002	-0.001	-0.208	0.044	-0.039	-0.058
GEM		-0.152	-0.138	-0.471	-0.234	-0.260	-0.307	-0.679	-0.686	-1.281	-0.305	-0.327	-0.457	-0.955	-0.489	-0.371
DP-MERF		0.201	-0.166	-0.082	0.126	0.047	0.066	-0.652	-1.213	-0.944	-0.123	0.111	-0.066	0.002	0.042	**0.134**
PrivBayes		0.024	-0.040	-0.088	-0.098	-0.030	-0.007	-0.314	-0.083	-0.237	-0.026	-0.142	-1.289	-2.779	-0.877	-0.004
PrAda-GAN	1.0	**0.495***	**0.480***	**0.434***	**0.443***	**0.427***	**0.240***	**0.185***	**0.198***	**0.196***	**0.160**	0.127	**0.073***	**0.103**	0.106*	**0.096**
AIM		0.005	0.017	-0.212	-0.038	-0.009	0.157	0.124	0.076	-0.049	0.145	0.042	-0.117	0.101	0.012	-0.054
PrivMRF		0.003	-0.009	-0.212	-0.022	-0.013	0.0004	-0.004	-0.028	-0.058	-0.002	0.024	-0.195	0.075	-0.003	-0.071
GEM		-0.041	-0.079	-0.345	-0.087	-0.070	-0.093	-0.181	-0.248	-0.421	-0.041	0.102	-0.157	-0.107	0.027	0.038
DP-MERF		0.290	-0.220	-0.037	0.254	0.157	0.126	-0.534	-0.472	-0.411	0.065	**0.149***	-0.003	0.085	**0.112**	0.090
PrivBayes		0.019	-0.017	-0.075	-0.088	-0.005	0.005	-0.450	-0.003	-0.114	-0.018	0.013	-0.717	-3.596	-0.505	-0.033
PrAda-GAN	5.0	0.615	**0.607***	**0.577**	0.580	**0.587***	0.261	0.205	0.225	0.221	**0.218**	0.171	**0.158***	**0.161**	**0.167**	0.090
AIM		**0.642***	0.489	0.567	**0.583**	0.527	**0.343***	0.184	**0.320***	**0.277***	0.207	**0.212***	-0.005	0.151	0.165	**0.150***
PrivMRF		0.057	-0.046	-0.166	0.010	0.107	0.317	**0.232***	0.305	0.185	0.197	0.037	-0.246	0.079	0.006	-0.074
GEM		-0.029	-0.049	-0.343	-0.093	-0.046	-0.042	-0.102	-0.124	-0.364	-0.006	0.179	-0.023	0.112	0.145	0.067
DP-MERF		0.298	-0.078	0.173	0.300	0.021	0.112	-0.573	-0.044	-0.356	0.088	0.174	-0.327	0.148	0.143	0.088
PrivBayes		0.459	0.398	0.419	0.405	0.469	0.014	-0.708	-0.011	-0.093	-0.021	0.020	-0.497	-3.620	-0.408	-0.046
Ground Truth		0.856	0.809	0.812	0.818	0.657	0.548	0.524	0.540	0.510	0.309	0.979	0.952	0.977	0.968	0.203

Table 1: Real data comparison for downstream machine learning efficacy (R^2). Best results are in **bold**; second-best are underlined. The best results that hold significance over the others have a *.

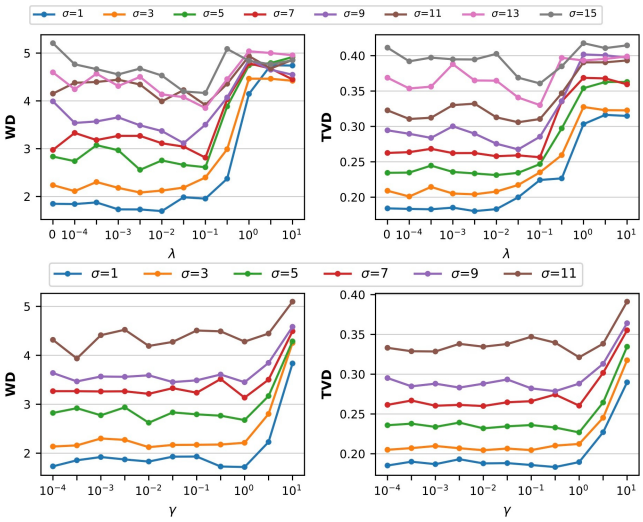


Figure 3: Average WD and TVD under varying λ and γ.

5.2 Real Data Comparison

Experiment Setup We compare PrAda-GAN with Bayesian network–based methods (AIM (McKenna et al. 2022), PrivMRF (Cai et al. 2021), PrivBayes (Zhang et al. 2017)) and deep learning approaches for differentially private synthetic data generation (GEM (Liu, Vietri, and Wu 2021), DP-MERF (Harder, Adamczewski, and Park 2021)). Experiments are conducted on three continuous OpenML datasets (Vanschoren et al. 2013): California, House-16H, and Cpu-act, with dataset statistics reported in the appendix (Jia et al. 2025). Each experiment is repeated 10 times with randomness arising from initialization, data partitioning, and shuffling. In each trial, data are randomly split 6:4 into training and testing sets, and hyperparameters are predetermined using a validation set from the training data.

