Frugal Generative Modeling for Tabular Data

Alice Lacan, Blaise Hanczar, Michele Sebag

Frugal Generative Modeling for Tabular Data

Alice Lacan[1,2] (✉), Blaise Hanczar[1], and Michele Sebag[2]

[1] IBISC, U. Evry, Université Paris-Saclay
{alice.lacan,blaise.hanczar}@univ-evry.fr
[2] TAU, CNRS-INRIA-LISN, Université Paris-Saclay Michele.Sebag@lri.fr

Abstract. This paper presents a generative modeling approach called GMDA designed for tabular data, adapted to its arbitrary feature correlation structure. The generative model is trained so that sampled regions in the feature space contain the same fraction of true and synthetic samples, allowing true and synthetic data distributions to be aligned using a frugal and sound learning criterion. The merits of GMDA in terms of the usual performance indicators (pairwise correlation errors, precision, recall, predictive performance) are on par with or better than the state-of-the-art approaches for tabular data based on VAEs, GANs, or diffusion models. The key point is that it provides generative models with one or more orders of magnitude that are more frugal than baseline approaches.

Keywords: Generative modeling · Tabular data · Frugality

1 Introduction

The domain of generative modeling has been established for over a decade [15,8]. Its impressive results in language modeling [3] and image generation [23] rely on sophisticated embeddings, trained from massive data amounts and exploiting the specifics of image and text data [5,23].

This paper deals with the generative modeling of tabular data in response to the fact that tabular data is essential in most application areas, from healthcare to e-commerce or manufacturing. Generative modeling aims to address several problems. Firstly, data augmentation is required to train deep models when the original dataset is too restricted; for instance, manufacturing data are usually limited in size compared to the datasets commonly used to train deep neural nets. Secondly, synthetic data might be needed to accommodate privacy policy regimes [31,2]. Lastly, the synthetic data can populate hardly visited regions of the data space [6,20,25].

The challenge of generative modeling for tabular data is twofold. Firstly, tabular data is heterogeneous and contains features of mixed type (numerical and categorical); secondly, the relationships between the different features, i.e., the joint distribution of the data, can be arbitrarily complex, unlike text and image data. Except for a few approaches aimed to estimate the sought joint distribution through knowledge graphs [13] or Bayesian networks [7], most generative modeling approaches assume the independence of the features. After appropriately

preprocessing the features depending on their type (float, integer, or categorical), the generative model architecture comprises plain, fully connected layers [9]. The size of such models tends to increase rapidly with the number of features. This large model size, in turn, increases the risk of overfitting, especially since tabular datasets are typically an order of magnitude smaller than text or image datasets. Another challenge is that synthetic tabular data is difficult for domain experts to visually assess, unlike text and image data. Various ad hoc quality criteria, ranging from correlations to predictive performance (e.g., using Catboost [22]), have been developed to measure the fidelity of synthetic data to the true data. However, as noted by [1], outliers can easily mislead these criteria.

The main generative modeling approaches, namely variational autoencoders (VAEs) [15], generative adversarial networks (GANs) [8], and diffusion models [11] have been adapted to tabular data [30,32] (Section 2). These approaches typically involve large, fully connected architectures and require careful tuning of hyperparameters. The learning process faces optimization challenges when the size of the true dataset is limited, particularly in the case of adversarial training.

The contribution presented in this paper is a frugal approach to deep generative modeling for tabular data called *Generative Modeling with Density Alignment* (GMDA) (Section 3). Similar to [28], GMDA aims to align the true and synthetic data distributions. The difference lies in the learning criterion: whereas [28] minimizes an f-divergence between the two distributions, GMDA operates by i) sampling hyper-rectangles in the feature space; ii) minimizing the difference between the fraction of true and synthetic samples falling in these hyper-rectangles. Informally speaking, if these fractions are equal for all hyper-rectangles, both distributions are equal. The challenge is to sample the hyper-rectangles in a manner that provides sufficient coverage of the true data distribution while keeping the computational cost (which depends on the number of the hyper-rectangles) within acceptable limits.

After detailing the experimental setting (Section 4), we report on the extensive experimental validation of the approach on both artificial and real-world datasets, compared to baselines TVAE, CTGAN, and TabDDPM [30,16] (Section 5). Overall, GMDA produces generative models that perform on par with or better than the state of the art while being an order of magnitude more frugal in size. The scalability of GMDA w.r.t. high dimensionality settings is experimentally investigated using the TCGA and GTEX datasets with dimensions up to 1,000 [29,18]. The paper concludes with some perspectives for further research (Section 6).

2 Related Work

As mentioned, generative modeling has been less studied for tabular data than image and text data because of the diversity of the underlying joint distribution structure of the data and the difficulty in assessing the quality of a generative model.

In the last decade, the primary approaches have been based on oversampling techniques [4,19], exemplified by methods like SMOTE. These approaches use local heuristics (e.g., nearest neighbor interpolations) to create synthetic samples and overcome data gaps or class imbalances. Although these are the most commonly used techniques, they are unsuitable for high-dimensional problems, as nearest neighbor search suffers from the curse of dimensionality.

Another strategy is to estimate and exploit the joint distribution of data using probabilistic graphical models such as Bayesian networks (BNs) [27,7]. [33] also proposes an approximation of a set of low-dimensional marginals with PrivBayes, aiming at generating synthetic data in privacy-sensitive domains. The main challenge is to determine the Bayesian graph's structure to reflect the independencies and conditional dependencies between variables. These approaches face two main limitations: identifying a graph structure is known to be NP-hard, and the number of statistical tests required to determine dependency relationships is cubic in the number of variables, which raises the issue of multiple hypothesis testing.

Adaptations of the celebrated VAE [15] and GAN [8] approaches have been designed for tabular data. [30] presents two generative models for tabular data, respectively called TVAE and CTGAN, each employing specific type-dependent normalization for each feature. These models are shown to outperform BN-based approaches while being less prone to mode collapse than older GAN-based generative models for tabular data, e.g., TableGAN [21] and PATE-GAN [31].

Diffusion models have also been adapted to the tabular data setting, including TabDDPM [16] and TabSYN [32], both of which outperform TVAE and CTGAN. The performance of TabSYN is enhanced by combining the embedding of the continuous and categorical features into a latent continuous space and using a VAE in this latent space to seed the diffusion process.

