Distributional Learning of Variational AutoEncoder: Application to Synthetic Data Generation

Seunghwan An and **Jong-June Jeon**[*]
Department of Statistical Data Science, University of Seoul, S. Korea
{dkstmdghks79, jj.jeon}@uos.ac.kr

Abstract

The Gaussianity assumption has been consistently criticized as a main limitation of the Variational Autoencoder (VAE) despite its efficiency in computational modeling. In this paper, we propose a new approach that expands the model capacity (i.e., expressive power of distributional family) without sacrificing the computational advantages of the VAE framework. Our VAE model's decoder is composed of an infinite mixture of asymmetric Laplace distribution, which possesses general distribution fitting capabilities for continuous variables. Our model is represented by a special form of a nonparametric M-estimator for estimating general quantile functions, and we theoretically establish the relevance between the proposed model and quantile estimation. We apply the proposed model to synthetic data generation, and particularly, our model demonstrates superiority in easily adjusting the level of data privacy.

1 Introduction

Variational Autoencoder (VAE) [31, 51] and Generative Adversarial Networks (GAN) [22] are generative models that are used to estimate the underlying distribution of a given dataset. To avoid the curse of dimensionality, VAE and GAN commonly introduce a low-dimensional latent space on which a conditional generative model is defined. By minimizing an information divergence between the original data and its generated data, the generative models are learned to produce synthetic data similar to the original one. Accordingly, VAE and GAN have been applied in various applications, such as generating realistic images, texts, and synthetic tabular data for privacy preservation purposes [30, 60, 63, 65, 34].

However, the difference in the strength of the assumption about the generative distribution brings significant contrasts in the VAE and GAN generation performances [30, 7, 17]. In the GAN framework, the adversarial loss enables direct minimization of the Jensen-Shannon divergence between the ground-truth density function and the generative distribution under no distributional assumption [13, 59]. Roughly speaking, the GAN employs a nonparametric model as its conditional generative model defined on the latent space.

On the contrary, in the VAE framework, the Gaussianity assumption has been favored [31, 32, 15, 10, 39]. It is because Gaussianity gives us three advantages: 1) the reconstruction loss can be interpreted as the mean squared error that is one of the most popular losses in optimization theory, 2) generating a new sample is computationally straightforward, and 3) KL-divergence is computed in a simple closed form. However, these benefits have led us to pay the price for the distributional capacity of the generative model, in that the generative model of the VAE is constrained in the form of marginalization of the product of the two Gaussian distributions. Here, the distributional capacity means the expressive power of the distributional family. This restricted distributional capacity has

---

[*]Corresponding author.

been the critical limitation [8, 33] and leads to a heavy parameterization of the decoder mean vector to approximate complex underlying distributions.

To increase the distributional capacity in synthetic data generation, [63, 65] introduce the multi-modality in the distributional assumption of the decoder, which is known as the *mode-specific normalization technique*. Although the mixture Gaussian decoder modeling of [63, 65] allows handling more complex distributions of the observed dataset while preserving all of the advantages of Gaussianity, we numerically find that the mixture Gaussian is not enough to capture the complex underlying distribution.

Our main contribution is that, beyond Gaussianity, we propose a novel VAE learning method that directly estimates the conditional cumulative distribution function (CDF) while maintaining the objective of maximizing the Evidence Lower Bound (ELBO) of the observed dataset. It implies that we have a nonparametric distribution assumption on the generative model. We call this approach *distributional learning of the VAE*, which is enabled by estimating an infinite number of conditional quantiles [4, 20]. By adopting the *continuous ranked probability score* (CRPS) loss, the objective function of our proposed distribution learning method is computationally tractable [21, 43, 20].

In our proposed distributional learning framework, 1) the reconstruction loss is equivalent to the CRPS loss, which is a *proper scoring rule* [21, 43], 2) generating a new sample is still computationally straightforward due to the inverse transform sampling, and 3) KL-divergence is still computed in a simple closed form. To show the effectiveness of our proposed model in capturing the underlying distribution of the dataset, we evaluate our model for synthetic data generation with real tabular datasets.

2 Related Work

Modeling of the decoder and reconstruction loss. To increase the distributional capacity, many papers have focused on decoder modeling while not losing the mathematical link to maximize the ELBO. [57, 1] assume their decoder distributions as Student-t and asymmetric Laplace distributions, respectively, to mitigate the *zero-variance problem* that the model training becomes unstable if the estimated variance of the decoder shrinks to zero in Gaussian VAE [41, 55, 15]. [2] proposes a general distribution of the decoder, which allows improved robustness by optimizing the shape of the loss function during training. Recently, [5] proposes a reconstruction loss that directly minimizes the *blur error* of the VAE by modeling the covariance matrix of multivariate Gaussian decoder.

On the other hand, there exists a research direction that focuses on replacing the reconstruction loss without concern for losing the mathematical derivation of the lower bound. [38, 52, 46] replace the reconstruction loss with an adversarial loss of the GAN framework. [27] introduces a feature-based loss that is calculated with a pre-trained convolutional neural network (CNN). Another approach by [14] adopts Watson's perceptual model, and [28] directly optimizes the generative model in the frequency domain by a focal frequency reconstruction loss. Most of the above-mentioned methods aim to capture the properties of human perception by replacing the element-wise loss (L_1 or L_2-norm), which hinders the reconstruction of images [38].

Synthetic data generation. The GAN framework is widely adopted in the synthetic data generation task since it enables synthetic data generation in a nonparametric approach [12, 47, 63, 65]. [63, 65] assume that continuous columns of tabular datasets can be approximated by the Gaussian mixture distribution and model their decoder using Gaussian mixture distribution. Additionally, [63, 65] preprocess the continuous variables using the variational Gaussian mixture model [3], which is known as the mode-specific normalization technique. However, the preprocessing step requires additional computational resources and hyperparameter tuning of the number of modes. Other approaches by [47, 65] regularize the discrepancy between the first and second-order statistics of the observed and synthetic dataset. [12] proposes the GAN-based synthesizer, which focuses on generating high-dimensional discrete variables with the assistance of the pre-trained AutoEncoder.

3 Proposal

Let $\mathbf{x} \in \mathbb{R}^{p+q}$ be an observation consisting of continuous and discrete variables and $I = I_C \cup I_D = \{1, \cdots, (p+q)\}$ be an index set of the variables, where I_C and I_D correspond to index sets of

p continuous and q discrete variables. T_j denotes the number of levels for the discrete variables $\mathbf{x}_j, j \in I_D$. We denote the ground-truth underlying distribution (probability density function, PDF) as $p(\mathbf{x})$ and the ground-truth CDF as $F(\mathbf{x})$.

Let $\mathbf{z}$ be a latent variable, where $\mathbf{z} \in \mathbb{R}^d$ and $d < p + q$. The prior and posterior distribution of $\mathbf{z}$ are assumed to be $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ and $q(\mathbf{z}|\mathbf{x}; \phi) = \mathcal{N}\big(\mathbf{z}|\mu(\mathbf{x}; \phi), diag(\sigma^2(\mathbf{x}; \phi))\big)$, respectively. Here, $\mathbf{I}$ is $d \times d$ identity matrix, $\mu : \mathbb{R}^{p+q} \mapsto \mathbb{R}^d$, $\sigma^2 : \mathbb{R}^{p+q} \mapsto \mathbb{R}_+^d$ are neural networks parameterized with ϕ, and $diag(a), a \in \mathbb{R}^d$ denotes a diagonal matrix with diagonal elements a. Moreover, we consider $\alpha \in [0, 1]$ as a random variable having density $p(\alpha)$.

3.1 Distributional Learning

Our proposed model assumes that $p(\mathbf{x})$ is parametrized by an infinite mixture of asymmetric Laplace distribution (ALD) [4]. The ALD is characterized by two parameters: α, representing the asymmetry, and $\beta > 0$, representing the scale. By considering these parameters, along with the model parameter θ, we can define the probability model of $\mathbf{x}$ as follows:

$$p(\mathbf{x}; \theta, \beta) = \int \int_0^1 p(\mathbf{x}|\mathbf{z}, \alpha; \theta, \beta) p(\mathbf{z}, \alpha) d\alpha d\mathbf{z}.$$

Assumption 1. *(1) $\{\mathbf{x}_j\}_{j \in I}$ are conditionally independent given $\mathbf{z}$. (2) The discrete random variables $\{\mathbf{x}_j\}_{j \in I_D}$ are independent of α. (3) α and $\mathbf{z}$ are independent.*

By Assumption 1-(1), we model the dependency between $\mathbf{x}_j$s solely through the latent variable $\mathbf{z}$ [49]. Assumption 1-(2) implies that α is related only to the continuous variables. Then, the decoder of our VAE model denoted as $p(\mathbf{x}|\mathbf{z}, \alpha; \theta, \beta)$ is specified by equation (1):

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{z}, \alpha; \theta, \beta) &= \prod_{j \in I_C} p(\mathbf{x}_j|\mathbf{z}, \alpha; \theta_j, \beta) \cdot \prod_{j \in I_D} p(\mathbf{x}_j|\mathbf{z}; \theta_j) \qquad (1) \\
&= \prod_{j \in I_C} \frac{\alpha(1-\alpha)}{\beta} \exp\left(-\rho_\alpha\left(\frac{\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j)}{\beta}\right)\right) \cdot \prod_{j \in I_D} \prod_{l=1}^{T_j} \pi_l(\mathbf{z}; \theta_j)^{\mathbb{I}(\mathbf{x}_j = l)},
\end{aligned}
$$

where $\theta = (\theta_1, \cdots, \theta_{p+q})$, β is a non-trainable constant, $\rho_v(u) = u(v - \mathbb{I}(u < 0))$ (check function), and $\mathbb{I}(\cdot)$ denotes the indicator function. $D_j(\cdot, \cdot; \theta_j) : [0, 1] \times \mathbb{R}^d \mapsto \mathbb{R}$ is the location parameter of ALD, which is parameterized with θ_j [1]. For discrete variables, $\pi(\cdot; \theta_j) : \mathbb{R}^d \mapsto \Delta^{T_j - 1}$ is a neural network parameterized with θ_j, where $\Delta^{T_j - 1}$ is the standard $(T_j - 1)$-simplex for all $\mathbf{z} \in \mathbb{R}^d$, and the subscript l referes to the lth element of the output π.

Assumption 1-(3) leads our objective function,

$$
\begin{aligned}
\min_{\theta, \phi} \quad & \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{q(\mathbf{z}|\mathbf{x}; \phi)} \left[\sum_{j \in I_C} \int_0^1 \rho_\alpha\Big(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j)\Big) d\alpha - \sum_{j \in I_D} \sum_{l=1}^{T_j} \mathbb{I}(\mathbf{x}_j = l) \cdot \log \pi_l(\mathbf{z}; \theta_j) \right] \\
+ \quad & \beta \cdot \mathbb{E}_{p(\mathbf{x})}[\mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z}))], \qquad (2)
\end{aligned}
$$

where constant terms are omitted. In order to achieve balanced learning of the two reconstruction losses in (2), we have removed the weight β associated with the second reconstruction loss. We refer to our model as 'DistVAE.' The first term in (2) corresponds to the CRPS loss, which measures the accuracy of the proposed CDF approximation with respect to the ground-truth CDF of the underlying distribution [21, 43, 20].

Interestingly, (2) is the limit of the negative ELBO derived from a finite mixture of ALD. We introduce α, a discrete uniform random variable taking values on $\alpha_k = k/K$ for $k = 1, \cdots, K$. Then, the negative ELBO of $p(\mathbf{x}; \theta, \beta)$ scaled with β is written by

$$
\begin{aligned}
& \mathbb{E}_{q(\mathbf{z}|\mathbf{x}; \phi)} \left[\sum_{j \in I_C} \frac{1}{K} \sum_{k=1}^{K} \rho_{\alpha_k}\Big(\mathbf{x}_j - D_j(\alpha_k, \mathbf{z}; \theta_j)\Big) \right] - \beta \frac{p}{K} \sum_{k=1}^{K} \log \alpha_k (1 - \alpha_k) + \beta p \log \beta \\
- \quad & \beta \cdot \mathbb{E}_{q(\mathbf{z}|\mathbf{x}; \phi)} \left[\sum_{j \in I_D} \sum_{l=1}^{T_j} \mathbb{I}(\mathbf{x}_j = l) \cdot \log \pi_l(\mathbf{z}; \theta_j) \right] + \beta \cdot \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z})) \qquad (3)
\end{aligned}
$$

3

(see Appendix A.1 for detailed derivation). The reconstruction loss, which corresponds to the first term of (3), is a composite quantile loss for estimating the target quantiles α_ks [64, 35, 45, 62, 9]. This entails adopting a Bayesian perspective for α as a prior (the Bayesian modeling for estimating multiple quantiles). Furthermore, throughout the derivation of the reconstruction loss, the role of α is pivotal in ensuring the representation of the reconstruction loss. To prevent the observation $\mathbf{x}$ from influencing the distribution of α, α is only assigned with a prior distribution, and the resulting reconstruction loss becomes the CRPS loss, a proper scoring rule.

