TabDDPM: Modelling Tabular Data with Diffusion Models

Akim Kotelnikov [1 2] **Dmitry Baranchuk** [2] **Ivan Rubachev** [1 2] **Artem Babenko** [2]

Abstract

Denoising diffusion probabilistic models are becoming the leading generative modeling paradigm for many important data modalities. Being the most prevalent in the computer vision community, diffusion models have recently gained some attention in other domains, including speech, NLP, and graph-like data. In this work, we investigate if the framework of diffusion models can be advantageous for general tabular problems, where data points are typically represented by vectors of heterogeneous features. The inherent heterogeneity of tabular data makes it quite challenging for accurate modeling since the individual features can be of a completely different nature, i.e., some of them can be continuous and some can be discrete. To address such data types, we introduce TabDDPM — a diffusion model that can be universally applied to any tabular dataset and handles any feature types. We extensively evaluate TabDDPM on a wide set of benchmarks and demonstrate its superiority over existing GAN/VAE alternatives, which is consistent with the advantage of diffusion models in other fields. The source code of TabDDPM is available at GitHub.

1. Introduction

Denoising diffusion probabilistic models (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) have recently become an object of great research interest in the generative modeling community since they often outperform the alternative approaches both in terms of the realism of individual samples and their diversity (Dhariwal & Nichol, 2021). The most impressive successes of DDPM were demonstrated in the domain of natural images (Dhariwal & Nichol, 2021; Saharia et al., 2022; Rombach et al., 2022), where the advantages of diffusion models are successfully exploited in applications, such as colorization (Song et al., 2021), inpainting (Song et al., 2021), segmentation (Baranchuk et al., 2021), super-resolution (Saharia et al., 2021; Li et al., 2021), semantic editing (Meng et al., 2021) and others. Beyond computer vision, the DDPM framework is also investigated in other fields, such as NLP (Austin et al., 2021; Li et al., 2022), waveform signal processing (Kong et al., 2020; Chen et al., 2020b), molecular graphs (Jing et al., 2022; Hoogeboom et al., 2022), time series (Tashiro et al., 2021), testifying the universality of diffusion models across a wide range of problems.

Our work aims to investigate if the universality of DDPM can be extended to the case of general tabular problems, which are ubiquitous in various industrial applications that include data described by a set of heterogeneous features. For many such applications, the demand for high-quality generative models is especially acute because of the modern privacy regulations, like GDPR, which prevent publishing real user data, while the synthetic data produced by generative models can be shared. However, training a high-quality model of tabular data can be more challenging than in computer vision or NLP due to the heterogeneity of individual features and relatively small sizes of typical tabular datasets. This paper shows that despite these two intricacies, the diffusion models can successfully approximate typical distributions of tabular data, leading to state-of-the-art performance on most of the benchmarks. In more detail, the main contributions of this work are the following:

1. We introduce TabDDPM — a simple design of DDPM for tabular problems that can be applied to any tabular task and work with mixed data types including numerical and categorical features.

2. We demonstrate that TabDDPM outperforms the alternative approaches designed for tabular data, including GAN-based and VAE-based methods, and illustrate the sources of this advantage on several datasets.

3. We observe that shallow interpolation-based methods, e.g., SMOTE (Chawla et al., 2002), produce surprisingly effective synthetic data that provides competitively high ML efficiency. Compared with SMOTE, we show that TabDDPM's data is preferable for privacy-concerned scenarios when synthetic data are used to substitute the real user data that cannot be shared.

[1] HSE university, Moscow, Russia [2] Yandex, Moscow, Russia. Correspondence to: Akim Kotelnikov <ya@akotelnikov.ru>.

Proceedings of the 40^{th} *International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

2. Related Work

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) is a paradigm of generative modeling that aims to approximate the target distribution by the endpoint of the Markov chain, which starts from a given parametric distribution, typically a standard Gaussian. Each Markov step is performed by a deep neural network that effectively learns to invert the diffusion process with a known Gaussian kernel. Ho et al. demonstrated the equivalence of diffusion models and score matching (Song & Ermon, 2019; 2020), showing them to be two different perspectives on the gradual conversion of a simple known distribution into a target distribution via the iterative denoising process. Several recent works (Nichol, 2021; Dhariwal & Nichol, 2021) have developed more powerful model architectures as well as different advanced learning protocols, which led to the "victory" of DDPM over GANs in terms of generative quality and diversity in the computer vision field. In this work, we demonstrate that one can also successfully use diffusion models for tabular problems.

Generative models for tabular problems are currently an active research direction in the machine learning community since high-quality synthetic data is in great demand for many tabular tasks. First, the tabular datasets are often limited in size, unlike in vision or NLP problems, for which massive "extra" data is available on the Internet. Second, proper synthetic datasets do not contain actual user data. Therefore, they are not subject to GDPR-like regulations and can be publicly shared without violating anonymity. The recent works have developed a large number of models, including tabular VAEs (Xu et al., 2019) and GAN-based approaches (Xu et al., 2019; Engelmann & Lessmann, 2021; Jordon et al., 2018; Fan et al., 2020; Torfi et al., 2022; Zhao et al., 2021; Kim et al., 2021; Zhang et al., 2021; Nock & Guillame-Bert, 2022; Wen et al., 2022). By extensive evaluations on a large number of public benchmarks, we show that TabDDPM surpasses the existing alternatives, often by a large margin.

"Shallow" synthetics generation. Unlike unstructured images or natural texts, tabular data is typically structured, i.e., the individual features are often interpretable and it is unclear if their modeling requires several layers of "deep" architectures. Therefore, the simple interpolation techniques, like SMOTE (Chawla et al., 2002) (originally proposed to address class imbalance) can serve as simple and powerful solutions as demonstrated in (Camino et al., 2020), where SMOTE is shown to outperform tabular GANs for minor class oversampling. In the experiments, we demonstrate the advantage of TabDDPM's synthetics over synthetics produced with interpolation techniques from the privacy-preserving perspective.

3. Background

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are likelihood-based generative models that handle the data through forward and reverse Markov processes. The forward process $q\left(x_{1:T}|x_0\right) = \prod_{t=1}^{T} q\left(x_t|x_{t-1}\right)$ gradually adds noise to an initial sample x_0 from the data distribution $q\left(x_0\right)$ sampling noise from the predefined distributions $q\left(x_t|x_{t-1}\right)$ with variances $\{\beta_1, ..., \beta_T\}$.

The reverse process $p\left(x_{0:T}\right) = \prod_{t=1}^{T} p\left(x_{t-1}|x_t\right)$ gradually denoises a latent variable $x_T \sim q\left(x_T\right)$ and allows generating new data samples from $q(x_0)$. Distributions $p\left(x_{t-1}|x_t\right)$ are usually unknown and approximated by a neural network with parameters θ. These parameters are learned from the data by optimizing a variational lower bound:

$$\log q\left(x_0\right) \geq \mathbb{E}_{q(x_0)}\Big[\underbrace{\log p_\theta\left(x_0|x_1\right)}_{L_0} - \underbrace{KL\left(q\left(x_T|x_0\right)|q\left(x_T\right)\right)}_{L_T} - \\ - \sum_{t=2}^{T} \underbrace{KL\left(q\left(x_{t-1}|x_t, x_0\right)|p_\theta\left(x_{t-1}|x_t\right)\right)}_{L_t}\Big] \tag{1}$$

Gaussian diffusion models operate in continuous spaces ($x_t \in \mathbb{R}^n$) where forward and reverse processes are characterized by Gaussian distributions:

$$q\left(x_t|x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$
$$q\left(x_T\right) := \mathcal{N}\left(x_T; 0, I\right)$$
$$p_\theta\left(x_{t-1}|x_t\right) := \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right)$$

Ho et al. (2020) suggest using diagonal $\Sigma_\theta\left(x_t, t\right)$ with a constant σ_t and computing $\mu_\theta\left(x_t, t\right)$ as a function of x_t and $\epsilon_\theta(x_t, t)$:

$$\mu_\theta\left(x_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right)$$

