Learning What Not to Impute: An Uncertainty-Aware Diffusion Framework for Meaningful Missingness

Lixing Zhang[1], **Yidong Ouyang**[2], **Weifu Li**[1], **Guang Cheng**[2], **Shixiang Zhu**[3], **Liyan Xie**[1*]
[1]University of Minnesota [2]University of California, Los Angeles [3]Carnegie Mellon University

Abstract

Missing value imputation is a fundamental task in machine learning, with most existing methods assuming that all missing entries correspond to unobserved regular values. In many real-world datasets, however, missingness may arise from two distinct sources: some entries are *meaningfully missing* (intrinsically absent and semantically valid), while others are missing due to the observation process and should be imputed. We formalize this distinction as a *selective imputation* problem, where the goal is to jointly infer which missing entries should be preserved and which should be recovered. To address this challenge, we propose Diff-Joint, a diffusion-based framework that jointly models tabular data together with a latent missingness mask. The method alternates between conditional sampling and uncertainty-aware aggregation to iteratively refine both imputed values and missingness labels. Empirical results on synthetic and real-world datasets demonstrate that Diff-Joint effectively identifies meaningfully missing entries while achieving competitive imputation accuracy and improved downstream task performance. Our code is publicly available at https://github.com/lxzhang1/Diff-Joint.

1 Introduction

Missing value imputation is a common and important problem in machine learning, statistics, and data mining [1, 2]. In many applications, training datasets contain missing entries that must be imputed either as quantities of direct interest or as a preprocessing step for downstream modeling, analysis, and decision-making. A broad range of methods have been developed to recover missing entries from the observed data values, ranging from classical statistical procedures [3, 4, 5, 6] to modern deep generative models [7, 8, 9, 10].

Under the conventional imputation task, all missing entries are typically treated as unobserved regular values that need to be fully recovered. However, it is important to notice that in some datasets, the missing value can arise from two distinct sources: an entry may be meaningfully missing in the complete record, or a regular value that went missing during the observation process [11]. For example, in clinical records, a laboratory measurement may be missing because the test was not ordered, but the absence of the test order itself may carry clinical meaning [12]. In survey data, a response such as "n/a" may be a valid answer rather than an unobserved value; for instance, income from employment is genuinely not applicable for a respondent who is not employed [13]. In e-commerce or recommender-system data, the absence of a product attribute may indicate that the attribute is inapplicable to the item, rather than that the attribute was accidentally omitted [14, 15]. See Figure 1 for a conceptual illustration.

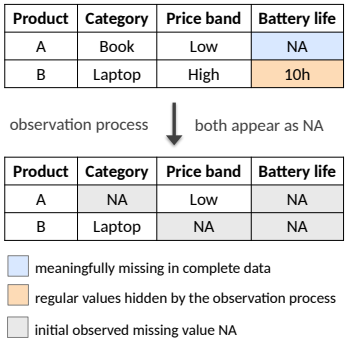

Figure 1: An illustrative example of "meaningful missingness."

*Corresponding author. Contact: zhan9503@umn.edu and liyanxie@umn.edu.

This distinction implies that not all observed missing entries should be treated in the same way. Some entries are missing because the missing state itself is part of the underlying record. When a missing entry is meaningful, imputing it with a regular value can distort the data distribution, remove useful semantic information, and introduce bias into the learned data distribution. The central goal is therefore not only to estimate the value of a missing entry, but also to determine whether the entry should be imputed. This problem is challenging because the two sources of missingness are not directly labeled in the observed data. Moreover, the distinction is entry-wise and context-dependent: the same missing token may be meaningful for one sample but observation-induced for another.

This calls for new imputation methods that can decide not only *how to impute*, but also *when not to impute*. In particular, the goal is to preserve missing entries that are meaningful while imputing only those entries that are missing due to the observation process. To this end, we propose a new learning framework that performs *selective imputation*: rather than completing all missing entries, the method jointly identifies meaningfully missing states and recovers observation-induced missing values. We instantiate this framework using diffusion models [16, 17], which provide a flexible backbone for modeling complex tabular distributions and imputation [10, 18].

The proposed method, Diff-Joint, introduces a joint diffusion state $(\mathbf{x}, \mathbf{c})$, where $\mathbf{x}$ represents the completed tabular values and $\mathbf{c}$ is a binary mask indicating which entries are meaningfully missing. Starting from a random initialization, Diff-Joint alternates between two steps. First, it trains a diffusion model on the current joint state. Second, it draws multiple conditional samples given the observed entries and aggregates these samples to update both the imputed values and the meaningfully-missing mask. The aggregation step leverages the entrywise *uncertainty scores*: high posterior uncertainty provides evidence that an observed "na" may correspond to a meaningful missing state rather than a recoverable regular value. Through this iterative refinement, the model learns both the data distribution and the missingness structure.

Our contributions are summarized as follows.

1. We formulate a selective imputation problem for tabular data with two latent sources of observed missingness: meaningfully missing entries that should be preserved, and observation-induced missing entries that should be imputed.
2. We propose Diff-Joint, a diffusion-based framework that jointly models completed tabular values and meaningfully-missing masks. By alternating between diffusion-model training and uncertainty-aware aggregation of conditional samples, Diff-Joint iteratively refines both the imputed values and the labels of meaningfully missing entries.
3. We evaluate the proposed method on both a synthetic Bayesian-network dataset and three real-world datasets from MIMIC-IV-ED and the BeyondArena benchmark. The results show that Diff-Joint can identify meaningfully missing entries while maintaining competitive imputation performance and improving downstream predictive performance.

Related Work. Missing-value imputation has been extensively studied, and existing approaches can be roughly divided into two categories: classical statistical methods and modern deep generative models. Classical missing-data theory separates the underlying data distribution from the observation process, and commonly categorizes missingness mechanisms as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) [11, 19]. This perspective underlies many classical imputation methods, including likelihood-based estimation with the EM algorithm [3], chained-equation imputation [5], and random-forest imputation for mixed-type data [6]. These methods mainly treat missing entries as unobserved regular values to be recovered.

Deep generative models now provide a flexible framework for imputation. Representative approaches include adversarial imputation with GAIN [7], latent-variable modeling with MIWAE [8], adaptive iterative imputation with HyperImpute [20], and masked-reconstruction methods such as ReMasker and CACTI [21, 22]. Diffusion and score-based generative models have become a powerful class of generative models [16, 23, 17, 24]. For missing-data problems, CSDI trains conditional score-based models for probabilistic time-series imputation, while TabCSDI adapts this idea to mixed-type tabular data [9, 10]. Forest-Diffusion combines diffusion or flow-based generative modeling with gradient-boosted trees for tabular generation and imputation [25]. Closest to our iterative training procedure, DiffPuter combines diffusion models with an EM-style refinement loop to learn from incomplete data and update missing-value estimates through conditional sampling [18]. These diffusion-based methods mainly focus on completing all missing values in the data rather than modeling the meaningful-missingness explicitly.

Another related line of work recognizes that missingness patterns themselves can carry useful information. In this direction, missing-data handling has been studied for robust prediction from incomplete inputs [26], and recurrent models for clinical time series incorporate masks and time gaps as predictive features [27]. In multimodal federated learning, learned missingness profiles are used to adapt representations for prediction [28]. Recent work on synthetic data generation also emphasizes that preserving missingness distributions can be important for downstream utility [29]. These works show that missingness should not always be ignored or naively filled. However, they typically use the observed missingness pattern as an auxiliary feature and do not explicitly model the entry-wise distinction between missing states. Our work makes this distinction explicit by introducing a meaningfully-missing mask and learning it jointly with the underlying data distribution.

2 Problem Setup and Preliminaries

We consider mixed-type tabular data with d features (columns). For the j-th feature, let $\mathcal{X}_j$ denote its domain of regular values, and define the augmented domain $\bar{\mathcal{X}}_j = \mathcal{X}_j \cup \{\mathrm{na}\}$, where na is used throughout the paper to denote the observed missing values. We represent a complete data point as $\mathbf{x}^{\text{true}} = (x_1, \ldots, x_d) \in \bar{\mathcal{X}}_1 \times \cdots \times \bar{\mathcal{X}}_d$, drawn from the underlying data-generating distribution on the augmented domain. If $x_j = \mathrm{na}$ in $\mathbf{x}^{\text{true}}$, then the j-th entry is *meaningfully missing* (MM), meaning that the value is intrinsically absent and should be treated as a valid state. To model such meaningfully missing entries, we associate each complete data instance $\mathbf{x}^{\text{true}}$ with a binary mask

$$\mathbf{c} = (c_1, \ldots, c_d) \in \{0,1\}^d, \quad c_j = \mathbb{1}\{x_j^{\text{true}} = \mathrm{na}\}. \tag{1}$$

In particular, $c_j = 1$ indicates that the j-th entry is intrinsically missing.

In addition to such intrinsically missing entries, an observed record may also exhibit *observation-induced missingness*, arising from the observation process after the underlying data instance is generated. Typical causes include incomplete data collection, recording errors, and data transfer failures. These observation-induced missing entries are also recorded using the same symbol na in the observed data. We encode this second source using another binary mask $\mathbf{r} = (r_1, \ldots, r_d) \in \{0,1\}^d$, where $r_j = 1$ indicates that the j-th entry is missing due to the observation process.

The final observation is then determined jointly by meaningful missingness and observation-induced missingness. We define the observation mask $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_d) \in \{0,1\}^d$ with $\omega_j = (1-c_j)(1-r_j)$. Given $(\mathbf{x}^{\text{true}}, \boldsymbol{\omega})$, the observed record $\mathbf{x}^{\text{obs}} = (x_1^{\text{obs}}, \ldots, x_d^{\text{obs}})$ is defined as $x_j^{\text{obs}} = x_j$ if $\omega_j = 1$, and $x_j^{\text{obs}} = \mathrm{na}$ otherwise. In other words, the same symbol na is observed regardless of whether the underlying cause is meaningful missingness or observation-induced missingness. Consequently, these two types of missingness are not directly distinguishable from the observed data alone. Our framework therefore assumes that the two missingness mechanisms induce distinct statistical properties in the conditional distribution of missing entries given the observed context. We formalize this mechanism-separation perspective and establish identifiability conditions in Section 3.2.

Given n observed records $\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}$, our goal is to infer which entries are meaningfully missing and to impute *only* those entries that are observation-induced missing. Specifically, we first aim to infer the meaningful-missingness mask $\widehat{\mathbf{c}}_i = (\widehat{c}_{i,1}, \ldots, \widehat{c}_{i,d})$ to determine which entries of each record should remain na. Second, for entries not identified as meaningfully missing, we estimate their regular values and output an imputed value $\widehat{x}_{i,j} \in \mathcal{X}_j$. Thus, the final imputed record preserves na for entries with $\widehat{c}_{i,j} = 1$, while replacing observation-induced missing entries with their imputed values $\widehat{x}_{i,j}$. We assume that the set $\mathcal{T} \subseteq \{1, \ldots, d\}$ of MM candidate columns is specified in advance.

2.1 Preliminaries: Diffusion Models

This work uses diffusion models to learn the data distribution [16]. Diffusion models are characterized by their forward and backward processes. The forward process perturbs the data distribution $p(\mathbf{x})$ by injecting Gaussian noise, as described by the following continuous-time equation [17]:

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \ t \in [0, T], \tag{2}$$

where $\mathbf{w}$ is the standard Brownian motion, $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is a drift coefficient, and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ is a diffusion coefficient. The marginal distribution of $\mathbf{x}_t$ at time t is denoted as $p_t(\mathbf{x}_t)$, and p_0 is the

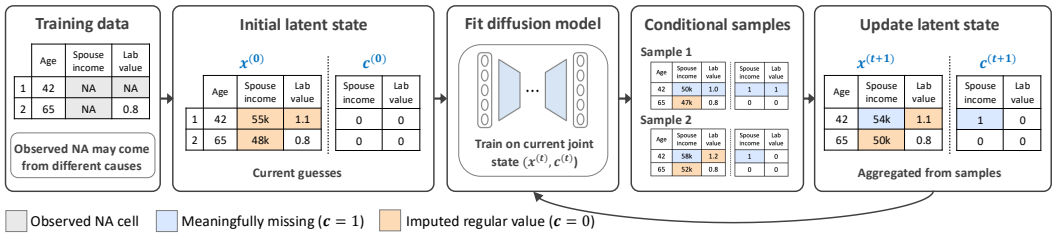

Figure 2: Overview of Diff-Joint. Starting from incomplete observations $\mathbf{x}^{\text{obs}}$, Diff-Joint alternates between diffusion-model training on the current joint state $\widehat{\mathbf{s}}^{(t)} = (\widehat{\mathbf{x}}^{(t)}, \widehat{\mathbf{c}}^{(t)})$, and latent state update via conditional sampling and aggregation.

distribution of the initial value $\mathbf{x}_0$, which equals the true data distribution. Then, we can reverse the forward process (2) for generation, defined as:

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{3}$$

where $\overline{\mathbf{w}}$ is a standard Brownian motion when time flows backwards from T to 0. The key of the backward process is to estimate the score function of each marginal distribution, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, by training a score network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ [30, 31, 32]

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\text{Unif}[0,T]} \left\{ \lambda(t) \mathbb{E}_{p_t(\mathbf{x}_t)} \left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2 \right] \right\}, \tag{4}$$

where $\lambda(t) : [0, T] \to \mathbb{R}_{>0}$ is a positive weighting function.

It is worthwhile mentioning that since tabular data may contain both continuous and discrete variables, we use an encoder described in Appendix D.1 to map each record into a continuous model space for diffusion-model training and sampling. The corresponding decoder maps generated samples back to the original mixed-type tabular space. For notational simplicity, we use $\mathbf{x}$ for both the original and encoded representations in the rest of the paper; the distinction is clear from context.

3 Methodology

3.1 Diff-Joint: Joint Diffusion for Selective Imputation

To model meaningful missingness, we define the joint diffusion state $\mathbf{s} = (\mathbf{x}, \mathbf{c})$, where $\mathbf{c}$ is the meaningful-missingness (MM) mask. Based on such joint-state representation, we propose Diff-Joint, an iterative framework for jointly modeling tabular values and meaningful-missingness patterns. As illustrated in Figure 2, given the observed dataset $\{\mathbf{x}_i^{\text{obs}}\}_{i=1}^n$, the Diff-Joint framework initializes the observed missing values by randomly filling the missing entries in each $\mathbf{x}_i^{\text{obs}}$ to obtain $\widehat{\mathbf{x}}_i^{(0)}$, and setting $\widehat{\mathbf{c}}_i^{(0)} = \mathbf{0}$. This yields the initial joint state $\widehat{\mathbf{s}}_i^{(0)} = (\widehat{\mathbf{x}}_i^{(0)}, \widehat{\mathbf{c}}_i^{(0)})$ for each data point. That is, all observed missing entries are initially treated as observation-induced missingness. Starting from $\{\widehat{\mathbf{s}}_i^{(0)}\}_{i=1}^n$, Diff-Joint alternates between the following two steps at each iteration t: (i) *Model Update:* train a diffusion model on the given collection of joint data state $\{\widehat{\mathbf{s}}_i^{(t)}\}_{i=1}^n$ to capture dependencies between data values and meaningful-missingness patterns; and (ii) *Latent-state Update:* draw multiple conditional samples from the current diffusion model and update the joint state to $\{\widehat{\mathbf{s}}_i^{(t+1)}\}_{i=1}^n$. In the following, we describe the latent-state update and present the overall procedure in Algorithm 1. Detailed algorithms for each step are provided in Appendix A.