However, for distributional learning of VAE, it is necessary to estimate conditional quantiles for an infinite number of quantile levels, denoted by $K \to \infty$ [50]. The subsequent Theorem 1 establishes the convergence of the negative ELBO (3) to our objective function (2) as $K \to \infty$ [4].

Theorem 1. *For all $j \in I_C$, suppose that $\int_0^1 \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\rho_\alpha\big(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j)\big)d\alpha < \infty$, and $\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}[\rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))]$ is continuous over $\alpha \in (0, 1)$. Then,*

$$\lim_{K \to \infty} \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\frac{1}{K}\sum_{k=1}^{K}\rho_{\alpha_k}\Big(\mathbf{x}_j - D_j(\alpha_k, \mathbf{z}; \theta_j)\Big)\right] = \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\int_0^1 \rho_\alpha\Big(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j)\Big)d\alpha\right],$$

and $\lim_{K \to \infty} \frac{1}{K}\sum_{k=1}^{K}\log\alpha_k(1 - \alpha_k) = \int_0^1 \log\alpha(1 - \alpha)d\alpha = -2$.

3.2 Theoretical Results

In this section, we aim to provide theoretical insights into the ability of DistVAE, utilizing the objective function (2), to recover the ground-truth distribution $p(\mathbf{x})$. To simplify the analysis without loss of generality, we consider the scenario where $\mathbf{x}$ comprises only p continuous random variables. Hence, we have $I = I_C = \{1, \cdots, p\}$, and $p(\mathbf{x})$ is defined over $\mathbf{x} \in \mathbb{R}^p$ with $p(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^p$. First, define a function $q(\mathbf{x}|\mathbf{z}; \phi)$ by

$$q(\mathbf{x}|\mathbf{z}; \phi) := \frac{p(\mathbf{x})q(\mathbf{z}|\mathbf{x}; \phi)}{q(\mathbf{z}; \phi)},$$

where $q(\mathbf{z}; \phi) := \int q(\mathbf{z}|\mathbf{x}; \phi)p(\mathbf{x})d\mathbf{x}$ is the aggregated posterior [58]. Clearly, $q(\mathbf{x}|\mathbf{z}; \phi)$ is a PDF of $\mathbf{x}$ for a given $\mathbf{z}$. $q(\mathbf{x}|\mathbf{z}; \phi)$ is a conditional PDF of $\mathbf{x}$ parametrized by ϕ and it is an approximated PDF of $\int_0^1 p(\mathbf{x}|\mathbf{z}, \alpha; \theta, \beta)d\alpha$. Since we assume that $\mathbf{x}_j$s are conditionally independent in Assumption 1, $q(\mathbf{x}|\mathbf{z}; \phi) = \prod_{j=1}^p q_j(\mathbf{x}_j|\mathbf{z}; \phi)$ and the conditional CDF is written as

$$F(\mathbf{x}|\mathbf{z}; \phi) = \prod_{j=1}^{p} F_j(\mathbf{x}_j|\mathbf{z}; \phi), \text{ where } F_j(\mathbf{x}_j|\mathbf{z}; \phi) := \int_{-\infty}^{\mathbf{x}_j} q_j(x|\mathbf{z}; \phi)dx. \tag{4}$$

For notational simplicity, we let

$$\theta^*(\phi) \in \arg\min_\theta \mathbb{E}_{p(\mathbf{x})q(\mathbf{z}|\mathbf{x};\phi)}\sum_{j=1}^{p}\int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))d\alpha,$$

where $\theta^*(\phi) = (\theta_1^*(\phi), \cdots, \theta_p^*(\phi))$.

Assumption 2. *(1) Given an arbitrary ϕ, $F_j(\cdot|\mathbf{z}; \phi) : \mathbb{R} \mapsto [0, 1]$ is absolutely continuous and strictly monotone increasing for all $j = 1, \cdots, p$, and $\mathbf{z} \in \mathbb{R}^d$. (2) Given an arbitrary θ, $D_j(\cdot, \mathbf{z}; \theta_j)$ is invertible and differentiable for all $j = 1, \cdots, p$ and $\mathbf{z} \in \mathbb{R}^d$. (3) The aggregated posterior $q(\cdot; \phi)$ is absolutely continuous to the prior distribution of $\mathbf{z}$.*

Theorem 2. *Under Assumption 2, for an arbitrary ϕ,*

$$\mathcal{KL}\left(p(\mathbf{x})\left\|\int\prod_{j=1}^{p}\frac{d}{d\mathbf{x}_j}D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi))q(\mathbf{z}; \phi)d\mathbf{z}\right)\right. = 0.$$

Theorem 2 shows that DistVAE is capable of recovering the ground-truth distribution $p(\mathbf{x})$, indicating its ability to facilitate distributional learning rather than data reconstruction. Nevertheless, relying on the aggregated posterior distribution may lead to overfitting [25, 42], and sampling from the

aggregated posterior can introduce computational challenges due to the absence of a straightforward closed-form representation for $q(\mathbf{z}; \phi)$. To address these concerns, we propose an alternative approach that leverages the prior distribution $p(\mathbf{z})$ instead of $q(\mathbf{z}; \phi)$, thereby enabling a computationally efficient synthetic generation process. This is substantiated by Theorem 3.

We define the estimated PDF $\hat{p}(\mathbf{x}; \theta^*(\phi))$ and CDF $\hat{F}(\mathbf{x}; \theta^*(\phi))$ as

$$\hat{p}(\mathbf{x}; \theta^*(\phi)) \;:=\; \int \prod_{j=1}^{p} \frac{d}{d\mathbf{x}_j} D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi)) p(\mathbf{z}) d\mathbf{z} \tag{5}$$

$$\hat{F}(\mathbf{x}; \theta^*(\phi)) \;:=\; \int \prod_{j=1}^{p} D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi)) p(\mathbf{z}) d\mathbf{z}. \tag{6}$$

Theorem 3. *Suppose that ϕ is given such that $\mathcal{KL}(q(\mathbf{z}; \phi) \| p(\mathbf{z})) < \epsilon$ for any $\epsilon > 0$. Then, under Assumption 2,*

$$\mathcal{KL}\left(p(\mathbf{x}) \,\middle\|\, \int \prod_{j=1}^{p} \frac{d}{d\mathbf{x}_j} D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi)) p(\mathbf{z}) d\mathbf{z} \right) < \epsilon.$$

Theorem 3 shows that even if we use the prior distribution $p(\mathbf{z})$ instead of the aggregated posterior $q(\mathbf{z}; \phi)$, it is feasible to minimize the KL-divergence between the ground-truth PDF $p(\mathbf{x})$ and our estimated PDF $\hat{p}(\mathbf{x}; \theta^*(\phi))$ of (5). This is achievable because the KL-divergence term in (2) is the upper bound of $\mathcal{KL}(q(\mathbf{z}; \phi) \| p(\mathbf{z}))$ and is minimized during the training process. Since each conditional distribution of the estimated PDF depends on the same latent variable, it can be seen that the correlation structure between covariates is modeled implicitly.

[24, 41, 37] have highlighted the role of the KL-divergence weight parameter β in controlling the precision of reconstruction. In our case, where the reconstruction loss is based on the CRPS loss, an increase in β leads to a less accurate estimation of the ground-truth CDF. It implies that a large β corresponds to lower-quality synthetic data, but it also enhances privacy level. Thus, β introduces a trade-off between the quality of synthetic data and the risk of privacy leakage. The privacy level can be adjusted by manipulating β [47], as demonstrated in the experimental results presented in Section 4.

3.2.1 Synthetic Data Generation

By estimating the conditional quantile functions, we can transform the synthetic data generation process into inverse transform sampling. This conversion offers a notable advantage as it provides a straightforward and efficient approach to generating synthetic data. We denote a synthetic sample of $\mathbf{x}_j$ as $\hat{x}_j$ for $j \in I$, and the synthetic data generation process can be summarized as follows:

1. Sampling from the prior distribution: $z \sim p(\mathbf{z})$.
2. Inverse transform sampling: For $j \in I_C$, $\hat{x}_j = D_j(u_j | z; \theta_j)$, where $u_j \sim U(0, 1)$.
3. Gumbel-Max trick [23]: For $j \in I_D$, $\hat{x}_j = \arg\max_{l=1, \cdots, T_j} \{\log \pi_l(z; \theta_j) + G_l\}$, where $G_l \sim Gumbel(0, 1)$, and $l = 1, \cdots, T_j$.

Note that both continuous and discrete variables share the same latent variable z. This shared latent variable allows for capturing the dependencies between variables. We numerically observe that the sampling, implemented using the Gumbel-Max trick, maintains the imbalanced ratio of the labels in the discrete variable.

3.2.2 Parameterization of ALD

As introduced in [20], for $j \in I_C$, we parameterize the function D_j, the location parameter of ALD, by a linear isotonic spline as follows:

$$D_j(\alpha, \mathbf{z}; \theta_j) = \gamma^{(j)}(\mathbf{z}) + \sum_{m=0}^{M} b_m^{(j)}(\mathbf{z})(\alpha - d_m)_+ \quad \text{s.t.} \quad \sum_{m=0}^{k} b_m^{(j)}(\mathbf{z}) \geq 0, k = 1, \cdots, M, \tag{7}$$

where $\gamma^{(j)}(\mathbf{z}) \in \mathbb{R}$, $b^{(j)}(\mathbf{z}) = (b_0^{(j)}(\mathbf{z}), \cdots, b_M^{(j)}(\mathbf{z})) \in \mathbb{R}^{M+1}$, $d = (d_0, \cdots, d_M) \in [0, 1]^{M+1}$, $0 = d_0 < \cdots < d_M = 1$, and $(u)_+ := \max(0, u)$. θ_j is a neural network parameterized mapping such that $\theta_j : \mathbb{R}^d \mapsto \mathbb{R} \times \mathbb{R}^{M+1}$, which takes $\mathbf{z}$ as input and outputs $\gamma^{(j)}(\mathbf{z})$ and $b^{(j)}(\mathbf{z})$. The constraint is introduced to ensure monotonicity. As demonstrated in [20], the reconstruction loss can be computed in a closed form by utilizing the parameterization of (7) (refer to Appendix A.5 for a detailed description of the loss function). This implies that our objective function (2) is computationally tractable. Note that the linear isotonic spline is not differentiable for finite points where the measure has no point mass.

4 Experiments

4.1 Overview

Dataset. For evaluation, we consider following real tabular datasets: `covertype`, `credit`, `loan`, `adult`, `cabs`, and `kings` (see Appendix A.8 for detailed data descriptions). We treat the ordinal variables as continuous variables and discretize the estimated CDF (see Appendix A.6 for the discretization algorithm). Synthetic samples of ordinal variables are rounded to the first decimal place.

Compared models. We compare DistVAE[†] with the state-of-the-art synthesizers; CTGAN [63], TVAE [63], and CTAB-GAN [65]. All models have the same size of the latent dimension ($d = 2$). The chosen latent space indeed limits the capacity of decoders for all models. However, we maintain a small and consistent number of parameters across all models during the experiment to isolate the performance differences in synthetic data generation to the methodologies of each synthesizer, specifically emphasizing the contribution of the decoder model's flexibility in estimating underlying distributions (see Table 10 in Appendix A.9 for a comprehensive comparison of the model parameters).

4.2 Evaluation Metrics

To assess the quality of the synthetic data, we employ three types of assessment criteria: 1) machine learning utility, 2) statistical similarity, and 3) privacy preservability. Each criterion is evaluated using multiple metrics, and the performance of synthesizers is reported by averaged metrics over the real tabular datasets. The synthetic dataset is generated to have an equal number of samples as the real training dataset.

Machine learning utility. The machine learning utility (MLu) is measured by the predictive performance of the trained model over the synthetic data. We consider three popular machine learning algorithms: linear (logistic) regression, Random Forest [6], and Gradient Boosting [19]. We measure the performance by utilizing the mean absolute relative error (MARE) for regression tasks [47] and the F_1 score for classification tasks [63, 65, 61, 29, 47, 12, 18].

Statistical similarity. The marginal distributional similarity between the real training and synthetic datasets is evaluated using two metrics: the Kolmogorov statistic and the 1-Wasserstein distance [18]. These metrics measure the distance between the empirical marginal CDFs [40]. The joint distributional similarity is assessed by comparing the correlation matrices [65]. To compute the correlation matrix and measure the L_2 distance between the correlation matrices of the real training and synthetic datasets, we employ the `dython` library [‡]. These enable a comprehensive evaluation of both marginal and joint distributional similarities between the real training and synthetic datasets.

