TabDiff: a Unified Diffusion Model for Multi-Modal Tabular Data Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Synthesizing high-quality tabular data is an important topic in many data science applications, ranging from dataset augmentation to privacy protection. However, developing expressive generative models for tabular data is challenging due to its inherent heterogeneous data types and intricate column-wise distributions. In this paper, we introduce TabDiff, a unified diffusion framework that models all multi-modal distributions of mixed-type tabular data in one model. Our key insight is to design different continuous-time diffusion processes for numerical and categorical data, and learn one model to simultaneously predict the noise for different modalities. To counter the high disparity of different feature distributions, we further introduce feature-wise learnable diffusion processes to optimally balance the generative performance. The entire framework can be efficiently optimized in an end-to-end fashion. Comprehensive experiments on seven datasets demonstrate that TabDiff achieves superior average performance over existing competitive baselines across five out of six metrics.

1 Introduction

Tabular data generation is a fundamental and important problem in many data processing and analysis tasks, such as training data augmentation (Fonseca & Bacao, 2023), data privacy protection (Assefa et al., 2021; Hernandez et al., 2022), and missing value imputation (You et al., 2020; Zheng & Charoenphakdee, 2022). The problem is highly challenging due to the inherent heterogeneous data types and intricate column-wise distributions. In the past few years, numerous deep generative models have been proposed for tabular data generation with autoregressive models (Borisov et al., 2023), VAEs (Liu et al., 2023), and GANs (Xu et al., 2019). Recently, with the rapid progress in diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022), researchers have also explored extending the framework for tabular data (Kim et al., 2022; Kotelnikov et al., 2023; Zhang et al., 2024). However, the advanced diffusion models are mainly designed for continuous data with Gaussian perturbation and cannot handle tabular categorical features. Existing methods typically rely on transforming these features into continuous space via various encoding techniques (Zheng & Charoenphakdee, 2022; Zhang et al., 2024) or learning separate discrete-time diffusion processes (Kotelnikov et al., 2023; Lee et al., 2023). However, it has been shown that these solutions either are trapped with suboptimal performance due to encoding overhead or cannot capture complex co-occurrence patterns of different modalities because of low model capacity. As a result, we seek to develop a unified and expressive diffusion model in the joint space of continuous and discrete features.

In this paper, we present TabDiff, a unified diffusion framework for tabular data generation. To handle heterogeneous data types, we propose a novel continuous-time diffusion process that perturbs numerical and categorical features jointly with continuous and discrete noise, and learn one model to simultaneously predict the noise for different modalities. To counteract the high heterogeneity in feature distributions, we further develop principled feature-wise learnable diffusion processes to

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

optimally allocate the generative capacity. We parameterize TabDiff with transformers processing different input types and optimize the entire framework efficiently in an end-to-end fashion. We conduct comprehensive experiments by comparing TabDiff with eight state-of-the-art methods on seven widely adopted tabular benchmarks. The experimental results demonstrate that TabDiff consistently outperforms previous methods over five out of six distinct evaluation metrics, suggesting our superior generative capacity on mixed-type tabular data.

2 Method

2.1 Overview

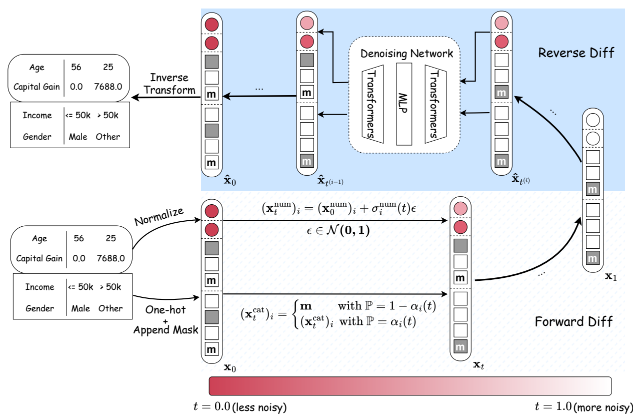

Figure 1: A high-level overview of TabDiff. TabDiff operates by normalizing numerical columns and converting categorical columns into one-hot vectors with an extra [MASK] class. Distinct forward diffusion processes are applied to each type, with each column's noise rate controlled by customized, learned schedules. News samples are generated via reverse diffusion, with the denoising network gradually denoising $\mathbf{x}_1$ into $\hat{\mathbf{x}}_0$ and followed by the inverse transform to recover the original format.

Notations. For a given mixed-type tabular dataset $\mathcal{T}$, we denote the number of numerical features and categorical features as M_{num} and M_{cat}, respectively. The dataset is represented as a collection of data entries $\mathcal{T} = \{\mathbf{x}\} = \{[\mathbf{x}^{\text{num}}, \mathbf{x}^{\text{cat}}]\}$, where each data entry $\mathbf{x}$ is a concatenated vector consisting of its numerical features x^{num} and categorical features $\mathbf{x}^{\text{cat}}$. We represent the i-th numerical feature as $\mathbf{x}_i^{\text{num}} \in \mathbb{R}$, and represent the j-th categorical feature as $\mathbf{x}_j^{\text{cat}} \in \{1, ..., C_j\}$ with C_j finite categories. Hence, we have $\mathbf{x}^{\text{num}} \in \mathbb{R}^{M_{\text{num}}}$ and $\mathbf{x}^{\text{cat}} \in \prod_{j=1}^{M_{\text{cat}}} \{1, ..., C_j\}$.

