MISSDIFF: TRAINING DIFFUSION MODELS ON TABULAR DATA WITH MISSING VALUES

Anonymous authors
Paper under double-blind review

ABSTRACT

The diffusion model has shown remarkable performance in modeling data distributions and synthesizing data. However, the vanilla diffusion model requires complete or fully observed training data. Incomplete data is a common issue in various real-world applications, including healthcare and finance, particularly when dealing with tabular datasets. This work considers learning a diffusion-based model from data with missing values for missing value imputations and generating synthetic complete data in a unified framework. With minimal assumptions on the missing mechanisms, our method models the score of complete data distribution by denoising score matching on data with missing values. We prove that the proposed method can recover the score of the complete data distribution, and the proposed training objective serves as an upper bound for the negative likelihood of observed data. Extensive experiments on imputation tasks together with generation tasks demonstrate that our proposed framework outperforms existing state-of-the-art approaches on multiple tabular datasets.

1 INTRODUCTION

Diffusion models have emerged as an effective tool for modeling the data distribution and synthesize various types of data, such as images (Ho et al., 2020; Song et al., 2021b; Dhariwal & Nichol, 2021; Rombach et al., 2021), videos (Ho et al., 2022), point clouds (Luo & Hu, 2021), and tabular data (Kim et al., 2023; Kotelnikov et al., 2022). These machine learning models typically rely on high-quality training data, which are usually expected to be free of missing values. In reality, it is often challenging to obtain complete data, particularly in healthcare, finance, recommendation systems, and social networks, due to privacy concerns, high cost or sampling difficulties, and the skewed distribution of user-generated content. For example, the respiratory rate of a patient may not have been measured, either because it was deemed unnecessary or was accidentally not recorded (Yoon et al., 2017; Alaa et al., 2016; Yoon et al., 2018a). Additionally, some information may be difficult or even dangerous to acquire, such as information obtained through a biopsy, which may not have been gathered for those reasons (Yoon et al., 2018b).

Moreover, deep generative models, particularly diffusion models, can be used to augment training data to protect the privacy of original tabular data and enhance the performance of machine learning models on tabular data (Kim et al., 2023; Xu et al., 2019; Kotelnikov et al., 2022; Zhang et al., 2023). Following this idea, we can achieve better performance for downstream tasks by utilizing generative model learning on incomplete data for synthetic data generation. Therefore, in this work, we focus on learning a generative model from training data containing missing values and synthesize *new complete data*, not just imputing the missing value.

Numerous studies have been proposed to deal with missing values in the training data. Some approaches use the variational lower bound on observed data to train a VAE-based model (Ipsen et al., 2021; Nazábal et al., 2018; Ma et al., 2020; Mattei & Frellsen, 2019; Valera et al., 2017). Other methods use adversarial training by optimizing a min-max objective to train a GAN-based model (Yoon et al., 2018a; Li et al., 2019; Li & Marlin, 2020). Most of the works mentioned above mainly focus on imputation tasks. They cannot be directly used for generating new complete samples [1]. One line of work first completes the data and then learns a generative model on imputed data. Some

[1] A detailed discussion can be found in Appendix B.1.

approaches delete instances or features with missing data or replace missing values with the mean of observed values for that feature. Other methods employ machine learning approaches (van Buuren & Groothuis-Oudshoorn, 2011; Bertsimas et al., 2017) or deep generative models for imputation tasks (Yoon et al., 2018a; Biessmann et al., 2019; Wang et al., 2020; Ipsen et al., 2022; Muzellec et al., 2020). It has been shown that imputation may reduce the diversity of the training data and may lead to biased performance in downstream tasks (Bertsimas et al., 2021; Ipsen et al., 2022). Another line of works first learns the generative model directly on the data with missing values by using the existing VAE-based or GAN-based model (Ipsen et al., 2021; Nazábal et al., 2018; Ma et al., 2020; Mattei & Frellsen, 2019; Valera et al., 2017; Yoon et al., 2018a; Li et al., 2019; Li & Marlin, 2020). After that, they first generate new samples containing missing values by removing different values in observed data and then apply the learned generative model to impute the missing data as described in Neves et al. (2022). In summary, these works require two-stage inference for synthesizing new complete samples, which might be biased (proven in Remark 3.1) or computationally expensive (detailed described in Section 3.4).

In this work, we propose a unified diffusion-based framework, which we call *MissDiff*, for both imputation and synthetic complete data generation without two-stage inference or training additional neural networks. *MissDiff* models the score (gradient log density) of complete data distribution by denoising score matching on data with missing values. We present the theoretical justification of *MissDiff* on recovering the oracle score function of the complete data and also upper bounding the negative likelihood of the observed data under mild assumptions.

We primarily utilize *tabular* data for the numerical experiments, as tabular data is a commonly encountered data type and frequently contains missing values in various applications Yoon et al. (2017); Alaa et al. (2016). Moreover, by considering tabular data as an example, we simultaneously study the missing value scenarios in categorical and continuous variables, which are both contained in tabular-type data.

To verify the effectiveness of *MissDiff*, we conduct a suite of numerical experiments under various missing mechanisms. For both imputation tasks and generation tasks, *MissDiff* outperforms existing state-of-the-art methods in most settings by a considerable margin.

Our contributions can be summarized as follows.

- We propose a diffusion-based unified framework, which we call *MissDiff*, for imputation and complete sample generation by learning from data with missing values.
- We provide the theoretical justifications of *MissDiff* on recovering the oracle score function of the complete data and upper bounding the negative likelihood of the observed data under mild assumptions.
- *MissDiff* outperforms existing state-of-the-art methods in most settings on both imputation tasks and generation tasks on multiple real tabular datasets under different missing mechanisms.

The rest of the paper is organized as follows. Section 2 reviews the setup of the missing data mechanism and the score-based generative model. Section 3 introduces the proposed method and theoretically characterizes the effectiveness of the proposed method. Numerical results are given in Section 4. We conclude the paper in Section 5. All proofs and additional numerical experiments are deferred to the appendix.

2 Problem Setup and Preliminaries

2.1 Training with Missing Data

We aim to learn a diffusion-based generative model from training data that may contain a certain proportion of missing values. Following the settings in Little & Rubin (1988); Li et al. (2019); Ipsen et al. (2022), we denote the underlying complete d-dimensional data as $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$ and assume it is sampled from the unknown true data-generating distribution $p_0(\mathbf{x})$. Here, each variable x_i, $i = 1, \ldots, d$, can be either categorical or continuous. For each data point $\mathbf{x}$, suppose there is a binary mask $\mathbf{m} = (m_1, \ldots, m_d) \in \{0,1\}^d$ which indicates the missing entry for the sample, i.e.,

$$m_i = \begin{cases} 1 & \text{if } x_i \text{ is observed,} \\ 0 & \text{if } x_i \text{ is missing.} \end{cases}$$

Then, the observed (incomplete) data $\mathbf{x}^{\text{obs}} = \mathbf{x} \odot \mathbf{m} + \text{na} \odot (\mathbf{1} - \mathbf{m})$, where na indicates the missing value, $\odot$ denotes element-wise multiplication, and $\mathbf{1}$ is the all-one vector.

Suppose we have n complete (unobservable) data points $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} p_0(\mathbf{x})$ and simultaneously n corresponding masks $\mathbf{m}_1, \ldots, \mathbf{m}_n$ generated from a specific missing data mechanism detailed later. Then, the observed data samples are denoted as $S^{\text{obs}} = \{\mathbf{x}_i^{\text{obs}}\}_{i=1}^n$. The missing mechanisms can be categorized based on the relationships between the mask $\mathbf{m}$ and the complete data $\mathbf{x}$ (Little & Rubin, 1988) as follows,

- Missing Completely At Random (MCAR): mask $\mathbf{m}$ is independent from complete data $\mathbf{x}$.
- Missing At Random (MAR): mask $\mathbf{m}$ only depends on the observed value $\mathbf{x}^{\text{obs}}$.
- Not Missing At Random (NMAR): $\mathbf{m}$ depends on the observed value $\mathbf{x}^{\text{obs}}$ and missing value.

Compared with previous work which typically develops their algorithms and theoretical foundations under the M(C)AR assumption Li et al. (2019); Ipsen et al. (2022); Yoon et al. (2018a); Li & Marlin (2020); Mattei & Frellsen (2019), our method and theoretical guarantees aim to provide a general framework for learning on incomplete data and generate complete data. By modeling the score of the complete data distribution from the observed data, we only require mild assumptions of missing mechanisms for recovering the oracle score (we refer to Theorem 3.2). In the following, we provide a brief introduction to the score-based generative model.

2.2 Score-Based Generative Model

In this work, we adopt the diffusion model[2] as the prototype for developing our proposed method. We propose to train the model with missing values directly without the need for prior imputation. We first briefly review the key components of score-based generative models (Ho et al., 2020; Song et al., 2021b).

Score-based generative models are a class of generative models that learn the score function, which is the gradient of the log density of the data distribution. These models have gained attention due to their flexibility and effectiveness in capturing complex data distributions. Following the notation in Song et al. (2021b), the score-based generative models are based on a forward stochastic differential equation (SDE), $\mathbf{x}(t)$ with $t \in [0, T]$, defined as (which corresponds to Eq (5) in Song et al. (2021b))

$$\mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is the standard Wiener process (Brownian motion), $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function called the drift coefficient of $\mathbf{x}(t)$, and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is a scalar function known as the diffusion coefficient of $\mathbf{x}(t)$.

The solution of a stochastic differential equation is a continuous trajectory of random variables $\{\mathbf{x}(t)\}_{t\in[0,T]}$. Let $p(\mathbf{x})$ denote the path measure for the trajectory $\mathbf{x}$ on $[0, T]$, $p_t(\mathbf{x})$ denote the marginal probability density function of $\mathbf{x}(t)$, and $p(\mathbf{x}(t)|\mathbf{x}(s))$ denote the conditional probability density of $\mathbf{x}(t)$ conditioned on $\mathbf{x}(s)$, where $s < t$ is a previous time point. When constructing the SDE, we let $p_0(\mathbf{x})$ be the true data distribution, and after perturbing the data according to the SDE, the data distribution becomes $p_T(\mathbf{x})$ which is close to a tractable noise distribution, usually set as the standard Gaussian distribution.

The data generation process is performed via the reverse SDE, i.e., first sampling data $\mathbf{x}_T$ from $p_T(\mathbf{x})$ and then generate $\mathbf{x}_0$ through the reverse of equation 1. For any SDE in equation 1, the corresponding backward/reverse process is as follows (we refer Anderson (1982) for detailed explanation):

$$\mathrm{d}\mathbf{x}(t) = \left[\mathbf{f}(\mathbf{x}(t), t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{2}$$

[2]We use the diffusion model and score-based generative model interchangeably as they are equivalent Song et al. (2021b).

where $\overline{\mathbf{w}}$ is a standard Wiener process when time flows backwards from T to 0, and $\mathrm{d}t$ is an infinitesimal negative time step.