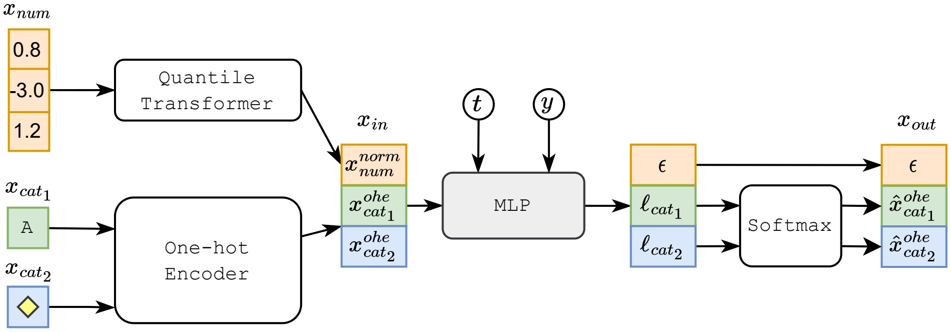
where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{i \leq t} \alpha_i$ and $\epsilon_\theta(x_t, t)$ predicts a "groundtruth" noise component ϵ for the noisy data sample x_t. In practice, the objective Equation 1 can be simplified to the sum of mean-squared errors between $\epsilon_\theta(x_t, t)$ and ϵ over all timesteps t:

$$L_t^{\text{simple}} = \mathbb{E}_{x_0, \epsilon, t}\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \tag{2}$$

Multinomial diffusion models (Hoogeboom et al., 2021) are designed to generate categorical data where $x_t \in \{0, 1\}^K$ is a one-hot encoded categorical variable with K values. The multinomial forward diffusion process defines $q\left(x_t|x_{t-1}\right)$ as a categorical distribution that corrupts the data by uniform noise over K classes:

$$q(x_t|x_{t-1}) := \text{Cat}\left(x_t; \left(1-\beta_t\right)x_{t-1} + \beta_t/K\right)$$
$$q\left(x_T\right) := \text{Cat}\left(x_T; 1/K\right)$$
$$q\left(x_t|x_0\right) = \text{Cat}\left(x_t; \bar{\alpha}_t x_0 + \left(1-\bar{\alpha}_t\right)/K\right)$$

Figure 1. TabDDPM scheme for classification problems; t, y and ℓ denote a diffusion timestep, a class label, and logits, respectively.



From the equations above, the posterior $q(x_{t-1}|x_t, x_0)$ can be derived:

$$q\left(x_{t-1}|x_t, x_0\right) = \text{Cat}\left(x_{t-1}; \pi / \sum_{k=1}^{K} \pi_k\right), \text{ where}$$

$$\pi = [\alpha_t x_t + (1 - \alpha_t)/K] \odot [\bar{\alpha}_{t-1} x_0 + (1 - \bar{\alpha}_{t-1})/K]$$

The reverse distribution $p_\theta\left(x_{t-1}|x_t\right)$ is parameterized as $q\left(x_{t-1}|x_t, \hat{x}_0(x_t, t)\right)$, where $\hat{x}_0$ is predicted by a neural network. Then, the model is trained to maximize the variational lower bound Equation 1.

4. TabDDPM

In this section, we describe the design of TabDDPM as well as its main hyperparameters, which affect the model's effectiveness.

TabDDPM uses the multinomial diffusion to model the categorical and binary features, and the Gaussian diffusion to model the numerical ones. In more detail, for a tabular data sample $x = [x_{\text{num}}, x_{\text{cat}_1}, ..., x_{\text{cat}_C}]$, that consists of N_{num} numerical features $x_{\text{num}} \in \mathbb{R}^{N_{\text{num}}}$ and C categorical features x_{cat_i} with K_i categories each, our model takes one-hot encoded versions of categorical features as an input (i.e. $x_{\text{cat}_i}^{\text{ohe}} \in \{0, 1\}^{K_i}$) and normalized numerical features. Therefore, the input x_0 has a dimensionality of $(N_{\text{num}} + \sum K_i)$. For preprocessing, we use the gaussian quantile transformation from the scikit-learn library (Pedregosa et al., 2011). Each categorical feature is handled by a separate forward diffusion process, i.e., the noise components for all features are sampled independently. The reverse diffusion step in TabDDPM is modeled by a multi-layer neural network that has an output of the same dimensionality as x_0, where the first N_{num} coordinates are the predictions of ϵ for the Gaussian diffusion and the rest are the predictions of $x_{\text{cat}_i}^{\text{ohe}}$ for the multinomial diffusions.

The TabDDPM model for the classification problems is schematically presented on Figure 1. The model is trained by minimizing a sum of mean-squared error L_t^{simple} (Equation 2) for the Gaussian diffusion term and the KL divergences L_t^i for each multinomial diffusion term (Equation 1). The total loss of multinomial diffusions is additionally divided by the number of categorical features.

$$L_t^{\text{TabDDPM}} = L_t^{\text{simple}} + \frac{\sum_{i \le C} L_t^i}{C} \tag{3}$$

For classification datasets, we use a class-conditional model, i.e., $p_\theta(x_{t-1}|x_t, y)$ is learned. For regression datasets, we consider a target value as an additional numerical feature, and the joint distribution is learned.

To model the reverse process, we use a simple MLP architecture adapted from (Gorishniy et al., 2021):

$$\begin{aligned} \texttt{MLP}(x) &= \texttt{Linear}\left(\texttt{MLPBlock}\left(\ldots\left(\texttt{MLPBlock}(x)\right)\right)\right) \\ \texttt{MLPBlock}(x) &= \texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(x))) \end{aligned} \tag{4}$$

As in (Nichol, 2021; Dhariwal & Nichol, 2021), a tabular input x_{in}, a timestep t, and a class label y are processed as follows:

$$\begin{aligned} t\_emb &= \texttt{Linear}(\texttt{SiLU}(\texttt{Linear}(\texttt{SinTimeEmb}(t)))) \\ y\_emb &= \texttt{Embedding}(y) \\ x &= \texttt{Linear}(x_{in}) + t\_emb + y\_emb \end{aligned} \tag{5}$$

where SinTimeEmb refers to a sinusoidal time embedding as in (Nichol, 2021; Dhariwal & Nichol, 2021) with a dimension of 128. All Linear layers in Equation 5 have a fixed projection dimension 128.

Hyperparameters in TabDDPM are essential since, in the experiments, we observed them having a strong influence on the model effectiveness. Table 1 lists the main hyperparameters and the search spaces for each of them, which we recommend using. The process of tuning is described in detail in the experimental section.

Table 1. The list of main hyperparameters for TabDDPM.

Hyperparameter	Search space
Learning rate	$\mathrm{LogUniform}[0.00001, 0.003]$
Batch size	$\mathrm{Cat}\{256, 4096\}$
Diffusion timesteps	$\mathrm{Cat}\{100, 1000\}$
Training iterations	$\mathrm{Cat}\{5000, 10000, 20000\}$
# MLP layers	$\mathrm{Int}\{2, 4, 6, 8\}$
MLP width of layers	$\mathrm{Int}\{128, 256, 512, 1024\}$
Proportion of samples	$\mathrm{Float}\{0.25, 0.5, 1, 2, 4, 8\}$
Dropout	0.0
Scheduler	cosine (Nichol, 2021)
Gaussian diffusion loss	MSE
Number of tuning trials	50

Table 2. Details on the datasets used in the evaluation.