Different from common data types such as images and text, developing generative models for tabular data is challenging as the distribution is determined by multi-modal data. We therefore propose TabDiff, a unified generative model for modeling the joint distribution $p(\mathbf{x})$ using a continuous-time diffusion framework. TabDiff can learn the distribution from finite samples and generate faithful, diverse, and novel samples unconditionally. We provide a high-level overview in Figure 1, which includes a forward *diffusion* process and a reverse *generative* process, both defined in continuous time. The diffusion process gradually adds noise to data, and the generative process learns to recover the data from prior noise distribution with neural networks parameterized by θ.

2.2 Unified Diffusion Model

Our unified diffusion framework is designed to directly operate on the data space and naturally handle each tabular column in its built-in datatype, both numerical and categorical. To counter the disparity in these datatypes, we thus introduce a hybrid forward process that gradually increases noise in both numerical and categorical column types with two different diffusion schedules $\boldsymbol{\sigma}$. Let $\{\mathbf{x}_t\}_{t=[0,1]}$ denote a sequence of data in the diffusion process indexed by a continuous time variable $t \in [0, 1]$, where $\mathbf{x}_0 \sim p_0$ are *i.i.d.* samples from real data distribution and $\mathbf{x}_1 \sim p_1$ are pure noise from prior distribution. The hybrid forward diffusion process can be then represented as (Ho et al., 2020):

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = q\left(\mathbf{x}_t^{\text{num}} \mid \mathbf{x}_0^{\text{num}}, \boldsymbol{\sigma}_{\text{num}}(t)\right) \cdot q\left(\mathbf{x}_t^{\text{cat}} \mid \mathbf{x}_0^{\text{cat}}, \boldsymbol{\sigma}_{\text{cat}}(t)\right). \tag{1}$$

Gaussian Diffusion for Numerical Features, The forward diffusion for continuous features is formulated as the solution to a stochastic differential equation (SDE) $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathrm{t})\mathrm{dt} + \mathrm{g}(\mathrm{t})\mathrm{d}\mathbf{w}$, where $\mathbf{f}(\cdot,t):\mathbb{R}^{M_{\text{num}}}\to\mathbb{R}^{M_{\text{num}}}$ is the drift coefficient, $g(\cdot):\mathbb{R}\to\mathbb{R}$ is the diffusion coefficient, and $\boldsymbol{w}$ is the standard Wiener process (a.k.a, Brownian motion). The reverse process can be formulated as a probability flow ordinary differential equation (ODE) $\mathrm{d}\mathbf{x}=\left[\mathbf{f}(\mathbf{x},\mathrm{t})-\frac{1}{2}\mathrm{g}(\mathrm{t})^2\nabla_{\mathbf{x}}\log\mathrm{p_t}(\mathbf{x})\right]\mathrm{dt}$, where $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ is the score function of $\mathbf{x}$ and this yields the backward trajectory of $\mathbf{x}$ as t goes from 1 to 0 (Song et al., 2021) . In this paper, we use the VE formulation (Song & Ermon, 2019; Song et al., 2021; Karras et al., 2022) with $\mathbf{f}(\cdot,t)=\mathbf{0}$ and $g(t)=\sqrt{2[\frac{d}{dt}\boldsymbol{\sigma}_{\text{num}}(t)]\boldsymbol{\sigma}_{\text{num}}(t)}$ such that the forward process can be written as:

$$\mathbf{x}_t^{\text{num}}=\mathbf{x}_0^{\text{num}}+\boldsymbol{\sigma}_{\text{num}}(t)\boldsymbol{\epsilon},\quad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{M_{\text{num}}}). \tag{2}$$

The reverse diffusion process can then be formulated accordingly as:

$$\mathrm{d}\mathbf{x}^{\text{num}}=-[\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{\sigma}_{\text{num}}(\mathrm{t})]\boldsymbol{\sigma}_{\text{num}}(\mathrm{t})\nabla_{\mathbf{x}}\log\mathrm{p_t}(\mathbf{x}^{\text{num}})\mathrm{dt}. \tag{3}$$

We train the diffusion model for numerical features via denoising score matching:

$$\mathcal{L}_{\text{num}}=\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x}_0)}\mathbb{E}_{t\sim p(t)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_{M_{\text{num}}})}\left\|\boldsymbol{\mu}_\theta^{\text{num}}(\mathbf{x}_t;\boldsymbol{x}_0,t)-\boldsymbol{\epsilon}\right\|_2^2, \tag{4}$$