The limitations of advanced generative models for tabular data are twofold. On the one hand, as noted by [16], they hardly outperform SMOTE in various contexts. On the other hand, these large-scale models are challenging to train on datasets of limited size, and their hyper-parameters must be carefully optimized to avoid overfitting or mode collapse.

Unsupervised sample-based indicators, such as precision and recall measures, have been proposed to address the challenging task of assessing data quality. These indicators respectively reflect the realism and diversity of the synthetic samples [17,1]. Among these metrics, some compare the local density of true and synthetic samples. However, since some widely adopted metrics have been shown to be biased in favor of simple memorization (duplication) of the real distribution, [12] proposed the FLD indicator to evaluate the overfitting of generative models. This indicator is based on approximating the synthetic distribution as a mixture of Gaussians and then measuring the likelihood of the true samples according to this estimated density.

The approach most related to ours is proposed by [28]. Within the GAN or Normalizing Flow frameworks, the generative model is trained by optimizing the f-divergence between the true and synthetic distributions, thereby tackling

a differentiable learning criterion. The choice of hyperparameters allows control over the precision-recall trade-off. A key difference from mainstream GANs is that the f functions involved in the f-divergence differ for the generator and the discriminator.

Discussion. A primary motivation for the proposed generative modeling approach is to comply with the AI frugality requirements [26,14]. More precisely, our goal is to develop generative models that perform comparably to the state of the art but are based on simpler architectures and require less computing power and data for learning.

For this purpose, the *Generative Modeling with Density Alignment* (GMDA) method is based on a stochastic loss, which estimates the alignment between the true and synthetic distributions. We found that a carefully designed stochastic sampler of hyper-rectangles in the feature space can effectively guide the generative model while being less computationally expensive and more robust than an adversarial module.

3 Overview of GMDA

This section introduces the proposed GMDA approach. It begins by outlining the fundamental principle of the approach, followed by a detailed description of its modules and the pseudo-code of the algorithm.[3]

Notations. In the following, the true distribution is the discrete distribution $\mathcal{D} = \{\mathbf{x}_1, \dots \mathbf{x}_n\}$ defined on the d-dimensional instance space $\mathcal{X} \subset \mathcal{R}^d$. Similarly, the generated or synthetic distribution $\mathcal{G}$ is defined on $\mathcal{X}$; for simplicity and convenience, a finite-sized sample generated from $\mathcal{G}$ is also denoted as $\mathcal{G}$.

Principle. As mentioned, performance metrics utilized to evaluate a generative modeling approach, such as precision and recall, reflect how closely the synthetic samples align with the true samples on average (realism) and how closely the true samples align with the synthetic samples on average (diversity). Consequently, optimizing these indicators entails a multi-objective optimization challenge: ensuring that most generated samples are situated in the high-density regions of the true distribution while adequately exploring the low-density regions to mitigate the risk of mode collapse. As pointed out by [28], this multi-objective issue can be reformulated as a single objective problem by minimizing an f-divergence between the true and synthetic distributions. GMDA minimizes a much simpler distance between both distributions based on density probes:

Definition 1. *Let $H \subset \mathcal{R}^d$ be a region in the instance space. The density probe related to H is defined as the difference between the percentage of true and synthetic samples falling within H, denoted as $d(\mathcal{D}, \mathcal{G}, H)$:*

$$d(\mathcal{D}, \mathcal{G}, H) = |Pr(\mathbf{x} \in \mathcal{D}, \mathbf{x} \in H) - Pr(\mathbf{x}' \in \mathcal{G}, \mathbf{x}' \in H)| \quad (1)$$

[3] The code and the supplementary material (SM) are publicly available at github.com/ablacan/gmda.

Given two distributions $\mathcal{A}$ and $\mathcal{B}$ on $\mathcal{X}$, it is straightforward to show that the expectation on $H \subset \mathcal{X}$ of $d(\mathcal{A}, \mathcal{B}, H)$ defines a distance among $\mathcal{A}$ and $\mathcal{B}$:

$$d(\mathcal{A}, \mathcal{B}) = \mathbb{E}_{H \subset \mathcal{X}} \quad d(\mathcal{A}, \mathcal{B}, H) \tag{2}$$

Based on this definition, one might consider devising an adversarial generative modeling framework comprising a generator and an adversary [8]. Here, the generator aims to formulate $\mathcal{G}$ while the adversary aims to identify regions H to maximize the corresponding density probes. Nevertheless, this adversarial strategy comes up against a fundamental drawback: optimizing Eq. 2 would result in memorizing $\mathcal{D}$, leading to an uninformative generative model. GMDA thus explores a non-adversarial approach, using a stochastic probe sampler instead of an adversary. Specifically, at each epoch, K regions $H_1, \ldots, H_K$ are randomly sampled, and GMDA works toward learning the generative model $\mathcal{G}$ that minimizes the pseudo-distance between $\mathcal{D}$ and $\mathcal{G}$, defined as the sum of the density probes $d(\mathcal{D}, \mathcal{G}, H_i)$ for $i = 1 \ldots K$.

GMDA primarily relies on two algorithmic components. The first aspect involves defining probe H in a manner that makes $d(\mathcal{D}, \mathcal{G}, H)$ differentiable and conducive to optimization through back-propagation. The second aspect entails establishing an effective sampling mechanism that selects informative probes $H_1 \ldots H_K$ aligning with the generative modeling objective.

3.1 Designing Differentiable Density Probes

For computational efficiency, the probes under consideration are hyper-rectangles, characterized as the Cartesian product of intervals $[a_i, b_i]$ for $i = 1 \ldots d$. The definition of the density probe $d(\mathcal{D}, \mathcal{G}, H)$ is based on the design of a smooth indicator function $I(x, [a, b])$ that approximates whether a scalar $x \in \mathcal{R}$ belongs to the interval $[a, b]$ while being differentiable (unlike the standard binary indicator). The proposed approach, illustrated in Fig. 1, defines $I(x, [a, b])$ as the product of two sigmoid functions:

Definition 2. *Let $x \in \mathcal{R}$ denote a real value, and $[a, b]$ be an interval in $\mathcal{R}$. The smooth indicator function $I(x, [a, b])$ is defined as:*

$$I(x, [a, b]) = \left(\frac{1}{1 + e^{-\lambda(x-a)}} \right) \times \left(\frac{1}{1 + e^{-\lambda(b-x)}} \right)$$

The slope $\lambda > 0$ of the sigmoid function governs the trade-off between $I(x, [a, b])$ closely approximating the binary indicator function (where a higher λ results in a more accurate approximation) and maintaining a bounded derivative range. In the experiments, the parameter λ is set to 5.