Distribution Similarity Representative results for privacy budgets $\varepsilon = 0.2, 1.0$, and 5.0 are shown in Table 1 of Jia et al. (2025), with additional metrics reported in the appendix. As shown in the table, PrAda-GAN achieves superior performance over competing methods in most scenarios. Among the baseline, AIM and PrivMRF emerge as the strongest competitors, while other deep learning methods fail to achieve comparable performance.

Machine Learning Efficacy Machine learning performance (R^2 scores) is reported in Table 1, with RMSE results in the appendix of Jia et al. (2025). PrAda-GAN consistently achieves higher utility across all scenarios, and demonstrates substantial advantages over the competitors under low privacy budgets, i.e., $\varepsilon = 0.2$ and 1.0.

6 Conclusion

This paper addresses the limitations of existing methods for differentially private tabular data synthesis by introducing a novel generative framework, PrAda-GAN. We provide theoretical analysis of PrAda-GAN, including a bound on the distance to the optimal generator and a generalization bound for the Wasserstein distance between the generated and true data distributions. Our results highlight the role of adaptive regularization in improving convergence. Empirical results on real-world datasets demonstrate that PrAda-GAN consistently outperforms existing methods while maintaining rigorous privacy guarantees.

Acknowledgments

The authors would like to thank the SPC and the reviewers for their constructive comments and recognition of this work. This work is supported by Beijing Social Science Fund (24GLC033).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *CCS*.

Andrey, P.; Bars, B. L.; and Tommasi, M. 2025. TAMIS: Tailored Membership Inference Attacks on Synthetic Data. arXiv:2504.00758.

Bassily, R.; Guzmán, C.; and Menart, M. 2021. Differentially private stochastic optimization: New results in convex and non-convex settings. *NeurIPS*.

Biau, G.; Cadre, B.; Sangnier, M.; and Tanielian, U. 2020. SOME THEORETICAL PROPERTIES OF GANS. *The Annals of Statistics*.

Bie, A.; Kamath, G.; and Singhal, V. 2022. Private Estimation with Public Data. In *NeurIPS*.

Bie, A.; Kamath, G.; and Zhang, G. 2023. Private gans, revisited. *arXiv preprint arXiv:2302.02936*.

Bu, Z.; Wang, Y.-X.; Zha, S.; and Karypis, G. 2023. Automatic clipping: Differentially private deep learning made easier and stronger. *NeurIPS*.

Cai, K.; Lei, X.; Wei, J.; and Xiao, X. 2021. Data synthesis via differentially private markov random fields. *PVLDB*.

Chen, D.; Orekondy, T.; and Fritz, M. 2020. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *NeurIPS*.

Chen, K.; Li, X.; GONG, C.; McKenna, R.; and Wang, T. 2025. Benchmarking Differentially Private Tabular Data Synthesis Algorithms. In *SynthData Workshop, ICLR*.

Chen, R.; Xiao, Q.; Zhang, Y.; and Xu, J. 2015. Differentially private high-dimensional data publication via sampling-based inference. In *KDD*.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *KDD*.

Dinh, V. C.; and Ho, L. S. 2020. Consistent feature selection for analytic deep neural networks. *NeurIPS*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*.

Feng, J.; and Simon, N. 2017. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2010. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

Gadre, S. Y.; Ilharco, G.; Fang, A.; Hayase, J.; Smyrnis, G.; Nguyen, T.; Marten, R.; Wortsman, M.; Ghosh, D.; Zhang, J.; et al. 2023. Datacomp: In search of the next generation of multimodal datasets. *NeurIPS*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *NeurIPS*.

Harder, F.; Adamczewski, K.; and Park, M. 2021. Dp-merf: Differentially private mean embeddings with random-features for practical privacy-preserving data generation. In *AISTATS*.

Hod, S.; Rosenblatt, L.; and Stoyanovich, J. 2025. Do You Really Need Public Data? Surrogate Public Data for Differential Privacy on Tabular Data. arXiv:2504.14368.

Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S. B.; Schirrmeister, R. T.; and Hutter, F. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*.

Hu, Y.; Wu, F.; Li, Q.; Long, Y.; Garrido, G. M.; Ge, C.; Ding, B.; Forsyth, D.; Li, B.; and Song, D. 2024. Sok: Privacy-preserving data synthesis. In *IEEE S&P*.

Huang, J.; Jiao, Y.; Li, Z.; Liu, S.; Wang, Y.; and Yang, Y. 2022. An error analysis of generative adversarial networks for learning distributions. *JMLR*.

Jia, K.; Ma, Y.; Li, Y.; and Wang, F. 2025. PrAda-GAN: A Private Adaptive Generative Adversarial Network with Bayes Network Structure. arXiv:2511.07997.

Jordon, J.; Yoon, J.; and Van Der Schaar, M. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*.

Kent, A.; Berrett, T. B.; and Yu, Y. 2024. Rate Optimality and Phase Transition for User-Level Local Differential Privacy. *arXiv preprint arXiv:2405.11923*.

