MissDiff: Training Diffusion Models on Tabular Data with Missing Values

Yidong Ouyang[1] **Liyan Xie**[1] **Chongxuan Li**[2] **Guang Cheng**[3]

Abstract

Diffusion models have shown remarkable performance in modeling data distributions and synthesizing data. The vanilla diffusion model typically requires complete or fully observed training data, while incomplete data is a common issue in various real-world applications, particularly in tabular data. This work presents a unified and principled diffusion-based framework for learning from data with missing values under various missing mechanisms. We first observe that the widely adopted "impute-then-generate" pipeline may lead to a biased learning objective. Then we propose to mask the regression loss of Denoising Score Matching in the training phase. We show that the proposed method is consistent in learning the score of data distributions, and the training objective serves as an upper bound for the negative likelihood in certain cases. The proposed framework is evaluated on multiple tabular datasets using realistic and efficacious metrics. It is demonstrated to outperform several baseline methods by a large margin.

1. Introduction

Diffusion models have emerged as an effective tool for modeling data distributions and synthesizing various types of data, including 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). It is known that such 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 sampling cost, etc (Yoon et al., 2018a;b).

In this work, we focus on learning a generative model from training data containing a significant amount of missing values, a problem that has been largely overlooked in the literature despite its widespread practical applications. Deep generative models, particularly diffusion models, can be used to augment training data and enhance the performance of image classification tasks (Azizi et al., 2023; You et al., 2023) and adversarial robustness (Gowal et al., 2021; Sehwag et al., 2022; Ouyang et al., 2022). Following this idea, we can achieve better performance for downstream tasks by utilizing the generative model learned on incomplete data for synthetic data generation. We will primarily take tabular data as examples, 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, the mixed-type property of tabular data will also be considered in the numerical experiments.

To deal with missing values in the training data, numerous studies propose to use various imputation methods and then train the model on the imputed data. Taking tabular data as an example, some approaches simply delete instances (rows) with missing data or replace missing values with mean imputation. 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., 2020a; Muzellec et al., 2020). It has been shown that imputation may reduce the diversity of the training data and thus lead to biased downstream performances (Bertsimas et al., 2021; Ipsen et al., 2020a).

In addition to imputation or simple deletion methods, previous work also studied learning from data with missing values and synthesizing complete data using GAN or VAE architectures (Li et al., 2019; Li & Marlin, 2020; Neves et al., 2022). Compared with our proposed framework, these methods involve training additional networks and impose certain assumptions on the missing mechanisms, and the unique challenges associated with tabular data are less investigated.

In this work, we propose a diffusion-based framework, which we call *MissDiff*, for generative model training from data with missing values. We present the theoretical justifications of *MissDiff* on recovering the oracle score function and upper bounding the negative likelihood on the data un-

[1]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China [2]Gaoling School of AI, Renmin University of China, China [3]Department of Statistics, University of California, Los Angeles, USA. Correspondence to: Liyan Xie <xieliyan@cuhk.edu.cn>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling 2023. Copyright 2023 by the author(s).

der mild assumptions on the missing mechanisms. To the best of our knowledge, this is the first work that learns a generative model from *mixed-type* data containing missing values, and the missing values are used *directly* in the training process without prior imputations. Finally, we conduct a suite of numerical experiments on mixed-type tabular data, comprising both continuous and categorical variables, under various missing mechanisms. Evaluated under several realistic and efficacious metrics, *MissDiff* consistently outperforms other baseline methods by a considerable margin.

2. Method

2.1. Problem Setup: 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., 2020a), 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, $i \in [d]$, can be either categorical or continuous.

For each data point $\mathbf{x}$, suppose there is a binary mask $\mathbf{m} = (m_1, \ldots, m_d) \in \{0, 1\}^d$ indicating the missing entries in $\mathbf{x}$, i.e., $m_i = 1$ if x_i is observed, and $m_i = 0$ if x_i is missing. Then, the observed data $\mathbf{x}^{\text{obs}} = \mathbf{x} \odot \mathbf{m} + \mathrm{na} \odot (\mathbf{1} - \mathbf{m})$, where na indicates the missing value[1], $\odot$ denotes element-wise multiplication, and $\mathbf{1}$ is the all-one vector.

Suppose we have n complete (unobservable) training 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 in the Appendix D.1. Then the observed data values are $S^{\text{obs}} = \{\mathbf{x}_i^{\text{obs}}\}_{i=1}^n$ with $\mathbf{x}_i^{\text{obs}} = \mathbf{x}_i \odot \mathbf{m}_i + \mathrm{na} \odot (\mathbf{1} - \mathbf{m}_i)$.

Our objective is to train a generative model p_ϕ, parametrized by the neural network parameters ϕ, using the observed data S^{obs}, such that p_ϕ is close to the true distribution $p_0(\mathbf{x})$ and we can efficiently generate synthetic data from p_ϕ. In the following, we mainly consider the score-based generative model as p_ϕ.