The smooth indicator function associated with a hyper-rectangle follows in a straightforward manner:

Definition 3. *Let $H \subset \mathcal{R}^d$ be defined as the Cartesian product of intervals $[a_i, b_i]$ for i in $1, \ldots d$. The smooth indicator function $I(\mathbf{x}, H)$ is defined as:*

$$\forall \mathbf{x} = (x_1, \ldots x_d) \in \mathcal{R}^d,\ I(\mathbf{x}, H) = \prod_{i=1}^{d} I(x_i, [a_i, b_i])$$

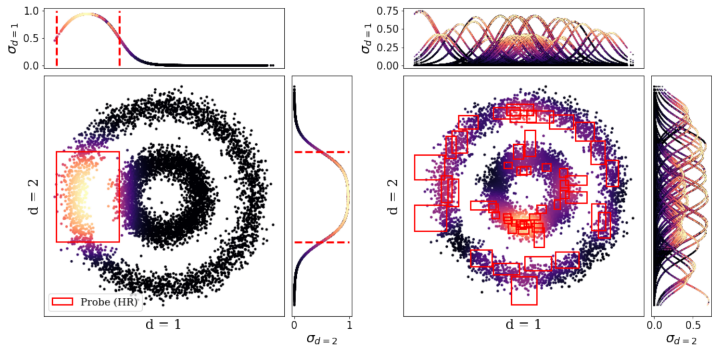

Fig. 1: Toy 2-circles 2d dataset: Indicator function $I(\mathbf{x}, H)$ linked to rectangles. Left: the value of $I(\mathbf{x}, H)$ for the depicted rectangle, with sigmoid slope $\lambda = 10$, and its marginals in the two dimensions. Right: a random sampling of 50 probes with a density of $\delta = 5\%$ (see text) showcasing densely visited areas (pink and orange) alongside vacant regions (black).

The effectiveness of a probe is assured by: i) setting $[a_i, b_i]$ to $[-\infty, \infty]$ for all coordinates except for a limited number identified as active coordinates (in the experiments, the number of active coordinates is fixed at 3); ii) appropriately sampling the probes. The selection of hyper-rectangles as probes (rather than, for instance, hyper-spheres) is rationalized by the ability to pre-calculate the cumulative density functions along each space coordinate. This facilitates an efficient probe sampling technique, as detailed below.

3.2 Designing a Probe Sampler

Each probe H is centered on a real sample $\mathbf{x} \in \mathcal{D}$, referred to as *seed*. The seed selection process aims to cover the true distribution support effectively (below). For a given seed $\mathbf{x}$, the hyper-rectangle $H(\mathbf{x})$ is determined by i) uniformly sampling the active coordinates in $\{1, \ldots, d\}$ without replacement; ii) defining the interval $[a_i, b_i]$ related to each active coordinate i.

The interval width is determined based on the density rate δ. This hyper-parameter controls the number $2m+1$ of true samples falling within the interval (where $m = \frac{\delta n - 1}{2}$, with n the number of true samples). The lower δ, the higher the number K of probes needed to cover the true distribution, and the better the training loss (defined below) estimates the distance between both true and synthetic distributions. It is important to note that this control of the interval width permits the size of the hyper-rectangle H to encompass, to some extent, the density of the real distribution. Across a specific coordinate, the intervals are narrower in dense areas and wider in sparser regions. The limitation is that the computation of interval $[a_i, b_i]$ considers each feature independently as if the joint distribution were the product of the marginals.

Algorithmically, given the (pre-computed) ordered list of the i-th coordinate values of the true samples, let $r_i(\mathbf{x})$ denote the rank of the i-th coordinate value for seed $\mathbf{x}$, with $\pi_i(k)$ the i-th coordinate value with a rank k. Subsequently, a_i and b_i are determined as follows:

$$a_i = \begin{cases} \pi_i(r_i(\mathbf{x}) - m) & \text{if } r_i(\mathbf{x}) > m \\ \pi_i(1) - \epsilon & \text{otherwise} \end{cases}$$
$$b_i = \begin{cases} \pi_i(r_i(\mathbf{x}) + m) & \text{if } r_i(\mathbf{x}) < n - m \\ \pi_i(n) + \epsilon & \text{otherwise} \end{cases}$$

with $\epsilon > 0$ a margin and n the number of true samples.

As could be expected, the regions of the instance space are neither equally dense nor equally easy to cover in general. The probe sampler thus aims to achieve a trade-off between uniformly sampling the seeds (exploration) and giving more consideration to challenging-to-cover regions (exploitation). At the initialization, the K probes are defined from K uniformly selected seeds in $\mathcal{D}$. In further epochs, a proportion η of the hyper-rectangles exhibiting the highest loss at iteration t are retained for iteration $t + 1$, where the probe persistence η is a hyperparameter of GMDA; other hyper-rectangles are defined from uniformly selected seeds. Note that this persistence mitigates the stochastic nature of the overall training loss. For $\eta = 0$, the loss boils down to a stochastic estimate of the distance among the true and synthetic distributions (Eq. 2).

