Masked Language Modeling Becomes Conditional Density Estimation for Tabular Data Synthesis

Seunghwan An[1], Gyeongdong Woo[1], Jaesung Lim[1], ChangHyun Kim[1], Sungchul Hong[3], Jong-June Jeon[2*]

[1]Department of Statistical Data Science, University of Seoul, S. Korea
[2]Department of Statistics, University of Seoul, S. Korea
[3]Department of Statistics, Changwon National University, S. Korea
{dkstmdghks79, dngudxor23, wotjd1410, hahaha503}@uos.ac.kr, shong@changwon.ac.kr, jj.jeon@uos.ac.kr

Abstract

In this paper, our goal is to generate synthetic data for heterogeneous (mixed-type) tabular datasets with high machine learning utility (MLu). Since the MLu performance depends on accurately approximating the conditional distributions, we focus on devising a synthetic data generation method based on conditional distribution estimation. We introduce MaCoDE by redefining the consecutive multi-class classification task of Masked Language Modeling (MLM) as histogram-based non-parametric conditional density estimation. Our approach enables the estimation of conditional densities across arbitrary combinations of target and conditional variables. We bridge the theoretical gap between distributional learning and MLM by demonstrating that minimizing the orderless multi-class classification loss leads to minimizing the total variation distance between conditional distributions. To validate our proposed model, we evaluate its performance in synthetic data generation across 10 real-world datasets, demonstrating its ability to adjust data privacy levels easily without re-training. Additionally, since masked input tokens in MLM are analogous to missing data, we further assess its effectiveness in handling training datasets with missing values, including multiple imputations of the missing entries.

Code — https://github.com/an-seunghwan/MaCoDE
Appendix — https://github.com/an-seunghwan/MaCoDE/blob/main/appendix.pdf

1 Introduction

There are two main objectives in synthetic tabular data generation: (1) preserving the statistical characteristics of the original dataset and (2) achieving comparable machine learning utility (MLu) to the original dataset. In this paper, our focus is on generating synthetic data with high MLu performance. Note that achieving high statistical fidelity does not guarantee high MLu performance (Hansen et al. 2023).

Given that MLu performance depends on accurately approximating conditional distributions, we focus on developing a synthetic data generation method based on conditional distribution estimation. However, two important properties of tabular data must be considered: (i) tabular data can consist of mixed types of data (Borisov et al. 2021; Shwartz-Ziv and Armon 2022), and (ii) the tabular data does not have an intrinsic ordering among columns (Gulati and Roysdon 2023).

(i) Considering the heterogeneous nature of tabular data and aiming to develop a method that addresses the challenges of modeling diverse distributions of continuous columns, we employ histogram-based non-parametric conditional density estimation through a multi-class classification task (Li, Bondell, and Reich 2019). This approach enables us to apply the classification loss uniformly across all types of columns. Since the histogram-based approach is theoretically valid only when continuous variables have bounded supports (Wasserman 2006; Li, Bondell, and Reich 2019), we transform continuous columns using the Cumulative Distribution Function (CDF) and constrain their values to the interval $[0, 1]$ (Li et al. 2021; Fang et al. 2022).

(ii) To learn the arbitrary generation ordering of columns, we utilize the Masked Language Modeling (MLM) approach (Devlin et al. 2019). By employing a masking scheme and the BERT model architecture, our proposed model enables the estimation of conditional densities across arbitrary combinations of target and conditional variables (Ghazvininejad et al. 2019; Ivanov, Figurnov, and Vetrov 2019; Nazábal et al. 2020). (Gulati and Roysdon 2023) proposed a similar method called TabMT, however, TabMT faces challenges in distributional learning because it relies on predicting the K-means cluster index of masked entries. Our approach contrasts with existing auto-regressive density estimators, which generate data in a fixed column order (Hansen 1994; Kamthe, Assefa, and Deisenroth 2021; Letizia and Tonello 2022). Additionally, (Germain et al. 2015; Papamakarios, Pavlakou, and Murray 2017) are also able to estimate conditional densities but differ from our approach by masking the model weights rather than the input.

Therefore, our proposed method redefines the consecutive multi-class classification task of MLM as histogram-based non-parametric conditional density estimation. We term our proposed model MaCoDE (Masked Conditional Density Estimation). The main contribution of our work is bridging the theoretical gap between distributional learning and the consecutive minimization of the multi-class classification loss within the MLM approach.

Specifically, we demonstrate that minimizing the *orderless* multi-class classification loss, when combined with the

*Corresponding author.

CDF transformation, provides theoretical validity for minimizing the discrepancy between conditional distributions in terms of total variation distance. This implies that we do not need to consider the ordering of bins, which could otherwise serve as a useful inductive bias. Note that, in the natural language domain, previous attempts to interpret MLM as distributional learning have been somewhat limited, relying on pseudo-likelihood or Markov random fields (Ghazvininejad et al. 2019; Wang and Cho 2019; Salazar et al. 2019; Ng, Cho, and Ghassemi 2020; Hennigen and Kim 2023).

We substantiate the effectiveness of our proposed method by evaluating its performance in synthetic data generation across 10 real-world tabular datasets, demonstrating its capability to adjust data privacy levels easily without re-training. Given that masked input tokens in MLM can be viewed as missing data, we also assess our model's effectiveness in handling training datasets with missing values. Moreover, as our proposed model estimates the conditional distribution while accommodating arbitrary conditioning sets, it can address various missingness patterns - an essential capability for generating samples and performing multiple imputations (Van Buuren 2018; Ivanov, Figurnov, and Vetrov 2019; Nazábal et al. 2020). Consequently, we further validate our method's effectiveness by evaluating its performance in multiple imputations across various missing data mechanisms.

2 Related Works

Existing methods using deep generative models aim to directly minimize the discrepancy between the multivariate ground-truth distribution and the generative model. These include CTGAN (Xu et al. 2019), TVAE (Xu et al. 2019), CTAB-GAN (Zhao et al. 2021), CTAB-GAN+ (Zhao et al. 2023), DistVAE (An and Jeon 2023), and TabDDPM (Kotelnikov et al. 2023).

In the realm of transformer-based synthesizers, methods such as TabPFGen (Ma et al. 2023), TabMT (Gulati and Roysdon 2023), and REaLTabFormer (Solatorio and Dupriez 2023) have been proposed. TabMT utilizes an MLM-based approach to generate synthetic data by predicting cluster indices from K-means clustering. TabPFGen is an energy-based model that leverages the Bayesian inference of TabPFN (Müller et al. 2022) framework. REaLTabFormer employs an autoregressive Transformer architecture akin to GPT-2 (Radford et al. 2019), applying natural language generation techniques to tabular data. Additionally, methods such as (Kamthe, Assefa, and Deisenroth 2021; Letizia and Tonello 2022; Drouin, Marcotte, and Chapados 2022; Ashok et al. 2024) use transformers for copula density estimation, specifically in time-series datasets.