Privacy preservability. The privacy-preserving capacity is measured using three metrics: the *distance to closest record* (DCR) [47, 65], *membership inference attack* [54, 12, 47], and *attribute disclosure* [12, 44]. As in [65], the DCR is defined as the 5th percentile of the L_2 distances between all real training samples and synthetic samples. Since the DCR is a L_2 distance-based metric, it is computed using only continuous variables. A higher DCR value indicates a more effective preservation of privacy, indicating a lack of overlap between the real training data and the synthetic samples. Conversely, an excessively large DCR score suggests a lower quality of the generated synthetic dataset. Therefore, the DCR metric provides insights into both the privacy-preserving capability and the quality of the synthetic dataset.

[†]We release the code at `https://github.com/an-seunghwan/DistVAE`.

[‡]`http://shakedzy.xyz/dython/modules/nominal/#associations`

The membership inference attack evaluation follows the steps detailed in Appendix A.7. The procedure is customized to be applied to a VAE-based synthesizer, such as DistVAE and TVAE. By transforming the problem into a binary classification task, we aim to identify the intricate relationship between the real training data and the synthetic samples. Higher binary classification scores indicate a higher vulnerability of the target synthesizer to membership inference attacks.

Attribute disclosure refers to the situation where attackers can uncover additional covariates of a record by leveraging a subset of covariates they already possess, along with similar records from the synthetic dataset. To quantify the extent to which attackers can accurately identify these additional covariates, we employ classification metrics. Higher attribute disclosure metrics indicate an increased risk of privacy leakage, implying that attackers can precisely infer unknown variables. In terms of privacy concerns, attribute disclosure can be considered a more significant issue than membership inference attacks, as attackers are assumed to have access to only a subset of covariates for a given record [12].

4.3 Results

Machine learning utility. We expect a high-quality synthesizer to generate synthetic data with comparable predictive performance to the real training dataset, denoted as the 'Baseline' in Table 1. The results in Table 1 demonstrate that DistVAE achieves a competitive MARE score and outperforms other methods in terms of the F_1 score. Furthermore, the performance of MLu improves as the value of β decreases, indicating that the quality of the generated synthetic data is controlled by β. For a comprehensive overview of the MLu scores for all tabular datasets, please refer to Appendix A.10.

Table 1: Averaged MLu metrics (MARE, F_1). Mean and standard deviation values are obtained from 10 repeated experiments. ↑ denotes higher is better and ↓ denotes lower is better.

Model	MARE ↓	F_1 ↑
Baseline	$0.150_{\pm 0.200}$	$0.814_{\pm 0.101}$
CTGAN	$0.321_{\pm 0.271}$	$0.672_{\pm 0.234}$
TVAE	$\mathbf{0.225}_{\pm 0.215}$	$0.594_{\pm 0.295}$
CTAB-GAN	$0.403_{\pm 0.392}$	$0.702_{\pm 0.162}$
DistVAE($\beta = 0.5$)	$0.349_{\pm 0.328}$	$\mathbf{0.769}_{\pm 0.128}$
DistVAE($\beta = 1$)	$0.344_{\pm 0.316}$	$\mathbf{0.762}_{\pm 0.134}$
DistVAE($\beta = 5$)	$0.392_{\pm 0.348}$	$0.679_{\pm 0.190}$

Table 2: Averaged correlation structural similarity. 'CorrDist' represents L_2 distance between the correlation matrices of synthetic and real training datasets. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

Model	CorrDist
CTGAN	$2.180_{\pm 0.467}$
TVAE	$2.739_{\pm 0.796}$
CTAB-GAN	$2.575_{\pm 0.513}$
DistVAE($\beta = 0.5$)	$\mathbf{1.473}_{\pm 0.398}$
DistVAE($\beta = 1$)	$1.730_{\pm 0.548}$
DistVAE($\beta = 5$)	$3.113_{\pm 1.119}$

Statistical similarity. The evaluation results for joint and marginal distributional similarities are presented in Table 2 and 3. In Table 2, DistVAE achieves the lowest CorrDist score, indicating its ability to accurately preserve the correlation structure of the real training dataset in the generated synthetic dataset. Furthermore, DistVAE surpasses other methods in Table 3 when it comes to marginal distributional similarity, suggesting that it successfully captures the underlying distribution of the observed dataset. Notably, reducing the value of β leads to an enhancement in the quality of the synthetic dataset, as evidenced by improvements in the correlation structure and similarity of the marginal distributions. Figure 1 provides visualizations of the estimated CDFs (6) for each continuous (or ordinal) variable in the cabs dataset. For detailed statistical similarity scores and additional visualizations of estimated CDFs for all tabular datasets, please refer to Appendix A.10.

Privacy preservability. The privacy preservability performances of synthesizers, as measured by the DCR, are presented in Table 4. DistVAE performs best in preserving privacy, with the highest DCR values compared to other methods. Notably, as the value of β increases in DistVAE, the DCR between the real training and synthetic datasets (R&S) also increases. This indicates that the risk of privacy leakage can be controlled by adjusting β, where higher values of β correspond to a higher level of privacy protection. Moreover, DistVAE consistently achieves large DCR values for the synthetic dataset (S) across all β values, indicating its ability to generate diverse synthetic samples. On the other hand, CTAB-GAN generates duplicated records in the synthetic dataset, resulting in

Table 3: Averaged marginal distributional similarity. K-S denotes the Kolmogorov-Smirnov statistic, and 1-WD represents the 1-Wasserstein distance. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

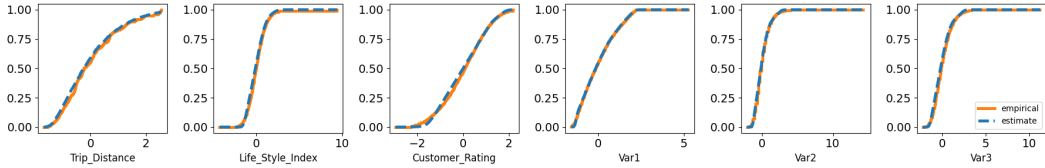| Model	Continuous		Discrete	
	K-S	1-WD	K-S	1-WD
CTGAN	$0.133_{\pm 0.106}$	$0.087_{\pm 0.025}$	$0.168_{\pm 0.195}$	$0.521_{\pm 0.532}$
TVAE	$0.196_{\pm 0.135}$	$0.220_{\pm 0.099}$	$0.385_{\pm 0.144}$	$1.681_{\pm 1.668}$
CTAB-GAN	$0.157_{\pm 0.089}$	$0.130_{\pm 0.037}$	$0.106_{\pm 0.083}$	$0.412_{\pm 0.378}$
DistVAE($\beta = 0.5$)	$0.090_{\pm 0.065}$	$\mathbf{0.075}_{\pm 0.026}$	$0.030_{\pm 0.017}$	$\mathbf{0.118}_{\pm 0.100}$
DistVAE($\beta = 1$)	$\mathbf{0.081}_{\pm 0.039}$	$0.083_{\pm 0.019}$	$\mathbf{0.027}_{\pm 0.021}$	$\mathbf{0.116}_{\pm 0.110}$
DistVAE($\beta = 5$)	$0.092_{\pm 0.037}$	$0.121_{\pm 0.058}$	$0.059_{\pm 0.034}$	$0.241_{\pm 0.163}$

Figure 1: `cabs` dataset. Empirical (solid orange) and estimated (dashed blue) CDFs of continuous and ordinal variables (Monte Carlo approximated with 5000 samples). We standardize covariates and remove observations outside the 1% and 99% percentile range.

relatively lower DCR scores for the synthetic dataset (S). For detailed DCR scores for all tabular datasets, please refer to Appendix A.10.

Table 4: Privacy preservability: Averaged distance to closest record (DCR) between real training and synthetic datasets (R&S), between the same real training datasets (R), and between the same synthetic datasets (S). Mean and standard deviation values are obtained from 10 repeated experiments. The DCR (R) score represents the baseline diversity of datasets. Higher is better.

Model	R&S	R	S
CTGAN	$0.426_{\pm 0.229}$	$0.237_{\pm 0.153}$	$0.356_{\pm 0.202}$
TVAE	$0.470_{\pm 0.181}$	$0.237_{\pm 0.153}$	$0.278_{\pm 0.195}$
CTAB-GAN	$0.508_{\pm 0.259}$	$0.237_{\pm 0.153}$	$0.039_{\pm 0.073}$
DistVAE($\beta = 0.5$)	$0.444_{\pm 0.250}$	$0.237_{\pm 0.153}$	$0.463_{\pm 0.288}$
DistVAE($\beta = 1$)	$0.463_{\pm 0.282}$	$0.237_{\pm 0.153}$	$0.479_{\pm 0.310}$
DistVAE($\beta = 5$)	$\mathbf{0.517}_{\pm 0.272}$	$0.237_{\pm 0.153}$	$\mathbf{0.511}_{\pm 0.335}$

To evaluate the membership inference attack, we prepare one attack model per class. The attack testing records comprise an equal number of real training and test records, distinguished by the labels *in* and *out*, respectively. Note that the test records are not employed in constructing the attack models. We employ gradient-boosting classifiers as the attack models, and for computational feasibility, we limit the number of attack models to one (i.e., $C = 1$).

For the membership inference attack evaluation, we utilize accuracy and AUC (Area Under Curve) as the evaluation metrics. Since the target labels (in/out) are balanced, and the task is a binary classification problem, these metrics are appropriate. The results presented in Table 5 reveal that both DistVAE and TVAE achieve nearly identical accuracy and AUC scores of 0.5. This indicates that the attack models can *not* distinguish between members of the real training and test datasets. Consequently, the membership inference attack is unsuccessful for both models. Therefore, DistVAE effectively generates synthetic datasets while ensuring privacy against membership inference attacks. A comprehensive assessment of membership inference attack performances for all tabular datasets can be found in Appendix A.10.

Table 5: Privacy preservability: Averaged membership inference attack performance. Mean and standard deviation values are obtained from 10 repeated experiments.

Model	Accuracy	AUC
TVAE	$0.495_{\pm 0.019}$	$0.495_{\pm 0.019}$
DistVAE($\beta = 0.5$)	$0.500_{\pm 0.003}$	$0.500_{\pm 0.003}$

Table 6: Privacy preservability: Averaged attribute disclosure performance with the F_1 score. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.262_{\pm 0.091}$	$0.282_{\pm 0.087}$	$0.275_{\pm 0.087}$
TVAE	$0.437_{\pm 0.162}$	$0.438_{\pm 0.160}$	$0.432_{\pm 0.162}$
CTAB-GAN	$\mathbf{0.257}_{\pm 0.123}$	$0.258_{\pm 0.114}$	$0.261_{\pm 0.111}$
DistVAE($\beta = 0.5$)	$0.328_{\pm 0.088}$	$0.328_{\pm 0.076}$	$0.310_{\pm 0.072}$
DistVAE($\beta = 1$)	$0.307_{\pm 0.073}$	$0.313_{\pm 0.068}$	$0.297_{\pm 0.066}$
DistVAE($\beta = 5$)	$0.265_{\pm 0.105}$	$\mathbf{0.253}_{\pm 0.103}$	$\mathbf{0.232}_{\pm 0.101}$

We present the attribute disclosure performance results in Table 6. For each value of k, we observe that as β increases, the F_1 score of DistVAE decreases. Also, DistVAE achieves the smallest F_1 score when k equals 10 and 100. Based on these results, we can conclude that DistVAE can generate synthetic datasets with a low risk of attribute disclosure, and the level of privacy preservation is controlled by β. Please refer to Appendix A.10 for a detailed evaluation of attribute disclosure performance for all tabular datasets.

Quantile estimation. To investigate the quantile estimation performance, we also evaluate DistVAE using the Vrate(α) metric [11]. The Vrate(α) is defined as $\frac{1}{|I_{test}|} \sum_{i \in I_{test}} I(x_i < \hat{Q}_\alpha)$, where $\alpha \in (0, 1)$, I_{test} is the set of indices for the test dataset, x_i is the i-th sample in the test dataset, and $\hat{Q}_\alpha$ is the empirical α-quantile of the synthetic data. Since Vrate(α) indicates the proportion of compliance samples in the test dataset, the Vrate(α) score should be close to α.

Table 7: Averaged Vrate(α) and $|\alpha - \text{Vrate}(\alpha)|$.