3.3 Learning criterion

A local loss $\mathcal{L}(H)$ is associated with each probe H, reflecting the disparity between the numbers of true and synthetic samples falling within H. These numbers are respectively approximated by the sum of $I(\mathbf{x}, H)$ for $\mathbf{x} \in \mathcal{D}$ and the sum of $I(\mathbf{x}, H)$ for $\mathbf{x} \in \mathcal{G}$. However, the absolute difference between the two numbers does not adequately reflect local density alignment: the same absolute difference should have a more significant impact in a low-density than in a high-density region. The absolute difference is consequently divided by a monotonic function of the density to address this observation. Following preliminary experiments, the loss associated with probe H is defined as:

$$\mathcal{L}(H) = \frac{\left|\sum_{\mathbf{x}\in\mathcal{D}} I(\mathbf{x}, H) - \sum_{\mathbf{x}'\in\mathcal{G}} I(\mathbf{x}', H)\right|}{\log\left(1 + \max\left(\sum_{\mathbf{x}\in\mathcal{D}} I(\mathbf{x}, H), \sum_{\mathbf{x}'\in\mathcal{G}} I(\mathbf{x}', H)\right)\right)} \tag{3}$$

To enforce the precision of the generative model and prevent synthetic samples from wandering in vacant regions, the learning criterion is augmented with an additional loss term called *dark probe* term, set to $w_{DH}\mathcal{L}(DH)$, with $w_{DH} \geq 0$ a hyperparameter of the approach, where DH is the complement of the union of probes $H_1, \ldots, H_K$, with $I(\mathbf{x}, DH)$ set to $\prod_{i=1}^{K}(1 - I(\mathbf{x}, H_i))$.[4]

[4] A small quantity is added to the denominator of $\mathcal{L}(DH)$ to prevent numerical instabilities.

Overall, the learning criterion of the generative model is the aggregate of the losses $\mathcal{L}(H)$ for H spanning the current probes $H_1, \ldots, H_K$, and the dark probe DH:

$$\mathcal{L} = \sum_{i=1}^{K} \mathcal{L}(H_i) + w_{DH}\mathcal{L}(DH) \tag{4}$$

Practically, GMDA takes a p-dimensional Gaussian noise vector noted $\mathbf{z}$ as input. The generative model is a neural net that produces a synthetic sample from $\mathbf{z}$, trained to minimize the loss (Eq. 4). Similarly to CTGAN [30], GMDA can produce conditional generative models, taking the concatenation of $\mathbf{z}$ and the encoding of a class variable as input and concatenating the class variable again to the latent vector in each layer; a class-dependent loss is considered (only focusing on the true samples within the given class), and the overall loss averages the class-dependent losses.

Algorithm 1 GMDA

Require: Data distribution $\mathcal{D}$, number of iterations T, number K of hyper-rectangles, hyperparameters: persistence η; density δ; dark probe weight w_{DH}
Initialization
Initialize θ, vector of the generator parameters
Select uniformly K distinct seeds $\mathbf{x}_1 \ldots \mathbf{x}_K$ in $\mathcal{D}$
for $k = 1, \ldots, K$ **do**
  Build $H_k = H(\mathbf{x}_k, \delta)$
  Compute $\mathcal{L}(H_k)$ ▷ Compute loss on H_k (Eq. 3)
end for
$\theta \leftarrow BP(\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}(H_k) + w_{DH}\mathcal{L}(DH))$ ▷ Update θ by back propagation
for $t = 2, \ldots, T$ **do**
  Exploitation
    Retain seeds associated with top ηK probes with highest loss
  Exploration
    Sample $(1 - \eta)K$ new seeds uniformly
  for $k = 1, \ldots, K$ **do**
    Build $H_k = H(\mathbf{x}_k, \delta)$
    Compute $\mathcal{L}(H_k)$ ▷ Compute loss on $H(\mathbf{x}_k)$
  end for
  $\theta \leftarrow BP(\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}(H_k) + w_{DH}\mathcal{L}(DH))$ ▷ Update θ by back-propagation
end for

3.4 Discussion

The GMDA learning criterion is, by design, a stochastic approximation of the distance between the true and the synthetic distribution (Eq. 2); the tightness of this approximation depends on the number K of probes and the density rate δ. For arbitrarily large values of n and K, and arbitrarily small δ, the

optimal generative model can thus yield optimal precision and accuracy (putting a synthetic sample in each true sample's neighborhood and no synthetic samples outside of these neighborhoods). The quality of the generative model is then bounded by the computational resources. Note that the loss can provide some direct and local assessment of the generative model, indicating the ill-covered regions:

Proposition 1. *Let $H(\mathbf{x})$ be the probe defined from seed $\mathbf{x}$ and density rate δ. If $H(\mathbf{x})$ does not contain any synthetic sample, then*

$$\mathcal{L}(H(\mathbf{x})) \geq 3$$

Proof: follows from Eq. 3.

4 Experimental setting

The primary goal of the experiments is to comparatively assess the performance of GMDA compared with the state of the art and to particularly investigate its scalability w.r.t. the dimension d of the problem. Our second goal is to assess its sensitivity w.r.t. the four hyper-parameters of the approach: the number K of probes, the density rate δ, the persistence rate η, and the weight w_{DH} of the dark probe. Lastly, the compliance of the approach with the Green AI requirements is assessed in terms of model size and computational cost.

Performance indicators. Both unsupervised and supervised indicators are considered. Unsupervised indicators include the pair-wise correlations error (evaluating the faithfulness of the synthetic data structure), the precision and recall indicators [17] measuring the realism and diversity of the synthetic distribution w.r.t. the true one, and the harmonic mean of these indicators (F1) [24].[5] Following [17], they are computed on the full dataset $\mathcal{D}$, and a synthetic $\mathcal{G}$ of same size n for the sake of stability, using a number $k = 5$ of nearest neighbors ($k = 10$ for high dimensional datasets). The supervised performance indicator, referred to as Machine Learning Efficiency (MLE), is the predictive accuracy on $\mathcal{D}$ (test set) of a classifier learned from $\mathcal{G}$, where $\mathcal{G}$ is learned from the training set of $\mathcal{D}$; the predictive accuracy (F1 score) is averaged over 20 runs following [16] (settings of the classifiers in SM G).

Benchmarks (Table 1). Besides 2d toy datasets used for illustration purposes, the experimental validation considers four well-studied datasets for the sake of comparison with the state-of-the-art, with a small or medium number of samples and features [16,32]. Two further gene expression datasets with a high number of features, originating from the Genotype-Tissue Expression (GTEx) project [18]

[5] Although such metrics are sensitive to outliers, we argue that they remain empirically more stable and interpretable than further formulations suggested by [1].

Table 1: The benchmark datasets (links to open source data in SM A).

Dataset	#Train	#Validation	#Test	Dim.	Classification
Diabetes	442	49	277	8	Binary
Gesture	5,687	632	1,724	32	Multiclass
Magic	15,406	1712	1,902	10	Binary
Wilt	2,786	310	1,743	5	Binary
GTEx	9,796	2,448	5,000	974	Multiclass
TCGA	6,499	1,625	1,625	978	Multiclass

and the Cancer Genome Atlas (TCGA) [29], are also considered. Following the common practice, the datasets are preprocessed (standardized to zero mean and unit variance for the first four datasets, using a quantile transformation for the last two). More details can be found in SM A.