3 Proposal

Notations. Let $\mathbf{x} \in \mathbb{R}^p$ denote an observation consisting of continuous and categorical (discrete) variables, and the jth variable (column) is denoted as $\mathbf{x}_j$. Here, subscript j refers to the jth element. I_C and I_D represent the index sets for continuous and categorical variables, where $I_C \cup I_D = \{1, \cdots, p\}$. The observed dataset is denoted as $\{\mathbf{x}^{(i)}\}_{i=1}^n$. $\mathbf{m} \in \{0,1\}^p$ is a binary vector indicating masked values, with $\mathbf{m}_j = 0$ indicating the jth column is masked (if $\mathbf{m}_j = 1$, then the jth column is not masked). F_j^* indicates the ground-truth CDF of the jth column, and $\hat{F}_j$ is an estimator of F_j^*.

Overview. Without loss of generality, we can consider an arbitrary conditional density function: for $j \in I_C$,

$$p_j^*(\mathbf{x}_j | \mathbf{x}_1, \cdots, \mathbf{x}_{j-1}). \tag{1}$$

Our primary objective is to estimate (1). However, there are two major challenges in estimating (1):

1. Modeling non-uniform distributions of continuous columns, $\mathbf{x}_j$.
2. Handling arbitrary combinations of conditional variables, $\mathbf{x}_1, \cdots, \mathbf{x}_{j-1}$.

Assumption 1. *For all $j \in I_C$, there exists $\hat{F}_j : \mathbb{R} \mapsto [0,1]$ such that $\hat{F}_j$ is invertible and differentiable.*

In this paper, we address these major challenges by unifying a histogram-based conditional density estimation and the MLM approach. And we assume that $\hat{F}_j$ satisfying Assumption 1 is given. For $j \in I_C$,

$$p_j^*(\mathbf{x}_j | \mathbf{x}_{-j}) \quad = \quad c_j^*(\hat{F}_j(\mathbf{x}_j) | \mathbf{x}_{-j}) \cdot \hat{p}_j(\mathbf{x}_j) \tag{2}$$

by the change of variable in terms of $\hat{F}_j$, where c_j^* is the conditional density of $\hat{F}_j(\mathbf{x}_j)$, $\mathbf{x}_{-j} \coloneqq (\mathbf{x}_1, \cdots, \mathbf{x}_{j-1})$, and $\hat{p}_j$ denotes the density of $\hat{F}_j$. In particular, we estimate c_j^* in (2) using a histogram-based approach.

3.1 Classification Target (Discretization)

The discretization is essential since the MLM-based approach hinges on multi-class classification tasks. In this section, we will describe how to transform the columns of a tabular dataset into classification targets. We denote the classification target (i.e., label) of the jth column as $\mathbf{y}_j$.

Continuous column. Firstly, since the histogram-based approach requires continuous columns to have bounded supports, we transform them into random variables within the $[0,1]$ range using their marginal CDFs. Then, we partition the $[0,1]$ interval with $L+1$ cut-points, $b_0, b_1, \cdots, b_L$, where $0 = b_0 < b_1 < b_2 < \cdots < b_{L-1} < b_L = 1$, resulting in L bins. We define the classification target $\mathbf{y}_j$ based on the interval within which $\hat{F}_j(\mathbf{x}_j)$ falls, as follows:

$$\mathbf{y}_j \coloneqq \sum_{s=0}^{L-1} \mathbb{I}\Big(b_s \leq \hat{F}_j(\mathbf{x}_j)\Big),$$

indicating that the classification target is the bin index, where $\mathbb{I}(\cdot)$ represents the indicator function, and $\mathbf{y}_j \in [L] \coloneqq \{1, 2, \cdots, L\}$. Thus, if $b_{l-1} \leq \hat{F}_j(\mathbf{x}_j) < b_l$, then $\mathbf{y}_j = l$.

Our discretization procedure presents an advantage: since the lower and upper bounds are naturally defined as 0 and 1 within our approach, determining the number of bins becomes more intuitive. For example, finer bins near the boundaries of 0 and 1 can be employed to achieve more accurate tail density estimation.

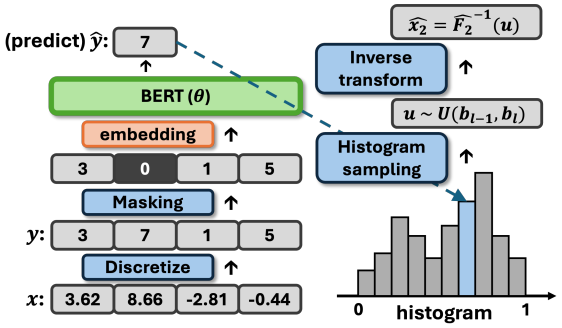

Figure 1: Overall structure of MaCoDE. In this case, the value of the second column is masked (replaced with '0') and predicted.

Categorical column. For categorical variables, without transformation procedures like those for continuous columns, we define $\mathbf{y}_j := \mathbf{x}_j$ for $j \in I_D$. To simplify notation, we denote the number of levels for all categorical variables as L. Allowing categorical variables with different numbers of categories is straightforward.

Definition 1. *The discretization function* $g : \mathbb{R}^p \mapsto [L]^p$ *is defined as*

$$g(\mathbf{x}; \hat{F})_j \quad := \quad \begin{cases} \sum_{s=0}^{L-1} \mathbb{I}(b_s \leq \hat{F}_j(\mathbf{x}_j)), & \text{if } j \in I_C \\ \mathbf{x}_j, & \text{if } j \in I_D \end{cases},$$

for $j = 1, \cdots, p$, *where* $\hat{F} := \{\hat{F}_j : j \in I_C\}$.

Finally, we transform the observations $\mathbf{x}$ into the discretized label $\mathbf{y}$ using the discretization function g of Definition 1, as $\mathbf{y}_j = g(\mathbf{x}; \hat{F})_j$ for all j.