Abbr	Name	# Train	# Validation	# Test	# Num	# Cat	Task type
AB	Abalone	2672	669	836	7	1	Regression
AD	Adult	26048	6513	16281	6	8	Binclass
BU	Buddy	12053	3014	3767	4	5	Multiclass
CA	California Housing	13209	3303	4128	8	0	Regression
CAR	Cardio	44800	11200	14000	5	6	Binclass
CH	Churn Modeling	6400	1600	2000	7	4	Binclass
DE	Default	19200	4800	6000	20	3	Binclass
DI	Diabetes	491	123	154	8	0	Binclass
FB	Facebook Comm. Vol.	157638	19722	19720	50	1	Regression
GE	Gesture Phase	6318	1580	1975	32	0	Multiclass
HI	Higgs Small	62751	15688	19610	28	0	Binclass
HO	House 16H	14581	3646	4557	16	0	Regression
IN	Insurance	856	214	268	3	3	Regression
KI	King	13832	3458	4323	17	3	Regression
MI	MiniBooNE	83240	20811	26013	50	0	Binclass
WI	Wilt	3096	775	968	5	0	Binclass

5. Experiments

In this section, we extensively evaluate TabDDPM against existing alternatives.

Datasets. For systematic investigation of the performance of tabular generative models, we consider a diverse set of 15 real-world public datasets. These datasets have various sizes, nature, number of features, and their distributions. Most datasets were previously used for tabular model evaluation in (Zhao et al., 2021; Gorishniy et al., 2021). The full list of datasets and their properties are presented in Table 2.

Baselines. Since the number of generative models proposed for tabular data is enormous, we evaluate TabDDPM only against the leading approaches from each paradigm of generative modeling. Also, we consider only the baselines with the published source code.

- **TVAE** (Xu et al., 2019) — the state-of-the-art variational auto-encoder for tabular data generation. To the best of our knowledge, there are no alternative VAE-like models that outperform TVAE and have public source code.
- **CTGAN** (Xu et al., 2019) — arguably the most popular and well-known GAN-based model for synthetic data generation.
- **CTABGAN** (Zhao et al., 2021) — a recent GAN-based model that is shown to outperform the existing tabular GANs on a diverse set of benchmarks. This approach cannot handle regression tasks.
- **CTABGAN+** (Zhao et al., 2022) — an extension of the **CTABGAN** model that was published in the very recent preprint. We are unaware of the GAN-based model for tabular data proposed after **CTABGAN+** and has a public source code.
- **SMOTE** (Chawla et al., 2002) — a "shallow" interpolation-based method that ”generates” a synthetic point as a convex combination of a real data point and its k-th nearest neighbor from the dataset. This method was originally proposed for minor class oversampling. Here, we generalize it to synthetic data generation as a simple sanity check, i.e., a new synthetic sample is ”generated” by interpolating two samples from the same class. For regression problems, we split data into two classes by the median of the target variable.

Evaluation measure. Our primary evaluation measure is *machine learning (ML) efficiency* (or utility) (Xu et al., 2019). In more detail, ML efficiency quantifies the performance of classification or regression models trained on synthetic data and evaluated on the real test set. Intuitively, models trained on high-quality synthetics should be competitive (or even superior) to models trained on real data. In our experiments, we use two evaluation protocols to compute ML efficiency. In the first protocol, which is more common in the literature (Xu et al., 2019; Zhao et al., 2021; Kim et al., 2022), we compute an average efficiency w.r.t. a set of diverse ML models (logistic regression, decision tree, and others). In the second protocol, we evaluate ML efficiency only w.r.t. the CatBoost model (Prokhorenkova et al., 2018), which is arguably the leading GBDT implementation providing state-of-the-art performance on tabular tasks (Gorishniy et al., 2021). In our experiments in subsection 5.2, we show that it is crucial to use the second protocol, while the first one can often be misleading.

Tuning process. To tune the hyperparameters of TabDDPM and the baselines, we use the Optuna library (Akiba et al., 2019). The tuning process is guided by the values of the ML efficiency (w.r.t. Catboost) of the generated synthetic data on a hold-out validation dataset (the score is averaged over five different sampling seeds). The search spaces for all hyperparameters of TabDDPM are reported in Table 1 (for baselines — in Appendix E). Additionally, we demonstrate that tuning the hyperparameters using the CatBoost guidance does not introduce any sort of "Catboost-biasedness",

Figure 2. The individual feature distributions for the real data and the data generated by TabDDPM, CTABGAN+, and TVAE. TabDDPM produces more realistic feature distributions than alternatives in most cases.

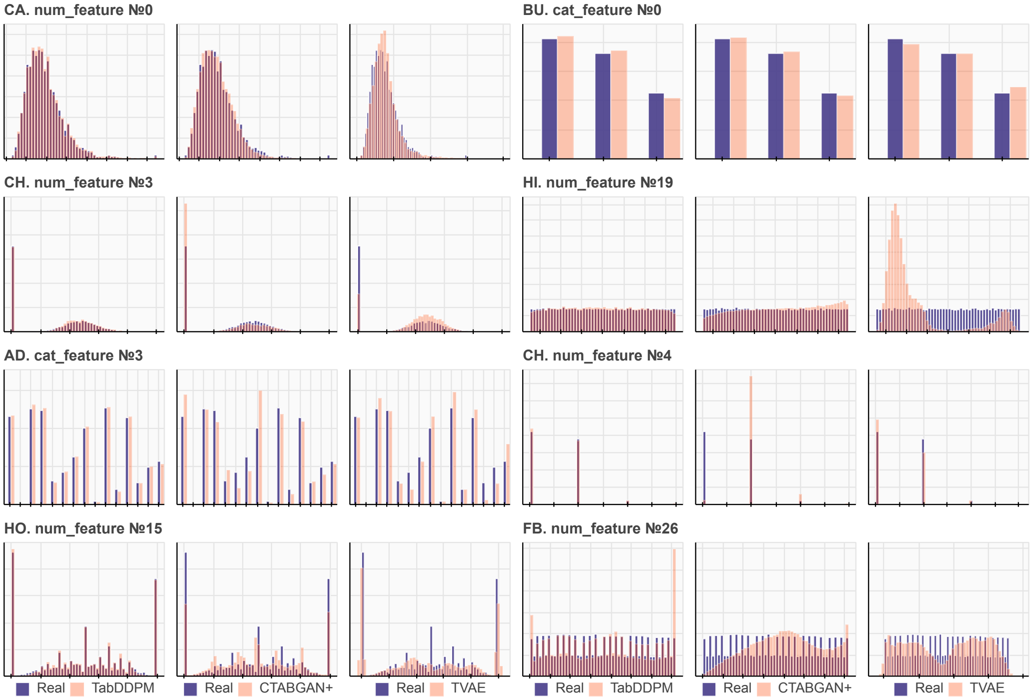
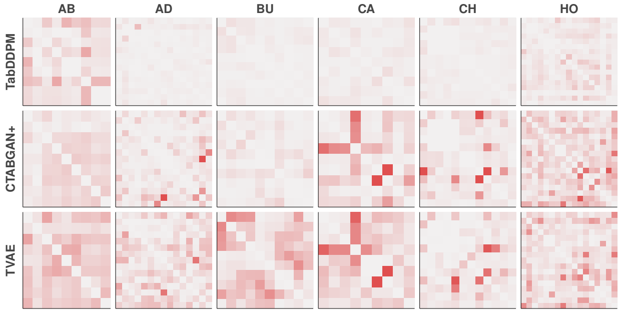


Figure 3. Absolute difference between correlation matrices computed on real and synthetic datasets. A more intensive red color indicates a higher difference between the real and synthetic correlation values. In most cases, TabDDPM captures feature correlations better.



and the Catboost-tuned TabDDPM produces synthetics that are also superior for other models, like MLP. These results are reported in Appendix A.

5.1. Qualitative comparison

Here, we qualitatively investigate the ability of TabDDPM to model the individual and joint feature distributions compared with the TVAE and CTABGAN+ baselines. In particular, for each dataset, we produce synthetic datasets from TabDDPM, TVAE, and CTABGAN+ of the same size as a real train set in a particular dataset. For classification datasets, each class is sampled according to its proportion in the real dataset. Then, we visualize the typical individual feature distributions for real and synthetic data in Figure 2. For completeness, the features of different types and distributions are presented.