We can generate new data by running backward the reverse-time SDE equation 2 when the score of each marginal distribution, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is known. Score Matching (Hyvärinen, 2005; Vincent, 2011; Song et al., 2019) can be used for training a score-based model $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$ to estimate the score:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{p(\mathbf{x}(0))} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \left[\left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \right\|_2^2 \right] \right\}, \tag{3}$$

where $\lambda : [0, T] \rightarrow \mathbb{R}_{>0}$ is a positive weighting function, t is uniformly sampled over $[0, T]$, $\mathbf{x}(0) \sim p_0(\mathbf{x})$ and $\mathbf{x}(t) \sim p(\mathbf{x}(t)|\mathbf{x}(0))$. The local consistency of score matching is shown in (Hyvärinen, 2005), i.e., $\mathbb{E}_{p(\mathbf{x}(0))}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})\|_2^2] = 0 \Leftrightarrow \boldsymbol{\theta} = \boldsymbol{\theta}^*$ under the assumption that there exists an unique $\boldsymbol{\theta}^*$ such that the true score function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ can be represented by $s_{\boldsymbol{\theta}^*}$. Vincent (2011) builds the connection between Denoising Score Matching and Score Matching, and Song et al. (2019) further proves Sliced Score Matching can learn the consistent estimator of the oracle score and the asymptotic normality for the Sliced Score Matching.

3 METHOD

In this section, we first discuss the room for improvement in existing frameworks for synthesizing new complete data in Section 3.1. Then, we propose a diffusion-based unified framework, *MissDiff*, for learning a generative model from incomplete data in Section 3.2. The theoretical guarantees of *MissDiff* are provided in Section 3.3 and the related work is summarized in 3.4.

3.1 THE LIMITATION OF "IMPUTE-THEN-GENERATE" FRAMEWORK

To learn a generative model from data with missing values for generating complete data, we can first construct a complete training dataset and then learn a generative model on the complete data, which is referred to as the "impute-then-generate" framework. We can either delete instances (rows) or features (columns) with missing data or adopt traditional imputation methods or training machine learning imputation models (van Buuren & Groothuis-Oudshoorn, 2011; Bertsimas et al., 2017) or deep generative models for imputation tasks (Vincent et al., 2008; Yoon et al., 2018a; Biessmann et al., 2019; Wang et al., 2020; Ipsen et al., 2022; Muzellec et al., 2020). However, this pipeline may bring bias to the training objective. We clarify this claim in remark 3.1.

Remark 3.1 ("Impute-then-generate" framework is biased). Inspired by the analysis pipeline of "impute-then-regress" (Bertsimas et al., 2021; Ipsen et al., 2022) for the prediction task, we can study a corresponding framework for the generation task. The generative model p_ϕ represents the probability distribution of the synthetic data $\mathbf{x}$. Under the maximum likelihood framework, $\phi^* := \arg\max_\phi \mathbb{E}_{\mathbf{x} \sim p_0(\mathbf{x})}[\log p_\phi(\mathbf{x})]$. When data has missing values, the general approach, known as "impute-then-generate", may be used in practice. In this approach, the observed data $\mathbf{x}^{\text{obs}}$ is first imputed using an imputation model f_φ, where $f_\varphi(\mathbf{x}^{\text{obs}})$ is trained by minimizing the regression loss $\mathbb{E}_{(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}}) \sim p_0(\mathbf{x})} \|f_\varphi(\mathbf{x}^{\text{obs}}) - \mathbf{x}^{\text{miss}}\|^2$ with $\mathbf{x}^{\text{miss}}$ as the ground truth value[3]. The optimal $f_\varphi^*(\mathbf{x}^{\text{obs}})$ satisfies $f_\varphi^*(\mathbf{x}^{\text{obs}}) = \mathbb{E}_{p_0(\mathbf{x}^{\text{miss}}|\mathbf{x}^{\text{obs}})}\left[\mathbf{x}^{\text{miss}}\right]$. Then, the generative model is trained by maximizing the likelihood of imputed data, i.e., $\max_\phi \log p_\phi(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}} := f_\varphi(\mathbf{x}^{\text{obs}}))$. In general, $\mathbb{E}_{p_0(\mathbf{x}^{\text{miss}}|\mathbf{x}^{\text{obs}})}[p_\phi(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}})] \neq p_\phi(\mathbf{x}^{\text{obs}}, \mathbb{E}_{p_0(\mathbf{x}^{\text{miss}}|\mathbf{x}^{\text{obs}})}[\mathbf{x}^{\text{miss}}])$. Therefore, this pipeline is biased because the optimal single imputation can no longer capture the data variability.

In this work, we show that modeling the score of the complete data distribution can help to form a unified way for both imputation and generation tasks. However, the vanilla diffusion model mentioned in Section 2.2 is unable to directly deal with data with missing values. Therefore, we propose a diffusion-based framework designed for training diffusion models on tabular data with missing values, which enjoys certain advantages as compared with aforementioned framework.

[3]Here the notation $(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}})$ means the complete data $\mathbf{x}$.

3.2 *MissDiff*: DENOISING SCORE MATCHING ON MISSING DATA

We propose the following Denoising Score Matching method for data with missing values. Instead of using Eq equation 3 to learn the score-based model $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$, we propose *MissDiff* as solution to

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} J_{DSM}(\boldsymbol{\theta}) \\ &:= \frac{T}{2}\mathbb{E}_t\Big\{\lambda(t)\mathbb{E}_{\mathbf{x}^{\text{obs}}(0)}\mathbb{E}_{\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)}\left[\Big\|\Big(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t),t) - \nabla_{\mathbf{x}^{\text{obs}}(t)}\log p(\mathbf{x}^{\text{obs}}(t)\mid \mathbf{x}^{\text{obs}}(0))\Big)\odot\mathbf{m}\Big\|_2^2\right]\Big\},\end{aligned} \tag{4}$$

where $\lambda(t)$ is a positive weighting function, $\mathbf{m} = \mathbb{1}\{\mathbf{x}^{\text{obs}}(0) \neq \text{na}\}$ indicated the observed entries in $\mathbf{x}^{\text{obs}}$ and $p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) = \mathcal{N}(\mathbf{x}^{\text{obs}}(t); \mathbf{x}^{\text{obs}}(0), \beta_t\mathbb{I})$ is the Gaussian transition kernel. More implementation details can be found in Appendix C.4.

More specifically, we mainly adopt the Variance Preserving (VP) SDE in this paper although Variance Exploding (VE) SDE (Song et al., 2021b) is also applicable. The forward diffusion process of the Variance Preserving SDE is defined as (which corresponds to Eq (11) in (Song et al., 2021b)):

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w},$$

where $\{\beta_t \in (0,1)\}_{t\in(0,T)}$ is the increasing sequence denoting the variance schedule. Algorithm 1 demonstrates the Denoising Score Matching objective on missing data[4].

As long as the score function of complete data distribution is learned by Algorithm 1, we can adopt Algorithm 2 for imputation and Algorithm 3 for generating complete samples, which are provided in the Appendix C.3.

Algorithm 1 *MissDiff*: Denoising Score Matching on Data with Missing Values

Require: Diffusion process hyperparameter β_t, σ_t, denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$.

1: **repeat**
2: Sample $\mathbf{x}_0^{\text{obs}}$ according to the data distribution and missing mechanism;
3: Infer mask $\mathbf{m} = \mathbb{1}[\mathbf{x}_0^{\text{obs}} \neq \text{na}]$;
4: $t \sim \text{Uniform}(\{1, \ldots, T\})$;
5: $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
6: Take gradient descent step on

$$\nabla_\theta \left\|\left(\epsilon_t - \mathbf{s}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{obs}} + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)\right)\odot\mathbf{m}\right\|^2.$$

7: **until** converged.

3.3 THEORETICAL GUARANTEES OF *MissDiff*

In this section, we examine the effectiveness of *MissDiff* by theoretically characterizing the Score Matching objective under mild conditions on the missing mechanisms and build a further connection between Score Matching and maximizing likelihood objective for training the diffusion model.

In the following theorem, we present our first theoretical result that verifies that Denoising Score Matching on missing data can learn the oracle score, i.e., the score on complete data. Theorem 3.2 states that the global optimal solution of Denoising Score Matching on missing data obtained by *MissDiff* is the same as the oracle score, as long as we do not have a variable that is completely missing in the training data. The proof can be found in Appendix A.1.

Theorem 3.2. *Denote $\boldsymbol{\rho}(\mathbf{x}) = [\rho_1, \ldots, \rho_d] = \mathbb{E}_{p(\mathbf{m}|\mathbf{x})}[\mathbf{1} - \mathbf{m}]$ as the missing probability of each entry when the complete data equals $\mathbf{x}$ [5]. Define $\rho_{max} := \max_{i=1,\ldots,d}\sup_{\mathbf{x}}\rho_i(\mathbf{x})$ as the supreme of missing rates and assume $\rho_{max} < 1$. Let $\boldsymbol{\theta}^*$ be the solution to the training objective of MissDiff defined in Eq equation 4. Then we have*

$$\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}(t), t) = \nabla_{\mathbf{x}(t)}\log p_t(\mathbf{x}(t)).$$

[4]We write $\mathbf{x}(t)$ as $\mathbf{x}_t$ in the algorithm box for simplicity.

[5]$\mathbf{1}$ denotes all one vector.

It is well known that with careful design of the weighting function λ_t, Denoising Score Matching can upper bound the negative log-likelihood of the diffusion model on the complete data (Song et al., 2021a). Therefore, it is straightforward to extend such a connection to incomplete data scenarios, which is detailed in the following theorem. These results provide insightful connections between the training objective of *MissDiff* and the maximum likelihood objective of the generative model on observed data.

Theorem 3.3. *The objective function of Denoising Score Matching on missing data is an upper bound for the negative likelihood of the generative model on observed data $\mathbf{x}^{obs}$ up to a constant, that is, for $\lambda_t = \beta_t$ and under the same condition of Theorem 3.2 and mild regularity conditions detailed in Appendix A.2, we have*

$$-\mathbb{E}_{p(\mathbf{x}^{obs})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \leq \frac{1}{1-\rho_{max}} J_{\mathrm{DSM}}\left(\boldsymbol{\theta}\right) + C_1,$$

where C_1 is a constant independent of $\boldsymbol{\theta}$.

The proof of Theorem 3.3 can be found in Appendix A.2. When there are missing values, Theorem 3.3 shows that the Denoising score matching on incomplete data still upper bounds the likelihood of the incomplete data up to a constant coefficient $1/(1-\rho_{\max})$. When there is no data missing, ρ is all zero vector, then we have $1/(1-\rho_{\max}) = 1$ and Theorem 3.3 degenerates to the Corollary 1 in Song et al. (2021a), i.e.,

$$-\mathbb{E}_{p(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})] \leq J_{\mathrm{DSM}}(\boldsymbol{\theta}; g(\cdot)^2) + C_1,$$

where the $J_{\mathrm{DSM}}(\boldsymbol{\theta}; g(\cdot)^2)$ is the Denoising Score Matching objective on complete data.