3.2 Masked Conditional Density Estimation

Target distribution. For all $j \in \{1, 2, \cdots, p\}$, the ground-truth conditional probability of $g(\mathbf{x}; \hat{F})_j = l$ given the other observed variables, i.e., $\{\mathbf{x}_k : \mathbf{m}_k = 1, k \neq j\}$, is defined as

$$\begin{aligned} & \Pr\Big(g(\mathbf{x}; \hat{F})_j = l \Big| \{\mathbf{x}_k : \mathbf{m}_k = 1, k \neq j\}\Big) \\ =: \quad & \pi^*_{jl}\Big(\{\mathbf{x}_k : \mathbf{m}_k = 1, k \neq j\}\Big), \end{aligned}$$

which also corresponds to the conditional probability of $\hat{F}_j(\mathbf{x}_j)$ in the lth bin, $[b_{l-1}, b_l)$.

Then, the ground-truth conditional distribution of $g(\mathbf{x}; \hat{F})_j$, which is our target distribution, is written accordingly:

$$\begin{aligned} & p^*_j\Big(g(\mathbf{x}; \hat{F})_j | \{\mathbf{x}_k : \mathbf{m}_k = 1, k \neq j\}\Big) \\ = \quad & \prod_{l=1}^{L} \pi^*_{jl}\Big(\{\mathbf{x}_k : \mathbf{m}_k = 1, k \neq j\}\Big)^{\mathbb{I}(g(\mathbf{x}; \hat{F})_j = l)}. \quad (3) \end{aligned}$$

MaCoDE. For j such that $\mathbf{m}_j = 0$, we parameterize (3) as follows:

$$\begin{aligned} & p_j\Big(g(\mathbf{x}; \hat{F})_j | g(\mathbf{x}; \hat{F}) \odot \mathbf{m}; \theta\Big) \\ = \quad & \prod_{l=1}^{L} \pi_{jl}\Big(g(\mathbf{x}; \hat{F}) \odot \mathbf{m}; \theta\Big)^{\mathbb{I}(g(\mathbf{x}; \hat{F})_j = l)}, \end{aligned}$$

where $\odot$ denotes element-wise multiplication and $\sum_{l=1}^{L} \pi_{jl}(g(\mathbf{x}; \hat{F}) \odot \mathbf{m}; \theta) = 1$ for all j. In this paper, we use the empirical CDF for $\hat{F}_j$. Here, θ represents all the parameters of the transformer encoder-based classifier, allowing us to process inputs of arbitrary lengths and accommodate different combinations of observed and masked variables. Additionally, we denote $(\pi_{j1}(\cdot; \theta), \cdots, \pi_{jL}(\cdot; \theta))$ as $\pi_j(\cdot; \theta)$.

Then, the objective function for a single observation is defined as the negative log-likelihood of masked entries:

$$\begin{aligned} & \mathcal{L}(\mathbf{y}, \mathbf{m}; \theta) \quad (4) \\ := \quad & -\sum_{j:\mathbf{m}_j=0} \log p_j(\mathbf{y}_j | \mathbf{y} \odot \mathbf{m}; \theta) \\ = \quad & -\sum_{j:\mathbf{m}_j=0} \sum_{l=1}^{L} \mathbb{I}(\mathbf{y}_j = l) \cdot \log \pi_{jl}(\mathbf{y} \odot \mathbf{m}; \theta), \end{aligned}$$

where the label $\mathbf{y}_j$ is defined as $\mathbf{y}_j = g(\mathbf{x}; \hat{F})_j$ for all j. Our objective function (4) estimates the conditional distribution of $\hat{F}_j(\mathbf{x}_j)$ in each bin, where its target distribution is (3). In other words, our objective is to approximate the true conditional probability π^*_{jl} by our estimated probability π_{jl}.

Similar to MLM, the bin index $(\mathbf{y} \odot \mathbf{m})_j$ serves as a 'word' index aligned with the jth column, encompassing its own vocabulary set of $L + 1$ words (i.e., $\{0, 1, 2, \cdots, L\}$), including '0' for the masked input. And π_{jl} represents the probability that the model outputs the word index l from the jth column.

For each j, the embedding layer preprocesses of $(\mathbf{y} \odot \mathbf{m})_j$ through `one-hot` encoding, with each bin index being assigned a learnable embedding vector. Note that our approach to handling masked entries shares similarities with the existing *zero imputation* technique, where all masked entries are replaced with zeros (Ivanov, Figurnov, and Vetrov 2019; Mattei and Frellsen 2019; Nazábal et al. 2020; Ipsen, Mattei, and Frellsen 2021). However, in our proposed method, although all masked entries are replaced with the same bin index, '0', each imputed zero value is embedded through distinct embedding vectors for each column.

Definition 2 (Mask distribution)**.** *The distribution of mask vector* $\mathbf{m}$ *is defined as:*

$$p(\mathbf{m}) := \int_0^1 p(\mathbf{m}|u)p(u)du,$$

where $p(u)$ *is the uniform distribution density, and* $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$ *follow conditionally independent Bernoulli distributions with probability* u *given* u.

Finally, we minimize the following objective function with respect to θ:

$$\min_{\theta} \quad \sum_{i=1}^{n} \mathbb{E}_{p(\mathbf{m})}\big[\mathcal{L}(\mathbf{y}^{(i)}, \mathbf{m}; \theta)\big], \quad (5)$$

Algorithm 1: Synthetic data generation

Initialize: For all j, $\hat{\mathbf{y}}_j \leftarrow 0$ and $\hat{\mathbf{m}}_j \leftarrow 0$.
Output: A synthetic sample $\hat{\mathbf{x}}$.

1: **for** $j = randperm\{1, 2, \cdots, p\}$ **do**
2: $\quad \hat{\mathbf{y}}_j \sim Cat\big(\pi_j(\hat{\mathbf{y}} \odot \hat{\mathbf{m}}; \theta)/\tau\big)$
3: $\quad \hat{\mathbf{m}}_j \leftarrow 1$
4: **for** $j = 1, 2, \cdots, p$ **do**
5: $\quad$ **if** $j \in I_C$ **then**
6: $\quad\quad u \sim U(b_{\hat{\mathbf{y}}_j - 1}, b_{\hat{\mathbf{y}}_j})$
7: $\quad\quad \hat{\mathbf{x}}_j \leftarrow \hat{F}_j^{-1}(u)$
8: $\quad$ **if** $j \in I_D$ **then**
9: $\quad\quad \hat{\mathbf{x}}_j \leftarrow \hat{\mathbf{y}}_j$

where the discretized training dataset is $\{\mathbf{y}^{(i)}\}_{i=1}^n = \{g(\mathbf{x}^{(i)}; \hat{F})\}_{i=1}^n$, and $\mathbb{E}_{p(\mathbf{m})}$ is approximated by Monte-Carlo and ancestral sampling. The distribution $p(\mathbf{m})$ can be defined by the user based on the specific problem requirements (Ivanov, Figurnov, and Vetrov 2019). As in BERT (Devlin et al. 2019), the task of our objective function (5) is to predict the original label for all masked inputs. The overall structure of MaCoDE is outlined in Figure 1.