α	0.1	0.3	0.5	0.7	0.9
Vrate(α)	0.204	0.373	0.533	0.725	0.908
$|\alpha - \text{Vrate}(\alpha)|$	0.104	0.083	0.04	0.032	0.008

The Vrate(α) evaluation results are presented in Table 7. Table 7 shows that the ratio of violated test samples ($|\alpha - \text{Vrate}(\alpha)|$) decreases as α increases, indicating a better performance for estimating larger quantiles. However, the ratio of violated test samples is relatively large for smaller α values, which may be due to extremely skewed continuous variables, such as `capital-gain` and `capital-loss` from `adult` dataset, that make the quantile estimation unstable.

5 Conclusion and Limitations

This paper introduces a novel distributional learning method of VAE, which aims to effectively capture the underlying distribution of the observed dataset using a nonparametric approach. Our proposed method involves directly estimating CDFs using the CRPS loss while maintaining the mathematical derivation of the lower bound.

In our study, we confirm that the proposed decoder enhances the performance of generative models for tabular data. However, this conclusion relies on the assumption of conditional independence among the observed variables given the latent variable. When the dimension of the latent variable is small, this assumption is prone to violation. Therefore, in cases where the size of the latent

space is limited, the proposed nonparametric fitting of the decoder might not accurately represent the underlying distribution. Particularly in the image domain, where adjacent pixel values exhibit significant dependence, it remains uncertain whether our proposed model would lead to notable improvements in image data generation with a low-dimensional latent space.

Nevertheless, classical image datasets, such as CIFAR-10 [36], often exhibit pixel value distributions that deviate considerably from Gaussian, with the frequencies of edge values (0 and 255) dominating more than other pixel values [16, 53]. Other image datasets, as presented in [48], demonstrate multi-modality in pixel value distributions. These experimental findings suggest that leveraging the capacity of distributional learning could be advantageous in approximating the ground-truth distribution of image data when the latent variable effectively captures conditional independence among the image pixels. Consequently, we expect that compromising biases arising from violating conditional independence and marginally misspecified distributions may further enhance our results, and we leave it for future research.

On the other hand, we consider two approaches to enhance the performance of quantile estimation. Firstly, we plan to extend the parameterization of conditional quantile functions to a more flexible monotonic regression model. Secondly, we intend to incorporate the Uniform Pessimistic Risk (UPR) [26] into the VAE framework to handle lower quantile levels better. Furthermore, we are exploring the expansion of DistVAE into a time-series distributional forecasting model by adopting the conditional VAE framework [56]. This extension will enable the application of our method to time-series data, opening new avenues for distributional forecasting.

Acknowledgments and Disclosure of Funding

References

[1] Haleh Akrami, Anand Joshi, Sergul Aydore, and Richard Leahy. Deep quantile regression for uncertainty estimation in unsupervised and supervised lesion detection. *Machine Learning for Biomedical Imaging*, 1:1–23, 2022.

[2] Jonathan T. Barron. A general and adaptive robust loss function. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4326–4334, 2017.

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859 – 877, 2016.

[4] Axel Brando, Jose A Rodriguez, Jordi Vitria, and Alberto Rubio Muñoz. Modelling heterogeneous distributions with an uncountable mixture of asymmetric laplacians. *Advances in neural information processing systems*, 32, 2019.

[5] Gustav Bredell, Kyriakos Flouris, Krishna Chaitanya, Ertunc Erdil, and Ender Konukoglu. Explicitly minimizing the blur error of variational autoencoders. In *The Eleventh International Conference on Learning Representations*, 2023.

[6] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[8] Yuri Burda, Roger Baker Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.

[9] Alex J. Cannon. Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic Environmental Research and Risk Assessment*, 32:3207–3225, 2018.

[10] Lluís Castrejón, Nicolas Ballas, and Aaron C. Courville. Improved conditional vrnns for video prediction. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7607–7616, 2019.

[11] Cathy W.S. Chen, Richard Gerlach, Bruce B.K. Hwang, and Michael McAleer. Forecasting Value-at-Risk using nonlinear regression quantiles and the intra-day range. *International Journal of Forecasting*, 28(3):557–574, 2012.

[12] E. Choi, Siddharth Biswal, Bradley A. Malin, Jon D. Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine Learning in Health Care*, 2017.

[13] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[14] Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. A loss function for generative neural networks based on watson's perceptual model. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[15] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019.

[16] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[17] Andrés Diaz-Pinto, Adrián Colomer, Valery Naranjo, Sandra Morales, Yanwu Xu, and Alejandro F Frangi. Retinal image synthesis for glaucoma assessment using dcgan and vae models. In *Ideal*, 2018.

[18] Kevin Fang, Vaikkunth Mugunthan, Vayd Ramkumar, and Lalana Kagal. Overcoming challenges of synthetic data generation. *2022 IEEE International Conference on Big Data (Big Data)*, pages 262–270, 2022.

[19] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[20] Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd international conference on artificial intelligence and statistics*, pages 1901–1910. PMLR, 2019.

[21] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359 – 378, 2007.

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[23] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

[24] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[25] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

[26] Sungchul Hong and Jong-June Jeon. Uniform pessimistic risk and optimal portfolio. *ArXiv*, abs/2303.07158, 2023.

[27] Xianxu Hou, L. Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141, 2016.

[28] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13899–13909, 2020.

[29] Sanket Kamthe, Samuel A. Assefa, and Marc Peter Deisenroth. Copula flows for synthetic data generation. *ArXiv*, abs/2101.00598, 2021.

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.

[31] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[32] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[33] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[34] A Kiran and S Saravana Kumar. A comparative analysis of gan and vae based synthetic data generators for high dimensional, imbalanced tabular data. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–6. IEEE, 2023.

[35] Roger W. Koenker, Limin Peng, Victor Chernozhukov, and Xuming He. Handbook of quantile regression. 2017.

[36] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[37] Abhishek Kumar and Ben Poole. On implicit regularization in β-vaes. In *International Conference on Machine Learning*, pages 5480–5490. PMLR, 2020.

[38] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[39] Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 741–752. Curran Associates, Inc., 2020.

[40] Erich Leo Lehmann. Elements of large-sample theory. 1998.

[41] James Lucas, G. Tucker, Roger Baker Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Neural Information Processing Systems*, 2019.

[42] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[43] James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22:1087–1096, 1976.

[44] Stan Matwin, Jordi Nin, Morvarid Sehatkar, and Tomasz Szapiro. A review of attribute disclosure control. In *Advanced Research in Data Privacy*, 2015.

[45] Sangjun Moon, Jong-June Jeon, Jason Eng Hun Lee, and Yongdai Kim. Learning multiple quantiles with neural networks. *Journal of Computational and Graphical Statistics*, 30:1238 – 1248, 2021.

[46] Prateek Munjal, Akanksha Paul, and N. C. Krishnan. Implicit discriminator in variational autoencoder. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[47] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11:1071–1083, 2018.

[48] V. Parmar, Bogdan Penkovsky, Damien Querlioz, and Manan Suri. Hardware-efficient stochastic binary cnn architectures for near-sensor computing. *Frontiers in Neuroscience*, 15, 2022.

[49] J. Peters, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: Foundations and learning algorithms. 2017.

[50] Kostantinos N Plataniotis and Dimitris Hatzinakos. Gaussian mixtures and their applications to signal processing. *Advanced signal processing handbook*, pages 89–124, 2017.

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[52] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *ArXiv*, abs/1706.04987, 2017.

[53] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

[54] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2016.

[55] Nicki Skafte, Martin Jø rgensen, and Sø ren Hauberg. Reliable training and estimation of variance networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[56] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015.

[57] Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Student-t variational autoencoder for robust density estimation. In *International Joint Conference on Artificial Intelligence*, 2018.

[58] Jakub Tomczak and Max Welling. Vae with a vampprior. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR, 09–11 Apr 2018.

[59] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4):588–598, 2017.

[60] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[61] Bingyang Wen, Yupeng Cao, Fan Yang, Koduvayur Subbalakshmi, and Rajarathnam Chandramouli. Causal-TGAN: Causally-aware synthetic tabular data generative adversarial network, 2022.

[62] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv: Machine Learning*, 2017.

[63] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[64] Keming Yu and Rana Moyeed. Bayesian quantile regression. *Statistics & Probability Letters*, 54:437–447, 2001.

[65] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y. Chen. Ctab-gan: Effective table data synthesizing. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 97–112. PMLR, 17–19 Nov 2021.

A Appendix

A.1 Derivation of ELBO

$$\log p(\mathbf{x}; \theta, \beta)$$

$$= \log \sum_{k=1}^{K} p(\alpha_k) \int p(\mathbf{x}|\mathbf{z}, \alpha_k; \theta, \beta) p(\mathbf{z}) d\mathbf{z}$$

$$= \log \sum_{k=1}^{K} p(\alpha_k) \int p(\mathbf{x}|\mathbf{z}, \alpha_k; \theta, \beta) \frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x}; \phi)} q(\mathbf{z}|\mathbf{x}; \phi) d\mathbf{z}$$

$$\geq \sum_{k=1}^{K} p(\alpha_k) \int q(\mathbf{z}|\mathbf{x}; \phi) \log \left(p(\mathbf{x}|\mathbf{z}, \alpha_k; \theta, \beta) \frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x}; \phi)} \right) d\mathbf{z}$$

$$= \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}[\log p(\mathbf{x}|\mathbf{z}, \alpha_k; \theta, \beta)] - \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z}))$$

$$= \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_C} \log p(\mathbf{x}_j|\mathbf{z}, \alpha_k; \theta_j, \beta) + \sum_{j \in I_D} \log p(\mathbf{x}_j|\mathbf{z}; \theta_j, \beta) \right] - \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z}))$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_C} \frac{1}{K} \sum_{k=1}^{K} \log \frac{\alpha_k(1 - \alpha_k)}{\beta} - \rho_{\alpha_k} \left(\frac{\mathbf{x}_j - D_j(\alpha_k|\mathbf{z}, \theta_j)}{\beta} \right) \right]$$

$$+ \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_D} \sum_{l=1}^{T_j} I(\mathbf{x}_j = l) \cdot \log \pi_l(\mathbf{z}; \theta_j) \right] - \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z}))$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[-\frac{1}{\beta} \cdot \sum_{j \in I_C} \frac{1}{K} \sum_{k=1}^{K} \rho_{\alpha_k} \left(\mathbf{x}_j - D_j(\alpha_k|\mathbf{z}, \theta_j) \right) \right] + \frac{p}{K} \sum_{k=1}^{K} \log \alpha_k(1 - \alpha_k) - p \cdot \log \beta$$

$$+ \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_D} \sum_{l=1}^{T_j} I(\mathbf{x}_j = l) \cdot \log \pi_l(\mathbf{z}; \theta_j) \right] - \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z}))$$

$$= -\frac{1}{\beta} \left(\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_C} \frac{1}{K} \sum_{k=1}^{K} \rho_{\alpha_k} \left(\mathbf{x}_j - D_j(\alpha_k|\mathbf{z}, \theta_j) \right) \right] - \beta \frac{p}{K} \sum_{k=1}^{K} \log \alpha_k(1 - \alpha_k) + \beta p \cdot \log \beta \right.$$

$$\left. - \beta \cdot \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)} \left[\sum_{j \in I_D} \sum_{l=1}^{T_j} I(\mathbf{x}_j = l) \cdot \log \pi_l(\mathbf{z}; \theta_j) \right] + \beta \cdot \mathcal{KL}(q(\mathbf{z}|\mathbf{x}; \phi)\|p(\mathbf{z})) \right),$$

by Jensen's inequality.

A.2 Proof of Theorem 1

Proof.

$$\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\frac{1}{K}\sum_{k=1}^{K}\rho_{\alpha_k}\Big(\mathbf{x}_j - D_j(\alpha_k|\mathbf{z},\theta_j)\Big)\right] = \frac{1}{K}\sum_{k=1}^{K}h(\alpha_k)$$

$$= \sum_{k=1}^{K}h(\alpha_k)\cdot(\alpha_k - \alpha_{k-1}),$$

where $\alpha_0 := 0$ and $\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\rho_{\alpha_k}\Big(\mathbf{x}_j - D_j(\alpha_k|\mathbf{z},\theta_j)\Big)\right]$ is denoted as $h(\alpha_k)$.