Masked Diffusion for Categorical Features, For categorical features, we borrow the most recently developed discrete diffusion schema (Sahoo et al., 2024). We define $\mathrm{Cat}(\cdot;\boldsymbol{\pi})$ as the categorical distribution over K classes with probabilities given by $\boldsymbol{\pi}\in\Delta^K$, where Δ^K is the K-simplex. Let the K-th category correspond to a special [MASK] token and $\mathbf{m}\in\{0,1\}^K$ be the one-hot vector for it, *i.e.*, $\mathbf{m}_K=1$. For forward masking, we set the target prior distribution $\boldsymbol{\pi}=\mathbf{m}$ as the masked absorbing state, and diffuse via interpolating between real data distribution and the prior:

$$\mathrm{q}(\mathbf{x}_\mathrm{t}^{\text{cat}}|\mathbf{x}_0^{\text{cat}})=\mathrm{Cat}(\mathbf{x}_\mathrm{t}^{\text{cat}};\alpha_\mathrm{t}\mathbf{x}_0^{\text{cat}}+(1-\alpha_\mathrm{t})\mathbf{m}), \tag{5}$$

where $\alpha_t\in[0,1]$ is a strictly decreasing function of t. Here we parameterize $\alpha_t=\exp(-\sigma_{\text{cat}}(t))$, where $\sigma_{\text{cat}}(t):[0,1]\to\mathbb{R}^+$. For the reverse process, we introduce a neural network model $\mathbf{x}_\theta(\mathbf{x}_t,t):\mathcal{V}\times[0,1]\to\Delta^K$ to estimate $\mathbf{x}_0$, through which we can approximate the unknown true posterior as:

$$p_\theta(\mathbf{x}_s^{\text{cat}}|\mathbf{x}_t^{\text{cat}})=\begin{cases}\mathrm{Cat}(\mathbf{x}_s^{\text{cat}};\mathbf{x}_t^{\text{cat}}) & \mathbf{x}_t^{\text{cat}}\neq\mathbf{m},\\ \mathrm{Cat}\left(\mathbf{x}_s^{\text{cat}};\frac{(1-\alpha_s)\mathbf{m}+(\alpha_s-\alpha_t)\boldsymbol{\mu}_\theta^{\text{cat}}(\mathbf{x}_t,t)}{1-\alpha_t}\right) & \mathbf{x}_t=\mathbf{m}.\end{cases} \tag{6}$$

where $s<t$ are any two arbitrary times over the continuous time. Previous works (Kingma et al., 2023) have shown that increasing discretization resolution can help approximate tighter evidence lower bound (ELBO). Therefore, we optimize the likelihood bound $\mathcal{L}_{\text{cat}}$ under continuous time limit:

$$\mathcal{L}_{\text{cat}}=\mathbb{E}_q\int_{t=0}^{t=1}\frac{\alpha_t'}{1-\alpha_t}\log\langle\boldsymbol{\mu}_\theta^{\text{cat}}(\mathbf{x}_t,t),\mathbf{x}_0^{\text{cat}}\rangle dt, \tag{7}$$

where α_t' is the first order derivative of α_t.

Consolidating $\mathcal{L}_{\text{num}}$ and $\mathcal{L}_{\text{cat}}$ we derive the total loss $\mathcal{L}$ with weight terms $\lambda_{\text{num}}(t)$ and $\lambda_{\text{cat}}(t)$ as:

$$\mathcal{L}=\lambda_{\text{num}}\mathcal{L}_{\text{num}}+\lambda_{\text{cat}}\mathcal{L}_{\text{cat}} \tag{8}$$

2.3 Adaptive Noise Schedule

To balance the trade-off between the learnable noise schedule's flexibility and robustness, we design two function families: the power mean numerical scheduler and the log-linear categorical scheduler.

Power-mean scheduler for numerical features, For the numerical noise scheduler $\boldsymbol{\sigma}_{\text{num}}(t)$ in eq. (2), we define $\boldsymbol{\sigma}_{\text{num}}(t)=[\sigma_i^{\text{num}}(t)]$. For $\forall i\in\{1,\cdots,M_{\text{num}}\}$:

$$\sigma_i^{\text{num}}(t)=(\sigma_{\text{min}}^{\frac{1}{\rho_i}}+t(\sigma_{\text{max}}^{\frac{1}{\rho_i}}-\sigma_{\text{min}}^{\frac{1}{\rho_i}})^{\rho_i}. \tag{9}$$

and we fix the same initial and final noise levels across all numerical features as $\sigma_i^{\text{num}}(0)=\sigma_{\text{min}}$ and $\sigma_i^{\text{num}}(1)=\sigma_{\text{max}}$.

Log-linear scheduler for categorical features, For the categorical noise scheduler $\boldsymbol{\sigma}_{\text{cat}}(t)$ in section 2.2, we define $\boldsymbol{\sigma}_{\text{cat}}(t)=[\sigma_j^{\text{cat}}(t)]$. For $\forall j\in\{1,\cdots,M_{\text{cat}}\}$:

$$\sigma_j^{\text{cat}}(t)=-\log(1-t^{k_j}) \tag{10}$$

We update $M_{\text{num}}+M_{\text{cat}}$ parameters $\rho_1,\cdots,\rho_{M_{\text{num}}}$ and $k_1,\cdots,k_{M_{\text{cat}}}$ via backpropagation without the need of modifying the loss function.

3 Experiment

3.1 Experimental Setup

Datasets. We conduct experiments on seven real-world tabular datasets consisting of both numerical and categorical attributes: Adult, Default, Shoppers, Magic, Faults, Beijing, News, and Diabetes. Detailed introduction of the datasets is in Appendix A.1.