2.2. Preliminaries: Score-Based Generative Model

In this work, we adopt the diffusion model as the prototype for our proposed method. We first briefly review the key components of score-based generative models (Ho et al., 2020; Song et al., 2021b). 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

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

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 (1). For any SDE in (1), the corresponding backward/reverse process is

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

where $\overline{\mathbf{w}}$ is a standard Wiener process when time flows backwards from T to 0.

We can generate new data by running backward the reverse-time SDE (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 true score, with consistency guarantees (Hyvärinen, 2005; Vincent, 2011; Song et al., 2019).

2.3. Proposed Method: *MissDiff*

In general, one common approach to learning a generative model from incomplete data is to construct a complete training data set first and then learn a generative model on the complete data. We can either delete instances with missing data or adopt "inpute-then-generative" paradigm, i.e., we complete the data by imputation methods (van Buuren & Groothuis-Oudshoorn, 2011; Bertsimas et al., 2017; Vincent et al., 2008; Yoon et al., 2018a; Biessmann et al., 2019; Wang et al., 2020; Ipsen et al., 2020a; Muzellec et al., 2020). It is noted that such pipeline may bring bias to the training objective, as commented in the following remark 2.1.

Remark 2.1 (Challenges with "inpute-then-generative" paradigm). Following the analysis of "inpute-then-regress" (Bertsimas et al., 2021; Ipsen et al., 2020a) for the prediction task, we can study a similar framework for the generation task. The generative model p_ϕ represents the

[1] The implementation details for na can be found in Section 2.

probability distribution of the synthetic data $\mathbf{x}$. When data has missing values, the "impute-then-generate" approach will first impute the observed data $\mathbf{x}^{\text{obs}}$ using an imputation model f_φ. Then, the generative model is trained on imputed data, i.e., $(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}} := f_\varphi(\mathbf{x}^{\text{obs}}))$ and with the special case of f_φ being the mean imputation, it becomes $(\mathbf{x}^{\text{obs}}, \mathbb{E}_{p_\phi(\mathbf{x}^{\text{miss}}|\mathbf{x}^{\text{obs}})}[\mathbf{x}^{\text{miss}}])$. Such pipeline is typically biased because with single imputation, the conditional distribution over the missing data is discarded, and the optimal single imputation can no longer capture the data variability, i.e., the distribution of the imputed data is different from the true data distribution.

Motivated by the above observation, we intend to train the model parameters ϕ by maximizing the data likelihood directly without imputation as the first step. Therefore, we propose *MissDiff*, a diffusion-based framework for learning on missing data. This approach incorporates the uncertainty of missing data directly into the learning process.

We propose the following Denoising Score Matching method for data with missing values. In *MissDiff*, the score model $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)$ is learned as solution to

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} J_{DSM}(\boldsymbol{\theta}) \\ &:= \frac{T}{2}\mathbb{E}_t\Big\{\lambda(t)\mathbb{E}_{p(\mathbf{x}^{\text{obs}}(0),\mathbf{m})}\mathbb{E}_{p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))} \\ &\Big\|\Big(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\text{obs}}(t), t) - \nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))\Big) \odot \mathbf{m}\Big\|_2^2\Big\},\end{aligned} \tag{3}$$

where $\lambda(t)$ is a positive weighting function, $\mathbf{m} = \mathbb{1}\{\mathbf{x}^{\text{obs}}(0) = \text{na}\}$ indicates the missing entries in $\mathbf{x}^{\text{obs}}(0)$ 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. To make $p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))$ and $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0))$ 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 (Nazábal et al., 2018; Ma et al., 2020). One-hot embedding is applied to discrete variables. More implementation details can be found in Appendix D.2.

We mainly adopt the Variance Preserving (VP) SDE in this paper although Variance Exploding (VE) SDE (Song et al., 2021b) is also applicable. Algorithm 1 and Algorithm 2 in Appendix C demonstrate the Denoising Score Matching objective on missing data and the sampling procedure.

2.4. Theoretical Results

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 state 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 2.2. *Denote ρ_i, $i \in \{1, 2, ..., d\}$ as the percentage of missing samples for the i-th entry in the training data. Suppose $\max_{i=1,...,d} \rho_i < 1$. Let $\boldsymbol{\theta}^*$ be the solution to the training objective of MissDiff defined in Eq (3). Then under mild conditions, we have*

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

Theorem 2.2 states that the global optimal solution of Denoising Score Matching on missing data obtained by *MissDiff* is the same as the oracle score. Detailed assumptions and the proof sketch can be found in Appendix E.1.

It is 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 2.3. *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, if $\lambda_t = \beta_t$ and ρ_i defined in Theorem 2.2, under mild regularity conditions detailed in Appendix E.2, we have*

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

where C_1 is a constant and $\rho_{\max} = \max_{i\in[d]} \rho_i$.

The proof sketch can be found in Appendix E.2.

3. Experiments

In this section, we demonstrate the effectiveness of the proposed method *MissDiff* using simulations and two real-world tabular datasets. We compare the proposed method with several baseline methods for synthetic data generation training on data with missing values. The *baseline methods* are described as follows, and the training details can be found in Appendix D.1.

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.