Latent-state Update. At each iteration t, conditioning on each observed training data $\mathbf{x}^{\text{obs}}$, we use the current updated joint-state diffusion model (parameterized by $\boldsymbol{\theta}^{(t)}$) to draw K samples $\{(\mathbf{x}^{(t,k)}, \mathbf{c}^{(t,k)})\}_{k=1}^K$. Here K is a pre-specified sample size. We then aggregate these K conditional samples to update both the imputed values and the meaningful-missingness mask for each data point.

First, for each missing j-th entry in $\mathbf{x}^{\text{obs}}$, we update its imputed value to $\widehat{x}_j^{(t+1)}$ as

$$\widehat{x}_j^{(t+1)} = \begin{cases} \text{mode}\big(\{x_j^{(t,1)}, \ldots, x_j^{(t,K)}\}\big), & \text{if feature } j \text{ is discrete,} \\ \frac{1}{K}\sum_{k=1}^K x_j^{(t,k)}, & \text{if feature } j \text{ is continuous,} \end{cases} \tag{5}$$

Algorithm 1 Diff-Joint for Selective Imputation

Input: Observed data table $X^{\text{obs}} = \{\mathbf{x}_i^{\text{obs}}\}_{i\in[n]}$, observation mask $\Omega = \{\boldsymbol{\omega}_i\}_{i\in[n]}$, MM search column set $\mathcal{T}$, number of samples K, number of reverse steps M, number of iterations $t_{\max}$.

Output: Final completed data $\widehat{X}$, meaningful-missingness mask $\widehat{C}$, and trained model parameters $\boldsymbol{\theta}$.

Initialize: randomly fill the missing entries in X^{obs} to obtain $\widehat{X}^{(0)}$.
Set the initial meaningful-missingness mask $\widehat{C}^{(0)} \leftarrow \mathbf{0}_{n\times d}$.
Form the initial joint representation $\widehat{S}^{(0)} \leftarrow (\widehat{X}^{(0)}, \widehat{C}^{(0)})$.
for $t = 0, 1, 2, \ldots, t_{\max} - 1$ **do**
  $\boldsymbol{\theta}^{(t)} \leftarrow \textsc{Model–Update}(\widehat{S}^{(t)})$ using Algorithm 2
  $\mathcal{S}^{(t)} \leftarrow \textsc{Conditional–Sample}(\boldsymbol{\theta}^{(t)}, X^{\text{obs}}, \Omega, K, M)$ using Algorithm 3
  $(\widehat{X}^{(t+1)}, \widehat{C}^{(t+1)}) \leftarrow \textsc{Aggregate}(\mathcal{S}^{(t)}, \Omega, \mathcal{T})$ using Algorithm 4
  Form $\widehat{S}^{(t+1)} \leftarrow (\widehat{X}^{(t+1)}, \widehat{C}^{(t+1)})$.
end for
$\widehat{X} \leftarrow \widehat{X}^{(t+1)} \odot (1 - \widehat{C}^{(t+1)}) + \text{na} \odot \widehat{C}^{(t+1)}$.
return $(\widehat{X}, \widehat{C}^{(t+1)}, \boldsymbol{\theta}^{(t)})$.

where $x_j^{(t,k)}$ denotes the j-th entry in the generated conditional sample $\mathbf{x}^{(t,k)}$.

Then, for each j such that the j-th entry of $\mathbf{x}^{\text{obs}}$ is missing, to quantify how uncertain the model is about this missing entry, we define the following uncertainty score:

$$u_j^{(t+1)} = \begin{cases} -\sum_{v\in\mathcal{X}_j} \hat{p}_j^{(t)}(v) \log \hat{p}_j^{(t)}(v), & \text{if feature } j \text{ is discrete,} \\ \sqrt{\frac{1}{K}\sum_{k=1}^{K}(x_j^{(t,k)} - \frac{1}{K}\sum_{\ell=1}^{K} x_j^{(t,\ell)})^2}, & \text{if feature } j \text{ is continuous,} \end{cases} \tag{6}$$

where $\hat{p}_j(v) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}(x_j^{(t,k)} = v)$ is the empirical probability mass function of these K samples. That is, the uncertainty score is defined as empirical entropy for discrete variables and empirical standard deviation for continuous ones. In both cases, a larger uncertainty score indicates greater posterior uncertainty about the missing entry, *i.e.*, the generated conditional samples are more diverse.

Finally, we compute the imputation (5) and the uncertainty score (6) for missing entries in each data point $\mathbf{x}_i^{\text{obs}}$, $i = 1, \ldots, n$. That is, let $\mathcal{M}_j \subset \{1, \ldots, n\}$ denote the set of samples whose j-th observed entry is na, then the above calculation yields the uncertainty score $\{u_{i,j}^{(t+1)} : i \in \mathcal{M}_j\}$ for the j-th variable. We then apply k-means clustering with $k = 2$ to separate these entries with *high* uncertainty from those with *low* uncertainty. Specifically, we denote the two resulting clusters by $\mathcal{C}_{0,j}^{(t+1)}$ and $\mathcal{C}_{1,j}^{(t+1)}$. Without loss of generality, we assume $\mathcal{C}_{1,j}^{(t+1)}$ is the cluster with higher average uncertainty values, thus treated as the meaningful-missingness cluster. In parallel, the sampled masks provide a direct MM signal through the majority vote that equals $\text{mode}\big(\{c_{i,j}^{(t,k)}\}_{k=1}^{K}\big)$. We combine these two signals conservatively to update the second part of the joint state:

$$\widehat{c}_{i,j}^{(t+1)} = \mathbb{1}\{i \in \mathcal{C}_{1,j}^{(t+1)}\} \vee \text{mode}\big(\{c_{i,j}^{(t,k)}\}_{k=1}^{K}\big). \tag{7}$$

Here $\vee$ denotes the logical OR operator. Thus, an entry is classified as meaningfully missing if either the sampled masks or the uncertainty pattern supports the MM interpretation.

The overall procedure can be viewed as an iterative scheme: the model-update step refits the diffusion model using the current joint-state estimates, while the latent-state-update step refines the MM indicators and imputed values using the learned model. Ablation studies in Appendix D.3 show that the full proposed algorithm outperforms both related baseline method and variants with individual components removed, demonstrating that iterative refinement, joint-state characterization, and the uncertainty-based aggregation rule are all essential and contribute to its strong performance.

3.2 Identifiability of Meaningful Missingness

In this section, we provide theoretical insights into the identifiability of meaningful missingness when it is mixed with observation-induced missingness in the observed record. The following Proposition formalizes two representative regimes under which meaningful missingness can be identified. The proof of Proposition 3.1 is provided in Appendix B.1.

Proposition 3.1 (Identifiability). *Assume that the meaningful missing mechanism satisfies* $\Pr(\mathbf{c}_j = 1 \mid \mathbf{x}^{true}_{-j}) = \phi_\theta(\mathbf{x}^{true}_{-j}; j)$ *with unknown parameter* θ*, and the function* ϕ_θ *satisfies that* $\phi_\theta \neq \phi_{\theta'}$ *for any* $\theta \neq \theta'$*. We consider the following two scenarios.*

(i) Assume the observation-induced missingness is MCAR with known probabilities $p_j = \Pr(r_j = 1) < 1$ *for all* j*. Then the meaningful missing is identifiable from the observed data* $\mathbf{x}^{\text{obs}}$*, i.e.,* θ *can be uniquely determined.*

(ii) Assume that there exists a known subset $S \subseteq \{1, \ldots, d\}$*,* $S^c \neq \emptyset$*, such that* $\Pr(c_j = 1) = 0$ *for* $j \notin S$*. That is, meaningful missingness will only occur in the columns in* S*. Assume further that the observation-induced missingness is MCAR with a common but unknown probability* $p_j = p < 1$*,* $j = 1, \ldots, d$*. Then the meaningful missing is identifiable from the observed data* $\mathbf{x}^{\text{obs}}$*, i.e.,* θ *can be uniquely determined.*

The two MCAR regimes are chosen and analyzed for technical simplicity, and the same principle should extend to more general settings where the observation-induced missingness is itself identifiable or can be separated from the MM mechanism through auxiliary structure, validation information, or certain parametric conditions. For example, prior work establishes MNAR identification under additional structural assumptions [33] or parametric restrictions [34], treating all missing entries as unobserved regular values. Adapting these results to identify the latent MM is a direction for future work. Moreover, Appendix B.3 provides MAR and MNAR counterexamples in which different MM mechanisms induce the same observed-data distribution. Thus, MAR or MNAR assumptions alone do not ensure identifiability when the observation-induced mechanism is unknown. Beyond these theoretical settings, our numerical experiments in Section 4 further show that the proposed algorithm remains fairly robust and maintains stable performance across various observation-induced missingness mechanisms.

Remark 3.2. We comment that Proposition 3.1 holds for general meaningfully-missing mechanisms ϕ_θ. One example that satisfies the assumption is the logistic scheme $\phi_\theta(\mathbf{x}; j) = \sigma(\alpha_j^\top \mathbf{x} + \beta_j)$, with parameters $\alpha_j \in \mathbb{R}^d$ and $\beta_j \in \mathbb{R}$ identifiable under assumptions in Proposition 3.1. We provide more discussions under this special case in the Appendix B.2.

Remark 3.3. Proposition 3.1 characterizes what is theoretically identifiable from the observed-data distribution, while practical recovery depends on the empirical effectiveness of the algorithm used. The uncertainty score is our key design for separating MM entries from observation-induced missing entries in practice. It relies on a heuristic uncertainty-gap assumption: observation-induced missing entries correspond to regular values and tend to have concentrated conditional predictive distributions, whereas MM entries may not be well explained by any single regular state and thus tend to have larger uncertainty scores. We show in Appendix B.2 that, under certain conditions, the first-step iteration can guarantee that those MM-entries exhibit higher uncertainty scores in expectation than non-MM entries, thus can be separated. Moreover, as the diffusion model better approximates the joint distribution, the K conditional samples $\{\mathbf{c}^{(t,k)}\}_{k=1}^K$ become more informative about the true MM indicators, enabling the logical OR update in Eq. (7) to more effectively complement the uncertainty-gap-based update. We also note that when reliable rules for meaningful missingness and all required context are available, those rules should be applied directly to infer the MM entries; our framework targets settings with unknown or incomplete rules, missing context, or stochastic meaningful missingness.

4 Numerical Experiments

Datasets and Numerical Setup. We evaluate Diff-Joint on four mixed-type tabular datasets: a synthetic Bayesian-network dataset [23] and three real-world datasets constructed from MIMIC-IV-ED [35], HELOC [36], and Hotel Booking Demand [36]. Table 6 in Appendix C summarizes the main statistics of the four datasets. For the Bayesian Network and MIMIC-IV-ED datasets, we construct ground-truth meaningful missingness using pre-specified mechanisms. For HELOC and Hotel Booking Demand, we derive reference MM labels from dataset-native semantic-missing states. These two datasets therefore evaluate recovery of naturally occurring MM without introducing synthetic MM. Dataset construction and preprocessing details are provided in Appendix C.

We then introduce an additional observation-induced missing layer. We evaluate our method under missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) mechanisms. For the Bayesian-network dataset, we report results under all three ordinary-missingness

Table 1: Method comparison on Bayesian Network data under different missing ratios and missing mechanisms.

Method	Ratio (%)	MCAR		MAR		MNAR	
		ACC_{out} (%) ↑	RMSE_{out} ↓	ACC_{out} (%) ↑	RMSE_{out} ↓	ACC_{out} (%) ↑	RMSE_{out} ↓
Diff-Joint	10	**78.34**	5.09	**75.32**	5.15	**72.51**	5.16
CMAE	10	35.21	**4.52**	24.67	**4.98**	30.43	**5.05**
DiffPuter	10	31.74	5.07	23.54	5.26	26.90	5.29
missForest	10	30.40	4.76	19.12	5.17	23.26	5.16
Mean/Mode	10	30.40	4.85	19.12	5.04	23.26	5.08
Diff-Joint	20	**69.35**	5.08	**66.16**	5.24	**65.85**	**4.88**
CMAE	20	47.18	**4.74**	34.25	**5.03**	42.14	5.07
DiffPuter	20	44.61	5.20	34.79	5.32	37.48	5.36
missForest	20	42.40	5.01	30.98	5.16	34.44	5.14
Mean/Mode	20	42.40	4.96	30.98	5.07	34.44	5.07
Diff-Joint	30	**63.65**	5.06	**57.06**	5.33	**67.91**	5.32
CMAE	30	52.26	**4.88**	40.83	5.06	45.29	5.08
DiffPuter	30	51.89	5.28	43.48	5.24	45.04	5.27
missForest	30	49.23	5.04	41.19	6.40	42.40	5.20
Mean/Mode	30	49.23	5.02	41.19	**5.05**	42.40	**5.05**
Diff-Joint	40	**65.92**	**4.99**	**67.99**	5.42	**60.38**	5.21
CMAE	40	54.09	5.44	45.15	5.06	50.30	5.05
DiffPuter	40	56.18	5.20	45.27	5.27	49.47	5.22
missForest	40	53.83	5.21	42.08	5.13	48.26	5.12
Mean/Mode	40	53.83	5.00	42.08	**5.06**	48.26	**5.02**

mechanisms. For the real-world datasets, we use MCAR ordinary missingness with different masking ratios. The formal definitions of these observation-induced missingness mechanisms and their specific implementations are given in Appendix C.5.

Evaluation Criteria. We evaluate Diff-Joint along two dimensions: meaningful-missingness (MM) identification and observation-induced-missing-value imputation. For MM identification, we report precision and recall over the missing entries, $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$, where TP, FP, and FN are true positives, false positives, and false negatives, computed by treating meaningful missingness as the positive class. For discrete variables, we additionally report token-level recovery accuracy, which is defined as $\text{ACC} = \frac{1}{|\mathcal{M}_{\text{disc}}|} \sum_{(i,j)\in\mathcal{M}_{\text{disc}}} \mathbb{1}\{\hat{x}_{i,j} = x_{i,j}\}$, where $\mathcal{M}_{\text{disc}}$ denotes the set of missing entries in discrete MM candidate columns; $x_{i,j} \in \bar{\mathcal{X}}_j$ and $\hat{x}_{i,j} \in \bar{\mathcal{X}}_j$ denote the true and imputed values, respectively. We note that this metric is stricter than MM-label accuracy: for non-MM entries, the imputed discrete value must also match the ground truth. For continuous variables, we report RMSE (and MAE) on ordinary-missing entries in continuous columns that are not allowed to contain meaningful missingness. In all tables, "out" denotes test-set performance.