Since $\alpha_k \in [\alpha_{k-1}, \alpha_k]$ and $h(\cdot) : [0,1] \mapsto \mathbb{R}$ is a continuous function, for $j \in I_C$,

$$\lim_{K\to\infty}\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\frac{1}{K}\sum_{k=1}^{K}\rho_{\alpha_k}\Big(\mathbf{x}_j - D_j(\alpha_k|\mathbf{z},\theta_j)\Big)\right] = \lim_{K\to\infty}\sum_{k=1}^{K}h(\alpha_k)\cdot(\alpha_k - \alpha_{k-1})$$

$$= \int_0^1 h(\alpha)d\alpha$$

$$= \int_0^1 \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\rho_\alpha\Big(\mathbf{x}_j - D_j(\alpha|\mathbf{z},\theta_j)\Big)\right]d\alpha$$

$$= \mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\phi)}\left[\int_0^1 \rho_\alpha\Big(\mathbf{x}_j - D_j(\alpha|\mathbf{z},\theta_j)\Big)d\alpha\right],$$

by the definition of the Riemann integral and the Fubini-Tonelli theorem. The proof is complete. $\square$

A.3 Proof of Theorem 2

Proof.

$$\min_\theta \mathbb{E}_{p(\mathbf{x})q(\mathbf{z}|\mathbf{x};\phi)}\sum_{j=1}^{p}\int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha,\mathbf{z};\theta_j))d\alpha = \mathbb{E}_{q(\mathbf{z};\phi)q(\mathbf{x}|\mathbf{z};\phi)}\sum_{j=1}^{p}\int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha,\mathbf{z};\theta_j))d\alpha$$

$$= \sum_{j=1}^{p}\mathbb{E}_{q(\mathbf{z};\phi)}\mathbb{E}_{q(\mathbf{x}_j|\mathbf{z};\phi)}\int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha,\mathbf{z};\theta_j))d\alpha.$$

So, for all $j = 1,\cdots,p$,

$$\theta_j^*(\phi) \in \arg\min_{\theta_j}\mathbb{E}_{q(\mathbf{z};\phi)}\mathbb{E}_{q(\mathbf{x}_j|\mathbf{z};\phi)}\int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha,\mathbf{z};\theta_j))d\alpha,$$

and it is proper scoring rule relative to $F_j(\cdot|\mathbf{z};\phi)$ for all $\mathbf{z} \in \mathbb{R}^d$. It implies that $D_j^{-1}(\mathbf{x}_j,\mathbf{z};\theta_j^*(\phi)) = F_j(\mathbf{x}_j|\mathbf{z};\phi)$, and $\frac{d}{d\mathbf{x}_j}D_j^{-1}(\mathbf{x}_j,\mathbf{z};\theta_j^*(\phi)) = q_j(\mathbf{x}_j|\mathbf{z};\phi)$, for all $\mathbf{x}_j \in \mathbb{R}$, by Assumption 2.

It follows that

$$\int\prod_{j=1}^{p}\frac{d}{d\mathbf{x}_j}D_j^{-1}(\mathbf{x}_j,\mathbf{z};\theta_j^*(\phi))q(\mathbf{z};\phi)d\mathbf{z} = \int\prod_{j=1}^{p}q_j(\mathbf{x}_j|\mathbf{z};\phi)q(\mathbf{z};\phi)d\mathbf{z}$$

$$= \int q(\mathbf{x}|\mathbf{z};\phi)q(\mathbf{z};\phi)d\mathbf{z}$$

$$= \int\frac{p(\mathbf{x})q(\mathbf{z}|\mathbf{x};\phi)}{q(\mathbf{z};\phi)}q(\mathbf{z};\phi)d\mathbf{z}$$

$$= \int p(\mathbf{x})q(\mathbf{z}|\mathbf{x};\phi)d\mathbf{z}$$

$$= p(\mathbf{x}).$$

The proof is complete. $\square$

A.4 Proof of Theorem 3

Proof.

$$\min_\theta \mathbb{E}_{p(\mathbf{x})q(\mathbf{z}|\mathbf{x};\phi)} \sum_{j=1}^p \int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))d\alpha \quad = \quad \mathbb{E}_{q(\mathbf{z};\phi)q(\mathbf{x}|\mathbf{z};\phi)} \sum_{j=1}^p \int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))d\alpha$$

$$= \quad \sum_{j=1}^p \mathbb{E}_{q(\mathbf{z};\phi)} \mathbb{E}_{q(\mathbf{x}_j|\mathbf{z};\phi)} \int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))d\alpha.$$

So, for all $j = 1, \cdots, p$,

$$\theta_j^*(\phi) \in \arg\min_{\theta_j} \mathbb{E}_{q(\mathbf{z};\phi)} \mathbb{E}_{q(\mathbf{x}_j|\mathbf{z};\phi)} \int_0^1 \rho_\alpha(\mathbf{x}_j - D_j(\alpha, \mathbf{z}; \theta_j))d\alpha,$$

and it is proper scoring rule relative to $F_j(\cdot|\mathbf{z}; \phi)$ for all $\mathbf{z} \in \mathbb{R}^d$. It implies that $D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi)) = F_j(\mathbf{x}_j|\mathbf{z}; \phi)$, and $\frac{d}{d\mathbf{x}_j} D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi)) = q_j(\mathbf{x}_j|\mathbf{z}; \phi)$, for all $\mathbf{x}_j \in \mathbb{R}$, by Assumption 2.

It follows that

$$\mathcal{KL}\left(p(\mathbf{x}) \left\| \int \prod_{j=1}^p \frac{d}{d\mathbf{x}_j} D_j^{-1}(\mathbf{x}_j, \mathbf{z}; \theta_j^*(\phi))p(\mathbf{z})d\mathbf{z}\right.\right) \quad = \quad \mathcal{KL}\left(p(\mathbf{x}) \left\| \int \prod_{j=1}^p q_j(\mathbf{x}_j|\mathbf{z}; \phi)p(\mathbf{z})d\mathbf{z}\right.\right)$$

$$= \quad \mathcal{KL}\left(p(\mathbf{x}) \left\| \int q(\mathbf{x}|\mathbf{z}; \phi)p(\mathbf{z})d\mathbf{z}\right.\right)$$

$$= \quad \mathcal{KL}\left(p(\mathbf{x}) \left\| \int \frac{p(\mathbf{x})q(\mathbf{z}|\mathbf{x}; \phi)}{q(\mathbf{z}; \phi)}p(\mathbf{z})d\mathbf{z}\right.\right)$$

$$= \quad \mathcal{KL}\left(p(\mathbf{x}) \left\| p(\mathbf{x}) \int \frac{p(\mathbf{z})q(\mathbf{z}|\mathbf{x}; \phi)}{q(\mathbf{z}; \phi)}d\mathbf{z}\right.\right)$$

$$= \quad \mathbb{E}_{p(\mathbf{x})}\left[-\log \int \frac{p(\mathbf{z})q(\mathbf{z}|\mathbf{x}; \phi)}{q(\mathbf{z}; \phi)}d\mathbf{z}\right]$$

$$\leq \quad \mathbb{E}_{p(\mathbf{x})}\left[\int q(\mathbf{z}|\mathbf{x}; \phi)\log\frac{q(\mathbf{z}; \phi)}{p(\mathbf{z})}d\mathbf{z}\right]$$

$$= \quad \iint p(\mathbf{x})q(\mathbf{z}|\mathbf{x}; \phi)\log\frac{q(\mathbf{z}; \phi)}{p(\mathbf{z})}d\mathbf{z}d\mathbf{x}$$

$$= \quad \iint p(\mathbf{x})q(\mathbf{z}|\mathbf{x}; \phi)\log\frac{q(\mathbf{z}; \phi)}{p(\mathbf{z})}d\mathbf{x}d\mathbf{z}$$

$$= \quad \int q(\mathbf{z}; \phi)\log\frac{q(\mathbf{z}; \phi)}{p(\mathbf{z})}d\mathbf{z}$$

$$= \quad \mathcal{KL}(q(\mathbf{z}; \phi)\|p(\mathbf{z}))$$

$$< \quad \epsilon,$$

by Jensen's inequality, Fubini-Tonelli theorem, and the assumptions. The proof is complete. □

A.5 Closed Form Loss

$$2 \cdot \int_0^1 \rho_\alpha\left(\mathbf{x}_j - D_j(\alpha, \mathbf{z}, \theta_j)\right)d\alpha \quad = \quad (2\tilde{\alpha}_j - 1)\mathbf{x}_j + (1 - 2\tilde{\alpha}_j)\gamma^{(j)}(\mathbf{z})$$

$$+ \quad \sum_{m=1}^M b_m^{(j)}(\mathbf{z})\left(\frac{1 - d_m^3}{3} - d_m - \max(\tilde{\alpha}_j, d_m)^2 + 2\max(\tilde{\alpha}_j, d_m)d_m\right),$$

where $D_j(\tilde{\alpha}_j, \mathbf{z}; \theta_j) = \mathbf{x}_j$, $\tilde{\alpha}_j = \frac{\mathbf{x}_j - \gamma^{(j)}(\mathbf{z}) + \sum_{m=0}^{m_0} b_m^{(j)}(\mathbf{z})d_m}{\sum_{m=0}^{m_0} b_m^{(j)}(\mathbf{z})}$, and $D_j(d_{m_0}, \mathbf{z}; \theta_j) \leq \mathbf{x}_j \leq D_j(d_{m_0+1}, \mathbf{z}; \theta_j)$.

16

A.6 Discretization of Estimated CDF

To ensure appropriate discretization of the estimated CDF for ordinal variables, we propose a post-ad-hoc discretization step [53]. We focus on the case where $p = 1$ and $q = 0$ for brevity. We denote the set of observed possible values for the ordinal variable as $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$. The discretization algorithm for the estimated CDF is presented in Algorithm 1, and we provide an example of the discretization algorithm's outcome in Figure 2.

Algorithm 1 Discretization of Estimated CDF

Input $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$, Estimated CDF $\hat{F}(\cdot; \theta)$
Output Discretized CDF $\hat{F}^*(\cdot; \theta)$
(1) Compute $\hat{F}(x^{(i)} - 0.5; \theta)$ and $\hat{F}(x^{(i)} + 0.5; \theta)$ for $i = 1, \cdots, m$.
(2) Discretization: For $i = 1, \cdots, m$,

$$
\begin{aligned}
\hat{F}^*(x^{(i)}; \theta) &:= \hat{F}^*(x^{(i-1)}; \theta) \\
&+ \hat{F}(x^{(i)} + 0.5; \theta) - \hat{F}(x^{(i)} - 0.5; \theta),
\end{aligned}
$$

where $\hat{F}^*(x^{(0)}; \theta) := 0$.
(3) Ensure monotonicity: For $i = 1, \cdots, m-1$, if $\hat{F}^*(x^{(i)}; \theta) > \hat{F}^*(x^{(i+1)}; \theta)$,

$$
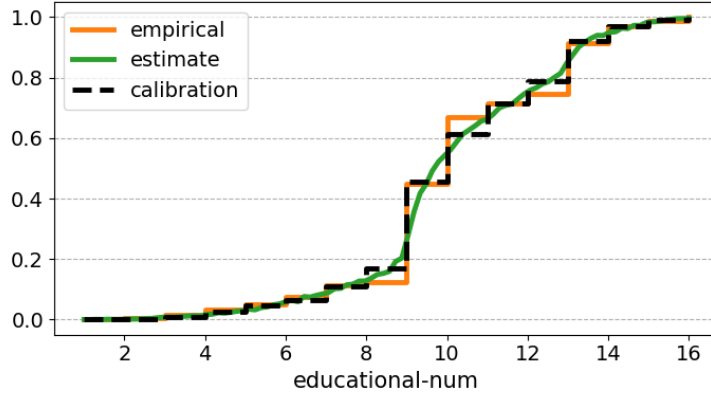\hat{F}^*(x^{(i+1)}; \theta) := \hat{F}^*(x^{(i)}; \theta).
$$

Figure 2: Discretized CDF for ordinal variable `educational-num` of `adult` dataset. 'estimate' indicates $\hat{F}(\cdot; \theta)$, 'calibration' indicates $\hat{F}^*(\cdot; \theta)$, and 'empirical' indicates the empirical CDF of the observed dataset.

A.7 Membership Inference Attack

[47] propose the customized *membership inference attack* method of [54] to attack the GAN-based synthesizer. Similarly, we propose the customized membership inference attack method of [54] to attack the VAE-based synthesizer.

Assumption 3 ([54]). *In the membership inference attack, the attacker attacks the target model under the following assumptions:*

> *(A1) The attacker is only allowed for black-box access, where the attacker can only supply inputs to the model and receive the model's output(s).*
>
> *(A2) The attacker can obtain as many outputs as they want from a target model to attack.*
>
> *(A3) The real and synthetic datasets should not have common records.*
>
> *(A4) The attacker knows the algorithm and architecture of the target model.*

Denote D^*_{train} and D^*_{test} as the real training and test datasets. Under Assumption 3, the overall steps of the membership inference attack are outlined below:

1. Generate shadow training and test datasets $D^{(i)}_{train}, D^{(i)}_{test}, i = 1, \cdots, C$ from M^* by (A1), where M^* is the model attacker wants to attack. By (A2), the attacker is allowed to obtain shadow datasets such as $|D^{(i)}_{train}| = |D^*_{train}|$ and $D^{(i)}_{train} \cap D^{(j)}_{test} = \varnothing$, for $i = 1, \cdots, C$. Under (A3), $D^{(i)}_{train} \cap D^*_{train} = \varnothing$, for $i = 1, \cdots, C$.