Remark 1. *We want to emphasize that our objective function* (5) *does not imply an assumption of conditional independence among masked entries given the observed data. Instead, by ensuring that $p(\mathbf{m})$ has full support over $\{0, 1\}^p$, sampling $\mathbf{m}$ from $p(\mathbf{m})$ allows us to learn conditional densities encompassing all possible combinations of conditioning sets and target variables (Ivanov, Figurnov, and Vetrov 2019; Gulati and Roysdon 2023). Furthermore, this training scheme remains invariant to the missing data scenario.*

Synthetic data generation. Tabular data lacks the inherent ordering between columns, unlike natural language. Therefore, as outlined in Algorithm 1, MaCoDE randomly generates one column at a time, conditioned on masked subset sizes from p to 1, in descending order ($p \rightarrow p-1 \rightarrow \cdots \rightarrow 2 \rightarrow 1$).

Controllable privacy level. Adjusting the privacy level during synthetic data generation is crucial in tabular domain (Park et al. 2018). Similar to TabMT (Gulati and Roysdon 2023), we can also regulate the privacy level using a single hyper-parameter τ without the need for re-training, while other existing synthesizers have fixed trade-offs between synthetic data quality and privacy level (Xu et al. 2019; Zhao et al. 2023; Kotelnikov et al. 2023) or require re-training (An and Jeon 2023).

By increasing the temperature parameter τ in $\hat{\mathbf{y}}_j \sim Cat\big(\pi_j(\hat{\mathbf{y}} \odot \hat{\mathbf{m}}; \theta)/\tau\big)$ at line 2 of Algorithm 1, we can mitigate the risk of privacy leakage. Figure 2 shows that MaCoDE allows for a trade-off between synthetic data quality (feature selection performance) and privacy preservability (DCR, Distance to Closest Record) as τ increases (see the Appendix for detailed results).

3.3 Theoretical Results

In this section, we aim to provide theoretical insights into MaCoDE's capabilities in conditional density estimation.

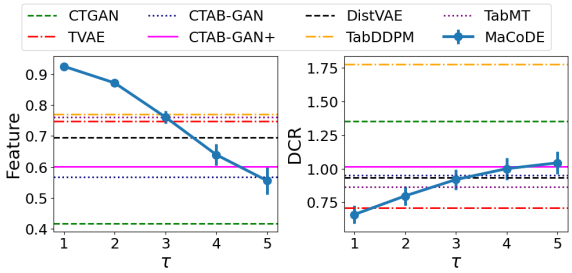

Figure 2: **Trade-off between quality and privacy.** Left: feature selection performance. Right: DCR. Error bars represent standard errors. The horizontal lines represent the mean values of metric scores for baseline models, which are independent of the temperature parameter.

Firstly, we consider a factorization of the ground-truth joint PDF $p^*(\mathbf{x}_1, \cdots, \mathbf{x}_p)$ according to an arbitrary permutation $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_p]$ of the indices $\{1, 2, \cdots, p\}$:

$$p^*_{\sigma_1}(\mathbf{x}_{\sigma_1}) \cdot p^*_{\sigma_2}(\mathbf{x}_{\sigma_2}|\mathbf{x}_{\sigma_1}) \cdots p^*_{\sigma_p}(\mathbf{x}_{\sigma_p}|\mathbf{x}_{\sigma_1}, \cdots, \mathbf{x}_{\sigma_{p-1}})$$

. As discussed in Remark 1, our objective function (5) facilitates the estimation of $p^*_{\sigma_j}(\mathbf{x}_{\sigma_j}|\mathbf{x}_{\sigma_1}, \cdots, \mathbf{x}_{\sigma_{j-1}})$ for any σ and j by utilizing the mask distribution of Definition 2.

Without loss of generality, in this section, let σ be an identity permutation. Since c^*_j is the conditional density of $\hat{F}_j(\mathbf{x}_j)$, the conditional probability of $\hat{F}_j(\mathbf{x}_j)$ in the lth bin, $\pi^*_{jl}(\mathbf{x}_{-j})$, is defined as

$$\pi^*_{jl}(\mathbf{x}_{-j}) = \int_{b_{l-1}}^{b_l} c^*_j(v|\mathbf{x}_{-j})dv,$$

where $\mathbf{x}_{-j} \coloneqq (\mathbf{x}_1, \cdots, \mathbf{x}_{j-1})$.

Assumption 2. *For all j, there exists $K_j \geq 0$ such that c^*_j is K_j-Lipschitz.*

Assumption 2 implies that c^*_j is constrained in its rate of change, which allows us to estimate the conditional density using a piece-wise constant function (Wasserman 2006; Tsybakov and Tsybakov 2009).

Based on (2), by the change of variable under Assumption 1, we define our conditional density estimator for p^*_j as follows:

$$\hat{p}_j(\mathbf{x}_j|\mathbf{x}_{-j}; \theta) = \hat{c}_j(\hat{F}_j(\mathbf{x}_j)|\mathbf{x}_{-j}; \theta) \cdot \hat{p}(\mathbf{x}_j)$$
$$= \left(\sum_{l=1}^{L} \frac{\mathbb{I}(\hat{F}_j(\mathbf{x}_j) \in [b_{l-1}, b_l))}{1/L} \cdot \pi_{jl}\right) \cdot \hat{p}(\mathbf{x}_j), \quad (6)$$

where π_{jl} denotes $\pi_{jl}(g(\mathbf{x}; \hat{F}) \odot \mathbf{m}^{(j)}; \theta)$ and $\mathbf{m}^{(j)}$ is a masking vector such that

$$\mathbf{m}^{(j)}_k \coloneqq \begin{cases} 1, & \text{if } k \in \{1, \cdots, j-1\} \\ 0, & \text{otherwise} \end{cases}.$$

Note that $\hat{c}_j$ is the histogram-based conditional density estimator. Synthetic samples can be generated from (6) as follows: Firstly, perform histogram sampling from $\hat{c}_j$ by choosing a histogram bin according to π_{jl} and uniformly sampling

from that bin interval. Then, apply inverse transform sampling of $\hat{p}_j$ with respect to the output of histogram sampling. This procedure is outlined in Algorithm 1.