In most cases, TabDDPM produces more realistic feature distributions compared with TVAE and CTABGAN+. The advantage is more pronounced (1) for numerical features, which are uniformly distributed, (2) for categorical features with high cardinality, and (3) for mixed-type features that combine continuous and discrete distributions. Then, we also visualize the differences between the correlation matrices computed on real and synthetic data for different datasets, see Figure 3. To compute the correlation matrices, we use the Pearson correlation coefficient for numerical-numerical correlations, the correlation ratio for categorical-numerical cases, and Theil's U statistic between categorical features. In comparison with CTABGAN+ and TVAE, TabDDPM generates synthetic datasets with more realistic pairwise correlations. These illustrations indicate that our TabDDPM model is more flexible than alternatives and produces superior synthetic data. We also follow (Zhao et al., 2021) and measure the Wasserstein distance between numerical features and the Jensen–Shannon divergence between categorical ones. Additionally, we report an L2 distance between correlation matrices (quantitative results for Fig-

Table 3. Average ranks (lower is better) over all datasets in terms of Wasserstein distance (WD) between numerical features, Jensen–Shannon divergence between categorical features and L2 distance between correlation matrices. Distances are calculated between generated data and real data.

	WD (Num.)	JS (Cat.)	L2 (Corr. matrix)
CTGAN	3.33	4.77	3.47
TVAE	4.20	3.92	4.40
CTABGAN+	3.87	2.54	3.40
SMOTE	**1.67**	2.15	2.00
TabDDPM	1.93	**1.62**	**1.73**

ure 3). The results are presented in Table 3 as an average rank across all datasets (lower is better). Lower rank indicates lower WD, JS divergence and L2 distance. The exact numbers can be found in Appendix B.

5.2. Machine Learning efficiency

In this section, we compare TabDDPM to alternative generative models in terms of machine learning efficiency. From each generative model, we sample a synthetic dataset with the size of a real train set in proportion from Table 1. This synthetic data is then used to train a classification/regression model, which is then evaluated using the real test set. In our experiments, classification performance is evaluated by the F1 score, and regression performance is evaluated by the R2 score. We use two protocols:

1. First, we compute average ML efficiency for a diverse set of ML models, as performed in previous works (Xu et al., 2019; Zhao et al., 2021; Kim et al., 2022). This set includes Decision Tree, Random Forest, Logistic Regression (or Ridge Regression) and MLP models from the scikit-learn library (Pedregosa et al., 2011) with the default hyperparameters except for: "max-depth" equals to 28 for Decision Tree and Random Forest, "maximum iterations" equals to 500 for Logistic and Ridge regressions, and "maximum iterations" equals to 100 for MLPs.
2. Second, we compute ML efficiency w.r.t. the current state-of-the-art model for tabular data. Specifically, we consider CatBoost (Prokhorenkova et al., 2018) and MLP architecture from (Gorishniy et al., 2021) for evaluation. CatBoost and MLP hyperparameters are thoroughly tuned on each dataset using the search spaces from (Gorishniy et al., 2021). We argue that this evaluation protocol demonstrates the practical value of synthetic data more reliably since in most real scenarios practitioners are not interested in using weak and suboptimal classifiers/regressors.

Main results. The ML efficiency values computed by both protocols are presented in Tables 4, 5. The ML efficiency for the tuned MLP is reported in Appendix A. To compute each value, we average the results over five random seeds for synthetics generation, and for each generated dataset, we average over ten random seeds for training classifiers/regressors. The key observations are described below:

- In both evaluation protocols, TabDDPM significantly outperforms TVAE and CTABGAN+ on most datasets, which highlights the advantage of diffusion models for tabular data as well as demonstrated for other domains in prior works.
- The interpolation-based SMOTE method demonstrates the performance competitive to TabDDPM and often significantly outperforms the GAN/VAE approaches. Interestingly, most of the prior works on generative models for tabular data do not compare against SMOTE, while it appears to be a simple baseline, which is challenging to beat.
- While many prior works use the first evaluation protocol to compute the ML efficiency, we argue that the second one (which uses the state-of-the-art model) is more appropriate. Tables 4, 5 show that the absolute values of classification/regression performance are much lower for the first protocol, i.e., weak classifiers/regressors are substantially inferior to CatBoost on the considered benchmarks. Therefore, one can hardly use these suboptimal models instead of CatBoost and their performance values are uninformative for practitioners. Moreover, in the first protocol, training on synthetic data is often advantageous compared to training on real data. This creates an impression that the data produced by generative models are more valuable than the real ones. However, it is not the case when one uses the tuned ML model, as in most practical scenarios. Appendix A confirms this observation for the properly tuned MLP model.

Overall, TabDDPM provides state-of-the-art generative performance and can be used as a source of high-quality synthetic data. Interestingly, in terms of ML efficiency, a simple "shallow" SMOTE method is competitive to TabDDPM, which raises the question if sophisticated deep generative models are needed. In the section below, we provide an affirmative answer to this question.

5.3. Privacy

Here, we investigate TabDDPM in privacy-concerned settings, e.g., sharing the data without disclosure of personal or sensitive information. In these setups, one is interested in high-quality synthetic data that does not reveal the records from the original dataset.

We measure the privacy of the generated data as a mean Dis-

Table 4. The values of machine learning efficiency computed w.r.t. five weak classification/regression models. Negative scores denote negative R2, which means that performance is worse than an optimal constant prediction.

	AB (*R2*)	AD (*F1*)	BU (*F1*)	CA (*R2*)	CAR (*F1*)	CH (*F1*)	DE (*F1*)	DI (*F1*)
TVAE	0.238±.012	0.742±.001	0.779±.004	−13.0±1.51	0.693±.002	0.684±.003	0.643±.003	0.712±.010
CTABGAN	–	0.737±.007	0.786±.008	–	0.684±.003	0.636±.010	0.614±.007	0.655±.015
CTABGAN+	0.316±.024	0.730±.007	0.837±.006	−7.59±.645	**0.708±.002**	0.650±.008	0.648±.008	**0.727±.023**
SMOTE	**0.400±.009**	0.750±.004	0.842±.003	0.667±.006	0.693±.001	0.690±.003	0.649±.003	0.677±.013
TabDDPM	**0.392±.009**	**0.758±.005**	**0.851±.003**	**0.695±.002**	0.696±.001	**0.693±.003**	**0.659±.003**	0.675±.011
Real	0.423±.009	0.750±.006	0.845±.004	0.663±.002	0.683±.002	0.679±.003	0.648±.003	0.699±.012
	FB (*R2*)	GE (*F1*)	HI (*F1*)	HO (*R2*)	IN (*R2*)	KI (*R2*)	MI (*F1*)	WI (*F1*)
TVAE	$\ll 0$	0.372±.006	0.590±.004	0.174±.012	0.470±.024	0.666±.006	**0.880±.002**	0.497±.001
CTABGAN	–	0.339±.009	0.539±.006	–	–	–	0.856±.003	0.656±.011
CTABGAN+	$\ll 0$	0.373±.009	0.598±.004	0.222±.042	0.669±.018	0.197±.051	0.867±.002	0.653±.027
SMOTE	**0.651±.002**	**0.478±.005**	0.664±.003	0.394±.006	0.709±.008	**0.751±.005**	0.860±.001	**0.793±.004**
TabDDPM	0.527±.005	0.462±.005	**0.670±.002**	**0.426±.007**	**0.734±.007**	0.611±.013	0.850±.004	**0.792±.004**
Real	0.645±.005	0.431±.005	0.663±.002	0.415±.007	0.708±.007	0.768±.013	0.850±.004	0.684±.004

Table 5. The values of machine learning efficiency computed w.r.t. the state-of-the-art tuned CatBoost model.