3.4 Related Work

Learning from data with missing value: Numerous studies have been proposed to deal with missing values in the training data. Variational Autoencoder (VAE) based models (Ipsen et al., 2021; Nazábal et al., 2018; Ma et al., 2020; Mattei & Frellsen, 2019; Valera et al., 2017; Ipsen et al., 2022) maximize the evidence low bound of the observed data, while Generative Adversarial Network (GAN) based models (Yoon et al., 2018a; Li et al., 2019; Li & Marlin, 2020) employ adversarial training for both the generative and discriminative models; Trevor et al. adopt flow-based model for imputation (Richardson et al., 2020).. Recently, Tashiro et al. (2021) proposes the conditional score-based generative model for time series imputation and Zheng & Charoenphakdee (2022) adopts the conditional score-based diffusion model proposed in Tashiro et al. (2021) for imputing tabular data. However, all of the above works mainly focus on imputation tasks. They either need two-stage inference for generating new complete samples, such as learning a generative model on imputed data or imputing the generated data containing missing values, or require training additional networks[6]. For example, Li et al. (2019) trains two generator-discriminator pairs for the masks and data respectively, which increases the computational cost, and Li & Marlin (2020) adopts Partial Bidirectional GAN, which requires an encoding and decoding network for the generator. Moreover, Nazábal et al. (2018); Ma et al. (2020) require training a different VAE independently of each data dimension. *MissDiff* is a diffusion-based unified framework for imputation and generation tasks without two-stage inference or training additional networks. There are some concurrent works that adopt gradient-boosted decision trees (Jolicoeur-Martineau et al., 2023), diffusion model (Zhang et al., 2024), and autoregression modeling (McCarter, 2024). In (Jolicoeur-Martineau et al., 2023), the authors adopt XGBoost to estimate the score. Zhang et al. (Zhang et al., 2024) leverages the Expectation-Maximization that first learns the joint distribution of both the observed and currently estimated missing data and then re-estimates the missing data based on the conditional probability given the observed data. And McCarter wt al. (McCarter, 2024) adopts tree-based autoregressive modeling of tabular data.

Generative model for tabular data: Tabular data, as mixed-type data that typically contains both categorical and continuous variables, has attracted significant attention in the field of machine learning. The presence of mixed variable types and class imbalance for discrete variables make it a challenging task to model tabular data. Recently, several deep learning models have been proposed

[6]Additional network means the extra network needed compares with the same model dealing with complete data.

for tabular data generation (Xu et al., 2019; Choi et al., 2017; Srivastava et al., 2017; Park et al., 2018; Kim et al., 2021; Finlay et al., 2020; Kim et al., 2023; Kotelnikov et al., 2022). Among these methods, (Kotelnikov et al., 2022) employs Gaussian transitions for continuous variables and multinomial transitions for discrete variables, while (Kim et al., 2023) proposes a self-paced learning technique and a fine-tuning strategy for score-based models and achieves state-of-the-art performance in tabular data generation. Moreover, the discrete Score Matching methods proposed in Meng et al. (2022) and Sun et al. (2023) can also be employed to handle discrete variables in tabular data. However, all of the methods mentioned above did not take missing values in the training data into consideration.

4 Experiments

In this section, we demonstrate the effectiveness of the proposed *MissDiff* against existing state-of-the-art models. Since most of the approaches dealing with missing data work on imputation tasks, we compare with them in Section 4.1. Then, we mainly focus on the complete synthetic data generation task, which was much less evaluated in the literature with missing data. We present a careful experimental setup, including datasets, baseline models, and evaluation criterion, in Section 4.2. The detailed experimental results under different missing mechanisms are in Section 4.3.

Table 1: Evaluation on imputation tasks. The standard deviations of five independent trials are shown in the parenthesis. The *lower* the RMSE, the *better* the performance.

Method	Census	Breast	Wine	Concrete	Libras	diabetes
Mean /Mode	0.120(0.003)	0.263(0.009)	0.076(0.003)	0.217(0.007)	0.099(0.001)	0.222(0.003)
MICE(linear)	0.101(0.002)	0.154(0.011)	0.065(0.003)	0.153(0.006)	0.034(0.001)	0.263(0.002)
MissForest	0.112(0.004)	0.163(0.014)	0.060(0.002)	0.173(0.005)	0.024(0.001)	0.216(0.003)
GAIN	0.123(0.057)	0.165(0.006)	0.072(0.004)	0.203(0.007)	0.089(0.006)	0.202(0.003)
MIWAE	0.113(0.042)	0.1874(0.079)	0.074 (0.005)	0.195(0.006)	0.083(0.003)	0.194(0.081)
CSDI_T	0.099(0.003)	0.153(0.003)	0.065(0.004)	**0.131(0.008)**	**0.011(0.001)**	0.197(0.001)
MissDiff	**0.089(0.006)**	**0.136(0.002)**	**0.053(0.001)**	0.161(0.001)	0.0787(0.002)	**0.051(0.004)**

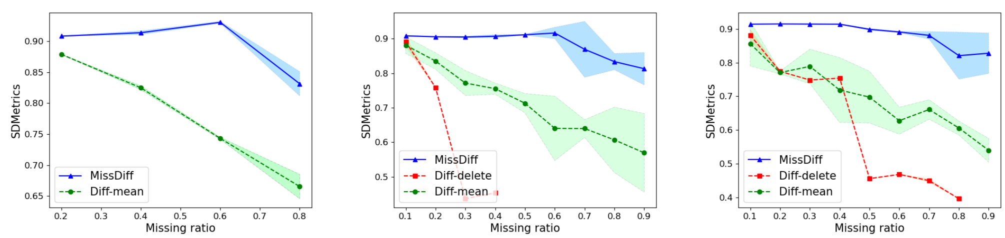
4.1 Experimental for Imputation Tasks

We follow the experimental setup as Zheng & Charoenphakdee (2022), which is evaluating *MissDiff* on six UCI Machine Learning Repository (Kelly et al.), e.g., Census (Kohavi & Becker, 1996), Breast (WIlliam, 1992), Wine (Paulo et al., 2009), Concrete (I-Cheng, 2007), Libras (Daniel et al., 2009), and Diabetes Dataset (Kohavi & Becker). We compare *MissDiff* with (i) the simple imputation method that uses mean values for continuous values and mode values for discrete variables (Mean / Mode), (ii) Multiple Imputation by Chained Equations (MICE) with linear regression (MICE_linear) (White et al., 2011), (iii) MissForest (Stekhoven, 2015), (iv) GAN-based imputation model, GAIN (Yoon et al., 2018a), (v) VAE-based imputation model, MIWAE (Mattei & Frellsen, 2019), and (vi) Diffusion-based imputation model, CSDI_T (Zheng & Charoenphakdee, 2022). We either adopt the results and hyperparameters from Zheng & Charoenphakdee (2022) or use the open source implementation from hyperimpute (Jarrett et al., 2022) concerning the baselines methods in Table 1. We evaluate these methods under the same criterion as Zheng & Charoenphakdee (2022), i.e., Root Mean Squared Error (RMSE) between the predicted value with the oracle missing value. The details of the missing mechanism can be found in Appendix C.1.

The performance comparison of *MissDiff* with state-of-the-art imputation approaches is presented in Table 1. For most datasets, *MissDiff* achieves the lowest RMSE. We provide some explanations about why *MissDiff* can achieve better performance than previous methods in the following. VAE-based imputation methods maximize the variational lower bound on observed data that may not have the guarantees on complete data, while *MissDiff* recovers the oracle score on complete data by Theorem 3.2. *MissDiff* avoids the instability caused by adversarial training, which might be the reason for achieving better results than the GAN-based method. Compared with the Diffusion-based imputation model, CSDI (Tashiro et al., 2021) and its tabular variant CSDI_T (Zheng & Charoenphakdee, 2022), that use conditional score matching, *MissDiff* achieves better results for the following two reasons. Conditional scores (depending on which information is conditioned) are difficult to learn and analyze. Therefore, there were no theoretical guarantees on whether the learned conditional score satisfied the optimality condition similar to Theorem 3.2 and 3.3. Moreover, although conditional score matching performs better in time series imputation tasks than unconditional score matching, it is not necessarily the case for tabular data. There may exist some complex or irregular dependencies between different columns in tabular data, e.g., some features might be redundant (uninformative). *MissDiff* achieves better results than CSDI_T.

4.2 Experimental Setup for Generation task

Datasets: We present a suite of numerical evaluations of the proposed *MissDiff* approach on a simulated Bayesian Network data, a real Census tabular dataset (Kohavi & Becker, 1996), and the MIMIC4ED tabular dataset (Xie et al., 2022), with various proportions of missing values. The details of the missing mechanism can be found in Appendix C.2.



(a) Row missing (b) Column missing (c) Independent missing

Figure 1: *Fidelity* evaluation of *MissDiff* on data generated by Bayesian Network under different missing ratios. We shade the area between mean ± std. More discussions are provided in Appendix C.5.

Baseline Methods: Since few previous works provide the experimental results of the generative models learned on tabular data with missing values for generating new complete samples, we mainly compare with the following five baseline methods:

1. *Diff-delete*: Learn a vanilla diffusion model after deleting rows containing missing values.
2. *Diff-mean*: Learn a vanilla diffusion model after imputing missing values using the mean value in that column.
3. STaSy (Kim et al., 2023) with the above two data completion methods. STaSy is the state-of-the-art diffusion model on tabular data, which outperforms MedGAN (Choi et al., 2017), VEEGAN (Srivastava et al., 2017), CTGAN (Xu et al., 2019), TVAE (Xu et al., 2019), TableGAN (Park et al., 2018), OCTGAN (Kim et al., 2021), RNODE (Finlay et al., 2020) by a large margin.
4. CSDI_T (Zheng & Charoenphakdee, 2022) learns a conditional diffusion on missing data.

Remark 4.1. MIWAE (Mattei & Frellsen, 2019) cannot be used for generation tasks directly. We provide the detailed discussion in Appendix C.5. CSDI_T can be used for generation tasks. However, no information can be conditioned on, which makes CSDI_T degenerate to *MissDiff*. Moreover, using CSDI_T for generation task exists a mismatch between training and generation, which makes the performance of CSDI_T worse than *MissDiff*.

In the following experiments, we use the variance-preserving SDE with the time duration $T = 100$ for the Bayesian Network and Census dataset and $T = 150$ for the MIMIC4ED dataset. We adopt four layers residual network as the backbone of the diffusion model. The dimension of the diffusion embedding is 128 with channels as 64. We use the standard pre/post-processing of tabular data to deal with mixed-type data (Kim et al., 2023; Kotelnikov et al., 2022; Zheng & Charoenphakdee, 2022), i.e., we use the min-max normalization for the continuous variables and reverse its scalar when generation. We use one-hot embedding for the discrete variables and use the rounding function after the softmax function when generation. We train the diffusion model for 250 epochs with batch size 64. For more details, please refer to Appendix C.4.

Evaluation Criterion: Following Xu et al. (2019); Kim et al. (2023); Kotelnikov et al. (2022), we use two types of criteria, *fidelity* and *utility*, to evaluate the quality of the synthetic data generated. To evaluate the *fidelity* of synthetic data compared with real data, we adopt a model-agnostic library, SDMetrics (Dat, 2023). The result is a float number range from 0 to 100%. The larger the score, the better the overall quality of synthetic data is.

To evaluate the *utility* of synthetic data, we follow the same pipeline of Kim et al. (2023), i.e., training various models, including Decision Tree, AdaBoost, Logistic/Linear Regression, MLP classifier/regressor, RandomForest, and XGBoost, on synthetic data, and validate the model on original training data, and test them with real test data. For classification tasks, we mainly use classification accuracy and also report AUROC, F1, and Weighted-F1 in Appendix C.6. For regression tasks, we mainly use RMSE and also report R^2 in the Appendix C.6. All the experiments are obtained from 3 repetitions.

4.3 Experiment Results for Generation task

4.3.1 Simulation Study

Q1: How does MissDiff perform on different missing ratios against the vanilla diffusion model learned on the data completed by two baseline methods mentioned in Section 4.2?