Baselines. We compare Diff-Joint against representative baselines from four families: simple statistical imputation, classical iterative imputation, masked autoencoding, and diffusion-based generative imputation. Mean/Mode imputes each column independently using the empirical mean for continuous variables and the empirical mode for categorical variables. We use missForest [6] as a strong tree-based iterative baseline for mixed-type tabular data, providing a computationally practical classical alternative on the large-scale MIMIC-IV-ED dataset. As modern deep-learning baselines, we include CACTI [22], a recent strong masked-autoencoding method reported to improve over several prior autoencoding baselines, and DiffPuter [18], a recent diffusion-based imputation method reported to outperform several prior diffusion baselines. On the Bayesian Network dataset, whose column names carry no semantic information, we use the non-embedding variant of CACTI, denoted CMAE. These baselines are designed to impute missing values rather than to identify meaningful missingness; therefore, their MM precision and recall are not measurable and we only report their imputation error and downstream performance. Additional implementation details are provided in Appendix E.

Results on the Synthetic Dataset. Table 1 reports out-of-sample results on the Bayesian-network synthetic dataset under three observation-induced missingness mechanisms: MCAR, MAR, and MNAR. It is worthwhile noting that Diff-Joint is the only method that explicitly identifies meaningful missingness. In terms of discrete token-level recovery, Diff-Joint achieves the best accuracy across

Table 2: Precision and recall of Diff-Joint on the Bayesian Network dataset under different missing ratios and missing mechanisms.

Method	Ratio	MCAR		MAR		MNAR	
		$\text{Recall}_{\text{out}} \uparrow$	$\text{Precision}_{\text{out}} \uparrow$	$\text{Recall}_{\text{out}} \uparrow$	$\text{Precision}_{\text{out}} \uparrow$	$\text{Recall}_{\text{out}} \uparrow$	$\text{Precision}_{\text{out}} \uparrow$
Diff-Joint	10	90.86	73.62	87.25	80.39	90.24	71.67
Diff-Joint	20	82.39	58.67	88.09	65.42	86.03	58.76
Diff-Joint	30	78.69	49.40	65.13	43.98	76.66	55.00
Diff-Joint	40	76.79	45.88	74.60	57.59	77.14	41.34

Table 3: Downstream performance on Bayesian Network for variables D2 and D3.

Method	Ratio	D2				D3			
		Macro-F1	ROC-AUC	Weighted-F1	ACC	Macro-F1	ROC-AUC	Weighted-F1	ACC
Diff-Joint	10	**75.54**	**94.81**	**71.42**	**89.38**	**79.16**	**86.01**	**77.48**	**84.72**
Mean/Mode	10	46.50	73.38	40.21	72.65	41.69	52.92	36.32	62.23
missForest	10	45.70	72.12	39.24	72.39	41.74	52.94	36.38	62.27
CMAE	10	40.82	70.39	32.02	70.10	38.85	56.64	32.54	61.42
DiffPuter	10	45.97	74.01	39.58	72.52	41.44	55.33	36.03	62.10
Diff-Joint	20	**75.44**	**95.26**	**71.23**	**89.72**	**79.16**	**86.25**	**77.48**	**84.72**
Mean/Mode	20	40.03	69.54	32.47	71.25	41.74	52.83	36.38	62.27
missForest	20	39.59	68.44	31.97	70.83	41.74	52.88	36.38	62.25
CMAE	20	42.10	72.79	33.48	70.72	40.88	54.82	35.12	62.25
DiffPuter	20	46.38	73.92	40.07	72.62	41.45	53.98	36.04	62.13
Diff-Joint	30	**76.55**	**95.49**	**72.57**	**90.00**	**78.89**	**85.91**	**77.17**	**84.65**
Mean/Mode	30	37.12	68.25	28.97	70.75	41.21	53.21	35.76	62.00
missForest	30	36.54	66.04	28.36	69.90	40.97	53.23	35.49	61.88
CMAE	30	45.25	72.42	37.12	71.85	41.66	56.51	36.26	62.23
DiffPuter	30	46.47	74.72	40.17	72.65	41.59	53.68	36.21	62.18
Diff-Joint	40	**61.11**	**94.77**	**53.94**	**86.15**	**75.13**	**86.42**	**72.73**	**83.15**
Mean/Mode	40	35.11	70.68	26.49	70.89	35.27	53.15	28.83	59.69
missForest	40	33.97	68.49	25.22	69.94	34.76	52.96	28.24	59.42
CMAE	40	38.85	69.95	29.62	69.08	39.83	63.21	33.43	62.23
DiffPuter	40	43.23	74.30	36.11	72.18	41.62	53.33	36.23	62.23

all ratios, substantially outperforming standard imputation baselines. This shows that modeling na as a semantic state improves recovery of the final entry state. Moreover, Diff-Joint maintains stable MM recovery performance, indicating that the uncertainty-based update is not tied to a specific missing pattern. For continuous imputation, Diff-Joint is competitive but not always the best point imputer, since its objective jointly balances ordinary-value recovery and MM identification. Furthermore, Table 2 presents the MM precision and recall of Diff-Joint and it can be seen that Diff-Joint achieves consistently high MM recall (90.86% to 76.79%) as the observation-induced missing ratio increases from 10% to 40%. Precision decreases at higher missing ratios, reflecting the increasing difficulty of distinguishing semantic absence from randomly masked entries. Additional precision and recall results under other missing mechanisms and dataset are provided in Appendix D.2.

We further evaluate the effectiveness of Diff-Joint via two downstream multi-class classification tasks for target variables D2 and D3. The results are reported in Table 3 using Macro-F1, Weighted-F1, ROC-AUC, and accuracy as classification metrics. More details can be found in Appendix D.1. It can be seen that the proposed method achieves the best performance among all baseline methods, indicating that the selective imputation can significantly improve the downstream task performance when na is explicitly considered as a meaningfully missing state.

Results on the MIMIC-IV-ED Dataset. Table 4 presents the results on the MIMIC-IV-ED dataset under MCAR observation-induced missing with varying missing ratios. Diff-Joint still achieves the best accuracy in terms of discrete token-level recovery for relatively small missing ratios, and it is the only method that can explicitly identify meaningful missingness. For continuous imputation, even though Diff-Joint does not always have the best performance, it remains competitive and stays as the second-best result in most cases, with very little difference from the best-performing method.

This demonstrates that Diff-Joint enjoys a much better tradeoff between imputation performance for continuous features and the overall accuracy for discrete features.

Furthermore, we also evaluate downstream predictive performance in Table 5, where each method is used as a preprocessing step before predicting four target discrete outcome variables. Diff-Joint achieves consistently stronger Macro-F1 and ROC-AUC across the outcomes, with especially large gains on predicting `Critical` and `ICU transfer 12h`. This suggests that preserving meaningful missingness provides useful predictive signal for clinically severe outcomes, rather than simply improving cell-level imputation.

Table 4: Evaluation on MIMIC-IV-ED under MCAR.

Method	Ratio (%)	$\text{MAE}_{\text{out}} \downarrow$	$\text{RMSE}_{\text{out}} \downarrow$	ACC_{out} (%) $\uparrow$
Diff-Joint	10	4.59 ± 0.03	8.31 ± 0.07	$\mathbf{72.00 \pm 0.50}$
DiffPuter	10	4.61 ± 0.04	8.30 ± 0.08	61.54 ± 0.78
CACTI	10	$\mathbf{4.55 \pm 0.03}$	$\mathbf{8.24 \pm 0.07}$	63.85 ± 0.61
Mean/Mode	10	6.32 ± 0.00	11.39 ± 0.00	57.98 ± 0.00
missForest	10	4.69 ± 0.02	8.60 ± 0.03	62.29 ± 0.12
Diff-Joint	20	4.78 ± 0.03	8.68 ± 0.04	$\mathbf{72.46 \pm 0.20}$
DiffPuter	20	4.80 ± 0.04	8.68 ± 0.07	69.00 ± 0.71
CACTI	20	$\mathbf{4.73 \pm 0.03}$	$\mathbf{8.60 \pm 0.06}$	70.48 ± 0.59
Mean/Mode	20	6.31 ± 0.00	11.37 ± 0.00	65.32 ± 0.00
missForest	20	5.02 ± 0.02	9.23 ± 0.03	69.63 ± 0.14
Diff-Joint	30	4.99 ± 0.03	9.05 ± 0.05	71.84 ± 0.66
DiffPuter	30	5.01 ± 0.03	9.13 ± 0.07	71.39 ± 0.84
CACTI	30	$\mathbf{4.94 \pm 0.03}$	$\mathbf{9.04 \pm 0.06}$	$\mathbf{72.71 \pm 0.57}$
Mean/Mode	30	6.31 ± 0.00	11.38 ± 0.00	68.26 ± 0.00
missForest	30	5.36 ± 0.02	9.86 ± 0.04	72.07 ± 0.15
Diff-Joint	40	5.20 ± 0.03	9.39 ± 0.05	69.54 ± 0.52
DiffPuter	40	5.27 ± 0.03	9.50 ± 0.06	73.11 ± 0.89
CACTI	40	$\mathbf{5.05 \pm 0.03}$	$\mathbf{9.25 \pm 0.06}$	73.52 ± 0.64
Mean/Mode	40	6.32 ± 0.00	11.39 ± 0.00	69.77 ± 0.00
missForest	40	5.70 ± 0.02	10.44 ± 0.04	$\mathbf{73.92 \pm 0.17}$

Results on the HELOC Dataset. We further evaluate Diff-Joint on HELOC, a credit-risk dataset with native semantic-missing states. We treat the special code `-7` ("Condition not Met") in two credit-history variables as meaningful missingness; dataset details are provided in Appendix C.3. Table 12 reports the complete results. Across observation-induced masking ratios of 10%–40%, Diff-Joint achieves the highest ACC among the compared methods, with MM recall ranging from 83.83% to 88.04%. It also achieves the lowest RMSE at 10% and 20% masking. However, its continuous-value imputation errors are higher than those of both baselines at 30% and 40%.

Results on the Hotel Booking Demand Dataset. We also evaluate Diff-Joint on Hotel Booking Demand, where native `NULL` entries in the `Agent` column are operationally interpreted as indicating that no agency ID is applicable. The dataset and this MM-labeling convention are described in Appendix C.4. As shown in Table 13, Diff-Joint achieves the highest ACC among the compared methods at all four masking ratios, while recovering MM entries with recall ranging from 62.26% to 84.98%. Its MAE and RMSE are nevertheless higher than those of DiffPuter and CACTI throughout.

5 Conclusion

We introduced Diff-Joint, an uncertainty-aware diffusion framework for selective imputation that distinguishes meaningfully missing (MM) entries from observation-induced missing entries. By jointly modeling tabular values and MM masks, Diff-Joint learns when to preserve na as a semantic state and when to recover a regular value. Experiments on a synthetic Bayesian-network dataset and real-world datasets show that the proposed method effectively identifies meaningful missingness and yields strong downstream predictive performance.

One limitation of the current framework is that it assumes a known set $\mathcal{T}$ of MM candidate columns. Future work could explore semi-supervised approaches to identifying these columns, relaxing the requirement that $\mathcal{T}$ be specified in advance. A second limitation is its reliance on a detectable uncertainty gap between MM and observation-induced missing entries. Although our empirical results suggest that this gap is stable across the missingness mechanisms considered, further work could characterize when such separation holds and establish identifiability under weaker assumptions. Additionally, the iterative diffusion procedure is also more computationally expensive than non-iterative baselines. However, its cost can be controlled through the user-specified number of iterations, and our ablation study shows that only a small number of iterations is often sufficient. Future work

Table 5: Downstream performance on MIMIC-IV-ED across four target outcome variables.

Method	Ratio	Hospitalization				Critical			
		Macro-F1	ROC-AUC	Weighted-F1	ACC	Macro-F1	ROC-AUC	Weighted-F1	ACC
Diff-Joint	10	**49.14**	**70.98**	**43.71**	61.66	**88.25**	**98.56**	**84.58**	**95.92**
Mean/Mode	10	40.45	66.37	34.03	58.97	51.28	76.26	34.35	88.56
missForest	10	45.36	68.38	38.41	63.75	47.93	78.37	28.81	88.50
DiffPuter	10	45.18	68.01	38.19	63.62	48.06	79.12	29.18	88.44
CACTI	10	45.65	68.44	38.68	**64.11**	48.52	79.60	29.64	88.72
Diff-Joint	20	**48.99**	**70.36**	**44.34**	57.95	**90.11**	**98.35**	**86.79**	**96.87**
Mean/Mode	20	37.20	64.71	31.12	55.83	53.75	75.81	38.32	88.42
missForest	20	45.40	68.28	38.46	63.81	48.86	78.27	30.23	88.80
DiffPuter	20	45.03	67.94	38.08	63.54	48.47	78.91	30.12	88.52
CACTI	20	45.67	68.60	38.67	**64.11**	49.12	79.40	30.69	88.78
Diff-Joint	30	**49.24**	**71.42**	**45.22**	54.63	**85.82**	**98.70**	**80.84**	**96.04**
Mean/Mode	30	34.95	63.81	29.11	54.52	54.01	75.36	38.76	88.42
missForest	30	45.58	68.69	38.58	**64.19**	48.02	78.18	28.88	88.63
DiffPuter	30	44.91	67.88	37.94	63.41	48.19	79.08	29.57	88.31
CACTI	30	45.50	68.30	38.53	63.88	48.72	79.71	30.10	88.59
Diff-Joint	40	43.32	**70.37**	**40.36**	45.21	**81.79**	**97.86**	**75.29**	**95.14**
Mean/Mode	40	31.54	63.15	26.05	52.22	52.25	76.43	36.23	88.14
missForest	40	45.46	68.64	38.44	64.07	48.24	78.19	29.26	88.60
DiffPuter	40	45.12	68.07	38.16	63.73	48.56	79.21	29.88	88.47
CACTI	40	**45.73**	68.60	38.74	**64.21**	49.08	79.76	30.54	88.90

Method	Ratio	ICU Transfer 12h				CCI CHF			
		Macro-F1	ROC-AUC	Weighted-F1	ACC	Macro-F1	ROC-AUC	Weighted-F1	ACC
Diff-Joint	10	**90.99**	**99.65**	**87.81**	**97.32**	**47.92**	**83.44**	**33.32**	**80.60**
Mean/Mode	10	51.66	76.88	34.71	88.71	36.64	73.24	19.13	79.59
missForest	10	46.79	77.55	26.73	88.46	42.30	74.19	26.75	80.20
DiffPuter	10	47.64	80.21	28.42	88.19	42.76	73.91	27.21	79.84
CACTI	10	48.11	80.77	28.99	88.48	43.15	74.40	27.97	80.23
Diff-Joint	20	**90.65**	**99.54**	**87.41**	**97.20**	**47.50**	**82.62**	**32.58**	**80.13**
Mean/Mode	20	53.25	74.85	37.31	88.74	35.53	73.22	17.66	79.45
missForest	20	46.94	77.59	27.07	88.32	42.42	74.00	27.25	79.72
DiffPuter	20	48.51	79.88	29.96	88.11	42.91	73.16	27.84	79.71
CACTI	20	49.03	80.34	30.62	88.35	43.53	73.60	28.60	80.05
Diff-Joint	30	**88.91**	**99.63**	**84.71**	**97.32**	**48.08**	**82.69**	**33.32**	79.32
Mean/Mode	30	51.72	76.17	35.17	88.52	33.61	71.50	15.06	79.12
missForest	30	46.92	78.63	27.06	88.31	42.02	72.15	26.46	79.99
DiffPuter	30	46.88	79.31	27.21	88.06	42.54	73.22	27.36	79.87
CACTI	30	47.32	79.84	27.70	88.38	43.27	73.85	28.09	**80.26**
Diff-Joint	40	**85.09**	**99.37**	**79.42**	**96.43**	**46.30**	**82.44**	**30.84**	78.92
Mean/Mode	40	50.38	76.49	33.31	88.22	34.44	73.06	16.20	79.19
missForest	40	48.52	77.94	29.17	89.35	41.90	72.09	26.27	79.93
DiffPuter	40	47.41	80.46	28.02	88.39	42.08	73.31	26.79	79.66
CACTI	40	48.03	81.03	28.69	88.72	42.67	73.84	27.28	**80.12**

will strengthen theoretical guarantees, improve scalability, and evaluate the framework on more real-world datasets.