	AB (*R2*)	AD (*F1*)	BU (*F1*)	CA (*R2*)	CAR (*F1*)	CH (*F1*)	DE (*F1*)	DI (*F1*)
CTGAN	0.420±.004	0.789±.001	0.867±.003	0.686±.003	0.730±.001	0.723±.006	**0.699±.002**	0.459±.096
TVAE	0.433±.008	0.781±.002	0.864±.005	0.752±.001	0.717±.001	0.732±.006	0.656±.007	**0.714±.039**
CTABGAN	–	0.783±.002	0.855±.005	–	0.717±.001	0.688±.006	0.644±.011	**0.731±.022**
CTABGAN+	0.467±.004	0.772±.003	0.884±.005	0.525±.004	0.733±.001	0.702±.012	0.686±.004	**0.734±.020**
SMOTE	**0.549±.005**	0.791±.002	0.891±.003	**0.840±.001**	0.732±.001	0.743±.005	0.693±.003	0.683±.037
TabDDPM	**0.550±.010**	**0.795±.001**	**0.906±.003**	0.836±.002	**0.737±.001**	**0.755±.006**	0.691±.004	**0.740±.020**
Real	0.556±.004	0.815±.002	0.906±.002	0.857±.001	0.738±.001	0.740±.009	0.688±.003	0.785±.013
	FB (*R2*)	GE (*F1*)	HI (*F1*)	HO (*R2*)	IN (*R2*)	KI (*R2*)	MI (*F1*)	WI (*F1*)
CTGAN	0.443±.005	0.333±.013	0.575±.006	0.433±.005	0.745±.009	0.772±.005	0.783±.005	0.749±.015
TVAE	0.685±.003	0.434±.006	0.638±.003	0.493±.006	0.784±.010	0.824±.003	0.912±.001	0.501±.012
CTABGAN	–	0.392±.006	0.575±.004	–	–	–	0.889±.002	**0.906±.019**
CTABGAN+	0.509±.011	0.406±.009	0.664±.002	0.504±.005	0.797±.005	0.444±.014	0.892±.002	0.798±.021
SMOTE	**0.803±.002**	**0.658±.007**	**0.722±.001**	0.662±.004	**0.812±.002**	**0.842±.004**	0.932±.001	**0.913±.007**
TabDDPM	0.713±.002	0.597±.006	**0.722±.001**	**0.677±.010**	0.809±.002	**0.833±.014**	**0.936±.001**	**0.904±.009**
Real	0.837±.001	0.636±.007	0.724±.001	0.662±.003	0.814±.001	0.907±.002	0.934±.000	0.898±.006

tance to Closest Record (DCR) (Zhao et al., 2021). Specifically, for each synthetic sample, we get the minimum L2 distance to the real records. Mean DCR averages these distances over all generated samples.

Low DCR values indicate that synthetic samples essentially mimic some real datapoints and can violate privacy requirements. Higher DCR values denote that the generative model can produce "new" records rather than just near duplicates of the real data. Note that out-of-distribution data, e.g., random noise, will also provide high DCR. Therefore, DCR needs to be considered along with ML efficiency together.

Table 7 presents the DCR values for TabDDPM, SMOTE, CTABGAN+ and TVAE. We observe that TabDDPM is more private than SMOTE and less private than GAN/VAE alternatives. We attribute this to significantly lower ML utility of GAN/VAE-based baselines.

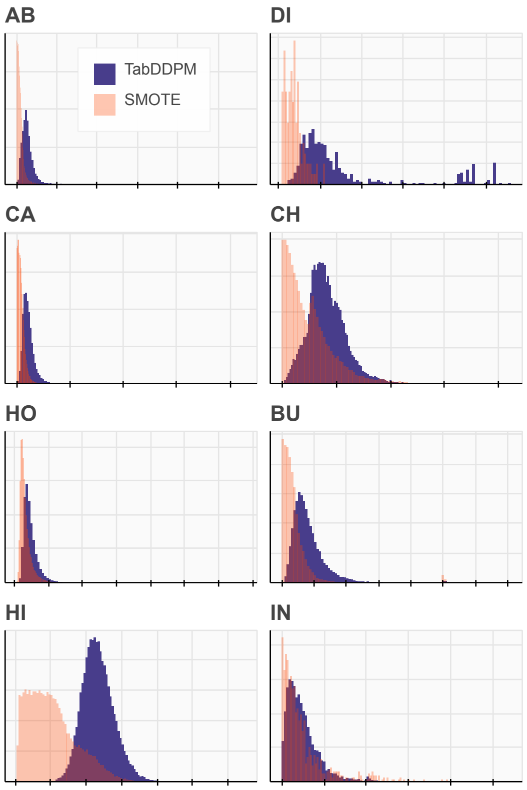
Since SMOTE computes convex combinations of the real records, the original DCR measure can pessimize SMOTE's privacy. To address this issue, we pretrain an MLP model on each dataset using real data. Then, we use this model to extract features from synthetic data and measure DCR in the latent space of the pretrained model. Table 14 provides mean DCR values on MLP features. The results are mostly consistent with Table 7 and do not alter our conclusions.

We also visualize histograms of the minimal synthetic-to-real distances in Figure 4. For SMOTE, most distance values are concentrated around zero, while TabDDPM samples are noticeably farther from real datapoints.

In addition, following (Chen et al., 2020a; Lee et al., 2021), we measure a success rate of a full black-box privacy attack (see Table 6). The attack aims to infer whether a record belongs to its original training data. The results show that

TabDDPM is more resistant to this full black-box attack than SMOTE. All these experiments confirm that TabDDPM significantly outperforms SMOTE in privacy-concerned scenarios and still provides state-of-the-art ML efficiency.

Figure 4. Histograms of minimal synthetic-to-real distances for TabDDPM and SMOTE. SMOTE values are concentrated around zero and, thus, SMOTE generates less private synthetic data.



Limitations and discussion

The proposed method does not pretend to be an all-in-one solution providing high privacy and high ML utility. Our experiments show that TabDDPM is more private than "shallow" SMOTE but do not give a definite answer if TabDDPM's data can satisfy real-world privacy-concerned applications. Therefore, the privacy problem of the DDPM-produced data needs to be further investigated. Moreover, DCR, used in this paper, is not an ultimate privacy measure and does not cover some critical use cases. For example, the L2 distance between records does not consider the importance of individual features and cannot detect leakage if some sensitive features coincide.

Table 6. Success rate of a full black-box privacy attack in terms of ROCAUC. A higher score indicates the higher success of attack. TabDDPM is significantly more robust than SMOTE.

	SMOTE	TabDDPM
AB	0.967	0.505
AD	0.619	0.511
BU	0.710	0.569
CA	0.986	0.516
CAR	0.721	0.506
CH	0.891	0.721
DE	0.679	0.497
DI	0.610	0.510
GE	0.864	0.533
HI	0.999	0.527
HO	0.826	0.546
IN	0.712	0.868
KI	0.748	0.517
MI	0.990	0.500
WI	0.954	0.516

Table 7. Comparison in terms of mean Distance to Closest Record (DCR) (higher is better). TabDDPM provides better DCR values compared with SMOTE but underperforms compared with TVAE and CTABGAN+. We attribute this to significantly lower ML efficiency of GAN/VAE-based alternatives.

	AB	AD	BU	CA	CAR	CH	DE	DI
TVAE	0.088	0.220	0.226	0.056	0.010	0.241	0.096	0.146
CTABGAN+	0.081	0.400	0.242	0.070	0.020	0.235	0.131	0.204
SMOTE	0.018	0.082	0.080	0.016	0.007	0.099	0.054	0.074
TabDDPM	0.061	0.295	0.168	0.045	0.016	0.166	0.061	0.308

	FB	GE	HI	HO	IN	KI	MI	WI
TVAE	1.418	0.171	0.497	0.127	0.102	0.200	0.025	0.020
CTABGAN+	0.666	0.169	0.533	0.129	0.124	0.390	10.761	0.027
SMOTE	0.264	0.041	0.209	0.066	0.050	0.090	0.012	0.009
TabDDPM	0.785	0.076	0.473	0.096	0.050	0.252	0.574	0.023

Also, in our work, we process categorical features using multinomial diffusion. However, alternative approaches exist, e.g., (Chen et al., 2022; Campbell et al., 2022; Zheng & Charoenphakdee, 2022). Each of these techniques is applicable to TabDDPM and can be an interesting direction to investigate. As for numerical features, the possible extension of TabDDPM can be inspired by (Nazabal et al., 2020) that distinguish different types of numerical variables, i.e., real-valued, positive real-valued or ordinal.