Figure 1 summarizes the SDMetrics score on the simulated Bayesian Network dataset example. With the same diffusion model architecture and the same training hyperparameter, *MissDiff* achieves consistently better results against the vanilla diffusion model deleting the incomplete row or using the mean value for imputation on various missing ratios. Moreover, the advantage of *MissDiff* becomes more obvious for large missing ratios. These experimental results verify the motivation of *MissDiff* proposed in Remark 3.1 that the learning objective of impute-then-generate is biased. Directly learning on the missing data can significantly enhance the performance of the learned generative model [7].

4.3.2 Real Tabular Datasets

Q2: How does MissDiff perform on more complicated real-world data and compared with state-of-the-art generative model on tabular data?

Table 2 demonstrates the effectiveness of *MissDiff* on the Census dataset under MCAR. STaSy is a state-of-the-art generative model for tabular data, which means *MissDiff* achieves quite good performance on learning from incomplete data and generating complete data. More importantly, *MissDiff* achieves better performance than *STaSy-delete* and *STaSy-mean* even without adopting the self-paced learning technique and the fine-tuning strategy used by STaSy. More experiments and discussions can be found in Appendix C.6.

Table 2: *Utility* (classification accuracy) evaluation of *MissDiff* on Census dataset. "-" denotes the corresponding method cannot applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data. The *larger* the accuracy, the *better* the performance.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*	CSDI_T
Row Missing	**79.48**%	-	78.45%	-	70.79%	79.15%
Column Missing	71.68%	72.89%	79.60%	68.96%	74.47%	**80.31%**
Independent Missing	**79.49**%	75.39%	75.96%	78.36%	77.34%	79.12%

Table 3 shows the performance of *MissDiff* on the MIMIC4ED dataset under MCAR. On this large dataset with dozens of continuous and discrete variables, *MissDiff* gives consistently better performance with the same training epochs (250 epochs).

Q4: How does MissDiff perform on other missing mechanisms beyond MCAR, i.e., MAR and NMAR?

Table 4 demonstrates the effectiveness of *MissDiff* on the Census dataset beyond MCAR. The results show the great potential of learning directly on the missing data when the missing mechanism is not

[7] We provide more discussions on the "Column missing" scenario in Appendix C.5.

Table 3: *Utility* (RMSE) evaluation of *MissDiff* on MIMIC4ED dataset. *Diff-delete* and *STaSy-delete* cannot be applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data. The *lower* the RMSE, the *better* the performance.

	MissDiff	*Diff-mean*	*STaSy-mean*	CSDI_T
Row Missing	**1.826**	2.166	1.894	1.853
Column Missing	**1.834**	2.011	1.935	1.874
Independent Missing	**1.852**	2.483	1.972	1.879

Table 4: *Utility* (classification accuracy) evaluation of *MissDiff* on Census dataset under MAR, NMAR. The *larger* the accuracy, the *better* the performance.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*	CSDI_T
MAR	**79.95**%	69.48%	77.43%	71.28%	73.65%	79.42%
NMAR	**80.95**%	66.50%	80.03%	78.11%	73.92%	80.23%

MCAR, which cannot be easily dealt with by previous methods (Li et al., 2019; Ipsen et al., 2022; Yoon et al., 2018a; Li & Marlin, 2020).

5 Conclusion

We propose a unified diffusion-based framework, called *MissDiff*, for synthetic data generation and imputation trained on data with missing values. Compared with the two-stage inference pipeline, *MissDiff* is an unbiased, and computationally friendly framework. The theoretical justification for *MissDiff*'s effectiveness is provided. Moreover, extensive numerical experiments demonstrate strong empirical evidence for the effectiveness of *MissDiff*.

Limitations and broader impact Overall, this research presents a promising direction for handling missing data in generative model training. The proposed framework, *MissDiff*, has potential applications in a wide range of domains where missing data is a common issue. A potential limitation of this work is that it has only been empirically validated on standard tabular data. For future directions, it would be interesting to see how *MissDiff* performs empirically with more complicated data types, e.g., tabular data that contains text information in medical diagnosis. Furthermore, further research could explore the theoretical effectiveness of *MissDiff* on the utility perspective or differential privacy perspective.

REFERENCES

Ahmed M. Alaa, Jinsung Yoon, Scott Hu, and Mihaela van der Schaar. Personalized risk scoring for critical care prognosis using mixtures of Gaussian processes. *IEEE Transactions on Biomedical Engineering*, 65:207–218, 2016.

Brian. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12:313–326, 1982. URL `https://api.semanticscholar.org/CorpusID:3897405`.

Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: An optimization approach. *J. Mach. Learn. Res.*, 18:196:1–196:39, 2017.

Dimitris Bertsimas, Arthur Delarue, and Jean Pauphilet. Prediction with missing data. *ArXiv*, abs/2104.03158, 2021.

Felix Biessmann, Tammo Rukat, Phillipp Schmidt, Prathik Naidu, Sebastian Schelter, Andrey Taptunov, Dustin Lange, and David Salinas. DataWig: Missing value imputation for tables. *J. Mach. Learn. Res.*, 20:175:1–175:6, 2019.

E. Choi, Siddharth Biswal, Bradley A. Malin, Jon D. Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete electronic health records using generative adversarial networks. *ArXiv*, abs/1703.06490, 2017.

Dias Daniel, Peres Sarajane, and Bscaro Helton. Libras Movement. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C5GC82.

Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alexandros G. Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *NIPS*, 2023.

Synthetic Data Metrics. DataCebo, Inc., 4 2023. URL `https://docs.sdv.dev/sdmetrics/`. Version 0.9.3.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. *ArXiv*, abs/2105.05233, 2021.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M. Oberman. How to train your neural ODE: the world of jacobian and kinetic regularization. In *International Conference on Machine Learning*, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005.

Yeh I-Cheng. Concrete Compressive Strength. UCI Machine Learning Repository, 2007. DOI: https://doi.org/10.24432/C5PK67.

Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. not-MIWAE: Deep generative modelling with missing not at random data. *ICLR*, 2021.

Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. How to deal with missing data in supervised deep learning? In *International Conference on Learning Representations*, 2022.

Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. *ArXiv*, abs/2206.07769, 2022. URL `https://api.semanticscholar.org/CorpusID:249712073`.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. In *International Conference on Artificial Intelligence and Statistics*, 2023. URL `https://api.semanticscholar.org/CorpusID:262046450`.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. https://archive.ics.uci.edu.

Jayoung Kim, Jinsung Jeon, Jaehoon Lee, Jihyeon Hyeong, and Noseong Park. OCT-GAN: Neural ODE-based conditional tabular GANs. *Proceedings of the Web Conference 2021*, 2021.

Jayoung Kim, Chae Eun Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. 2023.

Ronny Kohavi and Barry Becker. CDC. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset.

Ronny Kohavi and Barry Becker. Census income data set. https://archive.ics.uci.edu/ml/datasets/census+income, 1996.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. *ArXiv*, abs/2209.15421, 2022.

Steven Cheng-Xian Li and Benjamin M Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *International Conference on Machine Learning*, 2020.

Steven Cheng-Xian Li, Bo Jiang, and Benjamin M Marlin. MisGAN: Learning from incomplete data with generative adversarial networks. *ArXiv*, abs/1902.09599, 2019.

Roderick J. A. Little and Donald B. Rubin. Statistical analysis with missing data. 1988.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2836–2844, 2021.

Chao Ma, Sebastian Tschiatschek, José Miguel Hernández-Lobato, Richard E. Turner, and Cheng Zhang. VAEM: a deep generative model for heterogeneous mixed type data. *ArXiv*, abs/2006.11941, 2020.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, 2019.

Calvin McCarter. Unmasking trees for tabular data. *ArXiv*, abs/2407.05593, 2024. URL https://api.semanticscholar.org/CorpusID:271050499.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_RL7wtHkPJK.

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, 2020.

Alfredo Nazábal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using VAEs. *Pattern Recognit.*, 107:107501, 2018.

Diogo Neves, João Alves, Marcel Ganesh Naik, Alberto José Proença, and Fabian Prasser. From missing data imputation to data generation. *J. Comput. Sci.*, 61:101640, 2022.

Bernt Øksendal. Stochastic differential equations : an introduction with applications. *Journal of the American Statistical Association*, 82:948, 1987.

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.

Cortez Paulo, Cerdeira A., Almeida F., Matos T., and Reis J. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Trevor W. Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A. Bernal. Mcflow: Monte carlo flow models for data imputation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14193–14202, 2020. URL https://api.semanticscholar.org/CorpusID:214714171.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Neural Information Processing Systems*, 2021a.

Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021b.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *NIPS*, 2017.

Daniel J. Stekhoven. missforest: Nonparametric missing value imputation using random forest. 2015.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *ArXiv*, abs/2107.03502, 2021.

Isabel Valera, Melanie F. Pradier, Maria Lomeli, and Zoubin Ghahramani. General latent feature models for heterogeneous datasets. *J. Mach. Learn. Res.*, 21:100:1–100:49, 2017.

Stef van Buuren and Karin G. M. Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45:1–67, 2011.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011.

Pascal Vincent, H. Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, 2008.

Yufeng Wang, Dan Li, Xiang Li, and Min Yang. PC-GAIN: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural networks : the official journal of the International Neural Network Society*, 141:395–403, 2020.

Ian R. White, Patrick Royston, and Angela M. Wood. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, 30, 2011. URL https://api.semanticscholar.org/CorpusID:37379599.

Wolberg WIlliam. Breast Cancer Wisconsin (Original). UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5HP4Z.

Feng Xie, Jun Zhou, Jin Wee Lee, Mingrui Tan, Siqi Li, Logasan SO Rajnthern, Marcel Lucas Chee, Bibhas Chakraborty, A.I. Wong, Alon Dagan, Marcus Eng Hock Ong, Fei Gao, and Nan Liu. Benchmarking emergency department prediction models with machine learning and public electronic health records. *Scientific Data*, 9, 2022.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Neural Information Processing Systems*, 2019.

Jinsung Yoon, Camelia Davtyan, and Mihaela van der Schaar. Discovery and clinical decision support for personalized healthcare. *IEEE Journal of Biomedical and Health Informatics*, 21:1133–1145, 2017.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. *ArXiv*, abs/1806.02920, 2018a.

Jinsung Yoon, William R. Zame, Amitava Banerjee, Martin Cadeiras, Ahmed M. Alaa, and Mihaela van der Schaar. Personalized survival predictions via trees of predictors: An application to cardiac transplantation. *PLoS ONE*, 13, 2018b.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. *ArXiv*, abs/2310.09656, 2023. URL `https://api.semanticscholar.org/CorpusID:264146605`.

Hengrui Zhang, Liancheng Fang, and Philip S. Yu. Unleashing the potential of diffusion models for incomplete data imputation. *ArXiv*, abs/2405.20690, 2024. URL `https://api.semanticscholar.org/CorpusID:270199661`.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Improved techniques for maximum likelihood estimation for diffusion odes. In *ICML*, 2023.

Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *ArXiv*, abs/2210.17128, 2022.

A Proofs for Section 4

A.1 Proof of Theorem 3.2

In order to show Theorem 3.2, we aim to show that the optimal solution $\boldsymbol{\theta}^*$, which minimizes the objective function $J_{DSM}(\boldsymbol{\theta})$ satisfies $\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}(t),t) = \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))$, i.e., the optimal solution to the population loss function can recover the oracle score function.