Baselines. We compare the proposed TabDiff with eight popular synthetic tabular data generation methods under four categories. 1) GAN-based method: CTGAN (Xu et al., 2019). 3) VAE-based methods: TVAE (Xu et al., 2019) and GOGGLE (Liu et al., 2023). 4) Autoregressive Language Model: GReaT (Borisov et al., 2023). 5) Diffusion-based methods: STaSy (Kim et al., 2023), CoDi (Lee et al., 2023), TabDDPM (Kotelnikov et al., 2023) and TabSyn (Zhang et al., 2024).

Evalution Methods. Following previous methods (Zhang et al., 2024), We evaluate the quality of the synthetic data using six distinct metrics: Shape, Trend, α-Precision, β-Recall, Detection, and Machine Learning Efficiency (MLE). Among these metrics, Shape, Trend, α-Precision, β-Recall, and Detection evaluate if the synthetic data can faithfully recover the ground-truth data distribution, while MLE evaluates the synthetic data's utility on downstream tasks. A detailed introduction of these metrics is in Appendix A.2.

3.2 Results

In Table 1, we present the performance comparison of all methods using the five metrics. For each metric, we report the average score with standard deviation across the seven datasets. As demonstrated in the Table, TabDiff yields significant improvement over the competitive baselines on five out of the six metrics, except for the Machine Learning Efficiency task, where TabDiff achieves similar performance compared to TabSyn. Notably, even on Shape and Trend, where the state-of-the-art (SOTA) performance is already extremely high, leaving little room for improvement, TabDiff still achieved over 10% performance improvement. These results thoroughly demonstrate the capacity of TabDiff in modeling multi-modal multivariate joint distributions. The detailed experimental results on each dataset is presented in Appendix B.

Table 1: Comparison of the quality of synthetic data using six metrics. Each column represents the mean performance with std on each metric across seven datasets.

Methods	**Shape**↓	**Trend**↓	**α-Precision**↑	**β-Recall**↑	**Detection**↑	**MLE div**↓
CTGAN	$15.99_{\pm 4.72}$	$16.36_{\pm 15.72}$	$82.40_{\pm 13.19}$	$23.11_{\pm 10.45}$	$64.44_{\pm 10.72}$	$23.73_{\pm 39.80}$
TVAE	$15.97_{\pm 16.26}$	$16.43_{\pm 16.82}$	$75.85_{\pm 28.99}$	$25.32_{\pm 10.00}$	$52.50_{\pm 31.13}$	$20.15_{\pm 27.89}$
GOGGLE	$17.91_{\pm 18.07}$	$28.18_{\pm 25.33}$	$70.82_{\pm 26.24}$	$9.78_{\pm 6.62}$	$33.79_{\pm 34.33}$	$42.06_{\pm 51.94}$
GReaT	$14.20_{\pm 14.71}$	$40.52_{\pm 46.25}$	$80.87_{\pm 8.12}$	$42.86_{\pm 4.42}$	$51.18_{\pm 12.41}$	$13.31_{\pm 23.03}$
STaSy	$7.72_{\pm 7.01}$	$7.77_{\pm 6.43}$	$88.91_{\pm 2.98}$	$42.32_{\pm 8.66}$	$60.83_{\pm 10.98}$	$10.95_{\pm 21.64}$
CoDi	$21.56_{\pm 21.59}$	$23.23_{\pm 23.35}$	$84.29_{\pm 11.75}$	$27.12_{\pm}$	$34.35_{\pm 32.21}$	$30.18_{\pm 32.01}$
TabDDPM	$16.93_{\pm 19.47}$	$11.95_{\pm 13.44}$	$72.48_{\pm 43.18}$	$35.44_{\pm 26.17}$	$70.44_{\pm 44.19}$	$11.95_{\pm 16.88}$
TabSyn	$1.35_{\pm 1.44}$	$2.33_{\pm 2.39}$	$97.86_{\pm 1.58}$	$46.77_{\pm 8.30}$	$91.56_{\pm 15.27}$	$\mathbf{5.46}_{\pm 10.54}$
TabDiff	$\mathbf{1.17}_{\pm 1.26}$	$\mathbf{1.80}_{\pm 1.85}$	$\mathbf{98.16}_{\pm 1.35}$	$\mathbf{49.09}_{\pm 6.62}$	$\mathbf{97.87}_{\pm 2.34}$	$5.71_{\pm 12.27}$
Improv.	13.32%	22.64%	3.1%	4.9%	6.9%	–

4 Conclusion

In this paper, we introduced TabDiff, a unified diffusion framework for generating high-quality synthetic data. TabDiff combines a hybrid diffusion process to handle numerical and categorical features in their native formats. To address the disparate distributions of features and their interrelationships, we further introduced several key innovations, including learnable column-wise noise schedules. We conducted extensive experiments using a diverse set of datasets and metrics, comprehensively comparing TabDiff with existing approaches. The results demonstrate TabDiff's superior capacity in learning the original data distribution and generating faithful and diverse synthetic data to power downstream tasks.

References

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pp. 290–306. PMLR, 2022.

Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: Opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20. Association for Computing Machinery, 2021. ISBN 9781450375849.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

Joao Fonseca and Fernando Bacao. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10(1):115, 2023.

Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 26565–26577, 2022.

Jayoung Kim, Chaejeong Lee, Yehjin Shin, Sewon Park, Minjung Kim, Noseong Park, and Jihoon Cho. 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.

Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2023.

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023. URL https://arxiv.org/abs/2107.00630.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pp. 17564–17579. PMLR, 2023.

Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.

Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. Goggle: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth International Conference on Learning Representations*, 2021.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 7335–7345, 2019.

Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems*, 33: 19075–19087, 2020.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4Ay23yeuz0.

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

A Detailed Experiment Setups

A.1 Datasets

We use seven tabular datasets from UCI Machine Learning Repository[1]: Adult, Default, Shoppers, Magic, Beijing, News, and Diabetes, where each tabular dataset is associated with a machine-learning task. Classification: Adult, Default, Magic, Shoppers, and Diabetes. Regression: Beijing and News. The statistics of the datasets are presented in Table 2.

Table 2: Statistics of datasets. # Num stands for the number of numerical columns, and # Cat stands for the number of categorical columns.

Dataset	# Rows	# Num	# Cat	# Train	# Validation	# Test	Task
Adult	$48,842$	6	9	$28,943$	$3,618$	$16,281$	Classification
Default	$30,000$	14	11	$24,000$	$3,000$	$3,000$	Classification
Shoppers	$12,330$	10	8	$9,864$	$1,233$	$1,233$	Classification
Magic	$19,019$	10	1	$15,215$	$1,902$	$1,902$	Classification
Beijing	$43,824$	7	5	$35,058$	$4,383$	$4,383$	Regression
News	$39,644$	46	2	$31,714$	$3,965$	$3,965$	Regression
Diabetes	$101,766$	9	27	$61,059$	$2,0353$	$20,354$	Classification

A.2 Metrics

A.2.1 Shape and Trend

Shape and Trend are proposed by SDMetrics[2]. They are used to measure the column-wise density estimation performance and pair-wise column correlation estimation performance, respectively. Shape uses Kolmogorov-Sirnov Test (KST) for numerical columns and the Total Variation Distance (TVD) for categorical columns to quantify column-wise density estimation. Trend uses Pearson correlation for numerical columns and contingency similarity for categorical columns to quantify pair-wise correlation.

Shape. *Kolmogorov-Sirnov Test (KST)*: Given two (continuous) distributions $p_r(x)$ and $p_s(x)$ (r denotes real and s denotes synthetic), KST quantifies the distance between the two distributions using the upper bound of the discrepancy between two corresponding Cumulative Distribution Functions (CDFs):

$$\mathrm{KST} = \sup_x |F_r(x) - F_s(x)|, \tag{11}$$

where $F_r(x)$ and $F_s(x)$ are the CDFs of $p_r(x)$ and $p_s(x)$, respectively:

$$F(x) = \int_{-\infty}^{x} p(x)\mathrm{d}x. \tag{12}$$

Total Variation Distance: TVD computes the frequency of each category value and expresses it as a probability. Then, the TVD score is the average difference between the probabilities of the categories:

$$\mathrm{TVD} = \frac{1}{2}\sum_{\omega\in\Omega} |R(\omega) - S(\omega)|, \tag{13}$$

where ω describes all possible categories in a column Ω. $R(\cdot)$ and $S(\cdot)$ denotes the real and synthetic frequencies of these categories.

Trend. *Pearson Correlation Coefficient*: The Pearson correlation coefficient measures whether two continuous distributions are linearly correlated and is computed as:

$$\rho_{x,y} = \frac{\mathrm{Cov}(x,y)}{\sigma_x\sigma_y}, \tag{14}$$

[1] https://archive.ics.uci.edu/datasets

[2] https://docs.sdv.dev/sdmetrics

where x and y are two continuous columns. Cov is the covariance, and σ is the standard deviation.

Then, the performance of correlation estimation is measured by the average differences between the real data's correlations and the synthetic data's corrections:

$$\text{Pearson Score} = \frac{1}{2}\mathbb{E}_{x,y}|\rho^R(x,y) - \rho^S(x,y)|, \tag{15}$$

where $\rho^R(x,y)$ and $\rho^S(x,y))$ denotes the Pearson correlation coefficient between column x and column y of the real data and synthetic data, respectively. As $\rho \in [-1,1]$, the average score is divided by 2 to ensure that it falls in the range of $[0,1]$, then the smaller the score, the better the estimation.

Contingency similarity: For a pair of categorical columns A and B, the contingency similarity score computes the difference between the contingency tables using the Total Variation Distance. The process is summarized by the formula below:

$$\text{Contingency Score} = \frac{1}{2}\sum_{\alpha\in A}\sum_{\beta\in B}|R_{\alpha,\beta} - S_{\alpha,\beta}|, \tag{16}$$

where α and β describe all the possible categories in column A and column B, respectively. $R_{\alpha,\beta}$ and $S_{\alpha,\beta}$ are the joint frequency of α and β in the real data and synthetic data, respectively.

A.2.2 α-Precision and β-Recall

Following Liu et al. (2023) and Alaa et al. (2022), we adopt the α-Precision and β-Recall proposed in Alaa et al. (2022), two sample-level metric quantifying how faithful the synthetic data is. In general, α-Precision evaluates the fidelity of synthetic data – whether each synthetic example comes from the real-data distribution, β-Recall evaluates the coverage of the synthetic data, e.g., whether the synthetic data can cover the entire distribution of the real data (In other words, whether a real data sample is close to the synthetic data.)