6. Conclusion

In this paper, we have investigated the prospect of the diffusion modeling framework in the field of tabular data. In particular, we describe the DDPM design that can handle mixed data types consisting of numerical and categorical features. For the most considered benchmarks, the synthetic data produced by TabDDPM has consistently higher quality compared with the GAN/VAE-based rivals. Interestingly, shallow interpolation techniques like SMOTE have demonstrated competitive ML utility and need to be considered as a simple yet effective baseline. Nevertheless, TabDDPM outperforms SMOTE for the setups where the privacy of the data must be ensured.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Baldi, P., Sadowski, P., and Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5, 2014.

Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., and Babenko, A. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

Camino, R. D., Hammerschmidt, C. A., et al. Oversampling tabular data with deep generative models: Is it worth the effort? 2020.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chen, D., Yu, N., Zhang, Y., and Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pp. 343–362, 2020a.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020b.

Chen, T., Zhang, R., and Hinton, G. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. 2021.

Engelmann, J. and Lessmann, S. Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, 174:114582, 2021.

Fan, J., Liu, T., Li, G., Chen, J., Shen, Y., and Du, X. Relational data synthesis using generative adversarial networks: A design space exploration. *arXiv preprint arXiv:2008.12763*, 2020.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729*, 2022.

Jordon, J., Yoon, J., and Van Der Schaar, M. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Kelley Pace, R. and Barry, R. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

Kim, J., Jeon, J., Lee, J., Hyeong, J., and Park, N. Oct-gan: Neural ode-based conditional tabular gans. In *Proceedings of the Web Conference 2021*, pp. 1506–1515, 2021.

Kim, J., Lee, C., Shin, Y., Park, S., Kim, M., Park, N., and Cho, J. Sos: Score-based oversampling for tabular data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 762–772, 2022.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *KDD*, 1996.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Lee, J., Hyeong, J., Jeon, J., Park, N., and Cho, J. Invertible tabular gans: Killing two birds with one stone for tabular data synthesis. *Advances in Neural Information Processing Systems*, 34:4263–4273, 2021.

Li, H., Yang, Y., Chang, M., Feng, H., Xu, Z., Li, Q., and Chen, Y. Srdiff: Single image super-resolution with diffusion probabilistic models. 2021.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

Madeo, R. C. B., Lima, C. A. M., and Peres, S. M. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC*, 2013.

Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. Sdedit: Image synthesis and editing with stochastic differential equations. 2021.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pp. 7176–7185. PMLR, 2020.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

Nichol, Alex & Dhariwal, P. Improved denoising diffusion probabilistic models. *ICML*, 2021.

Nock, R. and Guillame-Bert, M. Generative trees: Adversarial and copycat. *ICML*, 2022.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. 2021.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Singh, K., Sandhu, R. K., and Kumar, D. Comment volume prediction using neural networks and decision trees. In *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim*, 2015.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *NeurIPS*, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. 2021.

Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

Torfi, A., Fox, E. A., and Reddy, C. K. Differentially private synthetic medical data generation using convolutional gans. *Information Sciences*, 586:485–500, 2022.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. Openml: networked science in machine learning. *arXiv*, 1407.7722v1, 2014.

Wen, B., Cao, Y., Yang, F., Subbalakshmi, K., and Chandramouli, R. Causal-tgan: Modeling tabular data using causally-aware gan. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhang, Y., Zaidi, N. A., Zhou, J., and Li, G. Ganblr: a tabular data generation model. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 181–190. IEEE, 2021.

Zhao, Z., Kunar, A., Birke, R., and Chen, L. Y. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pp. 97–112. PMLR, 2021.

Zhao, Z., Kunar, A., Birke, R., and Chen, L. Y. Ctab-gan+: Enhancing tabular data synthesis. *arXiv preprint arXiv:2204.00401*, 2022.

Zheng, S. and Charoenphakdee, N. Diffusion models for missing value imputation in tabular data. *arXiv preprint arXiv:2210.17128*, 2022.

Appendix

A. MLP evaluation and tuning

Here, we show that tuning the hyperparameters using the CatBoost guidance results in the TabDDPM models that produce synthetics that is also optimal for other classifiers/regressors. The results for a subset of datasets are presented on Table 8. The methods denoted with ”-CB” and ”-MLP” denote the CatBoost guidance and different types of evaluation (CatBoost and MLP, respectively). The ”-MLP-tune” suffix stands for the MLP guidance tuning and MLP evaluation.

Table 8. ML utility score with MLP evaluation and MLP tuning compared with CatBoost evaluation and CatBoost tuning. The table shows that tuning with CatBoost model provides useful synthetic for MLP.

	AB ($R2$)	AD ($F1$)	BU ($F1$)	CA ($R2$)	CAR ($F1$)	CH ($F1$)	DE ($F1$)	DI ($F1$)
TabDDPM-CB	0.550±.010	0.795±.001	0.906±.003	0.836±.002	0.737±.001	0.755±.006	0.691±.004	0.740±.020
Real-CB	0.556±.004	0.815±.002	0.906±.002	0.857±.001	0.738±.001	0.740±.009	0.688±.003	0.785±.013
TabDDPM-MLP	0.569±.010	0.794±.002	0.903±.003	0.809±.003	0.737±.001	0.750±.005	0.679±.008	0.754±.020
Real-MLP	0.581±.005	0.795±.001	0.905±.003	0.808±.002	0.739±.001	0.741±.006	0.688±.004	0.754±.017
TabDDPM-MLP-tune	0.559±.009	0.792±.002	0.901±.003	0.803±.004	0.737±.001	0.749±.006	0.674±.013	0.741±.018

	FB ($R2$)	GE ($F1$)	HI ($F1$)	HO ($R2$)	IN ($R2$)	KI ($R2$)	MI ($F1$)	WI ($F1$)
TabDDPM-CB	0.713±.002	0.597±.006	0.722±.001	0.677±.010	0.809±.002	0.833±.014	0.936±.001	0.904±.009
Real-CB	0.837±.001	0.636±.007	0.724±.001	0.662±.003	0.814±.001	0.907±.002	0.934±.000	0.898±.006
TabDDPM-MLP	–	0.595±.006	0.717±.002	0.643±.010	0.794±.008	0.804±.015	0.938±.001	0.921±.006
Real-MLP	–	0.607±.007	0.717±.002	0.614±.006	0.800±.003	0.882±.004	0.936±.001	0.905±.006
TabDDPM-MLP-tune	–	–	–	0.626±.009	0.800±.004	0.799±.018	–	0.914±.006

B. Additional results

Here, we follow (Zhao et al., 2021) and provide an additional quantitative comparison that shows how well individual feature distributions are modeled (Table 9, Table 10, Table 11). Also, we include density and coverage metrics from (Naeem et al., 2020) that are improved alternatives of precision and recall, respectively (Table 12, Table 13).

Table 9. Wasserstein distance between numerical features.

	AB	AD	BU	CA	CAR	CH	DE	DI
CTGAN	0.008	0.010	0.015	0.004	0.004	0.009	0.004	0.085
TVAE	0.020	0.016	0.039	0.007	0.027	0.049	0.009	0.044
CTABGAN+	0.008	0.011	0.016	0.019	0.003	0.046	0.022	0.016
SMOTE	**0.002**	0.003	0.005	**0.002**	0.001	0.006	**0.002**	0.020
TabDDPM	0.005	**0.002**	**0.003**	**0.002**	**0.000**	**0.005**	0.012	**0.008**

	FB	GE	HI	HO	IN	KI	MI	WI
CTGAN	0.004	0.010	**0.003**	0.005	0.021	0.022	0.004	0.013
TVAE	0.008	0.009	0.076	0.007	0.025	0.012	0.004	0.016
CTABGAN+	0.078	0.007	0.052	0.008	0.025	0.021	0.006	0.006
SMOTE	**0.000**	**0.004**	0.009	0.005	0.011	**0.004**	**0.000**	**0.002**
TabDDPM	0.089	0.011	0.003	**0.004**	**0.006**	0.014	0.001	**0.002**

Table 10. Jensen-Shannon divergence between categorical features.