For the Gaussian transition distribution that we used with the isotropic covariance matrix, the score on the incomplete data is equivalent to the score on the complete data when performing element-wise multiplication with mask, i.e., $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) \odot \mathbf{m} = \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \odot \mathbf{m}$[8], where $\mathbf{m} = \mathbb{1}\{\mathbf{x}^{\text{obs}}(0) \neq \text{na}\}$ indicated the missing entries in $\mathbf{x}^{\text{obs}}(0)$. Therefore, under certain conditions[9], we may first relate the Denosing Score Matching objective on missing data to the Denosing Score Matching objective on the complete data, i.e., the optimal solution of $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}^{\text{obs}}(0),\mathbf{m})}\mathbb{E}_{p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t),t) - \nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))) \odot \mathbf{m}\|_2^2]$ admits the same solution as $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}(0),\mathbf{m})}\mathbb{E}_{p(\mathbf{x}(t)|\mathbf{x}(0))}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t),t) - \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0))) \odot \mathbf{m}\|_2^2]$.

Moreover, notice that we have

$$\begin{aligned}
&\mathbb{E}_{p(\mathbf{x}(0),\mathbf{m})}\mathbb{E}_{p(\mathbf{x}(t)|\mathbf{x}(0))}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t),t) - \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0))) \odot \mathbf{m}\|_2^2] \\
&= \mathbb{E}_{p(\mathbf{x(0)},\mathbf{x(t)})}\|[(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t),t) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))) \odot \sqrt{\mathbb{E}_{p(\mathbf{m}|\mathbf{x}(0))}[\mathbf{m}]}\|_2^2],
\end{aligned}$$

where $\sqrt{\boldsymbol{z}}$ denotes the element-wise operation on vector $\boldsymbol{z}$. The last equation is because we take the conditional expectation of the binary mask $\mathbf{m}$ and since $\mathbf{m}_i \in \{0,1\}$ we have $\mathbb{E}[\mathbf{m}_i^2] = \mathbb{E}[\mathbf{m}_i]$ for any distribution of $\mathbf{m}$. Since $\mathbb{E}_{p(\mathbf{m}|\mathbf{x}(0))}[\mathbf{m}] = \mathbf{1} - \boldsymbol{\rho}$ with $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_d]$ and $\rho_i < 1$, $i \in \{1, 2, ..., d\}$ being the population percentage of missing samples for the i-th entry, we have $\mathbb{E}_{p(\mathbf{m}|\mathbf{x}(0))}[\mathbf{m}] > 0$ and thus we can show the global optimal of Denoising Score Matching on missing data is the same as the oracle score.

A.2 Proof of Theorem 3.3

The notations are defined as follows. We let π denote the pre-specified prior distribution (e.g., the standard normal distribution), $\mathcal{C}$ denote all continuous functions, and $\mathcal{C}^k$ denote the family of functions with continuous k-th order derivatives. Denote $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_d] = \mathbb{E}_{p(\mathbf{m}|\mathbf{x}(0))}[\mathbf{1} - \mathbf{m}]$ as the population percentage of missing samples for the i-th entry in the training data. Suppose $\max_{i=1,\ldots,d} \sup_{\mathbf{x}(0)} \rho_i < 1$. In addition, we make the same mild regularity assumptions as Song et al. (2021a) in the following.

Assumption A.1. (i) $p(\mathbf{x}) \in \mathcal{C}^2$ and $\mathbb{E}_{\mathbf{x}\sim p_0}[\|\mathbf{x}\|_2^2] < \infty$.

(ii) $\pi(\mathbf{x}) \in \mathcal{C}^2$ and $\mathbb{E}_{\mathbf{x}\sim\pi}[\|\mathbf{x}\|_2^2] < \infty$.

(iii) $\forall t \in [0,T] : f(\cdot,t) \in \mathcal{C}^1, \exists C > 0, \forall \mathbf{x} \in \mathbb{R}^d, t \in [0,T] : \|f(\mathbf{x},t)\|_2 \le C(1 + \|\mathbf{x}\|_2)$.

(iv) $\exists C > 0, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d : \|f(\mathbf{x},t) - f(\mathbf{y},t)\|_2 \le C\|\mathbf{x} - \mathbf{y}\|_2$.

(v) $g \in \mathcal{C}$ and $\forall t \in [0,T], |g(t)| > 0$.

(vi) For any open bounded set $\mathcal{O}$, $\int_0^T \int_{\mathcal{O}} \|p_t(\mathbf{x})\|_2^2 + dg(t)^2\|\nabla_{\mathbf{x}} p_t(\mathbf{x})\|_2^2 \,\mathrm{d}\mathbf{x}\mathrm{d}t < \infty$.

(vii) $\exists C > 0 \forall \mathbf{x} \in \mathbb{R}^d, t \in [0,T] : \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|_2 \le C(1 + \|\mathbf{x}\|_2)$.

(viii) $\exists C > 0, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d : \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \nabla_{\mathbf{y}} \log p_t(\mathbf{y})\|_2 \le C\|\mathbf{x} - \mathbf{y}\|_2$.

---

[8] Assume $p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) = \mathcal{N}(\mathbf{x}^{\text{obs}}(t); \mu^{\text{obs}}, \Sigma)$ and $p(\mathbf{x}(t)|\mathbf{x}(0)) = \mathcal{N}(\mathbf{x}(t); \mu, \Sigma)$, with $\Sigma = (1-\bar{\alpha}_t)\mathbb{I}$ and $\mu^{\text{obs}} = \mu \odot \mathbf{m}$. It is not hard to see $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) \odot \mathbf{m} = -\frac{1}{(1-\bar{\alpha}_t)}(\mathbf{x}^{\text{obs}}(t) - \mu^{\text{obs}}) \odot \mathbf{m} = -\frac{1}{(1-\bar{\alpha}_t)}(\mathbf{x}(t) - \mu) \odot \mathbf{m} = \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \odot \mathbf{m}$.

[9] We assume the score network $s_{\boldsymbol{\theta}}$ possesses sufficient approximation capability to encompass the true score function.

(ix) $\exists C > 0 \forall \mathbf{x} \in \mathbb{R}^d, t \in [0, T] : \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\|_2 \leq C(1 + \|\mathbf{x}\|_2)$.

(x) $\exists C > 0, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d : \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{y}, t)\|_2 \leq C\|\mathbf{x} - \mathbf{y}\|_2$.

(xi) Novikov's condition: $\mathbb{E}[\exp(\frac{1}{2}\int_0^T \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\|_2^2 \, \mathrm{d}t)] < \infty$.

(xii) $\forall t \in [0, T], \exists k > 0 : p_t(\mathbf{x}) = O(e^{-\|\mathbf{x}\|_2^k})$ as $\|\mathbf{x}\|_2 \to \infty$.

We mainly follow the proof strategy in Song et al. (2021a). Consider the predefined SDE on the observed data,

$$\mathrm{d}\mathbf{x}^{\text{obs}} = f(\mathbf{x}^{\text{obs}}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{5}$$

and the SDE parametrized by θ,

$$\mathrm{d}\hat{\mathbf{x}}_\theta^{\text{obs}} = \mathbf{s}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_\theta^{\text{obs}}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{6}$$

Let $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ denote the path measure of $\{\mathbf{x}^{\text{obs}}(t)\}_{t\in[0,T]}$ and $\{\hat{\mathbf{x}}_\theta^{\text{obs}}(t)\}_{t\in[0,T]}$, respectively. Therefore, the distribution of $p_0(\mathbf{x})$ and $p_\theta(\mathbf{x})$ can be represented by the Markov kernel $K(\{\mathbf{z}(t)\}_{t\in[0,T]}, \mathbf{y}) := \delta(\mathbf{z}(0) = \mathbf{y})$ as follow:

$$p_0(\mathbf{x}) = \int K(\{\mathbf{x}^{\text{obs}}(t)\}_{t\in[0,T]}, \mathbf{x})\mathrm{d}\boldsymbol{\mu}(\{\mathbf{x}^{\text{obs}}(t)\}_{t\in[0,T]}),$$
$$p_\theta(\mathbf{x}) = \int K(\{\hat{\mathbf{x}}_\theta^{\text{obs}}(t)\}_{t\in[0,T]}, \mathbf{x})\mathrm{d}\boldsymbol{\nu}(\{\hat{\mathbf{x}}_\theta^{\text{obs}}(t)\}_{t\in[0,T]}).$$

According to the data processing inequality with this Markov kernel, the Kullback–Leibler (KL) divergence between the distribution of $p_0(\mathbf{x})$ and $p_\theta(\mathbf{x})$ can be upper bounded, i.e.,

$$D_{\mathrm{KL}}(p_0\|p_\theta) = D_{\mathrm{KL}}\Big(\int K(\{\mathbf{x}^{\text{obs}}(t)\}_{t\in[0,T]}, \mathbf{x})\mathrm{d}\boldsymbol{\mu}\Big\| \int K(\{\hat{\mathbf{x}}_\theta^{\text{obs}}(t)\}_{t\in[0,T]}, \mathbf{x})\mathrm{d}\boldsymbol{\nu}\Big) \leq D_{\mathrm{KL}}(\boldsymbol{\mu}\|\boldsymbol{\nu}). \tag{7}$$

By the chain rule of KL divergences,

$$D_{\mathrm{KL}}(\boldsymbol{\mu}\|\boldsymbol{\nu}) = D_{\mathrm{KL}}(p_T\|\pi) + \mathbb{E}_{\mathbf{z}\sim p_T}[D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot \mid \mathbf{x}^{\text{obs}}(T) = \mathbf{z})\|\boldsymbol{\nu}(\cdot \mid \hat{\mathbf{x}}_\theta^{\text{obs}}(T) = \mathbf{z}))]. \tag{8}$$

Under assumptions (i) (iii) (iv) (v) (vi) (vii) (viii), the SDE in Eq equation 5 has a corresponding reverse-time SDE given by

$$\mathrm{d}\mathbf{x}^{\text{obs}} = [f(\mathbf{x}^{\text{obs}}, t) - g(t)^2\nabla_{\mathbf{x}^{\text{obs}}} \log p_t(\mathbf{x}^{\text{obs}})]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}. \tag{9}$$

Since Eq equation 9 is the time reversal of Eq equation 5, it induces the same path measure $\boldsymbol{\mu}$. As a result, $D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot \mid \mathbf{x}^{\text{obs}}(T) = \mathbf{z})\|\boldsymbol{\nu}(\cdot \mid \hat{\mathbf{x}}_\theta^{\text{obs}}(T) = \mathbf{z}))$ can be viewed as the KL divergence between the path measures induced by the following two (reverse-time) SDEs:

$$\mathrm{d}\mathbf{x}^{\text{obs}} = [f(\mathbf{x}^{\text{obs}}, t) - g(t)^2\nabla_{\mathbf{x}^{\text{obs}}} \log p_t(\mathbf{x}^{\text{obs}})]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \quad \mathbf{x}^{\text{obs}}(T) = \mathbf{x}^{\text{obs}},$$
$$\mathrm{d}\hat{\mathbf{x}}^{\text{obs}} = [f(\hat{\mathbf{x}}^{\text{obs}}, t) - g(t)^2\mathbf{s}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^{\text{obs}}, t)]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \quad \hat{\mathbf{x}}_\theta^{\text{obs}}(T) = \mathbf{x}^{\text{obs}}.$$