Baselines. For the first four datasets, the baselines are TVAE and CTGAN [30], and TabDDPM [16], using their hyperparameters as reported in the cited papers. For the last two datasets, the baselines are VAE [15] and WGAN-GP [10] (best settings in SM B).

GMDA **hyper-parameter setting.** GMDA is implemented as a three hidden layers neural net, with Leaky ReLU activations and batch normalization. All hyper-parameters of GMDA are adjusted using Bayesian optimization over 100 trials, maximizing the precision-recall trade-off (F1) between the generated distribution and the training set (details in SM B3).

5 Experiments

This section reports on the empirical results of GMDA compared with the state of the art (more results are presented in SM). All reported results are averaged on five runs.

5.1 Performance

The visual inspection of the 2D datasets (Fig. 2) reveals the main strengths and weaknesses of the approaches: TVAE tends to generate samples outside the true distribution of the moons, and the swiss roll; CTGAN fails to reproduce the shape for all 2D datasets. These observations confirm that VAEs tend to increase diversity at the cost of realism, while GANs might be prone to mode collapse issues. TabDDPM generates the most faithful data; GMDA-generated data is the most similar to that of TabDDPM.

The quality of the synthetic data structure is first assessed from the absolute difference between the true and the synthetic covariance matrices for the

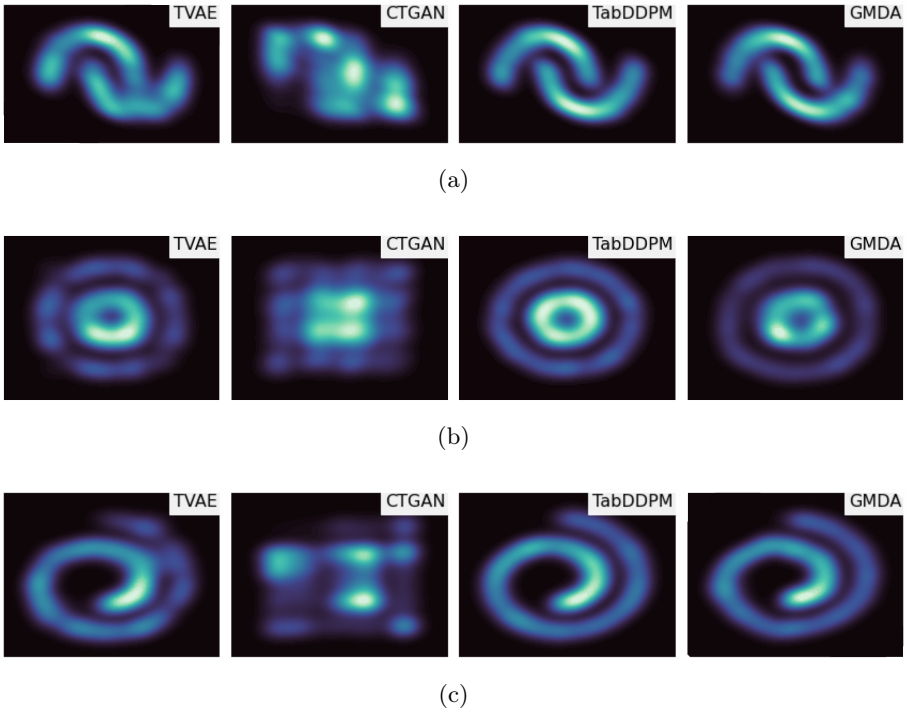

Fig. 2: The 2d datasets: Visual inspection of the generative models. From top to bottom: (a) moons, (b) circles, (c) swiss roll. From left to right: TVAE, CTGAN, TabDDPM, GMDA.

Table 2: Quality of the synthetic data structure (absolute difference of synthetic vs. true data correlations). Statistically significantly best results in bold.

Model	Diabetes	Gesture	Magic	Wilt	Avg.	Rank
TVAE	$2.3_{\pm 0.18}$	$3.12_{\pm 0.17}$	$1.68_{\pm 0.12}$	$2.41_{\pm 0.48}$	2.38%	2
CTGAN	$22.06_{\pm 1.16}$	$5.97_{\pm 0.16}$	$12.49_{\pm 0.15}$	$10.34_{\pm 0.33}$	12.72%	4
TabDDPM	$19.82_{\pm 2.88}$	$\mathbf{1.94_{\pm 0.13}}$	$1.32_{\pm 0.06}$	$22.11_{\pm 6.38}$	11.3%	3
GMDA	$\mathbf{2._{\pm 0.37}}$	$2.58_{\pm 0.03}$	$\mathbf{1.22_{\pm 0.13}}$	$\mathbf{1.9_{\pm 0.28}}$	**1.92%**	1

four medium-size datasets (visually represented in Fig. 3, and quantitatively in Table 2). GMDA ranks first (except on the Gesture dataset, where TabDDPM outperforms it) with an error rate of circa 2%. Interestingly, TabDDPM shows a high error rate variance and underperforms on Diabetes and Wilt. All other approaches dominate CTGAN, and TVAE ranks second overall.