2. Train shadow models $M_1, \cdots, M_C$ under (A4), i.e., each shadow model is trained similarly to the target model M^*.

3. For $i = 1, \cdots, C$,

   (a) Obtain representation vectors $\mathbf{z}$ from the encoder of M_i with the input of $D^{(i)}_{train}$. Then, attacking training records are $(\mathbf{y}, \mathbf{z}, in)$.

   (b) Obtain representation vectors $\mathbf{z}$ from the encoder of M_i with the input of $D^{(i)}_{test}$. Then, attacking training records are $(\mathbf{y}, \mathbf{z}, out)$.

   where $\mathbf{y}$ is the labels of shadow dataset records. And we assume that $\mathbf{y}$ consists of the MLu classification target.

4. Merge all attack training records, $(\mathbf{y}, \mathbf{z}, in/out)$.

5. For each class of $\mathbf{y}$, train *attack model* which is a binary classification model which classifies in/out based on the representation vectors $\mathbf{z}$.

6. Now it is ready to attack.

Note that we use the representation vector of VAE instead of the output of the GAN discriminator [47].

A.8 Dataset Descriptions

Websites

- `covertype:` `https://www.kaggle.com/datasets/uciml/forest-cover-type-dataset`
- `credit:` `https://www.kaggle.com/c/home-credit-default-risk`
- `loan:` `https://www.kaggle.com/datasets/teertha/personal-loan-modeling`
- `adult:` `https://www.kaggle.com/datasets/uciml/adult-census-income`
- `cabs:` `https://www.kaggle.com/datasets/arashnic/taxi-pricing-with-mobility-analytics?select=test.csv`
- `kings:` `https://www.kaggle.com/datasets/harlfoxem/housesalesprediction`

Table 8: Description of datasets. #C represents the number of continuous and ordinal variables. #D denotes the number of discrete variables.

Dataset	Train/Test Split	Regression Target	Classification Target	#C	#D
covertype	45k/5k	Elevation	Cover_Type	10	1
credit	45k/5k	AMT_CREDIT	TARGET	10	9
loan	4k/1k	Age	Personal Loan	5	6
adult	40k/5k	age	income	5	9
cabs	40k/1k	Trip_Distance	Surge_Pricing_Type	6	7
kings	20k/1k	long	condition	11	7

A.9 Experimental Settings

We run all experiments using Geforce RTX 3090 GPU, and our experimental codes are all available with `pytorch`.

Table 9: Hyper-parameter settings for tabular dataset experiments.

Model	epochs	batch size	learning rate	β (or decoder std range)	d	M
CTGAN	300	500	0.0002	-	2	-
TVAE	200	256	0.005	[0.1, 1]	2	-
CTAB-GAN	150	500	0.0002	-	2	-
DistVAE	100	256	0.001	0.5	2	10

Table 10: The number of model parameters for tabular dataset experiments.

Model	covtype	credit	adult	loan	cabs	kings
CTGAN	20k	32k	52k	13k	30k	51k
TVAE	10k	12k	13k	6k	10k	10k
CTAB-GAN	12k	13k	14k	5k	12k	15k
DistVAE	10k	12k	13k	6k	10k	16k

Table 11: Classifier and regressor used to evaluate synthetic data quality. The names of all parameters used in the description are consistent with those defined in corresponding packages.

Tasks	Model	Description
Regression	Linear Regression	Package: `statsmodels.api.sm.OLS`, setting: without intercept, defaulted values
	Random Forest	Package: `sklearn.ensemble.RandomForestRegressor`, setting: random_state=0, defaulted values
	Gradient Boosting	Package: `sklearn.ensemble.GradientBoostingRegressor`, setting: random_state=0, defaulted values
Classification	Logistic Regression	Package: `sklearn.linear_model.LogisticRegression`, setting: multi_class='ovr', fit_intercept=False, defaulted values
	Random Forest	Package: `sklearn.ensemble.RandomForestClassifier`, setting: random_state=0, defaulted values
	Gradient Boosting	Package: `sklearn.ensemble.GradientBoostingClassifier`, setting: random_state=0, defaulted values

A.10 Detailed Experimental Results
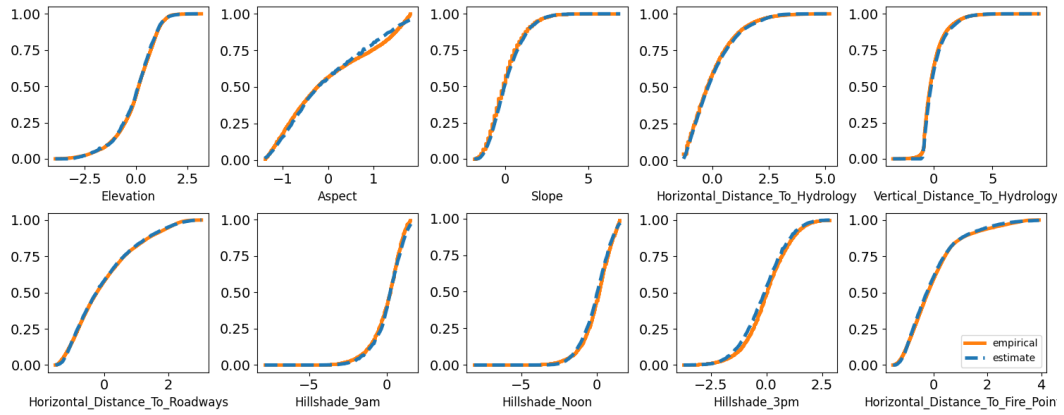
(a) covtype
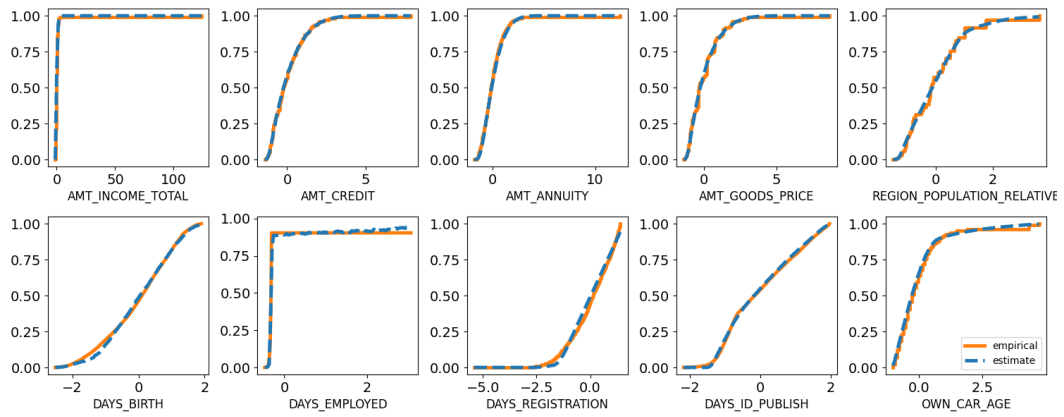
(b) credit
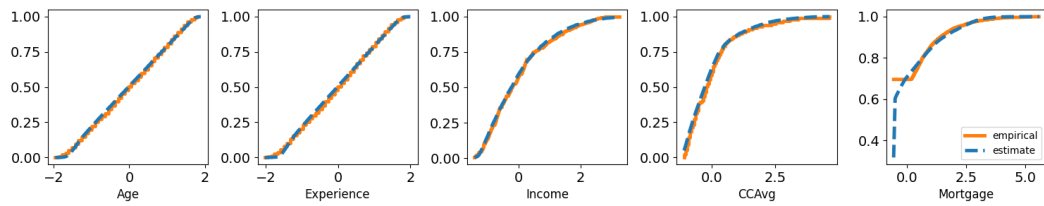
(c) loan

Figure 3: Empirical and estimated CDFs of continuous and count variables (Monte Carlo approximated with 5000 samples). We standardize covariates and remove observations outside the 1% and 99% percentile range.

(a) `adult`

(b) `cabs`

(c) `kings`
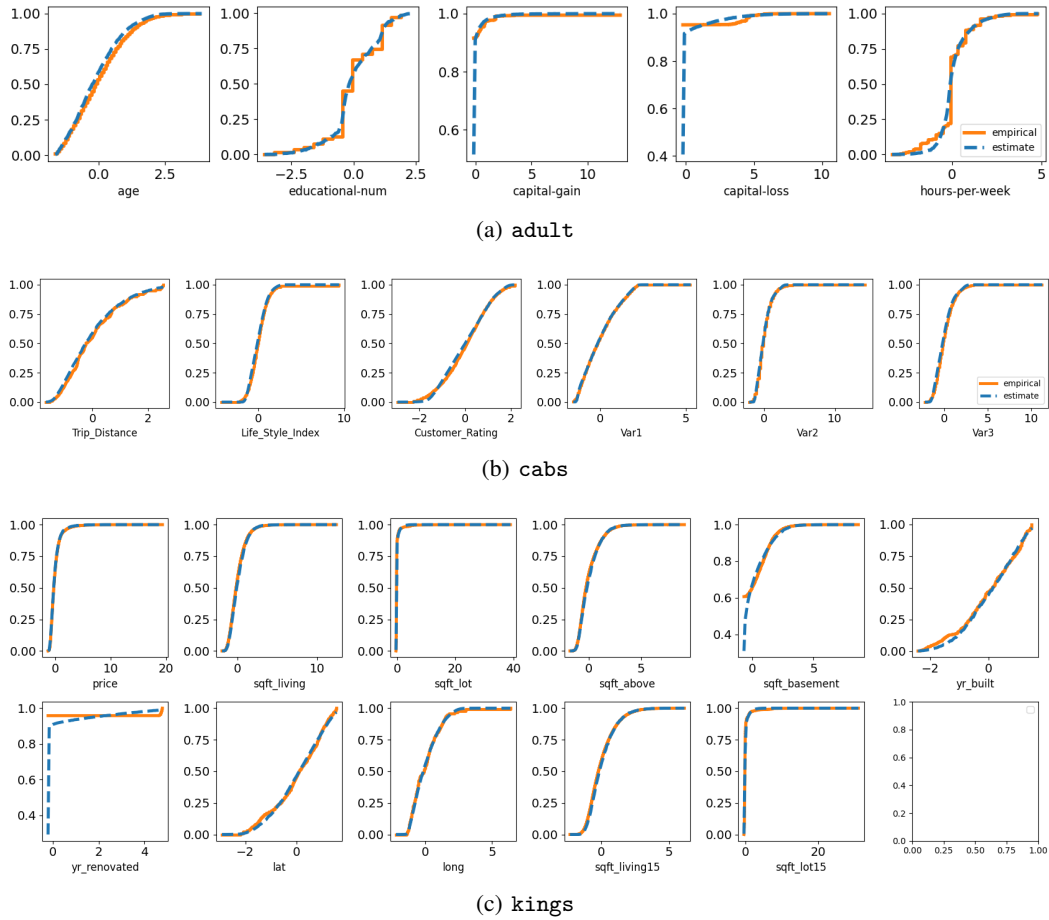
Figure 4: Empirical and estimated CDFs of continuous and count variables (Monte Carlo approximated with 5000 samples). We standardize covariates and remove observations outside the 1% and 99% percentile range.

Table 12: MLu metrics (MARE, F_1). Mean and standard deviation values are obtained from 10 repeated experiments. ↑ denotes higher is better and ↓ denotes lower is better.