Under assumptions (vii) (viii) (ix) (x) (xi), we apply the Girsanov Theorem II [(Øksendal, 1987), Theorem 8.6.6], together with the martingale property of Itô integrals, which yields

$$\begin{aligned}
J_{\mathrm{SM}}(\theta; g(\cdot)^2) &= \int_0^T \mathbb{E}_{\mathbf{m}, p_t(\mathbf{x}^{\text{obs}}(t))}[g(t)^2\|(\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p_t(\mathbf{x}^{\text{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t), t)) \odot \mathbf{m}(x)\|_2^2]\mathrm{d}t \\
&= \int_0^T \mathbb{E}_{p_t(\mathbf{x}^{\text{obs}}(t))}[g(t)^2\|(\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p_t(\mathbf{x}^{\text{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t), t)) \odot \sqrt{\mathbb{E}[\mathbf{m}(x)]}\|_2^2]\mathrm{d}t \\
&\geq 2(1 - \rho_{\text{max}})\mathbb{E}_{\boldsymbol{\mu}}[\frac{1}{2}\int_0^T g(t)^2\|\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p_t(\mathbf{x}^{\text{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t), t)\|_2^2 \, \mathrm{d}t] \\
&\geq 2(1 - \rho_{\text{max}})D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot \mid \mathbf{x}^{\text{obs}}(T) = \mathbf{z})\|\boldsymbol{\nu}(\cdot \mid \hat{\mathbf{x}}_\theta^{\text{obs}}(T) = \mathbf{z}))
\end{aligned} \tag{10}$$

where $\rho_{\text{max}} = \max_{i=1,\ldots,d} \sup_x \mathbb{E}[1 - \mathbf{m}_i(x)]$ denotes the supreme of missing rates, and $1 - \rho_{\text{max}} > 0$ by assumption. Combining Eqs. equation 7, equation 8 and equation 10, we have $D_{\mathrm{KL}}(p_0\|p_{\boldsymbol{\theta}}) \leq \frac{1}{1-\rho_{\text{max}}} J_{\mathrm{SM}}(\boldsymbol{\theta}; g(\cdot)^2) + D_{\mathrm{KL}}(p_T\|\pi)$, which further yields $-\mathbb{E}_{p(\mathbf{x}^{\text{obs}})}[\log p_\theta(\mathbf{x})] \leq \frac{1}{1-\rho_{\text{max}}} J_{\mathrm{DSM}}(\boldsymbol{\theta}; g(\cdot)^2) + C_1$ by Lemma A.2, where C_1 is a constant independent of θ.

Lemma A.2. *Denoising Score Matching on missing data is equivalent to Score Matching on missing data, i.e.,*

$$\begin{aligned}&\mathbb{E}_{p_t(\mathbf{x}^{obs})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{obs},t)-\nabla_{\mathbf{x}^{obs}}\log p_t(\mathbf{x}_t^{obs}))\odot\mathbf{m}\|_2^2]\\&=\mathbb{E}_{p(\mathbf{x}_0^{obs})}\mathbb{E}_{p(\mathbf{x}_t^{obs}|\mathbf{x}_0^{obs})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{obs},t)-\nabla_{\mathbf{x}_t^{obs}}\log p(\mathbf{x}_t^{obs}\mid\mathbf{x}_0^{obs}))\odot\mathbf{m}\|_2^2]+C,\end{aligned}\tag{11}$$

where $\mathbf{m}=\mathbb{1}\{\mathbf{x}_0^{obs}\neq\mathrm{na}\}$ indicated the missing entries in $\mathbf{x}^{obs}$ and C is a constant that does not depend on $\boldsymbol{\theta}$. We interchange $\mathbf{x}^{obs}(t)$ with $\mathbf{x}_t^{obs}$.

Proof. We begin with the Score Matching on the left-hand side of equation 11

$$\begin{aligned}\mathrm{LHS}&=\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t)-\nabla_{\mathbf{x}_t^{\mathrm{obs}}}\log p_t(\mathbf{x}_t^{\mathrm{obs}}))\odot\mathbf{m}\|_2^2]\\&=\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t)\odot\mathbf{m}\|^2]-S(\theta)+C_2,\end{aligned}\tag{12}$$

where $C_2=\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|\nabla_{\mathbf{x}_t^{\mathrm{obs}}}\log p_t(\mathbf{x}_t^{\mathrm{obs}})\odot\mathbf{m}\|^2]$ is a constant that does not depend on $\boldsymbol{\theta}$, and

$$\begin{aligned}S(\theta)&=2\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\nabla_{\mathbf{x}_t^{\mathrm{obs}}}\log p_t(\mathbf{x}_t^{\mathrm{obs}})\odot\mathbf{m}\rangle]\\&=2\int_{\mathbf{x}_t^{\mathrm{obs}}}p_t(\mathbf{x}_t^{\mathrm{obs}})\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\nabla_{\mathbf{x}_t^{\mathrm{obs}}}\log p_t(\mathbf{x}_t^{\mathrm{obs}})\odot\mathbf{m}\rangle\,\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}\\&=2\int_{\mathbf{x}_t^{\mathrm{obs}}}\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\nabla_{\mathbf{x}_t^{\mathrm{obs}}}p_t(\mathbf{x}_t^{\mathrm{obs}})\odot\mathbf{m}\rangle\,\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}\\&=2\int_{\mathbf{x}_t^{\mathrm{obs}}}\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\int_{\mathbf{x}_0^{\mathrm{obs}}}p_0(\mathbf{x}_0^{\mathrm{obs}})p(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})\odot\mathbf{m}\,\mathrm{d}\mathbf{x}_0^{\mathrm{obs}}\rangle\,\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}\\&=2\int_{\mathbf{x}_t^{\mathrm{obs}}}\int_{\mathbf{x}_0^{\mathrm{obs}}}p_0(\mathbf{x}_0^{\mathrm{obs}})p(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\frac{\mathrm{d}\log p(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\odot\mathbf{m}\rangle\,\mathrm{d}\mathbf{x}_0^{\mathrm{obs}}\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}\\&=2\mathbb{E}_{p(\mathbf{x}_t^{\mathrm{obs}},\mathbf{x}_0^{\mathrm{obs}})}[\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\frac{\mathrm{d}\log p(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\odot\mathbf{m}\rangle].\end{aligned}$$

Substituting this expression for $S(\theta)$ into Eq equation 12 yields

$$\begin{aligned}\mathrm{LHS}=&\,\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t)\odot\mathbf{m}\|^2]\\&-2\mathbb{E}_{p(\mathbf{x}_t^{\mathrm{obs}},\mathbf{x}_0^{\mathrm{obs}})}[\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\frac{\mathrm{d}\log p(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\odot\mathbf{m}\rangle]+C_2.\end{aligned}\tag{13}$$

On the other hand, we also have the Denoising Score Matching objective on the right-hand side of equation 11 is

$$\begin{aligned}\mathrm{RHS}=&\,\mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t)\odot\mathbf{m}\|^2]\\&-2\mathbb{E}_{p(\mathbf{x}_t^{\mathrm{obs}},\mathbf{x}_0^{\mathrm{obs}})}[\langle\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t),\frac{\mathrm{d}\log p_t(\mathbf{x}_t^{\mathrm{obs}}\mid\mathbf{x}_0^{\mathrm{obs}})}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\rangle\odot\mathbf{m}]+C_3,\end{aligned}\tag{14}$$

where $C_3=\mathbb{E}_{p(\mathbf{x}_t^{\mathrm{obs}},\mathbf{x}_0^{\mathrm{obs}})}[\|\frac{\mathrm{d}\log p_t(\mathbf{x}_t^{\mathrm{obs}}|\mathbf{x}_0^{\mathrm{obs}})}{\mathrm{d}\mathbf{x}_t^{\mathrm{obs}}}\odot\mathbf{m}\|^2]+C$ is a constant that does not depend on $\boldsymbol{\theta}$.

Comparing equations equation 13 and equation 14, we thus show that the two optimization objectives are equivalent up to a constant. □

B Discussion with Related Works

B.1 Related Works that can be Used for Imputation together with Generation Tasks

In the following, we provide a detailed discussion about which work about learning from missing data can be used for imputation together with generation tasks.

- HI-VAE (Nazábal et al., 2018) and VAEM (Ma et al., 2020) can be used for generation since they model each data dimension by a VAE, albeit at a high computational cost.

- GAN-based approaches (Li et al., 2019; Li & Marlin, 2020) can also be used for generation tasks, while (Li et al., 2019) trains two generator-discriminator pairs for the masks and data respectively, which increases the computational cost and (Li & Marlin, 2020) adopts Partial Bidirectional GAN, which requires an encoding and decoding network for the generator. (Yoon et al., 2018a) can be used for generation without additional computational cost. However, there exists a mismatch between the training and inference for GAIN. And the smaller the missing ratio of the observed data, the larger the discrepancy will be.
- MIWAE (Mattei & Frellsen, 2019) and non-MIWAE (Ipsen et al., 2021) do not have additional computational costs, but they are not suited for generation tasks due to their use of a student t distribution in the decoder $p(x^{\text{obs}}|z)$, which has limited capacity to accurately represent real distributions. The experimental results of directly using MIWAE for generation can be found in Table 6, column MIWAE in Appendix C.5.
- CSDI_T (Zheng & Charoenphakdee, 2022) is the previous SOTA method that can be used for generation tasks. We compared with CSDI_T in all imputation and generation tasks and discuss the advantages of our method at the end of Section 4.1.

B.2 Discussion with corrupted data based method

Missing value belongs to a special case of data corruption. Ambient Diffusion (Daras et al., 2023) generally studies how to solve the linear inverse problem $\mathbf{y} = A\mathbf{x}$. When the corruption matrices $\mathbf{A}$ is a diagonal matrix where each $\mathbf{A}_{ii} \sim \text{Ber}(1-p)$, then this can be used for solving Independent Missing under MCAR mechanism. Under this setting, we prove the equivalence between Eq (3.1) in Daras et al. (2023) and Denoising Score Matching on Missing Data (Eq equation 4) in our paper as follows:

$$\begin{aligned} J_{\text{naive}}^{\text{corr}}(\boldsymbol{\theta}) &= \frac{1}{2}\mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_t,\mathbf{A})}\left\|\mathbf{A}\left(\boldsymbol{h}_{\boldsymbol{\theta}}\left(\mathbf{A},\mathbf{A}\mathbf{x}_t,t\right)-\mathbf{x}_0\right)\right\|^2 \\ &= \frac{1}{2}\mathbb{E}_{(\mathbf{x}^{\text{obs}}(0),\mathbf{x}^{\text{obs}}(t))}\left\|\left(\boldsymbol{h}_{\boldsymbol{\theta}}\left(\mathbf{A},\mathbf{x}^{\text{obs}}(t),t\right)-\mathbf{x}^{\text{obs}}(0)\right)\odot\mathbf{m}\right\|^2, \end{aligned}$$

where $\mathbf{x}^{\text{obs}}(0) = \mathbf{A}\mathbf{x}_0 = \mathbf{x}_0 \odot \mathbf{m}$, $\mathbf{m} = \mathbb{1}\{\mathbf{x}^{\text{obs}}(0) \neq \text{na}\}$ is the mask representing missing indexes, and $\mathbf{x}^{\text{obs}}(t) = \mathbf{A}\mathbf{x}_t = \mathbf{x}_t \odot \mathbf{m}$.