	AB	AD	BU	CA	CA	CH	DE	DI
CTGAN	0.276	0.085	0.168	nan	0.076	0.039	0.120	nan
TVAE	0.027	0.095	0.072	nan	0.181	0.019	0.157	nan
CTABGAN+	0.035	0.052	0.037	nan	**0.009**	0.018	0.030	nan
SMOTE	**0.005**	0.074	0.072	nan	0.069	0.030	0.058	nan
TabDDPM	0.007	**0.019**	**0.026**	nan	0.011	**0.017**	**0.009**	nan

	FB	GE	HI	HO	IN	KI	MI	WI
CTGAN	**0.017**	nan	nan	nan	0.071	0.296	nan	nan
TVAE	0.246	nan	nan	nan	0.033	0.098	nan	nan
CTABGAN+	0.051	nan	nan	nan	0.023	**0.044**	nan	nan
SMOTE	0.027	nan	nan	nan	0.013	0.102	nan	nan
TabDDPM	0.046	nan	nan	nan	**0.008**	0.060	nan	nan

Table 11. L2 distance between correlation matrices.

	AB	AD	BU	CA	CA	CH	DE	DI
CTGAN	0.471	0.390	0.492	0.606	0.712	0.239	1.355	1.735
TVAE	0.517	0.636	0.569	0.753	2.437	0.564	1.965	0.736
CTABGAN+	0.283	0.576	0.164	0.749	0.738	0.727	1.496	**0.435**
SMOTE	**0.185**	0.482	0.245	0.127	0.599	**0.147**	**0.642**	0.838
TabDDPM	0.333	**0.133**	**0.068**	**0.090**	**0.202**	0.161	0.934	0.186

	FB	GE	HI	HO	IN	KI	MI	WI
CTGAN	5.651	5.301	1.413	0.742	0.196	1.530	43.815	0.538
TVAE	5.960	2.996	2.759	0.902	0.224	1.004	44.692	0.550
CTABGAN+	6.782	1.977	1.241	0.978	0.207	3.898	31.704	0.319
SMOTE	**1.596**	**0.560**	0.354	0.452	0.301	**0.569**	**0.258**	**0.059**
TabDDPM	16.120	1.192	**0.233**	**0.336**	**0.077**	3.623	9.185	0.375

Table 12. Density of synthetic data.

	AB	AD	BU	CA	CA	CH	DE	DI
CTGAN	0.224	0.708	0.780	0.586	0.938	0.865	0.698	0.238
TVAE	0.347	1.126	1.032	0.746	0.845	1.043	0.808	**1.565**
CTABGAN+	0.380	0.867	0.998	0.569	0.957	0.974	0.730	0.974
SMOTE	**1.389**	**1.415**	**1.226**	**1.329**	**1.200**	**1.238**	**1.282**	1.413
TabDDPM	0.904	1.008	1.116	1.027	1.011	1.148	0.810	0.831

	FB	GE	HI	HO	IN	KI	MI	WI
CTGAN	0.147	0.035	0.702	0.467	0.927	0.719	0.361	0.763
TVAE	0.005	0.248	0.960	0.604	1.072	0.868	0.747	0.919
CTABGAN+	0.187	0.448	0.730	0.565	1.052	0.186	0.110	0.831
SMOTE	**0.926**	**1.531**	**1.682**	**1.595**	**1.213**	**1.335**	**1.308**	**1.251**
TabDDPM	0.633	1.460	1.152	1.195	1.150	0.884	0.972	1.009

Table 13. Coverage of synthetic data.

	AB	AD	BU	CA	CA	CH	DE	DI
CTGAN	0.654	0.948	0.966	0.759	0.920	1.000	0.777	0.572
TVAE	0.769	0.886	0.585	0.922	0.208	0.991	0.672	0.978
CTABGAN+	0.960	0.951	0.999	0.459	0.960	0.830	0.841	**1.000**
SMOTE	**1.000**	0.970	0.968	**1.000**	0.866	1.000	0.962	0.841
TabDDPM	**1.000**	**0.994**	**1.000**	0.998	**0.978**	**1.000**	**0.967**	0.955

	FB	GE	HI	HO	IN	KI	MI	WI
CTGAN	0.238	0.029	0.871	0.839	0.986	0.739	0.576	0.986
TVAE	0.014	0.669	0.255	0.875	0.987	0.874	0.823	0.867
CTABGAN+	0.222	0.640	0.557	0.952	**1.000**	0.479	0.241	0.994
SMOTE	**0.928**	**1.000**	**0.999**	**1.000**	0.995	0.945	**0.991**	**1.000**
TabDDPM	0.782	0.997	0.980	**1.000**	**1.000**	**0.969**	0.956	**1.000**

C. Additional visualizations

Figure 5. The individual feature distributions for the real data and the data generated by TabDDPM, CTABGAN+, and TVAE. TabDDPM often models feature distributions more accurately than CTABGAN+ and TVAE.

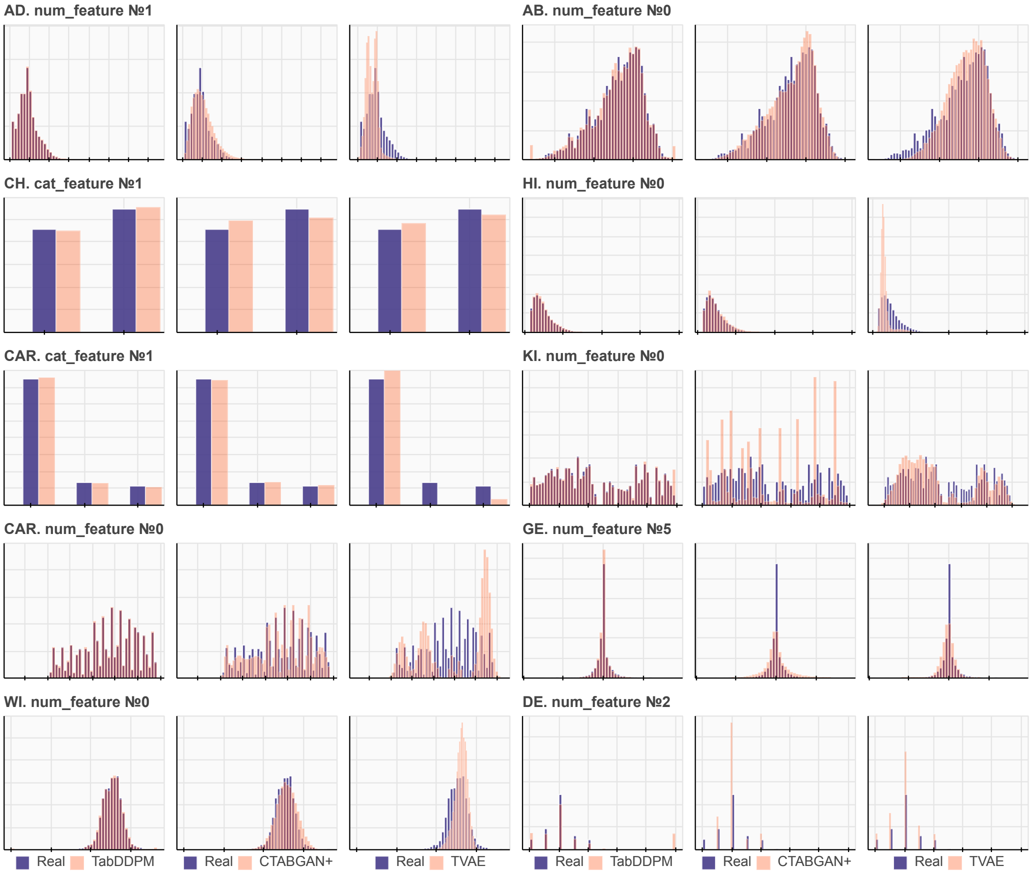
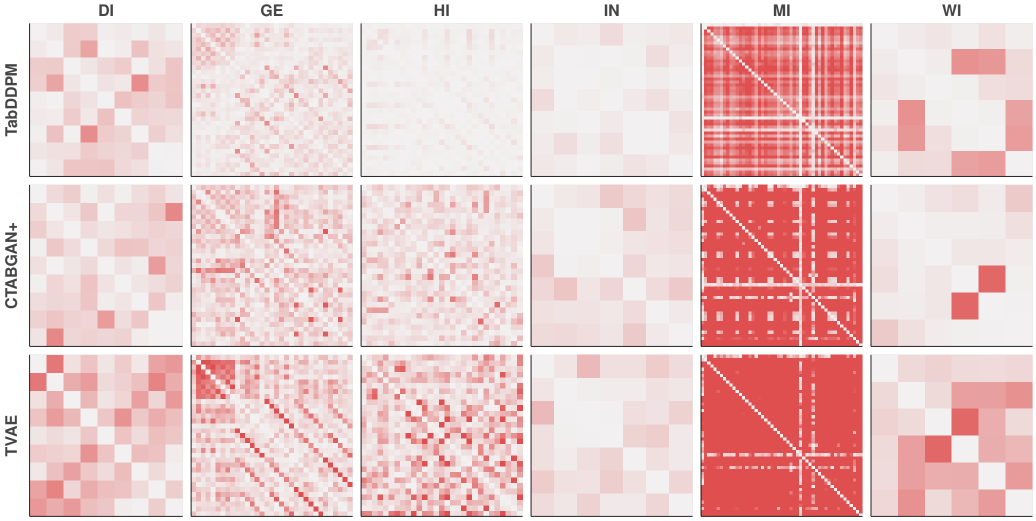