(a)

Dataset	covertype		credit		loan	
Model	MARE ↓	F_1 ↑	MARE ↓	F_1 ↑	MARE ↓	F_1 ↑
Baseline	0.035	0.718	0.064	0.927	0.020	0.948
CTGAN	$0.058_{\pm0.007}$	$0.227_{\pm0.030}$	$0.593_{\pm0.150}$	$0.914_{\pm0.006}$	$0.258_{\pm0.020}$	$0.842_{\pm0.109}$
TVAE	$0.079_{\pm0.007}$	$0.504_{\pm0.032}$	$0.260_{\pm0.135}$	$0.091_{\pm0.286}$	$0.124_{\pm0.033}$	$0.785_{\pm0.288}$
CTAB-GAN	$0.065_{\pm0.004}$	$0.493_{\pm0.027}$	$0.887_{\pm0.351}$	$0.913_{\pm0.005}$	$0.247_{\pm0.019}$	$0.887_{\pm0.026}$
DistVAE($\beta = 0.5$)	$0.044_{\pm0.002}$	$0.605_{\pm0.006}$	$0.763_{\pm0.068}$	$0.926_{\pm0.001}$	$0.249_{\pm0.007}$	$0.914_{\pm0.009}$
DistVAE($\beta = 1$)	$0.045_{\pm0.001}$	$0.557_{\pm0.007}$	$0.774_{\pm0.035}$	$0.926_{\pm0.000}$	$0.249_{\pm0.005}$	$0.897_{\pm0.002}$
DistVAE($\beta = 5$)	$0.063_{\pm0.001}$	$0.443_{\pm0.018}$	$0.870_{\pm0.022}$	$0.904_{\pm0.010}$	$0.249_{\pm0.005}$	$0.893_{\pm0.005}$

(b)

Dataset	adult		cabs		kings	
Model	MARE ↓	F_1 ↑	MARE ↓	F_1 ↑	MARE ↓	F_1 ↑
Baseline	0.216	0.854	0.565	0.743	0.001	0.695
CTGAN	$0.297_{\pm0.030}$	$0.796_{\pm0.022}$	$0.721_{\pm0.046}$	$0.674_{\pm0.024}$	$0.001_{\pm0.000}$	$0.579_{\pm0.035}$
TVAE	$0.238_{\pm0.006}$	$0.809_{\pm0.016}$	$0.642_{\pm0.035}$	$0.689_{\pm0.031}$	$0.010_{\pm0.005}$	$0.687_{\pm0.041}$
CTAB-GAN	$0.321_{\pm0.036}$	$0.730_{\pm0.069}$	$0.894_{\pm0.116}$	$0.582_{\pm0.047}$	$0.001_{\pm0.000}$	$0.608_{\pm0.022}$
DistVAE($\beta = 0.5$)	$0.232_{\pm0.004}$	$0.825_{\pm0.009}$	$0.803_{\pm0.129}$	$0.707_{\pm0.010}$	$0.001_{\pm0.000}$	$0.640_{\pm0.002}$
DistVAE($\beta = 1$)	$0.234_{\pm0.006}$	$0.822_{\pm0.003}$	$0.760_{\pm0.062}$	$0.725_{\pm0.004}$	$0.001_{\pm0.000}$	$0.644_{\pm0.003}$
DistVAE($\beta = 5$)	$0.327_{\pm0.008}$	$0.751_{\pm0.010}$	$0.839_{\pm0.042}$	$0.447_{\pm0.009}$	$0.002_{\pm0.000}$	$0.637_{\pm0.004}$
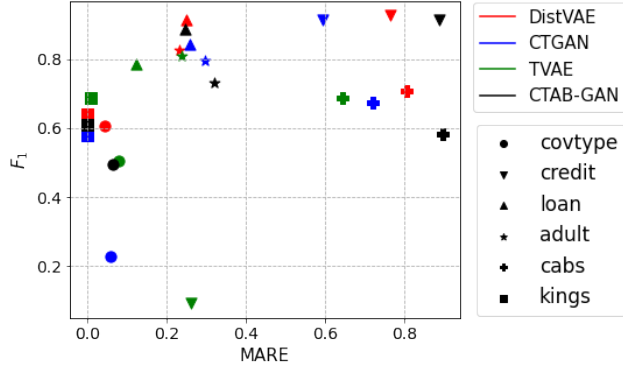
Figure 5: Machine learning utilities for compared models and real tabular datasets.

For a detailed comparison of the models and their performance, we present the paired (MARE, F_1) scores for all tabular datasets in Figure 5. A better score, indicating a superior performance in terms of MLu, is represented by a dot located in the upper left corner. Notably, Figure 5 consistently demonstrates that DistVAE achieves the best or at least competitive MLu across all tabular datasets. TVAE exhibits a notably low F_1 score in the credit dataset because it fails to handle the highly imbalanced discrete target variable. This comparative analysis highlights the strong MLu performance of DistVAE and the specific limitations of TVAE in certain scenarios.

Table 13: Correlation structure similarity. 'CorrDist' represents L_2 distance between the correlation matrix of synthetic and real datasets. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

Dataset	covertype	credit	loan	adult	cabs	kings
Model	CorrDist	CorrDist	CorrDist	CorrDist	CorrDist	CorrDist
CTGAN	$2.167_{\pm 0.419}$	$2.323_{\pm 0.362}$	$2.282_{\pm 0.177}$	$1.788_{\pm 0.217}$	$1.679_{\pm 0.129}$	$2.839_{\pm 0.246}$
TVAE	$1.969_{\pm 0.146}$	$4.021_{\pm 0.451}$	$2.404_{\pm 0.408}$	$2.231_{\pm 0.269}$	$3.136_{\pm 0.686}$	$2.665_{\pm 0.296}$
CTAB-GAN	$2.351_{\pm 0.185}$	$2.696_{\pm 0.275}$	$2.073_{\pm 0.110}$	$2.387_{\pm 0.470}$	$2.532_{\pm 0.225}$	$3.411_{\pm 0.399}$
DistVAE($\beta = 0.5$)	$1.179_{\pm 0.090}$	$2.072_{\pm 0.162}$	$1.654_{\pm 0.050}$	$0.830_{\pm 0.078}$	$1.481_{\pm 0.071}$	$1.559_{\pm 0.135}$
DistVAE($\beta = 1$)	$2.359_{\pm 0.018}$	$2.229_{\pm 0.102}$	$1.910_{\pm 0.019}$	$0.746_{\pm 0.042}$	$1.495_{\pm 0.047}$	$1.621_{\pm 0.149}$
DistVAE($\beta = 5$)	$2.946_{\pm 0.007}$	$3.161_{\pm 0.006}$	$2.113_{\pm 0.015}$	$3.186_{\pm 0.007}$	$1.930_{\pm 0.004}$	$5.339_{\pm 0.006}$

Table 14: Marginal statistical similarity. K-S denotes the KolmogorovSmirnov statistic and 1-WD represents the 1-Wasserstein distance. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

(a) Continuous

Dataset	covertype		credit		loan	
Model	K-S	1-WD	K-S	1-WD	K-S	1-WD
CTGAN	$0.080_{\pm 0.011}$	$0.108_{\pm 0.014}$	$0.132_{\pm 0.017}$	$0.088_{\pm 0.013}$	$0.112_{\pm 0.055}$	$0.061_{\pm 0.016}$
TVAE	$0.088_{\pm 0.008}$	$0.156_{\pm 0.018}$	$0.156_{\pm 0.036}$	$0.200_{\pm 0.035}$	$0.162_{\pm 0.024}$	$0.201_{\pm 0.024}$
CTAB-GAN	$0.073_{\pm 0.011}$	$0.096_{\pm 0.014}$	$0.140_{\pm 0.021}$	$0.111_{\pm 0.013}$	$0.181_{\pm 0.070}$	$0.153_{\pm 0.023}$
DistVAE($\beta = 0.5$)	$0.032_{\pm 0.003}$	$0.041_{\pm 0.005}$	$0.088_{\pm 0.014}$	$0.089_{\pm 0.004}$	$0.060_{\pm 0.010}$	$0.048_{\pm 0.003}$
DistVAE($\beta = 1$)	$0.049_{\pm 0.001}$	$0.079_{\pm 0.001}$	$0.084_{\pm 0.008}$	$0.093_{\pm 0.002}$	$0.056_{\pm 0.003}$	$0.054_{\pm 0.002}$
DistVAE($\beta = 5$)	$0.064_{\pm 0.001}$	$0.112_{\pm 0.000}$	$0.146_{\pm 0.003}$	$0.113_{\pm 0.001}$	$0.057_{\pm 0.003}$	$0.056_{\pm 0.002}$

(b) Continuous

Dataset	adult		cabs		kings	
Model	K-S	1-WD	K-S	1-WD	K-S	1-WD
CTGAN	$0.323_{\pm 0.126}$	$0.086_{\pm 0.017}$	$0.045_{\pm 0.007}$	$0.060_{\pm 0.011}$	$0.109_{\pm 0.012}$	$0.116_{\pm 0.016}$
TVAE	$0.477_{\pm 0.053}$	$0.414_{\pm 0.070}$	$0.098_{\pm 0.008}$	$0.139_{\pm 0.023}$	$0.195_{\pm 0.018}$	$0.213_{\pm 0.046}$
CTAB-GAN	$0.275_{\pm 0.122}$	$0.178_{\pm 0.053}$	$0.086_{\pm 0.008}$	$0.118_{\pm 0.015}$	$0.188_{\pm 0.035}$	$0.128_{\pm 0.019}$
DistVAE($\beta = 0.5$)	$0.209_{\pm 0.064}$	$0.114_{\pm 0.009}$	$0.044_{\pm 0.003}$	$0.067_{\pm 0.003}$	$0.110_{\pm 0.008}$	$0.089_{\pm 0.003}$
DistVAE($\beta = 1$)	$0.138_{\pm 0.041}$	$0.111_{\pm 0.007}$	$0.046_{\pm 0.002}$	$0.068_{\pm 0.003}$	$0.115_{\pm 0.007}$	$0.093_{\pm 0.002}$
DistVAE($\beta = 5$)	$0.115_{\pm 0.005}$	$0.234_{\pm 0.002}$	$0.052_{\pm 0.001}$	$0.074_{\pm 0.002}$	$0.120_{\pm 0.002}$	$0.137_{\pm 0.002}$

(c) Discrete

Dataset	covertype		credit		loan	
Model	K-S	1-WD	K-S	1-WD	K-S	1-WD
CTGAN	$0.591_{\pm 0.003}$	$1.629_{\pm 0.011}$	$0.061_{\pm 0.008}$	$0.147_{\pm 0.024}$	$0.070_{\pm 0.010}$	$0.076_{\pm 0.013}$
TVAE	$0.238_{\pm 0.042}$	$0.606_{\pm 0.033}$	$0.583_{\pm 0.045}$	$1.566_{\pm 0.116}$	$0.193_{\pm 0.028}$	$0.221_{\pm 0.043}$
CTAB-GAN	$0.052_{\pm 0.028}$	$0.180_{\pm 0.126}$	$0.034_{\pm 0.006}$	$0.076_{\pm 0.018}$	$0.039_{\pm 0.011}$	$0.046_{\pm 0.013}$
DistVAE($\beta = 0.5$)	$0.023_{\pm 0.010}$	$0.073_{\pm 0.046}$	$0.020_{\pm 0.003}$	$0.046_{\pm 0.006}$	$0.019_{\pm 0.005}$	$0.027_{\pm 0.008}$
DistVAE($\beta = 1$)	$0.011_{\pm 0.005}$	$0.036_{\pm 0.025}$	$0.018_{\pm 0.001}$	$0.042_{\pm 0.003}$	$0.011_{\pm 0.002}$	$0.015_{\pm 0.005}$
DistVAE($\beta = 5$)	$0.109_{\pm 0.003}$	$0.379_{\pm 0.014}$	$0.027_{\pm 0.001}$	$0.068_{\pm 0.004}$	$0.009_{\pm 0.002}$	$0.013_{\pm 0.005}$

(d) Discrete

Dataset	adult		cabs		kings	
Model	K-S	1-WD	K-S	1-WD	K-S	1-WD
CTGAN	$0.065_{\pm 0.008}$	$0.463_{\pm 0.051}$	$0.069_{\pm 0.016}$	$0.238_{\pm 0.036}$	$0.140_{\pm 0.018}$	$0.529_{\pm 0.064}$
TVAE	$0.479_{\pm 0.048}$	$5.228_{\pm 0.273}$	$0.411_{\pm 0.088}$	$1.202_{\pm 0.212}$	$0.405_{\pm 0.034}$	$1.262_{\pm 0.099}$
CTAB-GAN	$0.169_{\pm 0.018}$	$0.745_{\pm 0.294}$	$0.086_{\pm 0.013}$	$0.443_{\pm 0.074}$	$0.255_{\pm 0.027}$	$0.979_{\pm 0.107}$
DistVAE($\beta = 0.5$)	$0.037_{\pm 0.004}$	$0.248_{\pm 0.026}$	$0.060_{\pm 0.017}$	$0.241_{\pm 0.084}$	$0.022_{\pm 0.004}$	$0.071_{\pm 0.012}$
DistVAE($\beta = 1$)	$0.031_{\pm 0.004}$	$0.239_{\pm 0.019}$	$0.070_{\pm 0.003}$	$0.292_{\pm 0.010}$	$0.023_{\pm 0.003}$	$0.073_{\pm 0.008}$
DistVAE($\beta = 5$)	$0.070_{\pm 0.002}$	$0.457_{\pm 0.010}$	$0.083_{\pm 0.003}$	$0.326_{\pm 0.011}$	$0.059_{\pm 0.002}$	$0.204_{\pm 0.011}$

Table 15: Privacy preservability: Distance to closest record (DCR) between real training and synthetic datasets (R&S), between the same real training datasets (R), and between the same synthetic datasets (S). Mean and standard deviation values are obtained from 10 repeated experiments. Higher is better.