Evaluation Criterion Following (Xu et al., 2019; Kim et al., 2023; Kotelnikov et al., 2022), we use two types of criterion, *fidelity* and *utility*, to evaluate the quality of the synthetic data generated. To evaluate the *fidelity* of synthetic data, we adopt a model-agnostic library, SDMetrics (Dat, 2023). To evaluate the *utility*, we follow the same pipeline of (Kim et al., 2023), i.e., training various models, including Decision Tree, AdaBoost, Logistic 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 D.3. For regression tasks, we mainly use the Root Mean Squared Error (RMSE) and also report R^2 in the Appendix D.3. All the experiment results are obtained from 3 repetitions.

Experiment Results: Simulation Study 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 when the missing ratio varies from 0.1 to 0.9. These results align with the observation in Remark 2.1 that the learning objective of impute-then-generate may be biased. Directly learning on the missing data can significantly enhance the performance of the learned generative model. The detailed explanations of missing mechanisms, e.g., “Row Missing”, “Column Missing”, and “Indep (Independent) Missing” can be found in appendix D.1.

Experiment Results: Real Tabular Datasets Table 1 and 2 demonstrate the effectiveness of *MissDiff* on the Census dataset under Missing Completely At Random (MCAR)[2]. More details about the Census dataset can be found in appendix D.1. *MissDiff* performs better than STaSy when learning on incomplete data, and we believe the performance of *MissDiff* can be further improved by adopting the self-paced learning technique and the fine-tuning strategy, which is left as future work. Moreover, the results of *STaSy-delete* and *STaSy-mean* in Tables 1 and 2 are obtained by training diffusion model for 1000 epochs, compared with 250 epochs of *MissDiff*, *Diff-delete*, and *Diff-mean*. If we reduce the training epochs of *STaSy-delete* and *STaSy-mean* to 250 epochs, the performance will degrade significantly, which can be found in Appendix D.3.3. Compared with the state-of-the-art performance of STaSy for tabular data generation, its relatively worse performance here indicates that STaSy could be susceptible to missing data.

Table 3 and 4 show the performance of *MissDiff* on the MIMIC4ED dataset under MCAR. On this large dataset with dozens of continuous and discrete variables, *MissDiff* yields consistently better performance with the same training epochs (250 epochs).

Table 1. Fidelity 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 score, the *better* the overall quality of synthetic data is.

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

Table 2. Utility (classification accuracy) evaluation of *MissDiff* on Census dataset. The *larger* the accuracy, the *better* the performance.

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

Table 3. Fidelity evaluation of *MissDiff* on MIMIC4ED dataset.

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

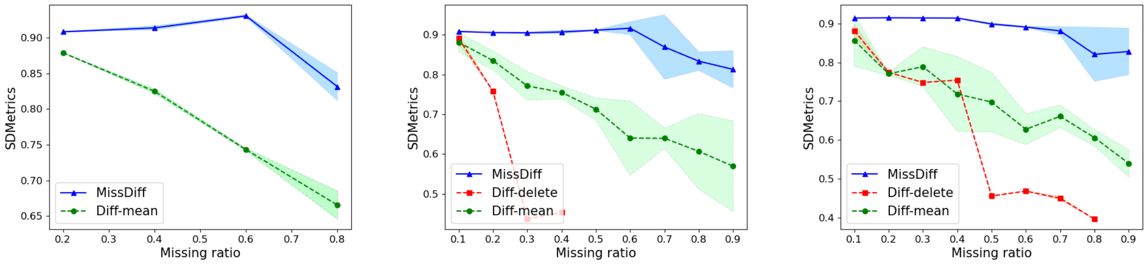
Table 4. Utility (RMSE) evaluation of *MissDiff* on MIMIC4ED dataset. The *lower* the RMSE, the *better* the performance.

	MissDiff	*Diff-delete*	*Diff-mean*	*STaSy-delete*	*STaSy-mean*
Row Missing	**1.826**	-	2.166	-	1.894
Rolumn Missing	**1.834**	-	2.011	-	1.935
Independent Missing	**1.852**	-	2.483	-	1.972

Ablation Study Table 5 and 6 demonstrate 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 MCAR, which has not been fully studied in previous methods (Li et al., 2019; Ipsen et al., 2020a; Yoon et al., 2018a; Li & Marlin, 2020).

Furthermore, we validate the performance of the propose *MissDiff* on imputation tasks for the Census dataset.

[2] More detailed explanation about the missing mechanisms MCAR, MAR, and NMAR can be found in Appendix B and D.1.



Row Missing

Column Missing

Indep Missing

Figure 1. Fidelity evaluation of *MissDiff* on data simulated from a pre-specified Bayesian Network under different missing ratios. We shade the area between mean ± std.

Table 5. Fidelity evaluation of *MissDiff* on Census dataset under MAR, NMAR with missing ratio 0.2.

	MissDiff	*Diff-delete*	*Diff-mean*
MAR	**77.45%**	73.78%	76.08%
NMAR	**77.88%**	75.72%	76.97%

Table 6. Utility (classification accuracy) evaluation of *MissDiff* on Census dataset under MAR, NMAR.