A.2.3 Detection

The detection measures the difficulty of detecting the synthetic data from the real data when they are mixed. We use the classifer-two-sample-test (C2ST) implemented by SDMetrics, where a logistic regression model plays the role of a detector.

A.2.4 Machine Learning Efficiency

In MLE, each dataset is first split into the real training and testing set. The generative models are learned on the real training set. After the models are learned, a synthetic set of equivalent size is sampled.

The performance of synthetic data on MLE tasks is evaluated based on the divergence of test scores using the real and synthetic training data. Therefore, we first train the machine learning model on the real training set, split into training and validation sets with a $8:1$ ratio. The classifier/regressor is trained on the training set, and the optimal hyperparameter setting is selected according to the performance on the validation set. After the optimal hyperparameter setting is obtained, the corresponding classifier/regressor is retrained on the training set and evaluated on the real testing set. The performance of synthetic data is obtained in the same way.

B Detailed Experiments Results

In the following sections, we present the detailed results on each metric and dataset.

B.1 Faithfulness

The faithfulness of synthetic data is measured across Shape, Trend, α-precision, β-recall, and CS2T scores. The corresponding detailed results measured on all datasets are presented in Tables 3 to 7.

B.2 Performance on Downstream Tasks

The generated data's utility on downstream tasks, measured by the Machine Learning Efficiency (MLE) is presented in Table 8.

Table 3: Error rates (%) of **Shape** in low-order statistics. **Red Bold Face** highlights the best score for each dataset. A lower error rate indicates a closer resemblance between the synthetic and real data in terms of column-wise density (i.e., superior results). On average TabDiff outperforms the best generative baseline model by $\mathbf{13.3}\%$.

Method	Adult	Default	Shoppers	Magic	Beijing	News	Diabetes	Average
CTGAN	16.84±0.03	16.83±0.04	21.15±0.10	9.81±0.08	21.39±0.05	16.09±0.02	9.82±0.08	15.99
TVAE	14.22±0.08	10.17±0.05	24.51±0.06	8.25±0.06	19.16±0.06	16.62±0.03	18.86±0.13	15.97
GOGGLE[1]	16.97	17.02	22.33	1.90	16.93	25.32	24.92	17.91
GReaT[2]	12.12±0.04	19.94±0.06	14.51±0.12	16.16±0.09	8.25±0.12	OOM	OOM	14.20
STaSy	11.29±0.06	5.77±0.06	9.37±0.09	6.29±0.13	6.71±0.03	6.89±0.03	OOM	7.72
CoDi	21.38±0.06	15.77±0.07	31.84±0.05	11.56±0.26	16.94±0.02	32.27±0.04	21.13±0.25	21.55
TabDDPM[3]	1.75±0.03	1.57±0.08	2.72±0.13	1.01±0.09	1.30±0.03	78.75±0.01	31.44±0.05	16.93
TabSyn	0.81±0.05	**1.01±0.08**	1.44±0.07	1.03±0.14	1.26±0.05	**2.06±0.04**	1.85±0.02	1.35
TabDiff	**0.63±0.05**	1.24±0.07	**1.28±0.09**	**0.78±0.08**	**1.03±0.05**	2.35±0.03	**0.89±0.23**	**1.17**
Improv.	22.2% ↓	0.0% ↓	11.11% ↓	14.29% ↓	18.25% ↓	0% ↓	46.39% ↓	13.3% ↓

[1] The results of baselines above TabSyn on datasets, except for Diabetes, are taken from Zhang et al. (2024).
[2] We encounter difficulty in reproducing TabSyn's results, so we report our own runs.
[3] GOOGLE set fixed random seed during sampling in the official codes, and we follow it for consistency.
[4] GReaT cannot be applied on News for maximum length limit.
[5] STaSy runs out of memory on Diabetes that has hight cardinality categorical columns
[6] TabDDPM cannot produce meaningful content on the News dataset.

Table 4: Error rates (%) of **Trend** in low-order statistics. **Red Bold Face** highlights the best score for each dataset. A lower error rate indicates a closer resemblance between the synthetic data and the testing in terms of pair-wise column correlation (i.e., superior results). On average TabDiff outperforms the best generative baseline model by $\mathbf{22.6}\%$.

Method	Adult	Default	Shoppers	Magic	Beijing	News	Diabetes	Average
CTGAN	20.23±1.20	26.95±0.93	13.08±0.16	7.00±0.19	22.95±0.08	5.37±0.05	18.95±0.34	16.36
TVAE	14.15±0.88	19.50±0.95	18.67±0.38	5.82±0.49	18.01±0.08	6.17±0.09	32.74±0.26	16.44
GOGGLE	45.29	21.94	23.90	9.47	45.94	23.19	27.56	28.18
GReaT	17.59±0.22	70.02±0.12	45.16±0.18	10.23±0.40	59.60±0.55	OOM	OOM	44.24
STaSy	14.51±0.25	5.96±0.26	8.49±0.15	6.61±0.53	8.00±0.10	3.07±0.04	OOM	7.77
CoDi	22.49±0.08	68.41±0.05	17.78±0.11	6.53±0.25	7.07±0.15	11.10±0.01	29.21±0.12	23.21
TabDDPM	3.01±0.25	4.89±0.10	6.61±0.16	1.70±0.22	2.71±0.09	13.16±0.11	51.54±0.05	11.95
TabSyn	1.93±0.07	2.81±0.48	2.13±0.10	0.88±0.18	3.13±0.34	1.52±0.03	3.90±0.04	2.33
TabDiff	**1.49±0.16**	**2.55±0.75**	**1.74±0.08**	**0.76±0.12**	**2.59±0.15**	**1.28±0.04**	**2.20±0.16**	**1.80**
Improve.	22.8% ↓	9.3% ↓	18.3% ↓	13.6% ↓	0.0% ↓	15.8% ↓	37.3% ↓	22.6% ↓