Proposition 1. *Under Assumption 1 and 2,*

$$TV\Big(p_j^*(\cdot|\mathbf{x}_{-j}), \hat{p}_j(\cdot|\mathbf{x}_{-j};\theta)\Big) \quad \leq \quad \frac{K_j}{2L} + \frac{\sqrt{Bias(\theta)}}{\sqrt{2}/L}$$

for all $j \in I_C$, and $\mathbf{x} \in \mathbb{R}^p$. Here, TV$(\cdot,\cdot)$ denotes the total variation distance, and $Bias(\theta)$ is defined as:

$$Bias(\theta) = \sum_{l=1}^{L} \pi_{jl}^*(\mathbf{x}_{-j}) \log \pi_{jl}^*(\mathbf{x}_{-j})$$
$$- \quad \mathbb{E}_{\mathbf{y}_j|\mathbf{x}_{-j}} \left[\sum_{l=1}^{L} \mathbb{I}(\mathbf{y}_j = l) \log \pi_{jl}(g(\mathbf{x};\hat{F}) \odot \mathbf{m}^{(j)};\theta) \right]$$

where $\mathbf{y}_j|\mathbf{x}_{-j}$ is a random variable having a categorical distribution such that $\Pr(\mathbf{y}_j = l|\mathbf{x}_{-j}) = \Pr(g(\mathbf{x};\hat{F})_j = l|\mathbf{x}_{-j}) = \pi_{jl}^*(\mathbf{x}_{-j})$ for all $l \in [L]$.

In the definition of $Bias(\theta)$, the second term on the right-hand side corresponds to the classification loss with respect to the target distribution given by (3). This implies that $Bias(\theta)$ can be minimized when the classification loss is minimized. And Proposition 1 demonstrates that the total variation distance between the ground-truth conditional density and our conditional density estimator $\hat{p}_j$ is upper bounded by $Bias(\theta)$.

Therefore, minimizing the orderless multi-class classification loss leads to minimizing the total variation distance between conditional distributions, making Algorithm 1 capable of generating theoretically valid synthetic samples under certain assumptions. Note that Proposition 1 holds for any arbitrary permutation σ, and the CDF transformation is crucial for our proposed method, as it ensures the validity of Proposition 1.

3.4 With Missing Data

Suppose the pattern of missingness varies individually for each observation and is defined by a corresponding missing indicator, denoted as $\mathbf{r}$, where the indicators for all observations are represented as $\{\mathbf{r}^{(i)}\}_{i=1}^n$. Here, $\mathbf{r}_j = 0$ indicates that $\mathbf{x}_j$ is missing, while $\mathbf{r}_j = 1$ denotes that $\mathbf{x}_j$ is observed. In cases where the training data contains missing entries (e.g., not a number), it is not feasible to input these missing entries and minimize their log-likelihoods. Therefore, to handle missing value inputs using g, irrespective of the value of $\mathbf{x}_j$, we further define g for Definition 1 as follows: for any F, $g(\mathbf{x};F)_j \coloneqq 0$ if $\mathbf{r}_j = 0$. This indicates the bin index '0' is assigned to a missing value input.

Therefore, the objective function with missing data is $\min_\theta \sum_{i=1}^n \mathbb{E}_{p(\mathbf{m})}\big[\mathcal{L}^*(\mathbf{x}^{(i)}, \mathbf{r}^{(i)}, \mathbf{m};\theta)\big]$, where

$$\mathcal{L}^*(\mathbf{x},\mathbf{r},\mathbf{m};\theta) \quad \coloneqq \quad - \sum_{j:\mathbf{m}_j=0,\mathbf{r}_j=1} \sum_{l=1}^{L} \mathbb{I}(g(\mathbf{x};\hat{F})_j = l)$$
$$\times \log \pi_{jl}(g(\mathbf{x};\hat{F}) \odot \min(\mathbf{m},\mathbf{r});\theta),$$

$\min(\mathbf{m},\mathbf{r})$ is element-wise minimum operation, and $\hat{F}$ is estimated using the observed dataset (Chenouri, Mojirsheibani, and Montazeri 2009). Note that, in $\mathcal{L}^*(\mathbf{x},\mathbf{r},\mathbf{m};\theta)$, we do not minimize the negative log-likelihood of missing entries.

Proposition 2. *Assuming $\mathbf{m} \perp\!\!\!\perp \mathbf{r}|\mathbf{x}$ and $\mathbf{m} \perp\!\!\!\perp \mathbf{x}$, if the data of $\mathbf{x}$ is MAR, then the following holds for the missing data model $p(\mathbf{m},\mathbf{r}|\mathbf{x})$:*

$$p(\mathbf{m},\mathbf{r}|\mathbf{x}) = p(\mathbf{m},\mathbf{r}|\mathbf{x}_{obs}),$$

where $\mathbf{x}_{obs}$ represents the observed covariates from $\mathbf{x}$, and the missingness pattern of $\min(\mathbf{m},\mathbf{r})$ also follows the MAR mechanism.

In $\mathcal{L}^*(\mathbf{x},\mathbf{r},\mathbf{m};\theta)$, the missingness pattern is described by $\min(\mathbf{m},\mathbf{r})$, indicating that the missing data model is determined by the joint distribution $p(\mathbf{m},\mathbf{r}|\mathbf{x})$. Thus, as the masking vector defined in Definition 2 satisfies the conditions in Proposition 2, the missingness pattern of our proposed model also conforms to the MAR mechanism according to Proposition 2 if the given data is MAR. It implies that our inference strategies based on the observed dataset can be justified (Mattei and Frellsen 2019). However, we empirically demonstrate in Section 4 that our proposed model is also applicable to other missing data scenarios.

4 Experiments

4.1 Overview

We conduct experiments in which we can provide answers to the following three experimental questions:

Q1. Does MaCoDE achieve state-of-the-art performance in synthetic data generation?

Q2. Can MaCoDE generate high-quality synthetic data even when faced with missing data scenarios?

Q3. Is MaCoDE capable of supporting multiple imputations for deriving statistically valid inferences from missing data?

Datasets. Similar to several recent studies (Gulati and Roysdon 2023; Kotelnikov et al. 2023), we utilize 10 publicly available real tabular UCI and Kaggle[1] datasets of varying sizes and the number of columns. Detailed statistics of these datasets are provided in the Appendix. Note that we include `covtype` dataset, which comprises approximately 580K rows, to demonstrate the scalability of our proposed model.

Baseline models. For MaCoDE, we set $L = 50$ and $\tau = 1$ for all datasets. Detailed hyperparameter settings are provided in the Appendix. [2] For Q1 and Q2, we compare MaCoDE with CTGAN (Xu et al. 2019), TVAE (Xu et al. 2019), CTAB-GAN (Zhao et al. 2021), CTAB-GAN+ (Zhao et al. 2023), DistVAE (An and Jeon 2023), TabDDPM (Kotelnikov et al. 2023), and TabMT (Gulati and Roysdon 2023). For Q3, we selected the following multiple imputation models that can handle mixed-type tabular datasets: MICE (van Buuren and Groothuis-Oudshoorn

[1] https://archive.ics.uci.edu/, https://www.kaggle.com/datasets/

[2] We run experiments using NVIDIA A10 GPU.