6 Acknowledgement

The work of Shixiang Zhu was partially supported by the National Science Foundation (Grant No. CAIG-2425888). The work of Guang Cheng was partially supported by the Office of Naval Research (Grant No. N00014-26-12466), a gift from CISCO, and the National Science Foundation (Grant No. CNS-2247795).

References

[1] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[2] Tlamelo Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago, and Oteng Tabona. A survey on missing data in machine learning. *Journal of Big data*, 8(1):140, 2021.

[3] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[4] Donald B Rubin. An overview of multiple imputation. In *Proceedings of the survey research methods section of the American statistical association*, volume 79, page 84, 1988.

[5] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.

[6] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[7] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5689–5698. PMLR, 2018.

[8] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pages 4413–4423. PMLR, 2019.

[9] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816, 2021.

[10] Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. In *NeurIPS 2022 First Table Representation Learning Workshop*, 2022.

[11] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[12] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[13] Paul D Allison. Missing data. *The SAGE handbook of quantitative methods in psychology*, 23:72–89, 2009.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. IEEE, 2008.

[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

[18] Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. DiffPuter: Empowering diffusion models for missing data imputation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[19] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2019.

[20] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. HyperImpute: Generalized iterative imputation with automatic model selection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9916–9937. PMLR, 2022.

[21] Tianyu Du, Luca Melis, and Ting Wang. Remasker: Imputing tabular data with masked autoencoding. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Aditya Gorla, Ryan Wang, Zhengtong Liu, Ulzee An, and Sriram Sankararaman. CACTI: Leveraging copy masking and contextual information to improve tabular data imputation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 20187–20225, 2025.

[23] Yidong Ouyang, Liyan Xie, Chongxuan Li, and Guang Cheng. Missdiff: Training diffusion models on tabular data with missing values. *arXiv preprint arXiv:2307.00467*, 2023.

[24] Namjoon Suh, Yuning Yang, Din-Yin Hsieh, Qitong Luan, Shirong Xu, Shixiang Zhu, and Guang Cheng. Timeautodiff: A unified framework for generation, imputation, forecasting, and time-varying metadata conditioning of heterogeneous time series tabular data. *Transactions on Machine Learning Research*, 2025.

[25] 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*, pages 1288–1296. PMLR, 2024.

[26] Pedro J. García-Laencina, José-Luis Sancho-Gómez, and Aníbal R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.

[27] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[28] Duong Nguyen, Nghia Hoang, Thanh Trung Huynh, Quoc Viet Hung Nguyen, and Phi Le Nguyen. Learning reconfigurable representations for multimodal federated learning with missing data. *Advances in Neural Information Processing Systems*, 38:28808–28845, 2025.

[29] Xinyue Wang, Hafiz Asif, and Jaideep Vaidya. Preserving missing data distribution in synthetic data. In *Proceedings of the ACM Web Conference 2023*, pages 2110–2121, 2023.

[30] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[31] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[32] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in artificial intelligence*, pages 574–584. PMLR, 2020.

[33] Chao Ma and Cheng Zhang. Identifiable generative models for missing not at random data imputation. *Advances in Neural Information Processing Systems*, 34:27645–27658, 2021.

[34] Wang Miao, Peng Ding, and Zhi Geng. Identifiability of normal and normal mixture models with nonignorable missing data. *Journal of the American Statistical Association*, 111(516):1673–1683, 2016.

[35] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. MIMIC-IV-ED. *PhysioNet*, 2021.

[36] Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond iid: How general are tabular foundation models, really?, 2026.

[37] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[38] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17564–17579. PMLR, 23–29 Jul 2023.

[39] Ulzee An, Ali Pazokitoroudi, Marcus Alvarez, Lianyun Huang, Silviu Bacanu, Andrew J. Schork, Kenneth Kendler, Päivi Pajukanta, Jonathan Flint, Noah Zaitlen, Na Cai, Andy Dahl, and Sriram Sankararaman. Deep learning-based phenotype imputation on population-scale biobank data increases genetic discoveries. *Nature Genetics*, 55:2269–2276, 2023.

[40] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, 2020.

A Algorithm Details

We provide the detailed algorithms for the MODEL–UPDATE step (Algorithm 2), CONDITIONAL–SAMPLE step (Algorithm 3), and the AGGREGATE step (Algorithm 4) used in Algorithm 1.

Algorithm 2 MODEL–UPDATE: Model Update

Input: Current joint representation $\widehat{S}^{(t)} = (\widehat{X}^{(t)}, \widehat{C}^{(t)})$.
Output: Updated diffusion model parameters $\boldsymbol{\theta}^{(t)}$.

while not converged **do**
  Sample a joint training example $\mathbf{s} \sim p_{\widehat{S}^{(t)}}(\mathbf{s})$.
  Sample diffusion time $t_{\text{diff}} \sim p(t)$.
  Sample Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  Form the noisy sample $\mathbf{s}_{t_{\text{diff}}} = \mathbf{s} + \sigma(t_{\text{diff}})\boldsymbol{\varepsilon}$.
  Compute the score-matching loss

$$\ell(\boldsymbol{\theta}) = \left\| \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{s}_{t_{\text{diff}}}, t_{\text{diff}}) - \frac{-\boldsymbol{\varepsilon}}{\sigma(t_{\text{diff}})} \right\|_2^2 .$$

  Update $\boldsymbol{\theta}$ via Adam.
end while
return $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}$.

B Theoretical Properties

We provide several basic theoretical properties of the proposed framework.

B.1 Proof of Proposition 3.1

Proof. Case 1. For each coordinate j, the observation process defines a known channel K_j from $\mathbf{x}_j^{\text{true}}$ to $\mathbf{x}_j^{\text{obs}}$:

$$K_j(\mathbf{x}_j^{\text{obs}} \mid \mathbf{x}_j^{\text{true}}) = \begin{cases} 1, & \mathbf{x}_j^{\text{true}} = \text{na},\ \mathbf{x}_j^{\text{obs}} = \text{na}, \\ 1 - p_j, & \mathbf{x}_j^{\text{true}} \in \mathcal{X}_j,\ \mathbf{x}_j^{\text{obs}} = \mathbf{x}_j^{\text{true}}, \\ p_j, & \mathbf{x}_j^{\text{true}} \in \mathcal{X}_j,\ \mathbf{x}_j^{\text{obs}} = \text{na}, \\ 0, & \text{otherwise.} \end{cases}$$

Since $p_j < 1$, the channel K_j is injective. Indeed, for any $a \in \mathcal{X}_j$,

$$\mathbf{P}(\mathbf{x}_j^{\text{obs}} = a) = (1 - p_j)\mathbf{P}(\mathbf{x}_j^{\text{true}} = a),$$

Algorithm 3 CONDITIONAL–SAMPLE

Input: Diffusion model $\boldsymbol{\theta}^{(t)}$, observed data X^{obs}, observation mask Ω, number of samples K, number of reverse steps M. $\widetilde{\boldsymbol{\omega}}$ denotes the observation mask in the joint-state space.
Output: Monte Carlo sample collection $\mathcal{S}^{(t)} = \{(X_k, C_k)\}_{k=1}^{K}$.

for $k = 1$ to K **do**
  Sample initial noise $\tilde{\mathbf{s}}_{t_M}^{(k)} \sim \mathcal{N}(\mathbf{0}, \sigma^2(t_M)\mathbf{I})$.
  for $i = M, M-1, \ldots, 1$ **do**
    Sample Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
    Compute forward-noised observed part: $\mathbf{s}_{t_{i-1}}^{\text{forward},(k)} = \mathbf{s}^{\text{obs}} + \sigma(t_{i-1})\boldsymbol{\varepsilon}$.
    Compute reverse update using the learned diffusion model:

$$\mathbf{s}_{t_{i-1}}^{\text{reverse},(k)} = \text{REVERSEUPDATE}_{\boldsymbol{\theta}^{(t)}}(\tilde{\mathbf{s}}_{t_i}^{(k)}, t_i, t_{i-1}).$$

    Merge observed and missing coordinates:

$$\tilde{\mathbf{s}}_{t_{i-1}}^{(k)} = \widetilde{\boldsymbol{\omega}} \odot \mathbf{s}_{t_{i-1}}^{\text{forward},(k)} + (1 - \widetilde{\boldsymbol{\omega}}) \odot \mathbf{s}_{t_{i-1}}^{\text{reverse},(k)}.$$

  end for
  Decode $\tilde{\mathbf{s}}_{t_0}^{(k)}$ into completed sample (X_k, C_k).
end for
return $\mathcal{S}^{(t)} = \{(X_k, C_k)\}_{k=1}^{K}$.

Algorithm 4 AGGREGATE: Aggregate Monte Carlo Samples

Input: Observed Data X^{obs}, observation mask Ω, Monte Carlo sample collection $\mathcal{S} = \{(\mathbf{x}_i^{(k)}, \mathbf{c}_i^{(k)})\}_{i\in[n],k\in[K]}$, MM search column set $\mathcal{T}$.
Output: Updated joint state $(\widehat{X}', \widehat{C}')$

Initialize $\widehat{X}' \leftarrow X^{\text{obs}}, \widehat{C}' \leftarrow \mathbf{0}_{n\times d}, \mathcal{M} \leftarrow \{(i,j) \mid i \in [n], j \in [d], \omega_{i,j} = 0\}$.
for all $(i,j) \in \mathcal{M}$ **do**
  Compute $\widehat{x}_{i,j}$, $u_{i,j}$, according to (5), (6).
  Set $\widehat{X}'_{i,j} \leftarrow \widehat{x}_{i,j}$.
end for
for all $j \in \mathcal{T}$ **do**
  Let $\mathcal{M}_j \leftarrow \{i : (i,j) \in \mathcal{M}\}$.
  Run k-means clustering with $k = 2$ on $\{u_{i,j} : i \in \mathcal{M}_j\}$ to obtain two clusters.
  Identify the high-uncertainty cluster $\mathcal{C}_{1,j}$.
  for all $i \in \mathcal{M}_j$ **do**
    Set $\widehat{C}'_{i,j} \leftarrow \widehat{c}_{i,j}$ according to (7).
  end for
end for
return $(\widehat{X}', \widehat{C}')$.

and hence

$$\mathbf{P}(\mathbf{x}_j^{\text{true}} = a) = \frac{\mathbf{P}(\mathbf{x}_j^{\text{obs}} = a)}{1 - p_j}.$$

Moreover,

$$\mathbf{P}(\mathbf{x}_j^{\text{obs}} = \text{na}) = \mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na}) + p_j \mathbf{P}(\mathbf{x}_j^{\text{true}} \in \mathcal{X}_j) = \mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na}) + p_j(1 - \mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na})),$$

so $\mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na})$ is also uniquely determined by the distribution of $\mathbf{x}_j^{\text{obs}}$ and the known value of p_j.

For the full vector, the observation channel is

$$K = K_1 \otimes \cdots \otimes K_d.$$

Since each K_j is injective, the product channel K is also injective. Therefore the complete-data distribution $\mathbf{P}(\mathbf{x}^{\text{true}})$ is uniquely determined by the observed-data distribution $\mathbf{P}(\mathbf{x}^{\text{obs}})$.

Now fix a coordinate j. Since

$$c_j = \mathbb{1}\{\mathbf{x}_j^{\text{true}} = \text{na}\},$$

we have, for any z such that $\mathbf{P}(\mathbf{x}_{-j}^{\text{true}} = z) > 0$,

$$\phi_\theta(z) = \mathbf{P}(c_j = 1 \mid \mathbf{x}_{-j}^{\text{true}} = z) = \mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na} \mid \mathbf{x}_{-j}^{\text{true}} = z).$$

The right-hand side can be written as

$$\mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na} \mid \mathbf{x}_{-j}^{\text{true}} = z) = \frac{\mathbf{P}(\mathbf{x}_j^{\text{true}} = \text{na},\ \mathbf{x}_{-j}^{\text{true}} = z)}{\mathbf{P}(\mathbf{x}_{-j}^{\text{true}} = z)}.$$

Both numerator and denominator are determined by the complete-data distribution $\mathbf{P}(\mathbf{x}^{\text{true}})$, which has already been shown to be identifiable from $\mathbf{P}(\mathbf{x}^{\text{obs}})$. Therefore $\phi_\theta(z)$ is identifiable for every z in the support of $\mathbf{x}_{-j}^{\text{true}}$.

Therefore ϕ_θ is identifiable: if another parameter θ' gives the same observed-data distribution, then it must satisfy $\phi_{\theta'}(z) = \phi_\theta(z)$ for all z with $\Pr(\mathbf{x}_{-j}^{\text{true}} = z) > 0$. By the assumed identifiability of the parametric family $\{\phi_\theta\}$, this implies $\theta' = \theta$. Therefore the meaningful-missingness mechanism is identifiable from the observed data.

Case 2. Recall that ω_j denotes the final observation indicator. First consider any $j \notin S$. By assumption, $c_j = 0$ almost surely. Therefore the only possible source of observed missingness in column j is observation-induced missingness. Hence $\omega_j = 1 - r_j$. Since $r_j \sim \text{Bernoulli}(p)$, we have $\mathbf{P}(r_j = 1) = p$. The left-hand side is determined by the observed-data distribution, so p is identifiable.