Table 5: Comparison of α-Precision scores. **Red Bold Face** highlights the best score for each dataset. Higher scores reflect better performance. TabDiff consistently achieves the best or second-best score on each dataset and surpasses all other baseline methods on average.

Methods	Adult	Default	Shoppers	Magic	Beijing	News	Diabetes	Average	Ranking
CTGAN	77.74±0.15	62.08±0.08	76.97±0.39	86.90±0.22	96.27±0.14	96.96±0.17	79.89±0.10	82.40	5
TVAE	98.17±0.17	85.57±0.34	58.19±0.26	86.19±0.48	97.20±0.10	86.41±0.17	19.24±0.15	75.85	7
GOGGLE	50.68	68.89	86.95	90.88	88.81	86.41	23.09	70.81	9
GReaT	55.79±0.03	85.90±0.17	78.88±0.13	85.46±0.54	**98.32±0.22**	OOM	OOM	80.87	6
STaSy	82.87±0.26	90.48±0.11	89.65±0.25	86.56±0.19	89.16±0.12	94.76±0.33	OOM	88.91	3
CoDi	77.58±0.45	82.38±0.15	94.95±0.35	85.01±0.36	98.13±0.38	87.15±0.12	64.80±0.53	84.29	4
TabDDPM	96.36±0.20	97.59±0.36	88.55±0.68	98.59±0.17	97.93±0.30	0.00±0.00	28.35±0.11	72.48	8
TabSyn	**99.39±0.18**	**98.65±0.23**	98.36±0.52	**99.42±0.28**	97.51±0.24	95.05±0.30	**96.61±0.24**	97.86	2
TabDiff	99.02±0.20	98.49±0.28	**99.11±0.34**	99.40±0.29	98.06±0.24	**97.36±0.17**	95.69±0.19	**98.21**	1

Table 6: Comparison of β-Recall scores. **Red Bold Face** highlights the best score for each dataset. Higher scores reflects better results. TabDiff consistently achieves the best or second-best β-Recall score on each dataset and surpasses all other baseline methods on average, indicating that the generated data spans a broad range of the real distribution. Though some baseline methods attained higher scores on specific datasets, they fail to demonstrate competitive performance on α-Precision, as models has to trade off fine-grained details in order to capture a broader range of features.

Methods	Adult	Default	Shoppers	Magic	Beijing	News	Diabetes	Average	Ranking
CTGAN	$30.80_{\pm 0.20}$	$18.22_{\pm 0.17}$	$31.80_{\pm 0.350}$	$11.75_{\pm 0.20}$	$34.80_{\pm 0.10}$	$24.97_{\pm 0.29}$	$9.42_{\pm 0.26}$	23.11	8
TVAE	$38.87_{\pm 0.31}$	$23.13_{\pm 0.11}$	$19.78_{\pm 0.10}$	$32.44_{\pm 0.35}$	$28.45_{\pm 0.08}$	$29.66_{\pm 0.21}$	$4.92_{\pm 0.13}$	25.32	7
GOGGLE	8.80	14.38	9.79	9.88	19.87	2.03	3.74	9.78	9
GReaT	$49.12_{\pm 0.18}$	$42.04_{\pm 0.19}$	$44.90_{\pm 0.17}$	$34.91_{\pm 0.28}$	$43.34_{\pm 0.31}$	OOM	OOM	43.34	3
STaSy	$29.21_{\pm 0.34}$	$39.31_{\pm 0.39}$	$37.24_{\pm 0.45}$	$53.97_{\pm 0.57}$	$54.79_{\pm 0.18}$	$39.42_{\pm 0.32}$	OOM	42.32	4
CoDi	$9.20_{\pm 0.15}$	$19.94_{\pm 0.22}$	$20.82_{\pm 0.23}$	$50.56_{\pm 0.31}$	$52.19_{\pm 0.12}$	$34.40_{\pm 0.31}$	$2.70_{\pm 0.06}$	27.12	6
TabDDPM	$47.05_{\pm 0.25}$	$47.83_{\pm 0.35}$	$47.79_{\pm 0.25}$	**$48.46_{\pm 0.42}$**	$56.92_{\pm 0.13}$	$0.00_{\pm 0.00}$	$0.03_{\pm 0.01}$	35.44	5
TabSyn	$47.92_{\pm 0.23}$	$46.45_{\pm 0.35}$	$49.10_{\pm 0.60}$	$48.03_{\pm 0.50}$	$59.15_{\pm 0.22}$	**$43.01_{\pm 0.28}$**	$33.72_{\pm 0.16}$	46.77	2
TabDiff	**$51.64_{\pm 0.20}$**	**$51.09_{\pm 0.25}$**	**$49.75_{\pm 0.64}$**	$47.67_{\pm 0.31}$	**$59.63_{\pm 0.23}$**	$42.10_{\pm 0.32}$	**$41.74_{\pm 0.17}$**	**49.35**	1

Table 7: Detection score (C2ST) using logistic regression classifier. Higher scores reflect superior performance. TabDiff consistently achieves the best or second-best performance across all datasets. Notably, TabDiff demonstrates exceptional performance on Diabetes, which contains many high-cardinality categorical features, highlighting its advanced capacity in generating faithful categorical data.