Figure 6. The absolute difference between correlation matrices computed on real and synthetic datasets. More intense red color indicates higher difference. Overall, TabDDPM captures correlations better.



D. Distance to Closest Record using pretrained MLP features

This section addresses the problem that DCR in the original feature space can be an unsuitable privacy measure for SMOTE. We pretrain the feature extractor on each dataset and compute DCR in the latent space of the MLP model. According to the results in Table 14, DCR calculated on MLP features brings similar conclusions to Table 7. SMOTE still significantly underperforms compared with TabDDPM.

Table 14. Comparison in terms of mean Distance to Closest Record (DCR) calculated on pretrained MLP features (higher is better). The results are consistent with Table 7.

	AB	AD	BU	CA	CAR	CH	DE	DI	FB	GE	HI	HO	IN	KI	MI	WI
TVAE	0.282	1.055	0.381	0.373	0.173	2.869	0.271	0.508	3.642	5.484	3.256	0.393	0.276	0.513	0.449	0.45
CTABGAN+	0.257	1.466	0.382	0.332	0.177	2.998	0.366	0.669	11.44	5.375	4.396	0.365	0.305	0.833	12.026	0.76
SMOTE	0.081	0.526	0.216	0.200	0.147	1.367	0.172	0.409	1.045	1.673	2.657	0.332	0.162	0.294	0.374	0.377
TabDDPM	0.195	1.246	0.330	0.290	0.160	2.240	0.168	1.232	30.46	3.85	3.557	0.336	0.172	0.889	7.993	0.620

E. Hyperparameters Search Spaces

Table 15. CatBoost hyperparameters space from (Gorishniy et al., 2021)

Parameter	Distribution
Max depth	$\mathrm{UniformInt}[3, 10]$
Learning rate	$\mathrm{LogUniform}[1e\text{-}5, 1]$
Bagging temperature	$\mathrm{Uniform}[0, 1]$
L2 leaf reg	$\mathrm{LogUniform}[1, 10]$
Leaf estimation iterations	$\mathrm{UniformInt}[1, 10]$
Number of tuning trials	100

Table 16. MLP hyperparameters space from (Gorishniy et al., 2021)

Parameter	Distribution
# Layers	$\mathrm{UniformInt}[1, 8]$
Layer size	$\mathrm{Int}\{64, 128, 256, 512, 1024\}$
Dropout	$\{0, \mathrm{Uniform}[0, 0.5]\}$
Learning rate	$\mathrm{LogUniform}[1e\text{-}5, 1e\text{-}2]$
Weight decay	$\{0, \mathrm{LogUniform}[1e\text{-}6, 1e\text{-}3]\}$
Number of tuning trials	100

Table 17. SMOTE hyperparameters search space. λ_{range} denotes the range of interpolation coefficient to sample from

Parameter	Distribution
k_neighbours	$\mathrm{Int}[5, 20]$
λ_{range}	$\mathrm{Float}[0, 1]$
Proportion of samples	$\mathrm{Float}\{0.25, 0.5, 1, 2, 4, 8\}$
Number of tuning trials	50

[4] https://github.com/Team-TUD/CTAB-GAN-Plus

[4] https://github.com/sdv-dev/CTGAN

Table 18. CTABGAN and CTABGAN+ hyperparameters search space. See an official implementation[2]

Parameter	Distribution
# claassif. layers	$\mathrm{UniformInt}[1, 4]$
Classif. layer size	$\mathrm{Int}\{64, 128, 256\}$
Training iterations	$\mathrm{Cat}\{1000, 5000, 10000\}$
Batch Size	$\mathrm{Int}\{512, 1024, 2048\}$
random_dim	$\mathrm{Int}\{16, 32, 64, 128\}$
num_channels	$\mathrm{Int}\{16, 32, 64\}$
Proportion of samples	$\mathrm{Float}\{0.25, 0.5, 1, 2, 4, 8\}$
Number of tuning trials	35

Table 19. TVAE hyperparameters search space. See an official implementation[4]

Parameter	Distribution
# claassif. layers	$\mathrm{UniformInt}[1, 6]$
Classif. layer size	$\mathrm{Int}\{64, 128, 256, 512\}$
Training iterations	$\mathrm{Cat}\{5000, 20000, 30000\}$
Batch Size	$\mathrm{Cat}\{456, 4096\}$
embedding_dim	$\mathrm{Int}\{16, 32, 64, 128, 256, 512, 1024\}$
loss factor	$\mathrm{LogUniform}[0.01, 10]$
Proportion of samples	$\mathrm{Float}\{0.25, 0.5, 1, 2, 4, 8\}$
Number of tuning trials	50

F. Datasets

We used the following datasets:

- Abalone (OpenML)
- Adult (income estimation, (Kohavi, 1996))
- Buddy (Kaggle)
- California Housing (real estate data, (Kelley Pace & Barry, 1997))
- Cardiovascular Disease dataset (Kaggle)
- Churn Modeling (Kaggle)
- Diabetes (OpenML)
- Facebook Comments Volume (Singh et al., 2015)
- Gesture Phase Prediction (Madeo et al., 2013)
- Higgs (simulated physical particles, (Baldi et al., 2014); we use the version with 98K samples available at the OpenML repository (Vanschoren et al., 2014))
- House 16H (OpenML)
- Insurance (Kaggle)
- King (Kaggle)
- MiniBooNE (OpenML)
- Wilt (OpenML)

G. Environment and Runtime

Experiments were conducted under Ubuntu 20.04 on a machine equipped with GeForce RTX 2080 Ti GPU and Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz. We used Pytorch 10.1, CUDA 11.3, scikit-learn 1.1.2 and imbalanced-learn 0.9.1 (for SMOTE).

As for runtime of the proposed method, it depends on the dataset and hyperparameters. We provide 3 examples below. All three examples use $T = 1000$ and $batch\_size = 4096$. Note that hyperparameters tuning contains 50 runs and takes usually 8-10 hours. "Sample time" is for the all *n_to_sample* number of samples.

Table 20. Training and sampling time for TabDDPM.

Dataset	input_dim	model_layers	train_steps	n_to_sample	train_time	sample_time
CH	16	[256,1024,1024, 1024,1024,512]	30k	26k	670s	6s
HI	28	[512,1024,1024, 1024,1024,512]	30k	502k	502s	430s
FB	146	[512,1024]	30k	1264k	783s	470s