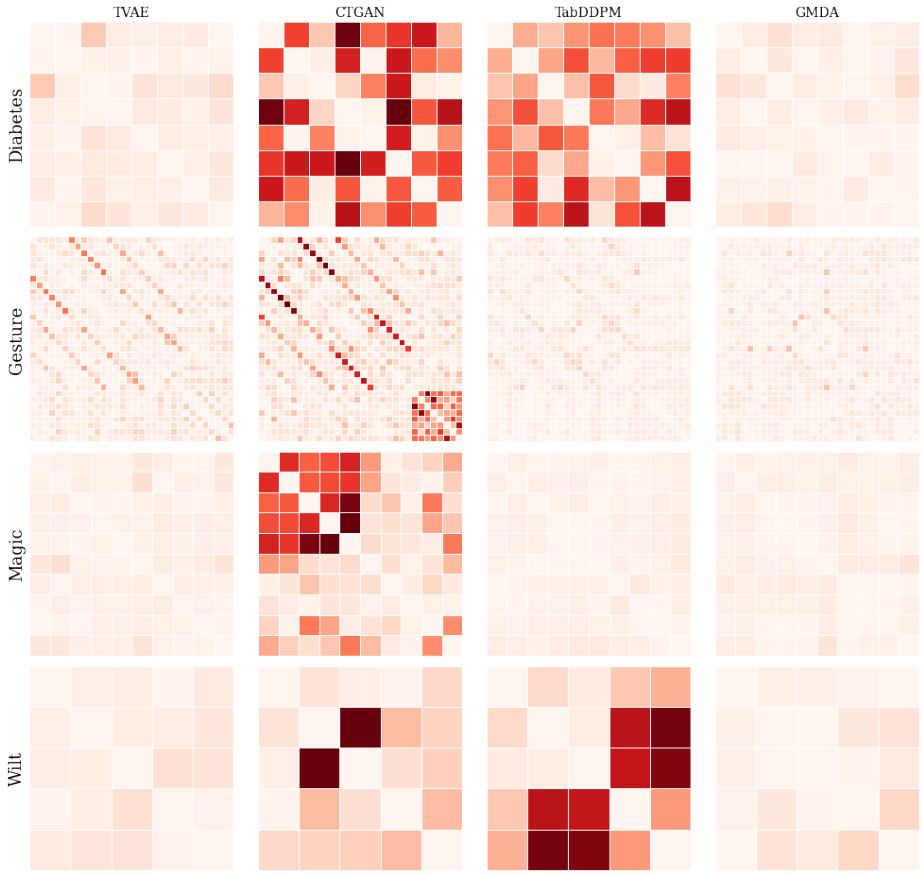

Fig. 3: Quality of the synthetic data structure (absolute difference of synthetic vs. true data correlations; the paler, the better). From top to bottom: Diabetes, Gesture, Magic and Wilt. From left to right: TVAE, CTGAN, TabDDPM, GMDA. The colors are normalized by dataset, i.e., darker red corresponds to higher absolute errors relative to the other feature pairs and generative models.

Table 3: Precision / Recall performance (F1 score (%); the higher, the better). Statistically significantly best results in bold.

Model	Diabetes	Gesture	Magic	Wilt	Avg.	Rank
TVAE	**95.37**$_{\pm 1.21}$	38.31$_{\pm 0.94}$	91.83$_{\pm 0.13}$	96.2$_{\pm 0.42}$	80.43%	3
CTGAN	1.45$_{\pm 0.78}$	0.13$_{\pm 0.12}$	33.39$_{\pm 0.33}$	18.8$_{\pm 1.}$	13.44%	4
TabDDPM	78.31$_{\pm 0.74}$	**84.61**$_{\pm 0.32}$	**96.75**$_{\pm 0.09}$	**98.36**$_{\pm 0.1}$	**89.51%**	1
GMDA	92.74$_{\pm 0.63}$	71.51$_{\pm 0.49}$	90.99$_{\pm 0.15}$	97.1$_{\pm 0.12}$	88.09%	2

Table 4: Machine learning efficiency: F1 score (%) of Catboost classifier on test data (the higher, the better). Statistically significantly best results in bold.

Model	Diabetes	Gesture	Magic	Wilt	Avg.	Rank
Baseline	$69.13_{\pm 1.08}$	$58.05_{\pm 0.62}$	$88.32_{\pm 0.28}$	$91.47_{\pm 0.72}$	-	-
TVAE	$64.87_{\pm 1.12}$	$35.6_{\pm 1.22}$	$82.89_{\pm 0.32}$	$86.42_{\pm 1.59}$	67.45%	3
CTGAN	$47.49_{\pm 7.33}$	$16.49_{\pm 3.07}$	$52.74_{\pm 0.58}$	$47.35_{\pm 1.17}$	41.02%	4
TabDDPM	$\mathbf{75.26_{\pm 0.89}}$	$\mathbf{50.36_{\pm 0.76}}$	$\mathbf{86.85_{\pm 0.24}}$	$88.83_{\pm 0.96}$	**75.32%**	1
GMDA	$67.69_{\pm 1.04}$	$43.58_{\pm 0.41}$	$85.58_{\pm 0.4}$	$\mathbf{92.84_{\pm 0.57}}$	72.42%	2

Table 5: High-dimensional datasets: Machine learning efficiency and Precision-Recall F1 score (the higher, the better). Statistically significantly best results in bold.

Model	GTEx		TCGA		Rank
	MLE	F1 (PR)	MLE	F1 (PR)	
MLP Class.	$99.32_{\pm 0.04}$	-	$93.59_{\pm 0.6}$	-	-
VAE	$\mathbf{98.98_{\pm 0.05}}$	$74.28_{\pm 0.1}$	$88.36_{\pm 0.97}$	$82.36_{\pm 0.06}$	3
WGAN-GP	$98.76_{\pm 0.09}$	$\mathbf{94.66_{\pm 0.1}}$	$\mathbf{92.04_{\pm 0.46}}$	$\mathbf{93.17_{\pm 0.17}}$	1
GMDA	$98.4_{\pm 0.6}$	$79.86_{\pm 0.25}$	$89.68_{\pm 0.4}$	$83.27_{\pm 0.34}$	2

The quality of the synthetic data is assessed from its realism-diversity trade-off, from the harmonic mean between precision and recall (Table 3). GMDA ranks second, being slightly but statistically significantly outperformed by TabDDPM. Detailed results (in SM C) show that GMDA is consistently the best approach in terms of precision (93% on average) on all datasets, while TabDDPM performs poorly on Diabetes, and both TVAE and CTGAN collapse on Gesture. The detailed PCA analysis of the true and synthetic data (in SM D) suggests that TVAE and TabDDPM gain diversity by generating data outside the true data manifold, i.e., with synthetic samples spread in empty regions for Gesture and Wilt. In contrast, the GMDA-generated data is similarly structured on the four datasets.

The quality of the synthetic data is last assessed in the supervised perspective, considering the Machine Learning Efficiency indicator (Table 4), reporting the predictive performance of the Catboost classifier trained from the synthetic data (results with XGBoost are presented in SM C). Likewise, GMDA ranks second, slightly outperformed by TabDDPM.