	MissDiff	*Diff-delete*	*Diff-mean*
MAR	**79.95%**	69.475%	77.425%
NMAR	**80.95%**	66.5%	80.025%

We compare *MissDiff* with state-of-the-art imputation approaches in Table 7. The result shows that although designed for generation tasks, *MissDiff* also performs well for imputation tasks.

Table 7. Imputation result comparisons on the Census dataset. The *lower* the RMSE, the *better* the performance.

Method	RMSE
Mean /Mode	0.120
MICE(linear)(van Buuren & Groothuis-Oudshoorn, 2011)	0.101
MissForest (Stekhoven, 2015)	0.112
GAIN(Yoon et al., 2018a)	0.123
CSDI_T (Zheng & Charoenphakdee, 2022)	0.099
MissDiff	**0.087**

4. Conclusion and Discussion

We propose a diffusion-based generative framework, called *MissDiff*, for synthetic data generation trained on data with missing values directly. *MissDiff* offers a promising alternative that directly handles missing data without the need for imputation or deletion. For future directions, we may refine the theoretical justifications for *MissDiff* and compare it with more baselines. We may also further study the downstream task performances with *MissDiff*.

5. Acknowledgement

Yidong Ouyang is partially supported by the Shenzhen Research Institute of Big Data PhD Fellowship. Yidong Ouyang and Liyan Xie are partially supported by UDF01002142 through The Chinese University of Hong Kong, Shenzhen. Guang Cheng is partially supported by Office of Naval Research, ONR (N00014-22-1-2680), NSF – SCALE MoDL (2134209), and Meta gift fund.

References

Alaa, A. M., Yoon, J., Hu, S., and van der Schaar, M. Personalized risk scoring for critical care prognosis using mixtures of gaussian processes. *IEEE Transactions on Biomedical Engineering*, 65:207–218, 2016.

Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J. Synthetic data from diffusion models improves imagenet classification. *ArXiv*, abs/2304.08466, 2023.

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

Bertsimas, D., Delarue, A., and Pauphilet, J. Prediction with missing data. *ArXiv*, abs/2104.03158, 2021.

Biessmann, F., Rukat, T., Schmidt, P., Naidu, P., Schelter, S., Taptunov, A., Lange, D., and Salinas, D. Datawig: Missing value imputation for tables. *J. Mach. Learn. Res.*, 20:175:1–175:6, 2019.

Choi, E., Biswal, S., Malin, B. A., Duke, J. D., Stewart, W. F., and Sun, J. Generating multi-label discrete electronic health records using generative adversarial networks. *ArXiv*, abs/1703.06490, 2017.

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

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. How to train your neural ode: the world of jacobian and kinetic regularization. In *International Conference on Machine Learning*, 2020.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. In *NeurIPS*, 2021.

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

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *ArXiv*, abs/2204.03458, 2022.

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

Ipsen, N. B., Mattei, P.-A., and Frellsen, J. How to deal with missing data in supervised deep learning? In *International Conference on Learning Representations*, 2020a.

Ipsen, N. B., Mattei, P.-A., and Frellsen, J. not-miwae: Deep generative modelling with missing not at random data. *ArXiv*, abs/2006.12871, 2020b.

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

Kim, J., Lee, C. E., and Park, N. Stasy: Score-based tabular data synthesis. 2023.

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

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: Modelling tabular data with diffusion models. *ArXiv*, abs/2209.15421, 2022.

Li, S. C.-X. and Marlin, B. M. Learning from irregularly-sampled time series: A missing data perspective. In *International Conference on Machine Learning*, 2020.

Li, S. C.-X., Jiang, B., and Marlin, B. M. Misgan: Learning from incomplete data with generative adversarial networks. *ArXiv*, abs/1902.09599, 2019.

Little, R. J. A. and Rubin, D. B. Statistical analysis with missing data. 1988.

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

Ma, C., Tschiatschek, S., Hernández-Lobato, J. M., Turner, R. E., and Zhang, C. VAEM: a deep generative model for heterogeneous mixed type data. *ArXiv*, abs/2006.11941, 2020.

Mattei, P.-A. and Frellsen, J. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, 2019.

Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_RL7wtHkPJK.

Muzellec, B., Josse, J., Boyer, C., and Cuturi, M. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, 2020.

Nazábal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. Handling incomplete heterogeneous data using vaes. *Pattern Recognit.*, 107:107501, 2018.

Neves, D., Alves, J., Naik, M. G., Proença, A. J., and Prasser, F. From missing data imputation to data generation. *J. Comput. Sci.*, 61:101640, 2022.

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

Ouyang, Y., Xie, L., and Cheng, G. Improving adversarial robustness by contrastive guided diffusion process. *ArXiv*, abs/2210.09643, 2022.

Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., and Kim, Y. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11:1071–1083, 2018.

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

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *ICLR*, 2022.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. In *Neural Information Processing Systems*, 2021a.

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021b.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, 2017.

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

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

Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *ArXiv*, abs/2107.03502, 2021.