(a)

Dataset	covertype		
Model	R&S	R	S
CTGAN	$0.715_{\pm 0.026}$	$0.329_{\pm 0.000}$	$0.514_{\pm 0.094}$
TVAE	$0.676_{\pm 0.031}$	$0.329_{\pm 0.000}$	$0.482_{\pm 0.025}$
CTAB-GAN	$0.892_{\pm 0.031}$	$0.329_{\pm 0.000}$	$0.011_{\pm 0.005}$
DistVAE($\beta = 0.5$)	$0.765_{\pm 0.008}$	$0.329_{\pm 0.000}$	$0.819_{\pm 0.010}$
DistVAE($\beta = 1$)	$0.878_{\pm 0.008}$	$0.329_{\pm 0.000}$	$0.906_{\pm 0.009}$
DistVAE($\beta = 5$)	$0.907_{\pm 0.012}$	$0.329_{\pm 0.000}$	$0.939_{\pm 0.008}$

(b)

Dataset	credit		
Model	R&S	R	S
CTGAN	$0.624_{\pm 0.033}$	$0.452_{\pm 0.000}$	$0.592_{\pm 0.061}$
TVAE	$0.627_{\pm 0.118}$	$0.452_{\pm 0.000}$	$0.423_{\pm 0.212}$
CTAB-GAN	$0.715_{\pm 0.025}$	$0.452_{\pm 0.000}$	$0.014_{\pm 0.006}$
DistVAE($\beta = 0.5$)	$0.692_{\pm 0.012}$	$0.452_{\pm 0.000}$	$0.742_{\pm 0.012}$
DistVAE($\beta = 1$)	$0.700_{\pm 0.006}$	$0.452_{\pm 0.000}$	$0.750_{\pm 0.005}$
DistVAE($\beta = 5$)	$0.718_{\pm 0.007}$	$0.452_{\pm 0.000}$	$0.757_{\pm 0.007}$

(c)

Dataset	loan		
Model	R&S	R	S
CTGAN	$0.249_{\pm 0.017}$	$0.109_{\pm 0.000}$	$0.243_{\pm 0.024}$
TVAE	$0.298_{\pm 0.057}$	$0.109_{\pm 0.000}$	$0.154_{\pm 0.026}$
CTAB-GAN	$0.272_{\pm 0.024}$	$0.109_{\pm 0.000}$	$0.076_{\pm 0.051}$
DistVAE($\beta = 0.5$)	$0.244_{\pm 0.025}$	$0.109_{\pm 0.000}$	$0.245_{\pm 0.020}$
DistVAE($\beta = 1$)	$0.238_{\pm 0.019}$	$0.109_{\pm 0.000}$	$0.241_{\pm 0.023}$
DistVAE($\beta = 5$)	$0.243_{\pm 0.015}$	$0.109_{\pm 0.000}$	$0.240_{\pm 0.025}$

(d)

Dataset	adult		
Model	R&S	R	S
CTGAN	$0.063_{\pm 0.017}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
TVAE	$0.277_{\pm 0.039}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
CTAB-GAN	$0.152_{\pm 0.064}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
DistVAE($\beta = 0.5$)	$0.060_{\pm 0.040}$	$0.000_{\pm 0.000}$	$0.005_{\pm 0.001}$
DistVAE($\beta = 1$)	$0.048_{\pm 0.027}$	$0.000_{\pm 0.000}$	$0.003_{\pm 0.000}$
DistVAE($\beta = 5$)	$0.177_{\pm 0.002}$	$0.000_{\pm 0.000}$	$0.001_{\pm 0.000}$

(e)

Dataset	cabs		
Model	R&S	R	S
CTGAN	$0.353_{\pm 0.006}$	$0.332_{\pm 0.000}$	$0.341_{\pm 0.010}$
TVAE	$0.339_{\pm 0.008}$	$0.332_{\pm 0.000}$	$0.195_{\pm 0.015}$
CTAB-GAN	$0.423_{\pm 0.025}$	$0.332_{\pm 0.000}$	$0.012_{\pm 0.023}$
DistVAE($\beta = 0.5$)	$0.364_{\pm 0.004}$	$0.332_{\pm 0.000}$	$0.368_{\pm 0.005}$
DistVAE($\beta = 1$)	$0.364_{\pm 0.004}$	$0.332_{\pm 0.000}$	$0.367_{\pm 0.004}$
DistVAE($\beta = 5$)	$0.368_{\pm 0.003}$	$0.332_{\pm 0.000}$	$0.365_{\pm 0.005}$

(f)

Dataset	kings		
Model	R&S	R	S
CTGAN	$0.550_{\pm 0.016}$	$0.199_{\pm 0.000}$	$0.447_{\pm 0.030}$
TVAE	$0.603_{\pm 0.091}$	$0.199_{\pm 0.000}$	$0.414_{\pm 0.053}$
CTAB-GAN	$0.596_{\pm 0.030}$	$0.199_{\pm 0.000}$	$0.122_{\pm 0.135}$
DistVAE($\beta = 0.5$)	$0.540_{\pm 0.009}$	$0.199_{\pm 0.000}$	$0.600_{\pm 0.013}$
DistVAE($\beta = 1$)	$0.552_{\pm 0.008}$	$0.199_{\pm 0.000}$	$0.605_{\pm 0.014}$
DistVAE($\beta = 5$)	$0.692_{\pm 0.013}$	$0.199_{\pm 0.000}$	$0.763_{\pm 0.015}$

Table 16: Privacy preservability: Membership inference attack performance. Mean and standard deviation values are obtained from 10 repeated experiments.

(a)

Dataset	covertype	
Model	Accuracy	AUC
TVAE	$0.499_{\pm 0.007}$	$0.499_{\pm 0.007}$
DistVAE($\beta = 0.5$)	$0.500_{\pm 0.003}$	$0.500_{\pm 0.003}$

(b)

Dataset	credit	
Model	Accuracy	AUC
TVAE	$0.500_{\pm 0.001}$	$0.500_{\pm 0.001}$
DistVAE($\beta = 0.5$)	$0.500_{\pm 0.001}$	$0.500_{\pm 0.001}$

(c)

Dataset	loan	
Model	Accuracy	AUC
TVAE	$0.497_{\pm 0.015}$	$0.497_{\pm 0.015}$
DistVAE($\beta = 0.5$)	$0.502_{\pm 0.006}$	$0.502_{\pm 0.006}$

(d)

Dataset	adult	
Model	Accuracy	AUC
TVAE	$0.493_{\pm 0.017}$	$0.493_{\pm 0.017}$
DistVAE($\beta = 0.5$)	$0.500_{\pm 0.000}$	$0.500_{\pm 0.000}$

(e)

Dataset	cabs	
Model	Accuracy	AUC
TVAE	$0.480_{\pm 0.033}$	$0.480_{\pm 0.033}$
DistVAE($\beta = 0.5$)	$0.498_{\pm 0.003}$	$0.498_{\pm 0.003}$

(f)

Dataset	kings	
Model	Accuracy	AUC
TVAE	$0.507_{\pm 0.025}$	$0.507_{\pm 0.025}$
DistVAE($\beta = 0.5$)	$0.502_{\pm 0.004}$	$0.502_{\pm 0.004}$

Table 17: Privacy preservability: Attribute disclosure performance with F_1 score. Mean and standard deviation values are obtained from 10 repeated experiments. Lower is better.

(a) `covtype`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.161_{\pm 0.030}$	$0.175_{\pm 0.029}$	$0.155_{\pm 0.041}$
TVAE	$0.356_{\pm 0.086}$	$0.357_{\pm 0.091}$	$0.349_{\pm 0.094}$
CTAB-GAN	$0.200_{\pm 0.035}$	$0.225_{\pm 0.042}$	$0.238_{\pm 0.033}$
DistVAE($\beta = 0.5$)	$0.308_{\pm 0.025}$	$0.338_{\pm 0.037}$	$0.313_{\pm 0.035}$
DistVAE($\beta = 1$)	$0.264_{\pm 0.024}$	$0.294_{\pm 0.036}$	$0.282_{\pm 0.029}$
DistVAE($\beta = 5$)	$0.141_{\pm 0.019}$	$0.135_{\pm 0.013}$	$0.115_{\pm 0.011}$

(b) `credit`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.319_{\pm 0.009}$	$0.330_{\pm 0.009}$	$0.323_{\pm 0.011}$
TVAE	$0.615_{\pm 0.147}$	$0.618_{\pm 0.143}$	$0.614_{\pm 0.145}$
CTAB-GAN	$0.322_{\pm 0.007}$	$0.332_{\pm 0.012}$	$0.331_{\pm 0.010}$
DistVAE($\beta = 0.5$)	$0.339_{\pm 0.013}$	$0.337_{\pm 0.011}$	$0.315_{\pm 0.012}$
DistVAE($\beta = 1$)	$0.339_{\pm 0.010}$	$0.324_{\pm 0.005}$	$0.301_{\pm 0.004}$
DistVAE($\beta = 5$)	$0.332_{\pm 0.007}$	$0.317_{\pm 0.007}$	$0.293_{\pm 0.007}$

(c) `loan`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.439_{\pm 0.027}$	$0.447_{\pm 0.022}$	$0.426_{\pm 0.033}$
TVAE	$0.611_{\pm 0.153}$	$0.602_{\pm 0.152}$	$0.596_{\pm 0.157}$
CTAB-GAN	$0.475_{\pm 0.048}$	$0.443_{\pm 0.027}$	$0.435_{\pm 0.030}$
DistVAE($\beta = 0.5$)	$0.505_{\pm 0.043}$	$0.465_{\pm 0.038}$	$0.439_{\pm 0.032}$
DistVAE($\beta = 1$)	$0.449_{\pm 0.038}$	$0.440_{\pm 0.035}$	$0.423_{\pm 0.031}$
DistVAE($\beta = 5$)	$0.458_{\pm 0.013}$	$0.441_{\pm 0.035}$	$0.416_{\pm 0.031}$

(d) `adult`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.234_{\pm 0.012}$	$0.255_{\pm 0.021}$	$0.261_{\pm 0.023}$
TVAE	$0.318_{\pm 0.081}$	$0.318_{\pm 0.080}$	$0.307_{\pm 0.080}$
CTAB-GAN	$0.199_{\pm 0.032}$	$0.202_{\pm 0.032}$	$0.205_{\pm 0.031}$
DistVAE($\beta = 0.5$)	$0.270_{\pm 0.018}$	$0.280_{\pm 0.021}$	$0.268_{\pm 0.015}$
DistVAE($\beta = 1$)	$0.264_{\pm 0.006}$	$0.276_{\pm 0.005}$	$0.260_{\pm 0.007}$
DistVAE($\beta = 5$)	$0.205_{\pm 0.005}$	$0.187_{\pm 0.003}$	$0.166_{\pm 0.002}$

(e) `cabs`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.238_{\pm 0.009}$	$0.246_{\pm 0.006}$	$0.231_{\pm 0.010}$
TVAE	$0.385_{\pm 0.079}$	$0.383_{\pm 0.079}$	$0.381_{\pm 0.081}$
CTAB-GAN	$0.235_{\pm 0.010}$	$0.236_{\pm 0.008}$	$0.237_{\pm 0.011}$
DistVAE($\beta = 0.5$)	$0.251_{\pm 0.010}$	$0.241_{\pm 0.007}$	$0.225_{\pm 0.005}$
DistVAE($\beta = 1$)	$0.246_{\pm 0.010}$	$0.240_{\pm 0.006}$	$0.227_{\pm 0.008}$
DistVAE($\beta = 5$)	$0.238_{\pm 0.010}$	$0.220_{\pm 0.009}$	$0.199_{\pm 0.005}$

(f) `kings`

Model	Number of neighbors (k)		
	1	10	100
CTGAN	$0.200_{\pm 0.009}$	$0.257_{\pm 0.032}$	$0.269_{\pm 0.036}$
TVAE	$0.338_{\pm 0.045}$	$0.349_{\pm 0.046}$	$0.345_{\pm 0.044}$
CTAB-GAN	$0.111_{\pm 0.068}$	$0.111_{\pm 0.088}$	$0.123_{\pm 0.114}$
DistVAE($\beta = 0.5$)	$0.293_{\pm 0.019}$	$0.310_{\pm 0.041}$	$0.301_{\pm 0.050}$
DistVAE($\beta = 1$)	$0.281_{\pm 0.016}$	$0.306_{\pm 0.039}$	$0.290_{\pm 0.041}$
DistVAE($\beta = 5$)	$0.215_{\pm 0.010}$	$0.221_{\pm 0.036}$	$0.205_{\pm 0.038}$