Now consider $j \in S$. By definition, we have $\omega_j = 0$ implies $c_j = 1$ or $r_j = 1$. Since r_j is MCAR and independent of $(c_j, \mathbf{x}_{-j}^{\text{true}})$,

$$\mathbf{P}(\omega_j = 0 \mid \mathbf{x}_{-j}^{\text{true}} = z) = \mathbf{P}(c_j = 1 \mid \mathbf{x}_{-j}^{\text{true}} = z) + \mathbf{P}(c_j = 0, r_j = 1 \mid \mathbf{x}_{-j}^{\text{true}} = z).$$

Thus

$$\mathbf{P}(\omega_j = 0 \mid \mathbf{x}_{-j}^{\text{true}} = z) = \phi_\theta(z) + p\{1 - \phi_\theta(z)\}.$$

Equivalently,

$$\mathbf{P}(\omega_j = 0 \mid \mathbf{x}_{-j}^{\text{true}} = z) = p + (1 - p)\phi_\theta(z).$$

Because $p < 1$ is identifiable from the non-MM columns, we obtain

$$\phi_\theta(z) = \frac{\mathbf{P}(\omega_j = 0 \mid \mathbf{x}_{-j}^{\text{true}} = z) - p}{1 - p}.$$

The right-hand side is identifiable from the observed-data distribution. Therefore ϕ_θ is identifiable on the support of $\mathbf{x}_{-j}^{\text{true}}$.

Finally, suppose another parameter θ' gives the same observed-data distribution. Then it gives the same function $\phi_{\theta'}(z)$ on the support of $\mathbf{x}_{-j}^{\text{true}}$. Hence $\phi_{\theta'}(z) = \phi_\theta(z)$ for all z in the support. By the identifiability assumption on the parametric family, this implies $\theta' = \theta$. Therefore θ is identifiable. □

B.2 More Discussions

Discussion of Remark 3.2 According to proof under Case 2, we obtain

$$\phi_\theta(z) = \sigma(\alpha_j^\top z + \beta_j) = \frac{\mathbf{P}(\omega_j = 0 \mid \mathbf{x}_{-j}^{\text{true}} = z) - p}{1 - p}.$$

In typical applications, the sample size is much larger than the number of covariates. Moreover, under a mild assumption that for any (α, β),

$$\alpha^\top Z + \beta = 0 \quad \text{a.s.}$$

implies $\alpha = \mathbf{0}$ and $\beta = 0$ [2], the linear predictor $\alpha_j^\top z + \beta_j$ is uniquely determined, and hence (α_j, β_j) is unique.

[2]This assumption can be induced by the general assumption $\phi_\theta \neq \phi_{\theta'}$ for any $\theta \neq \theta'$.

Discussion of Remark 3.3 We provide a more in-depth discussion of the uncertainty separation exhibited in the first iteration. For a distribution P on $\mathcal{X}_j$, define the population uncertainty functional

$$\mathcal{U}_j(P) = \begin{cases} -\sum_{v \in \mathcal{X}_j} P(v) \log P(v), & \text{if feature } j \text{ is discrete,} \\ \sqrt{\mathrm{Var}_{Y \sim P}(Y)}, & \text{if feature } j \text{ is continuous.} \end{cases}$$

Proposition B.1 (First-iteration weak separation)**.** *Fix a candidate feature j. For each random sample $\mathbf{x}^{\mathrm{obs}} := \{x_1^{\mathrm{obs}}, \ldots, x_d^{\mathrm{obs}}\}$, let $\mathcal{F}_j^{\mathrm{obs}} = \left(\mathbf{x}_{-j}^{\mathrm{obs}}, \boldsymbol{\omega}_{-j}\right)$ be the observed context excluding the target coordinate j. Suppose that the observation-induced missingness is MCAR, $r_j \sim \mathrm{Bernoulli}(\rho_j)$, and that the meaningful-missingness mechanism is MAR with respect to the observed context:*

$$\mathbf{P}(c_j = 1 \mid \mathcal{F}_j^{\mathrm{obs}}) = \pi_j(\mathcal{F}_j^{\mathrm{obs}}).$$

Let $\bar{\pi}_j = \mathbb{E}\left[\pi_j(\mathcal{F}_j^{\mathrm{obs}})\right]$. Define the population conditional distribution of the initialized first-round value by

$$P_j^{(0)}(A \mid F) := \mathbf{P}\left(\widehat{X}_j^{(0)} \in A \mid \mathcal{F}_j^{\mathrm{obs}} = F\right), \qquad A \subseteq \mathcal{X}_j.$$

The corresponding first-iteration population uncertainty score for the j-th coordinate is $U_j^{(1)} := \mathcal{U}_j\left(P_j^{(0)}(\cdot \mid \mathcal{F}_j^{\mathrm{obs}})\right)$. Assume that $\rho_j > 0$, $U_j^{(1)}$ has finite first moment and $\mathrm{Cov}\left(\pi_j(\mathcal{F}_j^{\mathrm{obs}}), U_j^{(1)}\right) > 0$. Then the first-iteration population uncertainty score is larger on average for meaningfully missing entries than for observation-induced missing entries:

$$\mathbb{E}\left[U_j^{(1)} \mid c_j = 1\right] > \mathbb{E}\left[U_j^{(1)} \mid c_j = 0, r_j = 1\right].$$

Proof. By definition, $U_j^{(1)} = \mathcal{U}_j\left(P_j^{(0)}(\cdot \mid \mathcal{F}_j^{\mathrm{obs}})\right)$, so $U_j^{(1)}$ is measurable with respect to $\mathcal{F}_j^{\mathrm{obs}}$. Since $\mathbf{P}(c_j = 1 \mid \mathcal{F}_j^{\mathrm{obs}}) = \pi_j(\mathcal{F}_j^{\mathrm{obs}})$, we have

$$\mathbb{E}\left[U_j^{(1)} \mid c_j = 1\right] = \frac{\mathbb{E}\left[U_j^{(1)} \mathbb{1}\{c_j = 1\}\right]}{\mathbf{P}(c_j = 1)} = \frac{\mathbb{E}\left[U_j^{(1)} \mathbf{P}(c_j = 1 \mid \mathcal{F}_j^{\mathrm{obs}})\right]}{\bar{\pi}_j} = \frac{\mathbb{E}\left[\pi_j(\mathcal{F}_j^{\mathrm{obs}}) U_j^{(1)}\right]}{\bar{\pi}_j}.$$

Similarly, since r_j is MCAR and independent of $(c_j, \mathcal{F}_j^{\mathrm{obs}})$,

$$\begin{aligned} \mathbb{E}\left[U_j^{(1)} \mid c_j = 0, r_j = 1\right] = \mathbb{E}\left[U_j^{(1)} \mid c_j = 0\right] &= \frac{\mathbb{E}\left[U_j^{(1)} \mathbb{1}\{c_j = 0\}\right]}{\mathbf{P}(c_j = 0)} \\ &= \frac{\mathbb{E}\left[U_j^{(1)} \left(1 - \pi_j(\mathcal{F}_j^{\mathrm{obs}})\right)\right]}{1 - \bar{\pi}_j}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\mathbb{E}\left[U_j^{(1)} \mid c_j = 1\right] - \mathbb{E}\left[U_j^{(1)} \mid c_j = 0, r_j = 1\right] = \frac{\mathbb{E}\left[\pi_j(\mathcal{F}_j^{\mathrm{obs}}) U_j^{(1)}\right]}{\bar{\pi}_j} - \frac{\mathbb{E}\left[\left(1 - \pi_j(\mathcal{F}_j^{\mathrm{obs}})\right) U_j^{(1)}\right]}{1 - \bar{\pi}_j} \\ &= \frac{\mathbb{E}\left[\pi_j(\mathcal{F}_j^{\mathrm{obs}}) U_j^{(1)}\right] - \bar{\pi}_j \mathbb{E}\left[U_j^{(1)}\right]}{\bar{\pi}_j(1 - \bar{\pi}_j)} = \frac{\mathrm{Cov}\left(\pi_j(\mathcal{F}_j^{\mathrm{obs}}), U_j^{(1)}\right)}{\bar{\pi}_j(1 - \bar{\pi}_j)}. \end{aligned}$$

□

By the results in Proposition B.1, there exists a threshold τ_j such that

$$\mathbf{P}\left(U_j^{(1)} > \tau_j \mid c_j = 1\right) > \mathbf{P}\left(U_j^{(1)} > \tau_j \mid c_j = 0, r_j = 1\right).$$

For such a threshold, the high-uncertainty group can thus be distinguished from the observation-induced missing group.

Remark B.2 (Interpretation of the covariance condition). The covariance condition in Proposition B.1 is natural under the random uniform initialization used by Diff-Joint. Let G_j denote the empirical initialization distribution $\mathrm{Unif}(\widehat{\mathcal{X}}_j)$ in Appendix D.1. For any measurable set $A \subseteq \mathcal{X}_j$, define

$$Q_j(A \mid F) := \mathbf{P}(X_j \in A \mid c_j = 0, \mathcal{F}_j^{\mathrm{obs}} = F),$$

the context-specific conditional distribution. Since every observed missing entry is initially filled from G_j, while an observed regular entry keeps its true value, the first-round initialized distribution satisfies

$$P_j^{(0)}(\cdot \mid F) = (1 - \alpha_j(F))Q_j(\cdot \mid F) + \alpha_j(F)G_j, \qquad \alpha_j(F) = \rho_j + (1 - \rho_j)\pi_j(F).$$

Thus, the mixture weight on the random-initialization component increases with the MM propensity $\pi_j(F)$. The initialization distribution G_j ignores the sample-specific context F and spreads mass over the column-level regular-value support, whereas $Q_j(\cdot \mid F)$ is constrained by the observed context and is typically more concentrated when the regular value is recoverable. Therefore, higher-MM-propensity contexts tend to induce larger first-iteration uncertainty, $U_j^{(1)} = \mathcal{U}_j(P_j^{(0)}(\cdot \mid \mathcal{F}_j^{\mathrm{obs}}))$, giving a positive association between $\pi_j(\mathcal{F}_j^{\mathrm{obs}})$ and $U_j^{(1)}$.

B.3 Non-identifiability under MAR and MNAR

The following counterexamples show that MAR or MNAR assumptions alone do not guarantee identification of meaningful missingness when the observation-induced mechanism is unknown. Therefore, additional assumptions or information are needed to ensure identifiability in these settings. Let $C = 1$ denote meaningful missingness, so that $X^{\text{true}} = \mathrm{na}$. The observed value X^{obs} is na whenever the entry is meaningfully missing or a regular value is masked.

Counterexample 1: MAR. Let $Z \sim \mathrm{Bernoulli}(1/2)$ be fully observed, with $\Pr(C = 1 \mid Z = z) = \pi$ and $X^{\text{true}} \mid (C = 0, Z = z) \sim \mathrm{Bernoulli}(1/2)$. Suppose observation-induced masking depends only on Z, with probability $p(z)$; hence, it is MAR. Consider

	π	$p(0)$	$p(1)$
Model A	0.2	0.4	0.7
Model B	0.4	0.2	0.6

For either model and $x \in \{0, 1\}$,

$$\Pr(X^{\mathrm{obs}} = x \mid Z = z) = \frac{1 - \pi}{2}\{1 - p(z)\}, \qquad \Pr(X^{\mathrm{obs}} = \mathrm{na} \mid Z = z) = \pi + (1 - \pi)p(z).$$

Substitution gives the same conditional distribution under both models:

	$X^{\mathrm{obs}} = 0$	$X^{\mathrm{obs}} = 1$	$X^{\mathrm{obs}} = \mathrm{na}$
$Z = 0$	0.24	0.24	0.52
$Z = 1$	0.12	0.12	0.76

Thus, the joint distribution of (Z, X^{obs}) is identical, although the MM probabilities are 0.2 and 0.4. Therefore, MAR alone does not identify the MM mechanism.

Counterexample 2: MNAR. Let $\Pr(C = 1) = \pi$ and $X^{\text{true}} \mid (C = 0) \sim \mathrm{Bernoulli}(1/2)$. Write $q(x)$ for the probability of masking a non-MM entry whose regular value is x. Consider

	π	$q(0)$	$q(1)$
Model A	0.2	0.5	0.75
Model B	0.4	1/3	2/3

Both models exhibit MNAR, with stronger masking for $X^{\text{true}} = 1$. Since

$$\Pr(X^{\mathrm{obs}} = x) = \frac{1 - \pi}{2}\{1 - q(x)\}, \qquad x \in \{0, 1\},$$

both models yield

$$\Pr(X^{\mathrm{obs}} = 0) = 0.2, \qquad \Pr(X^{\mathrm{obs}} = 1) = 0.1, \qquad \Pr(X^{\mathrm{obs}} = \mathrm{na}) = 0.7.$$

The observed-data distributions are therefore identical despite different MM probabilities, proving non-identifiability in this MNAR example.

These examples show that additional assumptions or information are needed to guarantee identification of the MM/observation-induced missingness decomposition. They do not imply that all MAR/MNAR settings are non-identifiable. General theoretical extensions beyond the analyzed MCAR regimes are left to future work.

C Dataset and Missingness Construction

This appendix describes the four datasets used in our experiments, as summarized in Table 6. For the Bayesian Network and MIMIC-IV-ED datasets, MM is constructed using pre-specified mechanisms; for HELOC and Hotel Booking Demand, reference MM labels are defined from native semantic-missing states under the conventions described below. Throughout the experiments, the MM candidate columns are assumed to be known, and MM identification is performed only on missing entries within these columns.

Table 6: Summary of the datasets used in the experiments. Feature and MM-candidate counts refer to raw columns before encoding.

Dataset	#Train	#Test	#Records	#Features	#MM candidate columns
Bayesian Network	14,000	6,000	20,000	5	3
MIMIC-IV-ED	353,150	88,287	441,437	30	4
HELOC	5,444	2,334	7,778	23	2
Hotel Booking Demand	56,992	24,426	81,418	26	1

C.1 Bayesian Network Synthetic Data Construction

We construct the synthetic tabular dataset from a Bayesian network so that both the underlying data-generating process and the meaningful-missingness mechanism are fully controlled. This synthetic data contains five variables: two continuous variables `C1` and `C2`, and three discrete variables `D1`, `D2`, and `D3`. The corresponding graph structure is shown in Figure 3.

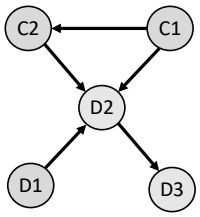

Figure 3: Bayesian network used to generate the synthetic tabular data. `C1` and `C2` are continuous variables, while `D1`, `D2`, and `D3` are discrete variables.