Method	Adult	Default	Shoppers	Magic	Beijing	News	Diabetes	Average
CTGAN	0.5949	0.4875	0.7488	0.6728	0.7531	0.6947	0.5593	0.6444
TVAE	0.6315	0.6547	0.2962	0.7706	0.8659	0.4076	0.0487	0.5250
GOGGLE	0.1114	0.5163	0.1418	0.9526	0.4779	0.0745	0.0912	0.3380
GReaT	0.5376	0.4710	0.4285	0.4326	0.6893	OOM	OOM	0.5118
STaSy	0.4054	0.6814	0.5482	0.6939	0.7922	0.5287	OOM	0.6083
CoDi	0.2077	0.4595	0.2784	0.7206	0.7177	0.0201	0.0008	0.3435
TabDDPM	0.9755	0.9712	0.8349	**0.9998**	0.9513	0.0002	0.1980	0.7044
TabSyn	0.9910	**0.9826**	0.9662	0.9960	0.9528	0.9255	0.5953	0.9156
TabDiff	**0.9950**	0.9774	**0.9843**	0.9989	**0.9781**	**0.9308**	**0.9865**	**0.9787**
Improv.	0.40% ↓	0.0% ↓	1.87% ↓	0.0% ↓	2.66% ↓	0.57% ↓	65.71% ↓	6.89% ↓

Table 8: Evaluation of **Machine Learning Efficiency**: AUC and RMSE are used for classification and regression tasks, respectively. ↑ (↓) denotes whether a higher or lower score shows better performance. TabDiff consistently achieves the best or second-best performance across all datasets.

Methods	Adult	Default	Shoppers	Magic	Beijing	News[1]	Diabetes	Average Gap
	AUC ↑	AUC ↑	AUC ↑	AUC ↑	RMSE ↓	RMSE ↓	AUC ↑	%
Real	$.927_{\pm .000}$	$.770_{\pm .005}$	$.926_{\pm .001}$	$.946_{\pm .001}$	$.423_{\pm .003}$	$.842_{\pm .002}$	$.704_{\pm .002}$	0%
CTGAN	$.886_{\pm .002}$	$.696_{\pm .005}$	$.875_{\pm .009}$	$.855_{\pm .006}$	$.902_{\pm .019}$	$.880_{\pm .016}$	$.569_{\pm .004}$	23.7%
TVAE	$.878_{\pm .004}$	$.724_{\pm .005}$	$.871_{\pm .006}$	$.887_{\pm .003}$	$.770_{\pm .011}$	$1.01_{\pm .016}$	$.594_{\pm .009}$	20.2%
GOGGLE	$.778_{\pm .012}$	$.584_{\pm .005}$	$.658_{\pm .052}$	$.654_{\pm .024}$	$1.09_{\pm .025}$	$.877_{\pm .002}$	$.475_{\pm .008}$	42.1%
GReaT	$.913_{\pm .003}$	$.755_{\pm .006}$	$.902_{\pm .005}$	$.888_{\pm .008}$	$.653_{\pm .013}$	OOM	OOM	13.3%
STaSy	$.906_{\pm .001}$	$.752_{\pm .006}$	$.914_{\pm .005}$	$.934_{\pm .003}$	$.656_{\pm .014}$	$.871_{\pm .002}$	OOM	10.9%
CoDi	$.871_{\pm .006}$	$.525_{\pm .006}$	$.865_{\pm .006}$	$.932_{\pm .003}$	$.818_{\pm .021}$	$1.21_{\pm .005}$	$.505_{\pm .004}$	30.2%
TabDDPM[2]	$.907_{\pm .001}$	$.758_{\pm .004}$	$.918_{\pm .005}$	$.935_{\pm .003}$	$.592_{\pm .011}$	$4.86_{\pm 3.04}$	$.521_{\pm .008}$	11.95%[1]
TabSyn	$.909_{\pm .001}$	**$.763_{\pm .002}$**	$.914_{\pm .004}$	**$.937_{\pm .002}$**	**$.547_{\pm .009}$**	**$.850_{\pm .024}$**	$.684_{\pm .002}$	**5.46%**
TabDiff	**$.912_{\pm .002}$**	**$.763_{\pm .005}$**	**$.921_{\pm .004}$**	$.936_{\pm .003}$	$.555_{\pm .013}$	$.866_{\pm .021}$	**$.689_{\pm .016}$**	5.76%

[1] As in CoDi (Lee et al., 2023), the continuous targets are standardized to avoid large values.
[2] TabDDPM fails to produce meaningful News data, so we exclude it from the average gap calculation.