Our score-matching objective is

$$J_{DSM}(\boldsymbol{\theta}) = \frac{T}{2}\mathbb{E}_t\Big\{\lambda(t)\mathbb{E}_{\mathbf{x}^{\text{obs}}(0)}\mathbb{E}_{\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)}\left[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t),t)-\nabla_{\mathbf{x}^{\text{obs}}(t)}\log p_t(\mathbf{x}^{\text{obs}}(t)))\odot\mathbf{m}\|_2^2\right]\Big\}.$$

The equivalence between $J_{\text{naive}}^{\text{corr}}(\boldsymbol{\theta})$ and $J_{DSM}(\boldsymbol{\theta})$ can be built upon the equivalence of score predictor and data predictor. Specifically, Theorem B.1 in Zheng et al. (2023) proves that the optimal data predictor satisfies $\boldsymbol{h}_{\boldsymbol{\theta}^*}(\mathbf{x}_t,t) = \mathbf{x}_t + \sigma_t^2\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_t,t)$.

In the context of dealing with missing data, Ambient Diffusion is very similar to CSDI which learns the complete data distribution in a self-supervised learning manner. The essence of Ambient Diffusion lies in modeling the conditional distribution $\mathbb{E}[\mathbf{x}_0|\tilde{A}\mathbf{x}_t,\tilde{A}]$ (or $p(\mathbf{x}|y)$ for the inverse problem).

At the end of Section 4.1, we discussed the advantages of utilizing unconditional score matching over conditional score matching, as employed by CSDI_T, for both imputation and generation tasks, which can be summarized as follows:

- Ambient Diffusion masks additional data by using the corruption matrix $\tilde{\mathbf{A}}$ and using the data predictor $\boldsymbol{h}_{\boldsymbol{\theta}^*}$ to predict the known masked value. *MissDiff* does not need to mask additional data.
- Ambient Diffusion models the conditional distribution $p(\mathbf{x}|\mathbf{x}^{\text{obs}})$, where *MissDiff* exactly models $p(\mathbf{x})$. Therefore, when using Ambient Diffusion to generate new complete samples, there exists a mismatch between training and generation, since there is no information that Ambient Diffusion can condition for generation tasks. We demonstrate this mismatch makes the performance of CSDI_T worse than *MissDiff* in all of the experiments in generation tasks.
- We also demonstrate modeling the conditional distribution $p(\mathbf{x}|\mathbf{x}^{\text{obs}})$ is not good as modeling unconditional distribution $p(\mathbf{x})$ for tabular data. There may exist some complex or irregular dependencies between different columns in tabular data, e.g., some features might be redundant (uninformative). We demonstrate this phenomenon by the experimental comparison of *MissDiff* against CSDI_T.

C More Details on Experiments

C.1 Datasets for Imputation Task

We adopt the same missing mechanism as Zheng & Charoenphakdee (2022), i.e., MCAR with the missing ratio of 0.2. To be more precise, the detailed implementation of MCAR is the "Row Missing" defined in paragraph C.2. We also provide the comparisons of imputation results under MAR and NMAR assumptions in the Table 5. Our method still achieves a smaller Mean Squared Error than CSDI_T under MAR and NMAR settings.

Table 5: The effectiveness of *MissDiff* on imputation tasks under MAR and NMAR.

Method	MAR	NMAR
CSDI_T	0.1205(0.004)	0.1274(0.005)
MissDiff	**0.1053(0.005)**	**0.1092(0.006)**

C.2 Datasets for Generation Task

Details of the Bayesian Network Figure 2 demonstrates the Bayesian Network for generating the tabular data. It contains two continuous variables C1, C2, and three discrete random variables D1, D2, and D3. The distribution of these variables is set as follows. The marginal distribution of C1 is $\mathcal{N}(25, 2)$, the conditional distribition of C2 given C1 is $\mathrm{C2}|\mathrm{C1} \sim \mathcal{N}(0.1 \cdot \mathrm{C1} + 50, 5)$, and the marginal distribution of D1 is $Bernoulli(0.3)$, where $Bernoulli(\xi)$ stands for the Bernoulli distribution with mean equal to ξ. The conditional distribution of D2, given C1, C2 and D1, is set as

$$\mathrm{D2}|\mathrm{C1},\mathrm{C2},\mathrm{D1} \sim \begin{cases} Ca(0.3, 0.6, 0.1) & \mathrm{C1} > 26, \mathrm{C2} > 55, \mathrm{D1} = 1; \\ Ca(0.2, 0.3, 0.5) & \mathrm{C1} > 26, \mathrm{C2} \leq 55, \mathrm{D1} = 1; \\ Ca(0.7, 0.1, 0.2) & \mathrm{C1} \leq 26, \mathrm{C2} > 55, \mathrm{D1} = 1; \\ Ca(0.1, 0.2, 0.7) & \mathrm{C1} \leq 26, \mathrm{C2} \leq 55, \mathrm{D1} = 1; \\ Ca(0.05, 0.05, 0.9) & \mathrm{D1} = 0, \end{cases}$$

where $Ca(p1, p2, 1 - p1 - p2)$ denotes the categorical (discrete) distribution for three pre-specified categories. The conditional distribution of D3 given D2 is

$$\mathrm{D3}|\mathrm{D2} \sim \begin{cases} Bernoulli(0.2) & \mathrm{D2} = 0; \\ Bernoulli(0.4) & \mathrm{D2} = 1; \\ Bernoulli(0.8) & \mathrm{D2} = 2. \end{cases}$$

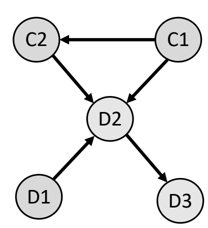


Figure 2: The demonstration of the Bayesian Network for generating the tabular data. "C1" and "C2" denote the continuous variables and "D1", "D2", "D3" denotes the discrete random variables. The marginal/conditional distributions for each node are detailed in Section C.2.

Choice of Masks under Different Missing Mechanisms To evaluate the performance of *MissDiff* on different missing mechanisms, we give a detailed explanation of the practical implementation of MCAR (Li et al., 2019; Yoon et al., 2018a), MAR(Ipsen et al., 2022; Li & Marlin, 2020), and NMAR (Muzellec et al., 2020; Ipsen et al., 2021).

- MCAR: there are three types of missing mechanisms in MCAR.
  - Row Missing. For a given missing ratio $\alpha \in (0,1)$, we have the number of elements missing in each row (i.e., for each sample $\mathbf{x}_i$) is $\lfloor d\alpha \rfloor$, where $\lfloor z \rfloor$ is the greatest integer less than z, and the location/index of the missing entries is randomly chosen according to the uniform distribution.
  - Column Missing. For a given missing ratio α, we have the number of elements missing in each column (for each feature) is $\lfloor n\alpha \rfloor$, and the location/index of the missing entries is randomly chosen according to the uniform distribution.
  - Independent Missing. Each entry in the table is masked missing according to the realization of a Bernoulli random variable with parameter α.
- MAR: a fixed subset of variables that cannot have missing values is first sampled. Then, the remaining variables will have missing values according to a logistic model with random weights, which takes the non-missing variables as inputs. The outcome of this logistic model is re-scaled to attain a given missing ratio α.
- NMAR: the same pipeline as MAR with the inputs of the logistic model are masked by the MCAR mechanism. We refer to Muzellec et al. (2020) for more detailed explanations.

Remark C.1. Under the three missing mechanisms in MCAR, with the missing ratio parameter set as $0 < \alpha < 1$, the condition in Theorem 3.2 can be satisfied with probability at least $1 - \delta$, where $\delta = \max\{(\frac{\alpha d - 1}{d})^n d, \alpha, \alpha^n d\}$ and it will be sufficiently small when α is small and n is sufficiently large.

Remark C.1 gives the guarantee that *MissDiff* can recover the oracle score under MCAR with high probability. For the data generated by the Bayesian Network in Section 4.3, there are only five variables (columns) (three categorical variables and two continuous variables). Therefore, in the row missing mechanism, we only have the missing ratio [0.2,0.4,0.6,0.8]. For the column missing or the independent missing mechanisms, we set the missing ratio to be [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]. In other datasets in Section 4.3, we adopt the missing ratio $\alpha = 0.2$ and XGBoost for the downstream tasks as the default setting. More experimental results can be found in Appendix C.6.

C.3 Algorithms for Imputation and Generation Tasks

MissDiff adopts the algorithm 2 for imputation task and algorithm 3 for generating new complete data. For the imputation, the key operation is in line 9. The element-wise multiplication guarantees the output $\mathbf{x}_0$ has the same value as $\mathbf{x}_{\text{obs}}$ in the observed entries. Therefore, in each iteration, the noising version of the observed data is used as the guidance.

Algorithm 2 *MissDiff* for Imputation

Require: Observed data $\mathbf{x}_0^{\text{obs}}$, Diffusion model $\mathbf{s}_{\boldsymbol{\theta}}$, hyperparameter β_t, σ_t, denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

1: Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$;
2: Infer mask $\mathbf{m} = \mathbb{1}[\mathbf{x}_0^{\text{obs}} \neq \text{na}]$;
3: $t = T$;
4: **while** $t \neq 0$ **do**
5: Sample $\epsilon_t^{\text{obs}} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ if $t > 1$, else $\epsilon_t^{\text{obs}} = \mathbf{0}$;
6: $\mathbf{x}_{t-1}^{\text{obs}} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0^{\text{obs}} + (1 - \bar{\alpha}_{t-1})\,\epsilon_t^{\text{obs}}$
7: Sample $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ if $t > 1$, else $\epsilon_t = \mathbf{0}$;
8: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)) + \sigma_t \epsilon_t$;
9: $\mathbf{x}_{t-1} = \mathbf{m} \odot \mathbf{x}_{t-1}^{\text{obs}} + (\mathbf{1} - \mathbf{m}) \odot \mathbf{x}_{t-1}$
10: $t = t - 1$;
11: **end while**
12: **return** $\mathbf{x}_0$.

Algorithm 3 *MissDiff* for Generation

Require: Diffusion model $\mathbf{s}_{\boldsymbol{\theta}}$, hyperparameter β_t, σ_t, denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.
1: Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$;
2: $t = T$;
3: **while** $t \neq 0$ **do**
4: Sample $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ if $t > 1$, else $\epsilon_t = \mathbf{0}$;
5: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)) + \sigma_t \epsilon_t$;
6: $t = t - 1$;
7: **end while**
8: **return** $\mathbf{x}_0$.

C.4 Implementation Details

To make the transition $p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))$ and the gradient $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t) \mid \mathbf{x}^{\text{obs}}(0))$ well defined for the mixed-type data, we use 0 to replace na for continuous variables and a new category to represent na for discrete variables, which is the same operation as in Nazábal et al. (2018); Ma et al. (2020) that can help to feed fixed dimensional data into neural networks. One-hot embedding is applied to discrete variables.