Valera, I., Pradier, M. F., Lomeli, M., and Ghahramani, Z. General latent feature models for heterogeneous datasets. *J. Mach. Learn. Res.*, 21:100:1–100:49, 2017.

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

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

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, 2008.

Wang, Y., Li, D., Li, X., and Yang, M. 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.

Xie, F., Zhou, J., Lee, J. W., Tan, M., Li, S., Rajnthern, L. S., Chee, M. L., Chakraborty, B., Wong, A., Dagan, A., Ong, M. E. H., Gao, F., and Liu, N. Benchmarking emergency department prediction models with machine learning and public electronic health records. *Scientific Data*, 9, 2022.

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

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

Yoon, J., Jordon, J., and van der Schaar, M. Gain: Missing data imputation using generative adversarial nets. *ArXiv*, abs/1806.02920, 2018a.

Yoon, J., Zame, W. R., Banerjee, A., Cadeiras, M., Alaa, A. M., and van der Schaar, M. Personalized survival predictions via trees of predictors: An application to cardiac transplantation. *PLoS ONE*, 13, 2018b.

You, Z., Zhong, Y., Bao, F., Sun, J., Li, C., and Zhu, J. Diffusion models and semi-supervised learners benefit mutually with few labels. *ArXiv*, abs/2302.10586, 2023.

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

A. More on Related Work

One line of research focuses on different types of generative models trained directly on data with missing values. These studies carefully modify the architecture and training objectives of Generative Adversarial Network (GAN) or Variational Autoencoder (VAE) to learn from incomplete data (Li et al., 2019; Li & Marlin, 2020; Ipsen et al., 2020a).

Another research direction explores learning generative models for imputing missing values in observed data (Yoon et al., 2018a; Neves et al., 2022; Ipsen et al., 2020b; Muzellec et al., 2020; Tashiro et al., 2021; Nazábal et al., 2018; Ma et al., 2020; Mattei & Frellsen, 2019; Valera et al., 2017). For example, (Tashiro et al., 2021) proposes the conditional score-based generative model for time series imputation. Moreover, (Zheng & Charoenphakdee, 2022) adapt the conditional score-based diffusion model proposed in (Tashiro et al., 2021) for imputing tabular data. Imputation methods cannot easily used for generating new complete data, which is the main difference with the first line of works.

Tabular data, as a mixed-type data that typically contains both categorical and continuous variables, has attracted significant attention in the field of machine learning. Tabular data synthesis has been a long-standing research topic in this area. 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-based models have been proposed for generating tabular data (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 random 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.

B. Missing Mechanism Examples

The missing mechanisms can be categorized based on the relationships between the mask $\mathbf{m}$ and the complete data $\mathbf{x}$ (Little & Rubin, 1988):

- Missing Completely At Random (MCAR): mask $\mathbf{m}$ is independent with the completed 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.

Unlike (Li et al., 2019; Ipsen et al., 2020a; Yoon et al., 2018a; Li & Marlin, 2020), which develop their algorithms and theoretical foundations under the MCAR assumption and leave the results beyond the MCAR for future work, our method and theoretical guarantees aim to provide a general framework for learning on incomplete data and generate complete data.

C. Algorithm Details

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

```
Require: Diffusion process hyperparameter $\beta_t$, denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.
repeat
    Sample $\mathbf{x}_0^{\text{obs}}$ according to the data distribution and missing mechanism;
    Infer mask $\mathbf{m} = \mathbf{1}[\mathbf{x}^{\text{obs}}(0) = \text{na}]$;
    $t \sim \text{Uniform}(\{1, \dots, T\})$;
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
    Take gradient descent step on
        $$\nabla_{\boldsymbol{\theta}} \left\| (\boldsymbol{\epsilon} - \mathbf{s}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^{\text{obs}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)) \odot \mathbf{m} \right\|^2.$$
until converged.
```

We mainly adopt the Variance Preserving (VP) SDE in this work. The forward diffusion process of the VP-SDE is defined as

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

Algorithm 2 Variance Preserving Sampling of *MissDiff*

```
Require: Diffusion process hyperparameter $\beta_t$, denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.
Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$;
$t = T$;
while $t \neq 1$ do
    Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$;
    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)) + \sqrt{\beta_t}\boldsymbol{\epsilon}_t$;
    $t = t - 1$;
end while
Return $\mathbf{x}_0$.
```

where $\{\beta_t \in (0, 1)\}_{t \in (0,T)}$ is the increasing sequence denoting the variance schedule.

Algorithm 1 and Algorithm 2 demonstrate the Denoising Score Matching objective on missing data and sampling procedure of *MissDiff*. We write $\mathbf{x}(t)$ as $\mathbf{x}_t$ in the algorithm box for simplicity.

D. Experiments Details

D.1. Experimental Setup

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), all with missing values. Each record in Census dataset contains the information(age, marital status, education level, and income) of a person from the 1994 US census database. MIMIC4ED datasets contains a vast amount of Electronic Health records (EHR) for patient data. The primary goal is to compare the effectiveness in synthetic data generation of the proposed *MissDiff* with the alternative baselines detailed later in this subsection.