Li, X.; Wang, C.; and Cheng, G. 2023. Statistical theory of differentially private marginal-based data synthesis algorithms. *arXiv preprint arXiv:2301.08844*.

Liu, R.; and Bu, Z. 2025. Towards hyperparameter-free optimization with differential privacy. In *ICLR*.

Liu, T.; Vietri, G.; and Wu, S. Z. 2021. Iterative methods for private synthetic data: Unifying framework and new methods. *NeurIPS*.

Liu, Y.; Peng, J.; James, J.; and Wu, Y. 2019. PPGAN: Privacy-preserving generative adversarial network. In *ICPADS*.

Long, Y.; Wang, B.; Yang, Z.; Kailkhura, B.; Zhang, A.; Gunter, C.; and Li, B. 2021. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. *NeurIPS*.

Lu, Y.; Shen, M.; Wang, H.; Wang, X.; van Rechem, C.; Fu, T.; and Wei, W. 2023. Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*.

Ma, C.; Li, J.; Ding, M.; Liu, B.; Wei, K.; Weng, J.; and Poor, H. V. 2023. RDP-GAN: A Rényi-differential privacy based generative adversarial network. *TDSC*.

Ma, Y.; Jia, K.; and Yang, H. 2024. Better locally private sparse estimation given multiple samples per user. In *Proceedings of the 41st ICML*.

Ma, Y.; and Yang, H. 2024. Optimal Locally Private Nonparametric Classification with Public Data. *JMLR*.

McKenna, R.; Mullins, B.; Sheldon, D.; and Miklau, G. 2022. AIM: an adaptive and iterative mechanism for differentially private synthetic data. *PVLDB*.

McKenna, R.; Sheldon, D.; and Miklau, G. 2019. Graphical-model based estimation and inference for differential privacy. In *ICML*.

Mironov, I. 2017. Rényi differential privacy. In *CSF*.

NIST. 2019. Differential Privacy Synthetic Data Challenge.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. *NeurIPS*.

Rojas-Carulla, M.; Schölkopf, B.; Turner, R.; and Peters, J. 2018. Invariant models for causal transfer learning. *JMLR*.

Simon, N.; Friedman, J.; Hastie, T.; and Tibshirani, R. 2013. A sparse-group lasso. *Journal of computational and graphical statistics*.

Su, J.; Hu, L.; and Wang, D. 2024. Faster rates of differentially private stochastic convex optimization. *JMLR*.

Sun, H.; Zhu, T.; Zhang, Z.; Jin, D.; Xiong, P.; and Zhou, W. 2021. Adversarial attacks against deep generative models on data: A survey. *TKDE*.

Tao, Y.; McKenna, R.; Hay, M.; Machanavajjhala, A.; and Miklau, G. 2021. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*.

Torkzadehmahani, R.; Kairouz, P.; and Paten, B. 2019. Dp-cgan: Differentially private synthetic data and label generation. In *CVPR Workshops*.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*.

Wang, J.; and Song, R. 2025. Dynamic Causal Structure Discovery and Causal Effect Estimation. *arXiv preprint arXiv:2501.06534*.

Wang, T.; Huang, J.; and Ma, S. 2024. Penalized Generative Variable Selection. *arXiv preprint arXiv:2402.16661*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Xia, Y. 2008. A multiple-index model and dimension reduction. *Journal of the American Statistical Association*.

Xie, L.; Lin, K.; Wang, S.; Wang, F.; and Zhou, J. 2018. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*.

Xu, K.; Li, C.; Zhu, J.; and Zhang, B. 2020. Understanding and stabilizing GANs' training dynamics using control theory. In *ICML*.

Yang, M.; Chi, C.-H.; Lam, K.-Y.; Feng, J.; Guo, T.; and Ni, W. 2024. Tabular data synthesis with differential privacy: A survey. *arXiv preprint arXiv:2411.03351*.

Yu, D.; Naik, S.; Backurs, A.; Gopi, S.; Inan, H. A.; Kamath, G.; Kulkarni, J.; Lee, Y. T.; Manoel, A.; Wutschitz, L.; Yekhanin, S.; and Zhang, H. 2022. Differentially Private Fine-tuning of Language Models. In *The Tenth ICLR*.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2021. Large scale private learning via low-rank reparametrization. In *ICML*.

Yuan, M.; and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2017. Privbayes: Private data release via bayesian networks. *TODS*.

Zhang, Q.; Tan, Y. S.; Tian, Q.; and Li, P. 2025. TabPFN: One Model to Rule Them All? *arXiv preprint arXiv:2505.20003*.

Zhang, X.; Ji, S.; and Wang, T. 2018. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*.

Zhao, Z.; Kunar, A.; Birke, R.; Van der Scheer, H.; and Chen, L. Y. 2024. Ctab-gan+: Enhancing tabular data synthesis. *Frontiers in big Data*.

Zheng, X.; Aragam, B.; Ravikumar, P. K.; and Xing, E. P. 2018. Dags with no tears: Continuous optimization for structure learning. *NeurIPS*.