We set the minimum noise level $\beta_1 = 0.0001$ and the maximum noise level $\beta_T = 0.5$ in Algorithm 1 and Algorithm 3 with quadratic schedule

$$\beta_t = \left(\frac{T-t}{T-1}\sqrt{\beta_1} + \frac{t-1}{T-1}\sqrt{\beta_T}\right)^2.$$

We mainly follow the hyperparameter in the previous works that train the diffusion model on tabular data Tashiro et al. (2021); Zheng & Charoenphakdee (2022). We use the Adam optimizer with MultiStepLR with 0.1 decay at $25\%, 50\%, 75\%$, and 90% of the total epochs and with an initial learning rate as 0.0005.

With regard to the baselines of STaSy, we adopt the same setting of its open resource implementation [10], i.e., Variance Exploding SDE with six layers ConcatSquash network as the backbone of the diffusion model and Fourier embedding, the adam optimizer with learning rate as 2e-03, training with batch size 64 and 250 epochs/1000 epochs with additional 50 finetuning epochs.

For the downstream classifier/regressor, we adopt the same base hyperparameters in [Kim et al. (2023), Table 26].

C.5 Additional Discussion for Generation Results

In this section, we provide more discussion on the experimental results of generation tasks.

Discussion 1: the performance of *MissDiff* as the missing ratio in range (0.1-0.6) In "Row missing" and "Column missing" in Figure 1, we can see the performance of *MissDiff* slightly increase when the missing rate increase in range (0.1-0.6). we conjecture that this is a phenomenon due to the unique structure of certain tabular datasets. For this simulated Bayesian network dataset, the dependencies between different columns are demonstrated in Figure 2. Some features might be uninformative, for instance, variables C1, C2, and D1 are all uninformative to the value of D3, given that D2 is observed. This implies that for some rows with missing C1, C2, and D3 values, the model still has enough information to learn the full dependence between variables D3 and D2. Moreover, the model can potentially learn the distribution of D3$|$D2 better in such cases since other redundant variables are excluded. Moreover, the performance starts to decrease when we increase the missing rate to 0.8, since in such case, we only have one variable left in each row and thus it is reasonable to expect worse performance.

Discussion 2: the performance of *MissDiff* in "Column missing" scenario in Census dataset In Table 2, *MissDiff* does not perform well on the "Column missing" scenario in the Census dataset. We believe the column missing mechanism described in Appendix C.2 is a special scenario. Most

[10] `https://openreview.net/forum?id=1mNssCWt_v`

specifically, the mask $\mathbf{m}$ (an indicator of missing values) for each row (sample) would depend on the masks of other rows as well, since the missing rate for each column is fixed. It leads to dependence between missing samples. We further note that in our population objective function Eq equation 4, as a standard practice, we regard the sample pair $(\mathbf{m}, \mathbf{x})$ are iid and the expectation in Eq equation 4 is taken with respect to this joint distribution. When the sample size of the dataset is relatively small, such sample dependence is more evident, and *MissDiff* is not as good as *Diff-mean*.

Discussion 3: the performance of MIWAE in Census dataset MIWAE models the distribution $p(x^{\text{obs}}|z)$ by a student t distribution with location, scale, and degrees of freedom outputted by the decoder, which has limited representation power for the real distribution. Directly using this learned distribution to generate samples has poor performance demonstrated in Table 6. A possible solution is using the "generate-then-impute" framework, i.e., randomly removing different values in observed data and then applying the learned model to impute the missing data. We refer to this method as MIWAE (modified) in the following table. *MissDiff* still achieves better results compared to other approaches together with the "generate-then-impute" framework.

Table 6: Comparison with MIWAE and "generate-then-impute" framework on Census dataset. "-" denotes the corresponding method cannot applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data. The *larger* the accuracy, the *better* the performance.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*	CSDI_T	MIWAE	MIWAE (modified)
Utility evaluation	**79.48**%	-	78.45%	-	70.79%	79.15%	23.7%	72.11%
Fidelity evaluation	**80.59**%	-	76.92%	-	56.75%	77.60%	59.11%	67.14%

C.6 Additional Experiential Results

C.6.1 Additional Results for Fidelity Evaluation

Table 7, 8, and 9 provide SDMetrics metric evaluation on *MissDiff*. They correspond to Table 2, 3, and 4 in Section 4.3.2.

Table 7: *Fidelity* evaluation of *MissDiff* on Census dataset. The *larger* the score, the *better* the overall quality of synthetic data is.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*	CSDI_T
Row Missing	**80.59**%	-	76.92%	-	56.75%	77.60%
Column Missing	**82.70**%	75.03%	76.17%	56.90%	51.54%	73.84%
Independent Missing	**83.16**%	74.94%	76.60%	56.07%	57.06%	82.56%

Table 8: *Fidelity* evaluation of *MissDiff* on MIMIC4ED dataset. *Diff-delete* and *STaSy-delete* cannot be applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data.

	MissDiff	*Diff-mean*	*STaSy-mean*	*CSDI_T*
Row Missing	**84.45**%	75.22%	82.94%	83.15%
Column Missing	**79.24**%	76.57%	79.03%	79.10%
Independent Missing	**78.01**%	76.16%	77.21%	77.53%

Table 9: *Fidelity* evaluation of *MissDiff* on Census dataset under MAR, NMAR.

	MissDiff	Diff-delete	Diff-mean	*STaSy-delete*	*STaSy-mean*	CSDI_T
MAR	77.45%	73.78%	76.08%	57.51%	50.06%	**78.14**%
NMAR	**77.88**%	75.72%	76.97%	54.11%	50.6%	77.51%

C.6.2 Additional Results of Other Criteria for *Utility* Evaluation

Table 10, 11, and 12 provide the additional experimental results for other criteria under *Utility* evaluation for Table 2, 3, and 4 in the main paper, i.e., the F1, Weighted-F1, AUROC for the classification task and R^2 for the regression task. A detailed explanation of the above-mentioned criteria can be found in Kim et al. (2023). To make our paper self-contained, we briefly restate it here.

1. Binary F1 for binary classification: sklearn.metrics.f1_score with 'average'='binary'.
2. Macro F1 for multi-class classification: sklearn.metrics.f1_score with 'average'='macro'.
3. Weighted-F1: $= \sum_{i=0}^{K} w_i s_i$, where K denotes the number of classes, the weight of i-th class w_i is $\frac{1-p_i}{K-1}$, p_i is the proportion of i-th class's cardinality in the whole dataset, and score s_i is a per-class F1 of i-th class (in a One-vs-Rest manner).
4. AUROC: sklearn.metrics.roc_auc_score.

From the results in Table 10, 11, and 12, it can be seen that the proposed *MissDiff* consistently outperforms the compared methods in most instances. For the column missing case, *MissDiff* tends to perform worse, which indicates the potential limitations of the proposed method for future investigations.

Table 10: *Utility* evaluation of *MissDiff* on Census dataset with other criteria.

Criterion	Missing Mechanism	*MissDiff*	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Binary F1	Row Missing	**0.344**	-	0.280	-	0.314
	Column Missing	0.141	0.063	0.413	**0.509**	0.383
	Independent Missing	**0.291**	0.045	0.225	0.274	0.241
Weighted-F1	Row Missing	0.470	-	0.423	-	**0.488**
	Column Missing	0.305	0.249	0.523	**0.571**	0.490
	Independent Missing	**0.431**	0.237	0.375	0.416	0.389
AUROC	Row Missing	**0.772**	-	0.685	-	0.731
	Column Missing	0.539	0.469	**0.757**	0.750	0.637
	Independent Missing	**0.650**	0.474	0.655	0.621	0.613

Table 11: *Utility* evaluation of *MissDiff* on MIMIC4ED dataset with R^2 criterion. *Diff-delete* and *STaSy-delete* cannot be applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data.

Missing mechanism	*MissDiff*	*Diff-mean*	*STaSy-mean*
Row Missing	**0.088**	0.057	0.067
Column Missing	**0.095**	0.023	0.073
Independent Missing	**0.156**	0.062	0.142

Table 12: *Utility* evaluation of *MissDiff* on Census dataset under MAR, NMAR with other criteria.

Criterion	Missing Mechanism	*MissDiff*	*Diff-delete*	*Diff-mean*
Binary F1	MAR	0.346	0.108	0.224
	NMAR	**0.464**	0.233	0.383
Weighted-F1	MAR	**0.473**	0.276	0.376
	NMAR	**0.564**	0.364	0.501
AUROC	MAR	**0.833**	0.441	0.774
	NMAR	**0.834**	0.499	0.746

C.6.3 Experiment Results for Different Classifiers/Regressors

As mentioned in Section 4.2, we train various models, including Decision Tree, AdaBoost, Logistic/Linear Regression, MLP classifier/regressor, RandomForest, and XGBoost, on synthetic data.

Table 13 to 17 present the corresponding results on different classifiers/regressors, from which we can see that *MissDiff* still performs well under most cases.

Table 13: *Utility* evaluation of *MissDiff* on Census dataset by Decision Tree.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**78.08**%	-	74.55%	-	60.74%
Column Missing	62.65%	69.10%	**78.88**%	65.38%	66.31%
independent	**80.68**%	72.68%	67.70%	76.35%	55.99%

Table 14: *Utility* evaluation of *MissDiff* on Census dataset by AdaBoost.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**80.38**%	-	79.28%	-	73.23%
Column Missing	72.18%	76.30%	**80.65**%	69.60%	42.24%
independent	**78.70**%	76.13%	75.96%	76.55%	78.39%

Table 15: *Utility* evaluation of *MissDiff* on Census dataset by Logistic Regression.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**79.20**%	-	77.08%	-	71.04%
Column Missing	73.50%	76.30%	**77.45**%	66.91%	69.08%
independent	76.20%	**76.30**%	76.25%	77.13%	69.68%

Table 16: *Utility* evaluation of *MissDiff* on Census dataset by Multi-layer Perceptron (MLP).

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**77.70**%	-	75.13%	-	49.78%
Column Missing	68.33%	65.75%	**75.00**%	70.97%	58.83%
independent	**75.33**%	72.18%	74.30%	76.81%	37.59%

Table 17: *Utility* evaluation of *MissDiff* on Census dataset by Random Forest.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**80.10**%	-	77.13%	-	72.68%
Column Missing	73.68%	76.33%	**79.88**%	74.70%	71.58%
independent	**79.33**%	76.30%	76.38%	76.31%	76.98%

C.6.4 Additional Results for *STaSy-delete* and *STaSy-mean*

The experimental results of *STaSy-delete* and *STaSy-mean* in Tables 2 and 7 are obtained by training diffusion model for 1000 epochs, compared with 250 epochs of *MissDiff*, *Diff-delete*, and *Diff-mean*. If we train *STaSy-delete* and *STaSy-mean* as the same training epochs (250 epochs) on the Census dataset under MCAR as *MissDiff*, their performance is demonstrated in Table 18 and 19. This observation highlights that the proposed *MissDiff* requires considerably fewer training epochs compared to STaSy in order to achieve satisfactory results when handling data with missing values.

Table 18: *Fidelity* evaluation of *MissDiff* on Census dataset with 250 training epochs.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**80.59**%	-	76.92%	-	50.08%
Column Missing	**82.70**%	75.03%	76.17%	52.49%	49.63%
independent	**83.16**%	74.94%	76.60%	53.7%	50.11%

Table 19: *Utility* evaluation of *MissDiff* on Census dataset with 250 training epochs.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**79.48**%	-	78.45%	-	60.96%
Column Missing	71.68%	72.89%	**79.60**%	56.19%	61.46%
independent	**79.49**%	75.39%	75.96%	49.78%	70.68%