5.2 Scalability w.r.t. the dimension d

First results showing the comparative performance (in MLE and Precision-Recall F1 score) of GMDA on the two high-dimensional GTEx and TCGA datasets are

displayed in Table 5, compared to the oracle baseline (standard MLP classifier trained on the true data, more in SM F), VAE [15] and WGAN-GP [10]. On both GTEx and TCGA, GMDA ranks second in precision-recall; it is dominated by other approaches in MLE.

These results first establish the feasibility of learning generative models in high dimension (d circa 1,000) based on density alignment with hyper-rectangular probes by significantly increasing the number of probes (circa 800 vs 100 for medium-size problems). On the other hand, GMDA shows a significantly higher variance than VAE and WGAN-GP. We shall return to this in Section 6.

5.3 Sensitivity study

As said, the optimal configuration for GMDA is determined by Bayesian optimization of the Precision-Recall F1 score. Around this configuration, the sensitivity of the MLE performance is analyzed by varying a single hyper-parameter (more results in SM E).

The sensitivity w.r.t. the number K of probes, ranging in $\{10, 100, 250, 500\}$, is depicted in Fig. 4. As expected, the performance increases when increasing K, yielding a more precise training loss. For all datasets but Diabetes, however, the performance reaches a plateau for a sufficiently high number of probes ($K \geq 100$).

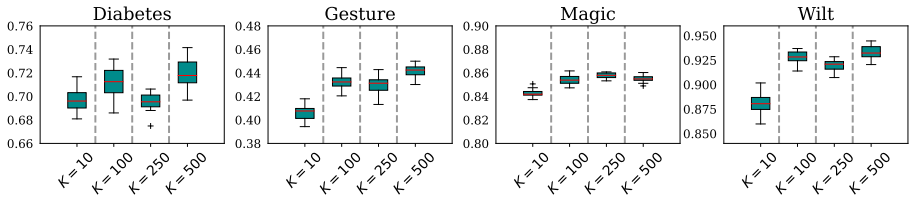

Fig. 4: GMDA: Sensitivity analysis of MLE w.r.t. the number K of probes (Catboost F1 score, quantiles on 5 runs).

The sensitivity w.r.t. the density rate δ, ranging in $\{.05, .1, .25, .35\}$, is depicted in Fig. 5. For Magic and Wilt, the performance is best for small values of δ, decreasing as δ increases. For Gesture, the performance is best for medium values of δ (δ in $[.10, .25]$) and degraded for lower or higher values. These results are interpreted in relation to the higher dimension of the Gesture dataset, suggesting that higher values of K should be considered in combination with smaller values for δ. The erratic sensitivity pattern on Diabetes is attributed to its low number of samples.

The sensitivity w.r.t. the other two parameters of GMDA, the persistence parameter η (ranging in $\{.0, .2, .5., .8\}$) and the dark probe weight w_{DH} (ranging in $\{0, 1\}$) is analyzed in SM D. In brief, the sensitivity is low w.r.t. η. In contrast,

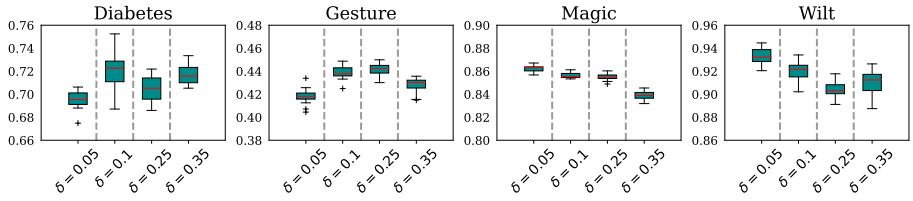

Fig. 5: GMDA: Sensitivity analysis of MLE w.r.t. the density rate δ parameter (Catboost F1 score, quantiles on 5 runs).

the dark probe weight significantly impacts the performance: its presence has a positive impact on Gesture (and Diabetes) and a negative impact on Wilt (and to a lesser extent on Magic).

Overall, the most sensitive parameters for GMDA are the number K of probes (that must be sufficiently high) and the density rate δ, controlling the precision of the coverage of the true distribution. The persistence parameter η shows a low sensitivity: the approach performs as well in pure stochastic mode ($\eta = 0$).[6]

5.4 Frugality analysis

All reported results are measured on an NVIDIA A40 GPU with 48 GB of RAM. The load of the generative modeling approaches (model size in number of parameters and computational training cost) is displayed in Fig. 6. In terms of computational time, TabDDPM ranks first, with GMDA coming in second; this performance is partly due to the use of pre-computation of the cumulative density functions used to define the probes (Section 3.2). Regarding model size, GMDA achieves a gain of one or more orders of magnitude compared to all other approaches. This improvement is due to the fact that GMDA does not incorporate an adversary module.

The frugality of GMDA in terms of model size is confirmed on the high-dimensional GTEx and TCGA datasets, showing a gain of one order of magnitude. For GTEx, the model size is 36×10^6 for VAE and 19×10^6 for WGAN-GP, compared to 3.2×10^6 for GMDA. For TCGA, the model size is 46×10^6 for VAE and 22×10^6 for WGAN-GP, compared to 2.6×10^6 for GMDA.

In terms of training time, the current implementation of GMDA is not as optimized as VAE or WGAN-GP, leading to a higher computational load. For GTEx, GMDA currently requires 3 hours (compared to 1 hour for VAE and 13 mn for WGAN-GP, due to a batch size of 1024); for TCGA, GMDA requires 2 hours (compared to 22 minutes for VAE and 1hour and 40mn for WGAN-GP).

[6] Along the same line, the use of refined heuristics for the selection of the seeds, accounting for how often these have been selected in the former epochs, did not improve the performance.

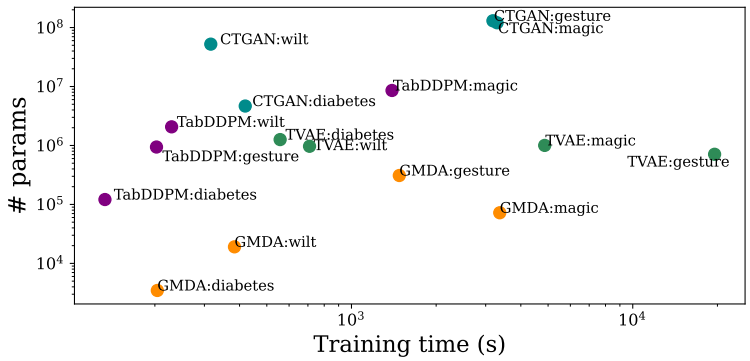

Fig. 6: Computational Training Load and Model Size (number of parameters) for TVAE, CTGAN, TabDDPM, and GMDA (in orange), over datasets Diabetes, Gesture, Magic and Wilt.