The detailed description of the dataset can be found in Table 8, which specifies the number of training data (#Train), the number of testing data (#Test), the number of categorical (discrete) variables in the tabular dataset (#Categorical), and the number of continuous variables (#Continuous). Moreover, the last column shows the evaluation task we adopted as detailed later.

Table 8. Synthetic and Real-World Datasets Used in Experiments.

Dataset	#Train	#Test	#Categorical	#Continuous	Utility
Bayesian Network	2000	20000	3	2	Multi-class classification
Census (Kohavi & Becker, 1996)	16000	4000	9	6	Binary classification
MIMIC4ED (Xie et al., 2022)	353150	88287	46	27	Regression

The details of the data generated from a Bayesian Network are as follows. 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 distribution 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

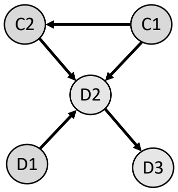


Figure 2. The demonstration of the Bayesian Network for generating the tabular data. “C1” and “C2” denote the continuous variables and “D1”, “D2”, “D3” denotes the discrete random variables. The marginal/conditional distributions for each node are detailed in Section **??**.

conditional distribution of D3 given D2 is

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

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; Ipsen et al., 2020a; Yoon et al., 2018a; Li & Marlin, 2020), MAR, and NMAR (Muzellec et al., 2020).

- 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 as missing according to the realization of a Bernoulli random variable with the 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 D.1. Under the three missing mechanisms in MCAR, with the missing ratio parameter set as $0 < \alpha < 1$, the condition in Theorem 2.2 can be satisfied with probability at least $1 - \delta$, where $\delta = \max\{(\frac{\alpha d - 1}{d})^n d, \alpha, \alpha^n d\}$ can be sufficiently small when α is small and n is sufficiently large.

Remark D.1 gives the guarantee that *MissDiff* can recover the oracle score under MCAR with high probability. In all tables in Sections 3, we adopt the missing ratio $\alpha = 0.2$ and XGBoost for the downstream tasks with no specific clarification. More experimental results can be found in Appendix D.3.

D.2. Implementation Details

We use the variance-preserving SDE with the time duration $T = 100$ for Bayesian Network and Census dataset and $T = 150$ for MIMIC4ED dataset. 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 scaler 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.

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 set the minimum noise level $\beta_1 = 0.0001$ and the maximum noise level $\beta_T = 0.5$ in Algorithm 1 and Algorithm 2 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 [3], i.e., Varaince 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].

D.3. Additional Experiental Results

D.3.1. Additional Results for Other Criteria for *Utility* Evaluation

Table 9, 10, and 11 provide the additional experimental results for other criteria under *Utility* evaluation for Table 2, 4, and 6 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 9, 10, and 11, 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.

D.3.2. Experiment Results for Different Classifiers/Regressors

As mentioned in section 3, we train various models, including Decision Tree, AdaBoost, Logistic Regression, MLP classifier/regressor, RandomForest, and XGBoost, on synthetic data. Table 12 to 16 present the corresponding results on different classifiers/regressors, from which we can see that *MissDiff* still performs well under most cases.

D.3.3. Additional Results for *STaSy-delete* and *STaSy-mean*

In section 3, we mentioned 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 significantly worse, which are demonstrated in Table 17 and 18. 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.

D.4. Computational Time

All the experiments are conducted on NVIDIA A100 Tensor Core GPUs. It takes around 30min for each experiment on Bayesian Network, around 5 hours for each experiment on the Census dataset, and around one day for each experiment on

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

Table 9. Utility evaluation of *MissDiff* on Census dataset with other criteria. "-" denotes the corresponding method cannot applied since no data $\mathbf{x}_i$ will be left after deleting the incomplete data.

Criterian	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 10. Utility evaluation of *MissDiff* on MIMIC4ED dataset with R^2 criterion.

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

Table 11. Utility evaluation of *MissDiff* on Census dataset under MAR, NMAR with other criteria.

Criterian	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

the MIMIC4ED dataset.

E. Proof Sketch for Section 4

E.1. Proof Sketch of Theorem 2.2

In order to show Theorem 2.2, 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., $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) \odot \mathbf{m} = \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \odot \mathbf{m}$[4], where $\mathbf{m} = \mathbb{1}\{\mathbf{x}^{\text{obs}}(0) = \text{na}\}$ indicated the missing entries in $\mathbf{x}^{\text{obs}}(0)$. Therefore, under certain conditions, we may first relate the Denosing Score Matching objective on missing data to the Denosing Score Matching objective on the complete data,

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

[4] 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 $\nabla_{\mathbf{x}^{\text{obs}}(t)} \log p(\mathbf{x}^{\text{obs}}(t)|\mathbf{x}^{\text{obs}}(0)) \odot \mathbf{m} = -(\mathbf{x}^{\text{obs}}(t) - \mu^{\text{obs}}) \odot \mathbf{m} = -(\mathbf{x}(t) - \mu) \odot \mathbf{m} = \nabla_{\mathbf{x}(t)} \log p(\mathbf{x}(t)|\mathbf{x}(0)) \odot \mathbf{m}$.