	Statistical fidelity				Machine learning utility			
Model	KL ↓	GoF ↓	MMD ↓	WD ↓	SMAPE ↓	F_1 ↑	Model ↑	Feature ↑
Baseline	$.016_{\pm.002}$	$.029_{\pm.002}$	$.002_{\pm.000}$	$1.019_{\pm.156}$	$.107_{\pm.008}$	$.686_{\pm.023}$	$.887_{\pm.018}$	$.956_{\pm.005}$
CTGAN	$.221_{\pm.014}$	$.561_{\pm.046}$	$.094_{\pm.007}$	$6.435_{\pm1.011}$	$.256_{\pm.016}$	$.411_{\pm.027}$	$.208_{\pm.048}$	$.417_{\pm.043}$
TVAE	$.066_{\pm.003}$	$.119_{\pm.005}$	$.016_{\pm.001}$	$\underline{1.631}_{\pm.173}$	$.192_{\pm.011}$	$.608_{\pm.021}$	$.486_{\pm.041}$	$.747_{\pm.027}$
CTAB-GAN	$.116_{\pm.008}$	$.196_{\pm.025}$	$.044_{\pm.004}$	$3.327_{\pm.460}$	$.218_{\pm.012}$	$.524_{\pm.026}$	$.263_{\pm.042}$	$.568_{\pm.041}$
CTAB-GAN+	$.136_{\pm.018}$	$.144_{\pm.010}$	$.054_{\pm.007}$	$3.971_{\pm.772}$	$.226_{\pm.017}$	$.530_{\pm.020}$	$.227_{\pm.048}$	$.601_{\pm.041}$
DistVAE	$.059_{\pm.007}$	$\underline{.070}_{\pm.004}$	$.016_{\pm.001}$	$2.272_{\pm.282}$	$.226_{\pm.017}$	$.588_{\pm.021}$	$.194_{\pm.048}$	$.695_{\pm.030}$
TabDDPM	$.696_{\pm.117}$	$.374_{\pm.087}$	$.057_{\pm.011}$	$42.916_{\pm8.127}$	$\underline{.161}_{\pm.011}$	$.576_{\pm.022}$	$.507_{\pm.039}$	$\underline{.770}_{\pm.027}$
TabMT	$\mathbf{.011}_{\pm.001}$	$\mathbf{.035}_{\pm.003}$	$\underline{.012}_{\pm.001}$	$2.299_{\pm.346}$	$.188_{\pm.013}$	$\underline{.622}_{\pm.024}$	$\underline{.528}_{\pm.039}$	$.761_{\pm.028}$
MaCoDE	$\underline{.034}_{\pm.004}$	$.072_{\pm.004}$	$\mathbf{.007}_{\pm.001}$	$\mathbf{1.630}_{\pm.245}$	$\mathbf{.158}_{\pm.010}$	$\mathbf{.635}_{\pm.023}$	$\mathbf{.599}_{\pm.035}$	$\mathbf{.925}_{\pm.007}$
MaCoDE(MCAR)	-	-	-	-	$.168_{\pm.011}$	$.623_{\pm.023}$	-	-
MaCoDE(MAR)	-	-	-	-	$.167_{\pm.011}$	$.626_{\pm.023}$	-	-
MaCoDE(MNARL)	-	-	-	-	$.169_{\pm.011}$	$.624_{\pm.023}$	-	-
MaCoDE(MNARQ)	-	-	-	-	$.164_{\pm.010}$	$.630_{\pm.023}$	-	-

Table 1: **Q1** and **Q2**. The means and the standard errors of the mean across 10 datasets and 10 repeated experiments are reported. Across all missingness patterns, a missingness rate of 0.3 is employed. ↑ (↓) denotes higher (lower) is better. The best value is bolded, and the second best is underlined.

2011), GAIN (Yoon, Jordon, and van der Schaar 2018), missMDA (Josse and Husson 2016), VAEAC (Ivanov, Figurnov, and Vetrov 2019), MIWAE (Mattei and Frellsen 2019), not-MIWAE (Ipsen, Mattei, and Frellsen 2021), and EGC (Zhao, Townsend, and Udell 2022). Detailed experimental settings for these baseline models are provided in the Appendix.

To assist in interpreting the metrics, we include a baseline synthetic dataset where the synthetic data comprises half of the real training dataset. This dataset, referred to as 'Baseline,' serves as a soft upper bound for evaluating the quality of the synthetic data.

Additional evaluations. Due to space constraints, (1) the results for controlling data privacy levels with varying temperature parameter τ, and (2) the results for **Q3** (including related works and sensitivity analysis regarding the missingness rate) are provided in the Appendix.

4.2 Evaluation Metrics

For all metrics, we report the mean and standard error of the mean (error bars) across 10 different random seeds and 10 datasets. For each random seed, the dataset is randomly split into training and testing sets with an 80% training and 20% testing ratio in the evaluation of Q1 and Q2. During evaluation, the synthetic dataset is generated to have the same number of samples as the real training dataset.

Q1. To evaluate the quality of generated synthetic data, we employ two metrics: statistical fidelity (Qian, Davis, and van der Schaar 2023) and machine learning utility (Hansen et al. 2023). For statistical fidelity, we utilize the Kullback–Leibler divergence (KL) and the Goodness-of-Fit (GoF) test (continuous: the two-sample Kolmogorov-Smirnov test statistic, categorical: the Chi-Squared test statistic) to assess marginal distributional similarity. Additionally, we employ the Maximum Mean Discrepancy (MMD) and 1-Wasserstein distance (WD) to measure joint distributional similarity. These metrics measure how well the synthetic data maintains statistical fidelity to the real training dataset.

Regarding machine learning utility, we use four metrics outlined in (Hansen et al. 2023): regression performance (SMAPE, symmetric mean absolute percentage error), classification performance (F_1), model selection performance (Model), and feature selection performance (Feature). These metrics are assessed by fitting the machine learning model on the real training and synthetic datasets individually and then comparing the performance of the two models on the test dataset. For a detailed evaluation procedure of Q1, refer to the Appendix.

Q2. To assess our proposed model's capability to generate high-quality synthetic datasets despite missing values in the training data, we evaluate the model trained on incomplete training datasets. Since other metrics require a complete training dataset for measurement, we only assess synthetic data quality using downstream regression and classification tasks on the test dataset, as these metrics can be evaluated using the test dataset alone, as in Q1. See the Appendix for a detailed evaluation procedure.