According to the underlying dependence in Figure 3, we first sample $\texttt{C1} \sim \mathcal{N}(25, 2)$. Next, conditional on `C1`, we generate

$$\texttt{C2} \mid \texttt{C1} \sim \mathcal{N}(0.1 \cdot \texttt{C1} + 50,\ 5).$$

We then generate $\texttt{D1} \sim \text{Bernoulli}(0.3)$, where $\text{Bernoulli}(\xi)$ denotes the Bernoulli distribution with mean ξ. The variable `D2` is generated conditionally on `C1`, `C2`, and `D1` as

$$\texttt{D2} \mid \texttt{C1}, \texttt{C2}, \texttt{D1} \sim \begin{cases} Ca(0.3, 0.6, 0.1), & \texttt{C1} > 26,\ \texttt{C2} > 55,\ \texttt{D1} = 1, \\ Ca(0.2, 0.3, 0.5), & \texttt{C1} > 26,\ \texttt{C2} \le 55,\ \texttt{D1} = 1, \\ Ca(0.7, 0.1, 0.2), & \texttt{C1} \le 26,\ \texttt{C2} > 55,\ \texttt{D1} = 1, \\ \text{na}, & \texttt{C1} \le 26,\ \texttt{C2} \le 55,\ \texttt{D1} = 1, \\ Ca(0.05, 0.05, 0.9), & \texttt{D1} = 0, \end{cases}$$

where $Ca(p_1, p_2, 1 - p_1 - p_2)$ denotes a categorical distribution over three regular categories.

Finally, D3 is generated conditionally on D2 as

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

After generating the five variables, we further introduce meaningful missingness on C1 by recoding large values of C1 into a semantic missing state:

$$\texttt{C1} = \begin{cases} \text{na}, & \texttt{C1} > 27, \\ \texttt{C1}, & \texttt{C1} \leq 27. \end{cases}$$

Overall, the data-generating distribution for this synthetic dataset has meaningfully missing variables in both the continuous variable C1 and two discrete variables D2, D3.

C.2 MIMIC-IV-ED Dataset

We extract the real-world dataset from MIMIC-IV-ED [35] by selecting a structured feature subset and then injecting synthetic meaningful missingness on pre-specified target columns.

Cohort and Feature Construction. Our real-data dataset is constructed from MIMIC-IV-ED using a selected subset of structured ED variables [35]. We retain 30 non-object columns, including 14 numerical variables and 16 discrete variables, covering triage measurements, last-recorded vital signs, prior utilization counts, clinical scores, outcomes, chief complaint indicators, and comorbidity indicators; the full list is given in Table 7.

The feature subset is chosen so that the recorded values can serve as reliable ground truth for controlled evaluation and so that the selected covariates support the construction of clinically meaningful MM mechanisms on a subset of target columns. Based on this subset, we later introduce synthetic meaningful missingness and additional observation-induced missingness to form the final dataset.

Table 7: Selected 30-column feature subset used to construct the MIMIC-IV-ED real-data dataset.

Group	Columns	Type
Triage vitals	triage_temperature, triage_heartrate, triage_resprate, triage_o2sat, triage_sbp, triage_dbp, triage_pain, triage_acuity	Continuous
Last recorded vitals	ed_temperature_last, ed_heartrate_last, ed_resprate_last, ed_o2sat_last, ed_sbp_last, ed_dbp_last	Continuous
Demographics / utilization	age, n_ed_365d, n_hosp_365d, n_icu_365d	Discrete
Clinical scores	score_CCI, score_NEWS	Discrete
Outcomes	outcome_hospitalization, outcome_critical, outcome_icu_transfer_12h	Discrete
Chief complaint indicators	chiefcom_chest_pain, chiefcom_shortness_of_breath, chiefcom_abdominal_pain, chiefcom_fever_chills	Discrete
Comorbidity indicators	cci_CHF, cci_Renal, eci_Pulmonary	Discrete

Meaningful-Missingness Mechanisms on MIMIC-IV-ED. After constructing the MIMIC-IV-ED cohort, we introduce synthetic meaningful missingness on a pre-specified subset of discrete variables. In the current dataset, the MM target columns are

outcome_critical, outcome_icu_transfer_12h, outcome_hospitalization, cci_CHF.

Table 8 reports the resulting MM rates, computed before adding the additional observation-induced missingness mask.

Table 8: Meaningful-missingness rates induced by the synthetic MM mechanisms on selected MIMIC-IV-ED target columns.

Feature	MM Rate (%)
`outcome_critical`	9.06
`outcome_icu_transfer_12h`	8.70
`outcome_hospitalization`	13.82
`cci_CHF`	15.17

MM on `outcome_critical`. We set $c_{i,j} = 1$ for $j =$ `outcome_critical` if $\texttt{triage_acuity}_i \leq 2$ and at least one of the following conditions holds:

$$\texttt{triage_o2sat}_i < 95, \qquad \texttt{triage_resprate}_i \geq 22, \qquad \texttt{triage_heartrate}_i \geq 110,$$

and set $c_{i,j} = 0$ otherwise.

MM on `outcome_icu_transfer_12h`. We set $c_{i,j} = 1$ for $j =$ `outcome_icu_transfer_12h` if $\texttt{triage_acuity}_i \leq 2$ and at least one of the following conditions holds:

$$\texttt{triage_o2sat}_i < 95, \qquad \texttt{triage_resprate}_i \geq 22, \qquad \texttt{n_icu_365d}_i \geq 1,$$

and set $c_{i,j} = 0$ otherwise.

MM on `outcome_hospitalization`. For hospitalization, we define the score

$$\begin{aligned} s_i^{\text{hosp}} =& \mathbf{1}\{\texttt{triage_acuity}_i \leq 2\} + \mathbf{1}\{\texttt{score_NEWS}_i \geq 2\} \\ &+ \mathbf{1}\{\texttt{age}_i \geq 70\} + \mathbf{1}\{\texttt{n_hosp_365d}_i \geq 1\}, \end{aligned}$$

and map it to a missingness probability $p_i^{\text{hosp}} = \text{clip}\Big(0.02 + 0.10\, s_i^{\text{hosp}},\, 0,\, 0.45\Big)$. We then sample

$$c_{i,j} \sim \text{Bernoulli}\Big(p_i^{\text{hosp}}\Big), \qquad j = \texttt{outcome_hospitalization}.$$

MM on `cci_CHF`. For the CHF comorbidity indicator, we define

$$\begin{aligned} s_i^{\text{CHF}} =& \mathbf{1}\{\texttt{age}_i \geq 75\} + \mathbf{1}\{\texttt{n_hosp_365d}_i \geq 2\} \\ &+ \mathbf{1}\{\texttt{triage_sbp}_i < 110\} + \mathbf{1}\{\texttt{score_CCI}_i \geq 4\}, \end{aligned}$$

and map it to a probability $p_i^{\text{CHF}} = \text{clip}\big(0.03 + 0.16\, s_i^{\text{CHF}},\, 0,\, 0.70\big)$. We then sample

$$c_{i,j} \sim \text{Bernoulli}\big(p_i^{\text{CHF}}\big), \qquad j = \texttt{cci_CHF}.$$

C.3 HELOC Dataset

HELOC is a credit-risk dataset obtained from the BeyondArena benchmark. After removing records containing the special codes `-8` or `-9`, it contains 7,778 records and 23 features. We treat the native `-7` ("Condition not Met") state in `MSinceMostRecentDelq` and `MSinceMostRecentInqexcl7days` as MM, representing the absence of an applicable elapsed-time value. The MM proportions in these two columns are 46.4% and 18.9%, respectively.

C.4 Hotel Booking Demand Dataset

Hotel Booking Demand is a hotel-reservation dataset obtained from the BeyondArena benchmark. After preprocessing, it contains 81,418 records and 26 features. We use `Agent` as the MM candidate column and operationally interpret its 8,886 native `NULL` entries (10.9% of records) as indicating that no agency ID is applicable.

C.5 Observation-induced Missingness Mechanisms

For both the synthetic and real-data datasets, we introduce an additional observation-induced ordinary missingness mask $R = (r_{i,j})$ on top of the underlying samples. Depending on the experiment, this mask is generated under MCAR, MAR, or MNAR mechanisms as defined below.

Missing Completely at Random (MCAR). Under the MCAR mechanism, each entry is masked independently with a fixed probability p, regardless of the data values. Formally, for each sample i and feature j, the mask variable $r_{i,j}$ is generated as

$$\mathbb{P}(r_{i,j} = 1 \mid X) = p.$$

Therefore, the resulting mask is completely random and has expected missing proportion p.

Missing at Random (MAR). Under the MAR mechanism, the missingness of some variables depends on a subset of variables that remain fully observed. Specifically, the features are first split into two disjoint sets:

$$\mathcal{J}_{\text{obs}} \cup \mathcal{J}_{\text{miss}} = \{1, \ldots, d\}, \qquad \mathcal{J}_{\text{obs}} \cap \mathcal{J}_{\text{miss}} = \emptyset,$$

where variables in $\mathcal{J}_{\text{obs}}$ are always observed, and variables in $\mathcal{J}_{\text{miss}}$ may be masked. For each $j \in \mathcal{J}_{\text{miss}}$, the masking probability is defined through a logistic model:

$$\mathbb{P}(r_{i,j} = 1 \mid X) = \mathbb{P}(r_{i,j} = 1 \mid X_{i,\mathcal{J}_{\text{obs}}}) = \sigma\big(X_{i,\mathcal{J}_{\text{obs}}}^{\top} w_j + b_j\big),$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, w_j is a randomly generated coefficient vector, and b_j is chosen so that the average masking rate is approximately p. Then the mask is sampled as

$$r_{i,j} \sim \text{Bernoulli}\big(\sigma\big(X_{i,\mathcal{J}_{\text{obs}}}^{\top} w_j + b_j\big)\big), \qquad j \in \mathcal{J}_{\text{miss}}.$$

Hence, the probability of missingness depends only on other observed variables, not on the masked entry itself, which is exactly the defining property of MAR.

Missing Not at Random (MNAR). Under the MNAR logistic mechanism, the features are first partitioned into two sets:

$$\mathcal{J}_{\text{param}} \cup \mathcal{J}_{\text{miss}} = \{1, \ldots, d\}, \qquad \mathcal{J}_{\text{param}} \cap \mathcal{J}_{\text{miss}} = \emptyset.$$

The variables in $\mathcal{J}_{\text{param}}$ are used as inputs to a logistic masking model, while variables in $\mathcal{J}_{\text{miss}}$ are masked according to

$$\mathbb{P}(r_{i,j} = 1 \mid X) = \sigma\big(X_{i,\mathcal{J}_{\text{param}}}^{\top} w_j + b_j\big), \qquad j \in \mathcal{J}_{\text{miss}}.$$

Thus,

$$r_{i,j} \sim \text{Bernoulli}\Big(\sigma\big(X_{i,\mathcal{J}_{\text{param}}}^{\top} w_j + b_j\big)\Big), \qquad j \in \mathcal{J}_{\text{miss}}.$$

After this, the input variables themselves are additionally masked at random:

$$r_{i,j} \sim \text{Bernoulli}(p), \qquad j \in \mathcal{J}_{\text{param}}.$$

Therefore, the missingness of variables in $\mathcal{J}_{\text{miss}}$ depends on values from $\mathcal{J}_{\text{param}}$, but those driving variables may themselves become missing. As a result, the masking probability depends on information that is not fully observed in the final dataset, so the mechanism is Missing Not At Random.

The tested MAR mechanism retains fully observed masking covariates, while the tested MNAR mechanism may retain some after random masking. This remaining context may keep predictions for regular missing values more concentrated than those for MM entries, helping preserve an uncertainty gap. It is worthwhile noting that this intuition is specific to the tested mechanisms and provides no general MAR/MNAR separation or identifiability guarantee.

D Implementation Details and Additional Numerical Results

D.1 Implementation Details

We implement Diff-Joint using the EDM diffusion framework [37]. At each outer iteration, the model is trained on the current joint state

$$\mathbf{s} = [\mathbf{x}_{\text{model}}, \mathbf{c}] \in \mathbb{R}^{d_x + d_c},$$

where $\mathbf{x}_{\text{model}}$ is the normalized encoded tabular vector and $\mathbf{c}$ is the current predicted meaningful-missingness mask in the original raw-column space. The mask coordinates are represented as scalar binary variables and concatenated directly with the encoded data representation.

We use the same TabDDPM-style MLP denoising backbone as DiffPuter [38, 18], applied to the joint diffusion state rather than to the encoded data vector alone. The EDM network consists of an input projection, a sinusoidal noise embedding passed through a two-layer time MLP and added to the projected state, three fully connected layers with SiLU activations, and a linear output layer matching the joint-state dimension. Thus, the denoiser is trained to reconstruct both the tabular coordinates and the MM-mask coordinates under the standard EDM preconditioning. Noise levels during training follow the EDM log-normal noise distribution, and sampling uses the standard EDM ρ-schedule with Heun's second-order sampler. We use the same architecture across all datasets and missingness settings.

Random Initialization. At initialization, all predicted meaningful-missingness indicators are set to zero, i.e., $\widehat{C}^{(0)} = 0$, so every observed missing entry is initially treated as observation-induced missingness. For each feature j, we form the empirical regular-value support from the observed entries,

$$\widehat{\mathcal{X}}_j = \{x_{\ell,j}^{\text{obs}} : \omega_{\ell,j} = 1,\ \ell \in [n]\}.$$

Then, for every missing entry (i, j), we initialize

$$\widehat{x}_{i,j}^{(0)} = G_{i,j}, \qquad G_{i,j} \sim \text{Unif}(\widehat{\mathcal{X}}_j),$$

independently across missing entries.

Discrete-variable Handling. We distinguish integer-valued and categorical discrete variables. Integer-valued columns are encoded as scalar coordinates in the diffusion model. During decoding, these coordinates are rounded to the nearest valid integer value. Categorical columns are expanded into one-hot blocks before training. After sampling, each categorical block is decoded by taking the $\arg\max$ over the block.

Diff-Joint Hyperparameters. We use 10 iterations for the Bayesian Network and HELOC and 3 iterations for MIMIC-IV-ED and Hotel Booking Demand. Except for the number of refinement iterations, we use a fixed hyperparameter configuration across datasets, missingness ratios, missingness mechanisms, and random seeds. The remaining default settings are summarized in Table 9.

Hardware and Runtime. All experiments were conducted on a single NVIDIA RTX 5090 GPU. Each combination of random seed, missingness mechanism, and missingness ratio takes approximately 15 minutes on the Bayesian-network dataset, 2 hours on MIMIC-IV-ED, 10 minutes on HELOC, and 30 minutes on Hotel Booking Demand.

Downstream Task Setup. All downstream tasks are formulated as multi-class classification problems on the augmented feature domain, where `na` may be treated as a valid semantic class when it appears in the target variable. We evaluate downstream utility using a train-on-generated, test-on-real classification protocol. For each method, dataset, missingness ratio, and random seed, let $\widehat{X}_{\text{tr}}$ denote the method-specific generated or completed training table obtained from the observed training data. Let the j-th variable be the target variable of the downstream classification task. The classifier is trained with $\widehat{X}_{\text{tr},-j}$ as features and $\widehat{X}_{\text{tr},j}$ as labels.