Table 12. 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 Missing	**80.68**%	72.68%	67.70%	76.35%	55.99%

Table 13. 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 Missing	**78.70**%	76.13%	75.96%	76.55%	78.39%

Table 14. 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 Missing	76.20%	**76.30**%	76.25%	77.13%	69.68%

Table 15. 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 Missing	**75.33**%	72.18%	74.30%	76.81%	37.59%

Table 16. 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 Missing	**79.33**%	76.30%	76.38%	76.31%	76.98%

Table 17. 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 Missing	**83.16**%	74.94%	76.60%	53.7%	50.11%

Table 18. 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 Missing	**79.49**%	75.39%	75.96%	49.78%	70.68%

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 $\mathbf{m}$. Assuming that $\mathbb{E}_{p(\mathbf{m}|\mathbf{x}(0))}[\mathbf{m}]\equiv\mathbf{1}-\boldsymbol{\rho}$ 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.

E.2. Proof Sketch of Theorem 2.3

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. Consider the MCAR missing mechanism. Denote ρ_i, $i\in\{1,2,...,d\}$ as the population percentage of missing samples for the i-th entry in the training data. Suppose $\max_{i=1,\ldots,d}\rho_i<1$. In addition, we make the same mild regularity assumptions as (Song et al., 2021a) in the following.

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

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

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

(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,

$$
\mathrm{d}\mathbf{x}^{\text{obs}}=f(\mathbf{x}^{\text{obs}},t)\mathrm{d}t+g(t)\mathrm{d}\mathbf{w}, \tag{4}
$$

and the SDE parametrized by θ,

$$
\mathrm{d}\hat{\mathbf{x}}_{\theta}^{\text{obs}}=\mathbf{s}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{\theta}^{\text{obs}},t)\mathrm{d}t+g(t)\mathrm{d}\mathbf{w}. \tag{5}
$$

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:

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

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.,

$$D_{\mathrm{KL}}(p_0\|p_\theta) = D_{\mathrm{KL}}\Big(\int K(\{\mathbf{x}^{\mathrm{obs}}(t)\}_{t\in[0,T]},\mathbf{x})\mathrm{d}\boldsymbol{\mu}\Big\|\int K(\{\hat{\mathbf{x}}^{\mathrm{obs}}_\theta(t)\}_{t\in[0,T]},\mathbf{x})\mathrm{d}\boldsymbol{\nu}\Big) \le D_{\mathrm{KL}}(\boldsymbol{\mu}\|\boldsymbol{\nu}). \tag{6}$$

By the chain rule of KL divergences,

$$D_{\mathrm{KL}}(\boldsymbol{\mu}\|\boldsymbol{\nu}) = D_{\mathrm{KL}}(p_T\|\pi) + \mathbb{E}_{\mathbf{z}\sim p_T}[D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot\mid\mathbf{x}^{\mathrm{obs}}(T)=\mathbf{z})\|\boldsymbol{\nu}(\cdot\mid\hat{\mathbf{x}}^{\mathrm{obs}}_\theta(T)=\mathbf{z}))]. \tag{7}$$

Under assumptions (i) (iii) (iv) (v) (vi) (vii) (viii), the SDE in Eq (4) has a corresponding reverse-time SDE given by

$$\mathrm{d}\mathbf{x}^{\mathrm{obs}} = [f(\mathbf{x}^{\mathrm{obs}},t) - g(t)^2\nabla_{\mathbf{x}^{\mathrm{obs}}}\log p_t(\mathbf{x}^{\mathrm{obs}})]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}. \tag{8}$$

Since Eq (8) is the time reversal of Eq (4), it induces the same path measure $\boldsymbol{\mu}$. As a result, $D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot\mid\mathbf{x}^{\mathrm{obs}}(T)=\mathbf{z})\|\boldsymbol{\nu}(\cdot\mid\hat{\mathbf{x}}^{\mathrm{obs}}_\theta(T)=\mathbf{z}))$ can be viewed as the KL divergence between the path measures induced by the following two (reverse-time) SDEs:

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

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

$$\begin{aligned}&D_{\mathrm{KL}}(\boldsymbol{\mu}(\cdot\mid\mathbf{x}^{\mathrm{obs}}(T)=\mathbf{z})\|\boldsymbol{\nu}(\cdot\mid\hat{\mathbf{x}}^{\mathrm{obs}}_\theta(T)=\mathbf{z}))\\ &= \mathbb{E}_{\boldsymbol{\mu}}[\frac{1}{2}\int_0^T g(t)^2\|\nabla_{\mathbf{x}^{\mathrm{obs}}(t)}\log p_t(\mathbf{x}^{\mathrm{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{obs}}(t),t)\|_2^2\,\mathrm{d}t]\\ &\le \frac{1}{2(1-\rho_{\max})}\int_0^T \mathbb{E}_{p_t(\mathbf{x}^{\mathrm{obs}}(t))}[g(t)^2\|\nabla_{\mathbf{x}^{\mathrm{obs}}(t)}\log p_t(\mathbf{x}^{\mathrm{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{obs}}(t),t)\odot\sqrt{\mathbf{1}-\boldsymbol{\rho}}\|_2^2]\mathrm{d}t\\ &= \frac{1}{2(1-\rho_{\max})}\int_0^T \mathbb{E}_{p_t(\mathbf{x}^{\mathrm{obs}}(t))}[g(t)^2\|\nabla_{\mathbf{x}^{\mathrm{obs}}(t)}\log p_t(\mathbf{x}^{\mathrm{obs}}(t)) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{obs}}(t),t)\odot\mathbf{m}\|_2^2]\mathrm{d}t = \frac{1}{1-\rho_{\max}}J_{\mathrm{SM}}(\theta;g(\cdot)^2),\end{aligned} \tag{9}$$

where $\rho_{\max} = \max_{i=1,\ldots,d}\rho_i$ and $1-\rho_{\max}>0$ by assumption. Combining Eqs. (6), (7) and (9), we have $D_{\mathrm{KL}}(p_0\|p_{\boldsymbol{\theta}}) \le \frac{1}{1-\rho_{\max}}J_{\mathrm{SM}}(\boldsymbol{\theta};g(\cdot)^2) + D_{\mathrm{KL}}(p_T\|\pi)$, which further yields $-\mathbb{E}_{p(\mathbf{x}^{\mathrm{obs}})}[\log p_\theta(\mathbf{x})] \le \frac{1}{1-\rho_{\max}}J_{\mathrm{DSM}}(\boldsymbol{\theta};g(\cdot)^2) + C_1$ by Lemma E.3, where C_1 is a constant independent of θ.

Remark E.2 (Interpretation of Theorem 2.3). When there is missing value, we can get 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 2.3 degenerates to the Corollary 1 in (Song et al., 2021a), i.e.,

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

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

Lemma E.3. *Denosing Score Matching on missing data is equivalent to Score Matching on missing data, i.e.,*

$$\begin{aligned}&\mathbb{E}_{p_t(\mathbf{x}^{obs})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{obs},t) - \nabla_{\mathbf{x}^{obs}}\log p_t(\mathbf{x}_t^{obs}))\odot\mathbf{m}\|_2^2]\\ &= \mathbb{E}_{p(\mathbf{x}_0^{obs})}\mathbb{E}_{p(\mathbf{x}_t^{obs}\mid\mathbf{x}_0^{obs})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{obs},t) - \nabla_{\mathbf{x}_t^{obs}}\log p(\mathbf{x}_t^{obs}\mid\mathbf{x}_0^{obs}))\odot\mathbf{m}\|_2^2] + C,\end{aligned} \tag{10}$$

where $\mathbf{m} = \mathbb{1}\{\mathbf{x}_0^{obs} = \mathrm{na}\}$ 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 (10)

$$\begin{aligned}\mathrm{LHS} &= \mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t) - \nabla_{\mathbf{x}_t^{\mathrm{obs}}}\log p_t(\mathbf{x}_t^{\mathrm{obs}}))\odot\mathbf{m}\|_2^2]\\ &= \mathbb{E}_{p_t(\mathbf{x}_t^{\mathrm{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\mathrm{obs}},t)\odot\mathbf{m}\|^2] - S(\theta) + C_2,\end{aligned} \tag{11}$$

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

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

Substituting this expression for $S(\theta)$ into Eq (11) yields

$$\begin{aligned}
\text{LHS} = {} & \mathbb{E}_{p_t(\mathbf{x}_t^{\text{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{obs}}, t) \odot \mathbf{m}\|^2] \\
& - 2\mathbb{E}_{p(\mathbf{x}_t^{\text{obs}}, \mathbf{x}_0^{\text{obs}})}[\langle \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{obs}}, t), \frac{\mathrm{d} \log p(\mathbf{x}_t^{\text{obs}} \mid \mathbf{x}_0^{\text{obs}})}{\mathrm{d}\mathbf{x}_t^{\text{obs}}} \odot \mathbf{m}\rangle] + C_2.
\end{aligned} \tag{12}$$

On the other hand, we also have the Denoising Score Matching objective on the right-hand side of (10) is

$$\begin{aligned}
\text{RHS} = {} & \mathbb{E}_{p_t(\mathbf{x}_t^{\text{obs}})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{obs}}, t) \odot \mathbf{m}\|^2] \\
& - 2\mathbb{E}_{p(\mathbf{x}_t^{\text{obs}}, \mathbf{x}_0^{\text{obs}})}[\langle \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{obs}}, t), \frac{\mathrm{d} \log p_t(\mathbf{x}_t^{\text{obs}} \mid \mathbf{x}_0^{\text{obs}})}{\mathrm{d}\mathbf{x}_t^{\text{obs}}}\rangle \odot \mathbf{m}] + C_3,
\end{aligned} \tag{13}$$

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

Comparing equations (12) and (13), we thus show that the two optimization objectives are equivalent up to a constant. □