Q3.[3] We assess the effectiveness of multiple imputations by employing interval inference for the population mean, which was proposed by Rubin (Rubin and Schenker 1986). We report the bias, coverage, and confidence interval length. The detailed evaluation procedure for multiple imputations and the missing value generation mechanisms (MCAR, MAR, MNARL, MNARQ) is provided in the Appendix.

[3]Since some off-the-shelf packages (`missMDA`, `gcimpute`) fail to deliver useful results, we include only 5 datasets that provide meaningful results: `abalone`, `banknote`, `breast`, `redwine`, and `whitewine`.

Model	Bias ↓	Coverage	Width ↓
MICE	$.010_{\pm.001}$	$.845_{\pm.019}$	$\mathbf{.040}_{\pm.002}$
GAIN	$.019_{\pm.002}$	$.633_{\pm.033}$	$\mathbf{.040}_{\pm.002}$
missMDA	$.015_{\pm.001}$	$.700_{\pm.022}$	$.043_{\pm.002}$
VAEAC	$\underline{.008}_{\pm.001}$	$.905_{\pm.016}$	$\mathbf{.040}_{\pm.002}$
MIWAE	$\mathbf{.006}_{\pm.000}$	$\underline{.952}_{\pm.012}$	$.043_{\pm.002}$
not-MIWAE	$\mathbf{.006}_{\pm.000}$	$\mathbf{.949}_{\pm.012}$	$\underline{.042}_{\pm.002}$
EGC	$\mathbf{.006}_{\pm.000}$	$.996_{\pm.004}$	$.058_{\pm.002}$
MaCoDE(MAR)	$\mathbf{.006}_{\pm.000}$	$.963_{\pm.009}$	$.051_{\pm.003}$

Table 2: **Q3** under MAR at 0.3 missingness. The means and standard errors of the mean across 5 datasets and 10 repeated experiments are reported. ↓ denotes lower is better. Coverage close to 0.95 indicates better performance. The best value is bolded, and the second best is underlined.

4.3 Results

Q1. As shown in Table 1, MaCoDE consistently achieves the highest metric scores in both joint distributional similarities and machine learning utility while also achieving competitive performance in marginal distributional similarity. This underscores the effectiveness of the synthetic data generation method based on estimating conditional distribution in preserving the joint statistical fidelity of the original data and enhancing the utility of synthetic data for downstream machine learning tasks. Notably, MaCoDE demonstrates remarkable performance in the feature selection downstream task.

Remark 2 (How does MaCoDE achieve the remarkable performance in feature selection?)**.** *We attribute the effectiveness of the feature selection downstream task to our emphasis on estimating 'conditional' distributions. In Random Forest (Breiman 2001), each node in a decision tree represents a conditional distribution of a variable conditioned on the splits made by the tree up to that node, and the feature importance is determined by the purity of nodes. Therefore, accurately estimating the conditional distribution can lead to higher performance in preserving the feature importance ranking.*

Q2. The missing data mechanism within the parentheses refers to the mechanism applied to the training dataset on which MaCoDE was trained. Despite encountering missing data scenarios such as MAR, MCAR, MNARL, and MNARQ, Table 1 illustrates MaCoDE's ability to generate high-quality synthetic data regarding machine learning utility while handling incomplete training datasets without significant performance degradation. Even in the presence of missing entries, MaCoDE achieves better metric scores than most baseline models in terms of SMAPE, except for TabDDPM. Additionally, concerning the F_1 score, MaCoDE either competes competitively or achieves a higher score than other baseline models.

Q3. The missing data mechanism within the parentheses refers to the mechanism applied to the dataset on which MaCoDE was trained. Table 2 indicates that MaCoDE consistently exhibits competitive performance against all baseline models across metrics assessing multiple imputation performances, including bias, coverage, and confidence interval length. This suggests our proposed approach can support multiple imputations for deriving statistically valid inferences from missing data with the MAR mechanism.

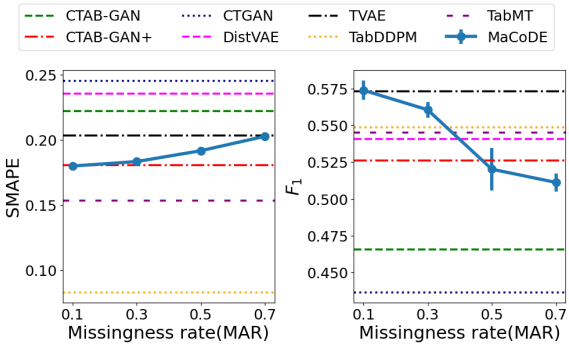

Figure 3: Sensitivity analysis with respect to missingness rate using `kings` dataset is performed for **Q2** under MAR. Results are reported as means and standard errors of the mean from 10 repeated experiments, with error bars representing the standard errors. The horizontal lines represent the mean values of metric scores for baseline models, which are independent of the missingness rate.

Sensitivity analysis. We also conducted a sensitivity analysis by varying the missingness rate of `kings` dataset. Figure 3 illustrates that, in terms of SMAPE, MaCoDE maintains competitive performance even as the missingness rate increases. Concerning the F_1 score, MaCoDE outperforms other models at missingness rates of 0.1 and 0.3, but its performance declines beyond a missingness rate of 0.5.

5 Conclusions and Limitations

This paper introduces an approach to generating synthetic data for mixed-type tabular datasets. Our proposed method integrates histogram-based non-parametric conditional density estimation and the MLM-based approach while bridging the theoretical gap between distributional learning and the consecutive multi-class classification task of MLM. Although our primary goal is to generate synthetic data with high MLu, we empirically demonstrate that we achieve high joint statistical fidelity and MLu simultaneously. Furthermore, empirical experiments validate that our proposed model can generate high-quality synthetic tabular datasets in terms of MLu even when incomplete training datasets are given.

Although MaCoDE demonstrates the ability to perform 'arbitrary' conditional density estimation by accommodating various combinations of conditioning sets and target variables, and despite empirical results showing its effectiveness in handling diverse distributions of continuous columns and generating high-quality synthetic data, the model has theoretical limitations. Specifically, it is valid under Lipschitz continuity (Assumption 2). Addressing the limitation of accommodating a broader range of continuous distributions is an important direction for future work.

Acknowledgements

Seunghwan An was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2024-00412613). Jong-June Jeon was supported by the National Research Foundation of Korea grant (2022R1A4A3033874, 2022M3J6A1084845, and 2022R1F1A1074758). The authors acknowledge the Urban Big data and AI Institute of the University of Seoul supercomputing resources (http://ubai.uos.ac.kr) made available for conducting the research reported in this paper.