Hyperparameters are selected on the real augmented-domain training split, and final performance is evaluated on the real augmented-domain test split. The same classifier family, feature-processing pipeline, and hyperparameter grid are used for all methods under the same downstream task.

When the target variable contains a meaningful-missing state, `na` is treated as a valid semantic class rather than as an unobserved label. Categorical `na` values in input features are encoded as ordinary categorical states, while continuous MM candidate columns are removed from the downstream feature set. On the Bayesian Network dataset, we evaluate two multi-class tasks with target variables `D2` and `D3`. On MIMIC-IV-ED, we evaluate four discrete clinical prediction tasks: `outcome_hospitalization`, `outcome_critical`, `outcome_icu_transfer_12h`, and `cci_CHF`. We use an XGBoost classifier for all tasks and select the model by validation Macro-F1 on the real training split.

Table 9: Default hyperparameters for Diff-Joint.

Hyperparameter	Value
Number of samples K	30
Sampling steps	50
MLP hidden/time-embedding dimension	1024
Batch size	4096
Optimizer	Adam
Learning rate	10^{-4}
Maximum epochs per outer iteration	1001
Early stopping patience	200
P_{mean}, P_{std}	$-1.2,\ 1.2$
σ_{data}	0.5
$\sigma_{\min}$, $\sigma_{\max}$, ρ	$0.002,\ 80,\ 7$
Inner inpainting repeats per noise level	10

We report Macro-F1, Weighted-F1, ROC-AUC, and accuracy. Macro-F1 is the unweighted average of class-wise F1 scores. Weighted-F1 denotes the class-balanced weighted F1 score used by our evaluator, where the class-wise F1 score for class ℓ is weighted by $\frac{1-p_\ell}{L-1}$, with L denoting the number of classes and p_ℓ denoting the class proportion. This weighting assigns larger relative weights to lower-support classes. ROC-AUC measures the ranking quality of predicted class probabilities and uses one-vs-rest aggregation for multi-class tasks. Accuracy is the fraction of correctly classified test examples.

D.2 Additional Results

MM Identification on MIMIC-IV-ED. Table 10 reports the MM precision and recall of Diff-Joint on MIMIC-IV-ED under MCAR. Precision decreases as the masking ratio increases, while recall remains between 55.34% and 65.12%.

Table 10: Precision and recall of Diff-Joint on MIMIC-IV-ED under MCAR.

Method	Ratio	$\text{Recall}_{\text{out}} \uparrow$	$\text{Precision}_{\text{out}} \uparrow$
Diff-Joint	10	61.06 ± 4.82	71.99 ± 0.67
Diff-Joint	20	55.34 ± 6.69	56.49 ± 0.81
Diff-Joint	30	59.90 ± 6.62	44.82 ± 2.58
Diff-Joint	40	65.12 ± 4.19	34.61 ± 1.24

Variability in CCI CHF Downstream Prediction. Table 11 reports the mean and standard deviation over five trials for predicting `cci_CHF` on MIMIC-IV-ED. Diff-Joint achieves the highest mean Macro-F1, ROC-AUC, and weighted F1 at all four masking ratios. Its mean accuracy is highest at 10% and 20%, with small differences from the competing methods, whereas CACTI achieves the highest mean accuracy at 30% and 40%.

Results on HELOC. Table 12 provides the complete HELOC results. Diff-Joint achieves the highest ACC at all four masking ratios and the lowest RMSE at 10% and 20%. Its MM recall remains above 83%, although precision decreases at higher masking ratios.

Results on Hotel Booking Demand. Table 13 reports the complete Hotel Booking Demand results. Diff-Joint achieves the highest ACC at each masking ratio, but its MAE and RMSE are higher than those of both baselines. MM precision and recall decrease as the masking ratio increases.

Performance at Higher Missingness Ratios. We extend the Bayesian Network evaluation to observation-induced masking ratios of 50% and 60%. Table 14 shows that Diff-Joint retains approximately 74% MM recall at both ratios, while precision decreases from 37.21% to 29.74%. At 60%, its ACC is lower than that of all compared baselines, showing the difficulty of selective imputation with less observed context.

Table 11: Downstream prediction of `cci_CHF` on MIMIC-IV-ED over five seeds.

Method	Ratio (%)	Macro-F1	ROC-AUC	Weighted-F1	ACC
Diff-Joint	10	$\mathbf{47.83 \pm 0.83}$	$\mathbf{83.50 \pm 0.25}$	$\mathbf{33.27 \pm 1.13}$	$\mathbf{80.43 \pm 0.14}$
Mean/Mode	10	36.76 ± 0.28	73.09 ± 0.22	19.38 ± 0.49	79.55 ± 0.07
missForest	10	42.09 ± 0.46	74.37 ± 0.42	27.06 ± 0.69	80.28 ± 0.19
DiffPuter	10	43.09 ± 0.63	73.69 ± 0.52	26.80 ± 0.94	79.95 ± 0.22
CACTI	10	42.99 ± 0.57	74.67 ± 0.55	28.31 ± 0.81	80.11 ± 0.24
Diff-Joint	20	$\mathbf{47.47 \pm 0.98}$	$\mathbf{83.14 \pm 0.36}$	$\mathbf{32.53 \pm 1.30}$	$\mathbf{80.23 \pm 0.29}$
Mean/Mode	20	35.35 ± 0.41	73.30 ± 0.19	17.45 ± 0.44	79.50 ± 0.11
missForest	20	42.68 ± 0.58	73.83 ± 0.36	27.70 ± 0.82	79.66 ± 0.14
DiffPuter	20	42.56 ± 0.81	73.38 ± 0.48	27.45 ± 0.87	79.89 ± 0.35
CACTI	20	43.75 ± 0.64	73.41 ± 0.46	29.01 ± 1.03	79.97 ± 0.17
Diff-Joint	30	$\mathbf{47.40 \pm 0.43}$	$\mathbf{81.99 \pm 0.39}$	$\mathbf{32.37 \pm 0.68}$	79.17 ± 0.40
Mean/Mode	30	33.76 ± 0.32	71.32 ± 0.35	15.37 ± 0.68	79.08 ± 0.09
missForest	30	41.79 ± 0.49	72.41 ± 0.59	26.12 ± 0.74	80.11 ± 0.28
DiffPuter	30	42.85 ± 0.68	72.87 ± 0.77	27.84 ± 1.18	79.73 ± 0.28
CACTI	30	42.89 ± 0.85	74.14 ± 0.69	27.68 ± 0.89	$\mathbf{80.41 \pm 0.36}$
Diff-Joint	40	$\mathbf{46.52 \pm 0.55}$	$\mathbf{81.56 \pm 1.07}$	$\mathbf{31.27 \pm 0.76}$	78.15 ± 1.23
Mean/Mode	40	34.19 ± 0.51	73.18 ± 0.29	15.92 ± 0.61	79.27 ± 0.21
missForest	40	42.23 ± 0.71	71.88 ± 0.48	26.69 ± 0.93	79.84 ± 0.22
DiffPuter	40	41.67 ± 0.95	73.59 ± 0.64	26.31 ± 1.05	79.82 ± 0.41
CACTI	40	42.92 ± 0.72	73.55 ± 0.62	27.74 ± 1.18	$\mathbf{80.24 \pm 0.29}$

Table 12: Evaluation on HELOC under different observation-induced masking ratios. Entries with $\pm$ report mean and standard deviation.

Method	Ratio (%)	MAE ↓	RMSE ↓	ACC (%) ↑	MM Precision (%) ↑	MM Recall (%) ↑
Diff-Joint	10	13.96 ± 0.42	$\mathbf{27.85 \pm 2.47}$	$\mathbf{63.09 \pm 0.61}$	85.96 ± 1.05	84.68 ± 2.34
DiffPuter	10	14.34 ± 0.48	30.35 ± 1.83	37.51 ± 0.42	–	–
CACTI	10	$\mathbf{13.74 \pm 0.35}$	30.47 ± 2.12	36.17 ± 0.58	–	–
Diff-Joint	20	14.29 ± 0.55	$\mathbf{27.86 \pm 1.24}$	$\mathbf{58.50 \pm 0.29}$	75.09 ± 0.64	83.83 ± 1.25
DiffPuter	20	14.33 ± 0.61	28.16 ± 1.35	43.34 ± 0.75	–	–
CACTI	20	$\mathbf{13.92 \pm 0.52}$	28.68 ± 1.48	42.23 ± 0.33	–	–
Diff-Joint	30	15.53 ± 0.45	30.01 ± 1.45	$\mathbf{54.64 \pm 0.60}$	66.38 ± 0.52	85.03 ± 1.87
DiffPuter	30	14.93 ± 0.39	29.54 ± 2.01	43.83 ± 0.51	–	–
CACTI	30	$\mathbf{13.98 \pm 0.47}$	$\mathbf{27.95 \pm 1.25}$	42.58 ± 0.69	–	–
Diff-Joint	40	16.49 ± 0.53	32.44 ± 1.10	$\mathbf{50.85 \pm 0.68}$	58.43 ± 0.44	88.04 ± 1.36
DiffPuter	40	15.94 ± 0.57	31.68 ± 1.62	42.24 ± 0.63	–	–
CACTI	40	$\mathbf{15.46 \pm 0.68}$	$\mathbf{30.33 \pm 1.94}$	41.13 ± 0.47	–	–

Sensitivity to MM Candidate-Set Misspecification. We compare the correct candidate set $\mathcal{T} = \{\texttt{C1}, \texttt{D2}, \texttt{D3}\}$ with the enlarged set $\mathcal{T}' = \mathcal{T} \cup \{\texttt{D1}\}$ on the Bayesian Network dataset, where `D1` contains no true MM. Table 15 shows that adding this column changes MAE, RMSE, and recall relatively little, but reduces precision and ACC through false MM predictions in `D1`.

Effect of Limited Training Data. We use a fixed Bayesian Network train/test split with 10% missingness and randomly retain subsets of the original 14,000 training records. Table 16 reports mean and standard deviation over five seeds. The empirical uncertainty gap is the mean uncertainty of MM entries minus that of OM entries. ACC and precision remain similar across sample sizes, while recall decreases from 91.56% to 86.54% when the training size is reduced to 500. The mean uncertainty gap remains positive at every tested sample size.

Two-Component Mixture Distribution. We construct an equally weighted mixture of the original Bayesian Network and a perturbed component with the same graph. In the perturbed component, `D2` exchanges its conditional distributions under $(\texttt{C1} \le 26, \texttt{C2} > 55, \texttt{D1} = 1)$ and $(\texttt{C1} \le 26, \texttt{C2} \le 55, \texttt{D1} = 1)$: the former produces na, and the latter follows $\mathrm{Ca}(0.7, 0.1, 0.2)$. We also set $\texttt{D3} \mid \texttt{D2} =$

Table 13: Evaluation on Hotel Booking Demand under different observation-induced masking ratios.

Method	Ratio (%)	MAE ↓	RMSE ↓	ACC (%) ↑	MM Precision (%) ↑	MM Recall (%) ↑
Diff-Joint	10	34.98 ± 1.70	45.46 ± 1.30	$\mathbf{53.79 \pm 0.13}$	72.79 ± 1.35	84.98 ± 2.81
DiffPuter	10	$\mathbf{31.75 \pm 1.84}$	$\mathbf{41.32 \pm 1.15}$	49.35 ± 0.18	–	–
CACTI	10	32.80 ± 2.11	42.82 ± 1.62	47.44 ± 0.25	–	–
Diff-Joint	20	36.41 ± 3.02	47.18 ± 2.51	$\mathbf{60.46 \pm 0.13}$	52.57 ± 2.12	83.70 ± 4.25
DiffPuter	20	$\mathbf{28.25 \pm 2.45}$	$\mathbf{39.00 \pm 2.03}$	57.82 ± 0.31	–	–
CACTI	20	33.03 ± 1.58	44.20 ± 1.91	56.77 ± 0.19	–	–
Diff-Joint	30	37.19 ± 2.88	48.53 ± 2.16	$\mathbf{61.41 \pm 0.27}$	34.70 ± 3.15	76.24 ± 8.47
DiffPuter	30	$\mathbf{31.22 \pm 1.92}$	$\mathbf{41.48 \pm 1.87}$	60.14 ± 0.22	–	–
CACTI	30	34.76 ± 2.37	45.64 ± 2.55	59.79 ± 0.34	–	–
Diff-Joint	40	42.03 ± 3.25	52.73 ± 2.78	$\mathbf{61.84 \pm 0.28}$	24.84 ± 0.98	62.26 ± 4.98
DiffPuter	40	$\mathbf{31.07 \pm 2.91}$	$\mathbf{41.80 \pm 2.34}$	58.63 ± 0.45	–	–
CACTI	40	36.57 ± 3.12	48.14 ± 1.88	59.58 ± 0.27	–	–

Table 14: Evaluation on the Bayesian Network dataset at higher observation-induced masking ratios. Entries with $\pm$ report mean and standard deviation.

Method	Ratio (%)	MAE ↓	RMSE ↓	ACC (%) ↑	MM Precision (%) ↑	MM Recall (%) ↑
Diff-Joint	50	4.05 ± 0.02	5.11 ± 0.03	58.48 ± 1.31	37.21 ± 0.89	73.67 ± 1.52
Mean/Mode	50	$\mathbf{3.99 \pm 0.00}$	$\mathbf{5.02 \pm 0.00}$	57.00 ± 0.00	–	–
missForest	50	4.32 ± 0.16	5.49 ± 0.21	57.00 ± 1.05	–	–
CMAE	50	4.00 ± 0.11	5.11 ± 0.14	55.05 ± 0.96	–	–
DiffPuter	50	4.07 ± 0.12	5.17 ± 0.15	$\mathbf{58.63 \pm 1.12}$	–	–
Diff-Joint	60	4.11 ± 0.06	5.17 ± 0.07	50.23 ± 1.97	29.74 ± 0.97	73.95 ± 1.66
Mean/Mode	60	3.97 ± 0.00	5.00 ± 0.00	59.31 ± 0.00	–	–
missForest	60	4.82 ± 0.24	6.04 ± 0.26	57.54 ± 1.76	–	–
CMAE	60	$\mathbf{3.94 \pm 0.12}$	$\mathbf{4.99 \pm 0.15}$	59.42 ± 1.05	–	–
DiffPuter	60	4.10 ± 0.13	5.17 ± 0.16	$\mathbf{60.83 \pm 1.18}$	–	–

Table 15: Sensitivity to MM candidate-set misspecification on the Bayesian Network dataset. The last column reports the MM false-positive rate for the non-MM column `D1`.

Candidate set	Ratio (%)	MAE ↓	RMSE ↓	ACC (%) ↑	Precision (%) ↑	Recall (%) ↑	D1 FPR (%) ↓
Correct	10	3.93	4.97	78.40	73.45	92.02	0.00
Over-specified	10	4.04	5.05	74.83	69.26	91.47	44.58
Correct	20	4.11	5.14	69.71	59.33	84.19	0.00
Over-specified	20	4.09	5.12	61.17	50.92	81.81	60.93

Table 16: Effect of training-set size on the Bayesian Network dataset at 10% MCAR missingness.