6 Conclusion

The primary motivation behind the presented GMDA was to design a generative modeling approach for tabular data that aligns more closely with the frugality requirements increasingly advocated for AI.

The contributions of the GMDA approach include: i) a sound learning criterion, based on the distance between the true and the learned distributions, precisely measured by the precision and recall performance indicators; ii) a stochastic approximation of this learning criterion, utilizing density probes and suitable for continuous optimization; iii) algorithmic pre-computations designed to facilitate the probe sampling task during the learning process.

The first two issues are closely related: employing a trained adversary for this pristine learning criterion would result in merely duplicating the target distribution, which is ineffective. The stochastic approximation of the probe, operating with bounded resources, is essential to prevent this duplication. In counterpart, the number and size of probes must match the dimension of the problem and the complexity of the target joint distribution.

The approach produces frugal generative models, significantly smaller in size compared to the state-of-the-art, with moderate or low sensitivity to its four hyperparameters. The GMDA performance on medium-sized tabular datasets is comparable to the state of the art, while on high-dimensional datasets, the results are currently promising, indicating potential for further research.

A first and short-term perspective involves extending GMDA to other (categorical and integer) data types.

A longer-term perspective is to address the challenge of high-dimensional generative modeling by exploiting the fact that the intrinsic dimension of the current datasets is usually significantly lower than the number of features. Drawing inspiration from [32], a potential approach could be integrating an auto-encoder with

GMDA, where GMDA operates in the latent space. The novelty of this approach would be replacing the conventional adversarial framework with a cooperative one, where the generative model collaborates with the auto-encoder to optimize a learning criterion defined in latent and feature spaces.

A medium-term perspective involves delving deeper into understanding the reasons behind the high variance of the GMDA results. Future work will explore the utilization of dynamic schemes to regulate the density rate δ and the number of probes throughout the learning process.

Acknowledgments. This research was supported by the Labex DigiCosme (University Paris-Saclay) and by a public grant overseen by the French National Research Agency (ANR) through the program UDOPIA, project funded by the ANR-20-THIA-0013-01.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alaa, A., Van Breugel, B., Saveliev, E.S., van der Schaar, M.: How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In: ICML (2022)
2. Bhanot, K., Qi, M., Erickson, J.S., Guyon, I., Bennett, K.P.: The problem of fairness in synthetic healthcare data. Entropy **23**(9), 1165 (Sep 2021)
3. Brown, T., Mann, B., et al.: Language models are few-shot learners. In: NeurIPS (2020)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (Jun 2002)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019)
6. Engelmann, J., Lessmann, S.: Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications **174**, 114582 (2021)
7. Gogoshin, G., Branciamore, S., Rodin, A.S.: Synthetic data generation with probabilistic bayesian networks. Mathematical Biosciences and Engineering **18**(6), 8603–8621 (2021)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
9. Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A.: Revisiting deep learning models for tabular data. In: NeurIPS (2021)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: NeurIPS (2017)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
12. Jiralerspong, M., Bose, J., Gemp, I., Qin, C., Bachrach, Y., Gidel, G.: Feature likelihood score: Evaluating the generalization of generative models using samples. In: NeurIPS (2023)

13. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2), 183–233 (1999)
14. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (Jul 2023)
15. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: ICLR, (2014)
16. Kotelnikov, A., Baranchuk, D., Rubachev, I., Babenko, A.: TabDDPM: Modelling tabular data with diffusion models. In: ICML (2023)
17. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. In: NeurIPS (2019)
18. Lonsdale, J., et al.: The genotype-tissue expression (gtex) project. Nature Genetics **45**(6), 580–585 (May 2013)
19. Nguyen, H.M., Cooper, E.W., Kamei, K.: Borderline over-sampling for imbalanced data classification. Int. J. Knowl. Eng. Soft Data Paradigms **3**, 4–21 (2009)
20. Onishi, S., Meguro, S.: Rethinking data augmentation for tabular data in deep learning. ArXiv (2023)
21. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. Proceedings of the VLDB Endowment **11**(10), 1071–1083 (Jun 2018)
22. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. In: NeurIPS (2018)
23. Radford, A., Kim, J.W., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
24. Sajjadi, M.S.M., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. In: NeurIPS (2018)
25. Schultz, K., Bej, S., Hahn, W., Wolfien, M., Srivastava, P., Wolkenhauer, O.: Convgen: A convex space learning approach for deep-generative oversampling and imbalanced classification of small tabular datasets. Pattern Recognition **147**, 110138 (2024)
26. Schwartz, R., Dodge, J., Smith, N., Etzioni, O.: Green ai. Communications of the ACM **63**, 54–63 (11 2020)
27. Sun, Y., Cuesta-Infante, A., Veeramachaneni, K.: Learning vine copula models for synthetic data generation. Proceedings of AAAI **33**(01), 5049–5057 (Jul 2019)
28. Verine, A., Negrevergne, B., Pydi, M.S., Chevaleyre, Y.: Precision-recall divergence optimization for generative modeling with GANs and normalizing flows. In: NeurIPS (2023)
29. Weinstein, J.N., Collisson, E.A., Mills, G.B., Shaw, K.R.M., Ozenberger, B.A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J.M.: The cancer genome atlas pan-cancer analysis project. Nature Genetics **45** (09 2013)
30. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. In: NeurIPS (2019)
31. Yoon, J., Jordon, J., van der Schaar, M.: PATE-GAN: Generating synthetic data with differential privacy guarantees. In: ICLR (2019)
32. Zhang, H., Zhang, J., Shen, Z., Srinivasan, B., Qin, X., Faloutsos, C., Rangwala, H., Karypis, G.: Mixed-type tabular data synthesis with score-based diffusion in latent space. In: ICLR (2024)
33. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: Private data release via bayesian networks. ACM Transactions on Database Systems **42**(4), 1–41 (Oct 2017)