References

An, S.; and Jeon, J.-J. 2023. Distributional Learning of Variational AutoEncoder: Application to Synthetic Data Generation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ashok, A.; Marcotte, É.; Zantedeschi, V.; Chapados, N.; and Drouin, A. 2024. TACTiS-2: Better, Faster, Simpler Attentional Copulas for Multivariate Time Series. In *The Twelfth International Conference on Learning Representations*.

Borisov, V.; Leemann, T.; Seßler, K.; Haug, J.; Pawelczyk, M.; and Kasneci, G. 2021. Deep Neural Networks and Tabular Data: A Survey. *IEEE transactions on neural networks and learning systems*, PP.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Chenouri, S.; Mojirsheibani, M.; and Montazeri, Z. 2009. Empirical measures for incomplete data with applications. *Electronic Journal of Statistics*, 3: 1021–1038.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*.

Drouin, A.; Marcotte, E.; and Chapados, N. 2022. TACTiS: Transformer-Attentional Copulas for Time Series. In *International Conference on Machine Learning*.

Fang, K.; Mugunthan, V.; Ramkumar, V.; and Kagal, L. 2022. Overcoming Challenges of Synthetic Data Generation. *2022 IEEE International Conference on Big Data (Big Data)*, 262–270.

Germain, M.; Gregor, K.; Murray, I.; and Larochelle, H. 2015. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, 881–889. PMLR.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Conference on Empirical Methods in Natural Language Processing*.

Gulati, M. S.; and Roysdon, P. F. 2023. TabMT: Generating tabular data with masked transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hansen, B. E. 1994. Autoregressive Conditional Density Estimation. *International Economic Review*, 35: 705–730.

Hansen, L.; Seedat, N.; van der Schaar, M.; and Petrovic, A. 2023. Reimagining Synthetic Tabular Data Generation through Data-Centric AI: A Comprehensive Benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hennigen, L. T.; and Kim, Y. 2023. Deriving Language Models from Masked Language Models. In *Annual Meeting of the Association for Computational Linguistics*.

Ipsen, N. B.; Mattei, P.-A.; and Frellsen, J. 2021. not-{MIWAE}: Deep Generative Modelling with Missing not at Random Data. In *International Conference on Learning Representations*.

Ivanov, O.; Figurnov, M.; and Vetrov, D. 2019. Variational Autoencoder with Arbitrary Conditioning. In *International Conference on Learning Representations*.

Josse, J.; and Husson, F. 2016. missMDA: A Package for Handling Missing Values in Multivariate Data Analysis. *Journal of Statistical Software*, 70(1): 1–31.

Kamthe, S.; Assefa, S. A.; and Deisenroth, M. P. 2021. Copula Flows for Synthetic Data Generation. *ArXiv*, abs/2101.00598.

Kotelnikov, A.; Baranchuk, D.; Rubachev, I.; and Babenko, A. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, 17564–17579. PMLR.

Letizia, N. A.; and Tonello, A. M. 2022. Copula Density Neural Estimation. *ArXiv*, abs/2211.15353.

Li, B.; Luo, S.; Qin, X.; and Pan, L. 2021. Improving GAN with inverse cumulative distribution function for tabular data synthesis. *Neurocomputing*, 456: 373–383.

Li, R.-B.; Bondell, H. D.; and Reich, B. J. 2019. Deep Distribution Regression. *Comput. Stat. Data Anal.*, 159: 107203.

Ma, J.; Dankar, A.; Stein, G.; Yu, G.; and Caterini, A. 2023. TabPFGen–Tabular Data Generation with TabPFN. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Mattei, P.-A.; and Frellsen, J. 2019. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. In *International Conference on Machine Learning*.

Müller, S.; Hollmann, N.; Arango, S. P.; Grabocka, J.; and Hutter, F. 2022. Transformers Can Do Bayesian Inference. In *International Conference on Learning Representations*.

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

Ng, N.; Cho, K.; and Ghassemi, M. 2020. SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1268–1283. Online: Association for Computational Linguistics.

Papamakarios, G.; Pavlakou, T.; and Murray, I. 2017. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30.

Park, N.; Mohammadi, M.; Gorde, K.; Jajodia, S.; Park, H.; and Kim, Y. 2018. Data Synthesis based on Generative Adversarial Networks. *Proc. VLDB Endow.*, 11: 1071–1083.

Qian, Z.; Davis, R.; and van der Schaar, M. 2023. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rubin, D. B.; and Schenker, N. 1986. Multiple Imputation for Interval Estimation from Simple Random Samples with Ignorable Nonresponse. *Journal of the American Statistical Association*, 81: 366–374.

Salazar, J.; Liang, D.; Nguyen, T. Q.; and Kirchhoff, K. 2019. Masked Language Model Scoring. In *Annual Meeting of the Association for Computational Linguistics*.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90.

Solatorio, A. V.; and Dupriez, O. 2023. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. *ArXiv*, abs/2302.02041.

Tsybakov, A. B.; and Tsybakov, A. B. 2009. Nonparametric estimators. *Introduction to Nonparametric Estimation*, 1–76.

Van Buuren, S. 2018. *Flexible imputation of missing data*. CRC press.

van Buuren, S.; and Groothuis-Oudshoorn, K. G. M. 2011. MICE: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45: 1–67.

Wang, A.; and Cho, K. 2019. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. In Bosselut, A.; Celikyilmaz, A.; Ghazvininejad, M.; Iyer, S.; Khandelwal, U.; Rashkin, H.; and Wolf, T., eds., *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 30–36. Minneapolis, Minnesota: Association for Computational Linguistics.

Wasserman, L. 2006. *All of nonparametric statistics*. Springer Science & Business Media.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling Tabular data using Conditional GAN. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2018. GAIN: Missing Data Imputation using Generative Adversarial Nets. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5689–5698. PMLR.

Zhao, Y.; Townsend, A.; and Udell, M. 2022. Probabilistic Missing Value Imputation for Mixed Categorical and Ordered Data. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Zhao, Z.; Kunar, A.; Birke, R.; and Chen, L. Y. 2021. CTAB-GAN: Effective Table Data Synthesizing. In Balasubramanian, V. N.; and Tsang, I., eds., *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, 97–112. PMLR.

Zhao, Z.; Kunar, A.; Birke, R.; and Chen, L. Y. 2023. CTAB-GAN+: Enhancing Tabular Data Synthesis. *Frontiers in Big Data*, abs/2204.00401.