Training size	ACC (%)↑	MM Precision (%)↑	MM Recall (%)↑	Uncertainty gap
500 (3.57%)	78.86 ± 0.68	75.07 ± 1.79	86.54 ± 1.86	0.18 ± 0.09
1,000 (7.14%)	78.54 ± 0.25	73.81 ± 0.70	88.52 ± 2.30	0.07 ± 0.07
2,800 (20.00%)	78.71 ± 0.50	73.92 ± 0.38	89.47 ± 1.45	0.14 ± 0.04
5,600 (40.00%)	78.63 ± 0.66	73.71 ± 0.77	90.92 ± 1.31	0.16 ± 0.06
14,000 (100.00%)	77.93 ± 0.36	73.33 ± 0.63	91.56 ± 2.31	0.17 ± 0.04

$1 \sim \mathrm{Bernoulli}(0.8)$ and $\texttt{D3} = \mathrm{na}$ when $\texttt{D2} = 2$; all other conditional distributions remain unchanged. Component labels are not supplied to the model, and the combined data undergo MCAR masking. Table 17 shows that the mean uncertainty gap remains positive, although MM recall becomes more variable across runs.

Table 17: Comparison of the original Bayesian Network and a two-component mixture under 10% and 40% MCAR missingness. The uncertainty gap is the mean MM uncertainty minus the mean OM uncertainty.

Distribution	Ratio (%)	RMSE ↓	ACC (%)↑	MM Precision (%)↑	MM Recall (%)↑	Uncertainty gap
Original	10	5.09 ± 0.13	78.34 ± 0.44	73.45 ± 0.34	92.02 ± 1.34	0.09 ± 0.04
Mixture	10	5.17 ± 0.22	73.41 ± 8.80	83.36 ± 0.74	72.10 ± 10.70	0.15 ± 0.03
Original	40	4.99 ± 0.06	65.92 ± 2.88	43.54 ± 2.26	76.99 ± 3.48	0.19 ± 0.09
Mixture	40	5.08 ± 0.08	67.33 ± 4.32	56.67 ± 2.64	80.27 ± 7.75	0.09 ± 0.01

D.3 Ablation Studies

Effect of Iterative Latent-State Refinement. We first study whether the refinement loop is necessary. Figure 4 reports performance as a function of the number of refinement iterations. Across both datasets, most improvements occur in the first few refinement iterations. MM-detection and final-state accuracy typically improve substantially from the first iteration to the third iteration, while observation-induced-missing-value imputation error stabilizes after a small number of iterations. We therefore use a fixed small number of refinement iterations in the main experiments.

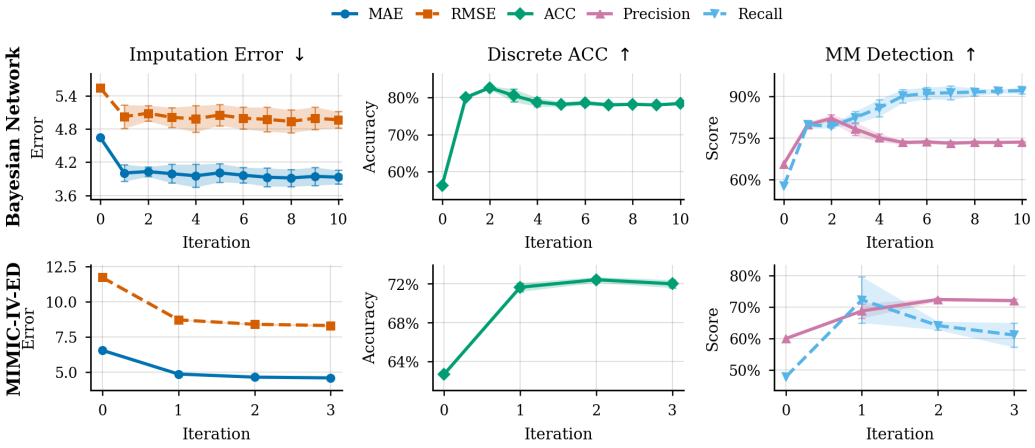

Figure 4: Effect of iterative latent-state refinement on Bayesian Network and MIMIC-IV-ED under ordinary MCAR at 10% missing ratio. Curves show the mean over five seeds, with shaded regions indicating one standard deviation. Results under the 0-th iteration are obtained with a randomly initialized denoising network, with all other components kept the same as in Diff-Joint.

Component Ablation. We ablate the main components of Diff-Joint on the Bayesian Network dataset under ordinary MCAR. DiffPuter is a standard diffusion imputer and does not output MM labels, whereas Diff-Joint models the joint state $(\mathbf{x}, \mathbf{c})$ to capture dependencies between tabular values and MM labels. We further compare a one-refinement variant with the full model, which uses 10 outer refinement iterations. Table 18 shows that joint $(\mathbf{x}, \mathbf{c})$ modeling substantially improves final-state recovery over standard diffusion imputation, and iterative refinement further improves MM F1, final-state accuracy, and RMSE.

Aggregation-Rule Ablation. We conduct an ablation study on the rule used to update the meaningful-missingness mask during iterative latent-state refinement. Let U denote the uncertainty-based signal obtained from the high-uncertainty cluster, and let V denote the sampled-mask signal obtained from the joint diffusion model. Table 19 reports the aggregation-rule ablation results. Majority-vote-only collapses from the all-zero MM-mask initialization, and the AND rule inherits this conservativeness. Uncertainty-only provides a nontrivial bootstrap signal, while OR fusion achieves the best MM F1 and final-state accuracy among the tested closed-loop rules. This sup-

Table 18: Component ablation on the Bayesian Network dataset under ordinary MCAR.

Missing Ratio	Method	Joint $(\mathbf{x}, \mathbf{c})$	Refinement	MM $F1_{out}$ (%) ↑	ACC_{out} (%) ↑	$RMSE_{out}$ ↓
20	DiffPuter	✗	✓	–	44.61	5.1962
20	Diff-Joint, 1-refine	✓	✗	69.85	71.11	5.1371
20	Diff-Joint, 10-iterations	✓	✓	**73.39**	**76.18**	**5.0832**
30	DiffPuter	✗	✓	–	51.89	5.2764
30	Diff-Joint, 1-refine	✓	✗	58.99	60.26	5.1461
30	Diff-Joint, 10-iterations	✓	✓	**64.76**	**72.67**	**4.9987**

ports using OR fusion as the default refinement rule, as it achieves the best overall performance in accurately identifying meaningful missingness.

Table 19: Aggregation-rule ablation on the Bayesian Network dataset under ordinary MCAR at missing ratio 30.

Rule	$Recall_{out}$ (%) ↑	$Precision_{out}$ (%) ↑	MM $F1_{out}$ (%) ↑	ACC_{out} (%) ↑	$RMSE_{out}$ ↓
Uncertainty-only	66.83	46.14	54.59	56.42	5.026
Majority-vote-only	0.00	0.00	0.00	51.27	5.006
Uncertainty ∧ majority vote	0.00	0.00	0.00	51.32	**4.942**
Uncertainty ∨ majority vote (ours)	**76.33**	**54.08**	**63.31**	**66.95**	5.168

Effect of Downstream Representation. We compare three representations of the Diff-Joint output for predicting D2 and D3 on the Bayesian Network dataset: selective imputation, which preserves na at predicted MM entries; the joint representation $(\widehat{\mathbf{x}}, \widehat{\mathbf{c}})$, which retains candidate values and appends the learned MM mask; and $(\widehat{\mathbf{x}}, \mathbf{1} - \boldsymbol{\omega})$, which instead appends the observed-missingness indicator. For each target, its value and corresponding mask coordinates are excluded from the classifier inputs. Table 20 shows that the learned-mask representation has higher mean performance than the observed-mask representation in 15 of the 16 metric comparisons.

Table 20: Downstream representation ablation on the Bayesian Network dataset.

Representation	Ratio	Target	Macro-F1	ROC-AUC	Weighted-F1	ACC
Selective	10	D2	75.54	94.81	71.42	89.38
Learned MM mask	10	D2	**80.28**	95.99	**77.32**	**89.63**
Observed mask	10	D2	80.25	**96.00**	77.30	89.59
Selective	10	D3	**79.16**	**86.01**	**77.48**	**84.72**
Learned MM mask	10	D3	72.29	82.03	69.91	78.56
Observed mask	10	D3	69.22	81.21	67.00	74.82
Selective	40	D2	61.11	**94.77**	53.94	**86.15**
Learned MM mask	40	D2	**73.75**	93.88	**71.40**	81.49
Observed mask	40	D2	72.20	92.86	70.08	78.65
Selective	40	D3	**75.13**	**86.42**	**72.73**	**83.15**
Learned MM mask	40	D3	66.05	83.17	63.44	72.67
Observed mask	40	D3	59.16	78.19	57.31	63.16

Remark D.1. The ablation studies in this section use different random seeds from the main experiments. Therefore, the reported values may differ slightly from those in the main tables, even under the same configurations.

E More Implementation Details on Baseline Methods

We compare against recent deep-learning baselines for tabular imputation. Specifically, we include CACTI [22], a recent strong masked-autoencoding method whose original benchmark reports improvements over autoencoding baselines such as ReMasker [21] and AutoComplete [39]. We also include DiffPuter [18], a recent diffusion-based imputation method that outperforms prior diffusion baselines such as TabCSDI [10] and MissDiff [23] on standard imputation benchmarks. On the Bayesian Network dataset, whose column names carry no semantic information, we use the non-embedding variant of CACTI, denoted CMAE.

All baselines are trained and evaluated on exactly the same observed datasets and the same train/test splits as Diff-Joint. When compatible, we use the same preprocessing pipeline as in the main experiments; otherwise, we follow the model-specific preprocessing required by the official implementation. Since these baselines are not designed to explicitly infer meaningful-missingness labels, MM precision and recall are not applicable and are therefore omitted for them.

Implementation Sources. We implement the baseline methods according to the following description or publicly available codebases.

- **Mean/Mode**: Missing continuous entries are filled with the empirical mean of the corresponding column, and missing discrete entries are filled with the empirical mode. Both statistics are computed from observed entries in the training split and then reused for test-set imputation. This baseline has no tunable hyperparameters.
- **missForest** [6]: We use the implementations provided in the official HyperImpute [20] repository: `https://github.com/vanderschaarlab/hyperimpute`.
- **CACTI/CMAE** [22]: We use the official implementation of CACTI at `https://github.com/sriramlab/CACTI`. On real-world datasets, we run CACTI with contextual column-name embeddings. Specifically, we precompute feature-name embeddings using `sentence-transformers/all-MiniLM-L6-v2` [40] and use the resulting `embeddings_colnames_MiniLM.npz` file as CACTI's column-context input. On the Bayesian Network dataset, the column names are synthetic and carry no semantic information. We therefore disable the column-name embedding module and denote the resulting non-contextual variant as CMAE. This change removes arbitrary column-name semantics while keeping the masked autoencoding architecture and the remaining model configuration unchanged.
- **DiffPuter** [18]: We use the official implementation at `https://github.com/hengruizhang98/DiffPuter`.

Hyperparameter Settings. We use a fixed hyperparameter protocol for all baseline methods. Hyperparameters are selected, when needed, using only the training split and an internal validation configuration. The validation criterion is ordinary-value imputation performance on held-out validation entries. We do not tune any method on the test set, downstream prediction performance, or MM precision/recall/F1.

Mean/Mode has no tunable hyperparameters. For CACTI/CMAE and DiffPuter, we follow the recommended or default configurations from the corresponding official implementations, except for dataset-specific changes required by input dimensionality, memory constraints, or contextual inputs. For missForest, we use the `hyperimpute` search space and perform hyperparameter selection separately for each random seed, using only the corresponding training and validation split. The detailed baseline configurations are listed in Table 21. Results reported with standard deviations are aggregated over five random seeds.

Table 21: Hyperparameter settings used for the baseline methods in the main experiments.

MODEL	HYPERPARAMETERS / CONFIGURATION
MEAN/MODE	No tunable hyperparameters.
MISSFOREST	Implemented using the `hyperimpute` package. For each random seed, we sample 50 hyperparameter configurations from the package-provided search space and report the best-performing configuration for that seed. Final results are summarized as mean and standard deviation over five seeds.
CMAE	Used on the Bayesian Network dataset. CMAE uses the CACTI masked-autoencoding architecture with the column-name embedding module disabled, because the synthetic column names carry no semantic information. The optimization settings are `epochs = 300`, `warmup_epochs = 50`, `batch_size = 128`, `lr = 1e-3`, `min_lr = 5e-6`, `weight_decay = 1e-3`, `grad_clip = 5.0`, `mask_ratio = 0.9`, `embed_dim = 64`, `nencoder = 10`, and `ndecoder = 4`.
CACTI	Used on MIMIC-IV-ED. CACTI uses the same optimization settings as CMAE: `epochs = 300`, `warmup_epochs = 50`, `batch_size = 128`, `lr = 1e-3`, `min_lr = 5e-6`, `weight_decay = 1e-3`, `grad_clip = 5.0`, `mask_ratio = 0.9`, `embed_dim = 64`, `nencoder = 10`, and `ndecoder = 4`. Column-name embeddings are precomputed using `sentence-transformers/all-MiniLM-L6-v2` [40].
DIFFPUTER	Bayesian Network: `max_iter = 10`. MIMIC-IV-ED: `max_iter = 10`. Common settings are `hid_dim = 1024`, `num_trials = 10`, `num_steps = 50`, `num_epochs = 1000`, `batch_size = 4096`, `learning_rate = 1e-4`, `scheduler_patience = 40`, and `early_stop_patience = 200`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize the paper's contributions and scope clearly in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of the work in the Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For our main theoretical results, we provide the assumptions and detailed proofs in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully describe our numerical setups that are sufficient to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and processed data are not publicly released at submission time, but will be made available upon reasonable request. The paper provides the synthetic Bayesian Network data generation model, MIMIC-IV-ED data design, model configurations, and testing details needed to reproduce the main empirical results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify all the training and testing details in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In some of our tables when applicable, we report the error bars (standard deviation) to show the statistical significance of the experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the computer resource used to implement the experiments in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work is about missing value imputation, and we discussed the potential positive impact of imputing missing values in real data. To the best of our knowledge, there is no direct negative societal impact of the work performed, except that the imputation for real data could be inaccurate as with all missing value imputation algorithms, and we reported the accuracy and imputation error in our numerical results.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: To the best of our knowledge, the paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We used one synthetic dataset generated through a simple Bayesian network simulation, and three real-world datasets selected from MIMIC-IV and BeyondArena, which we have cited properly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new public dataset or benchmark asset. The synthetic data are generated from a specified Bayesian Network described in the experimental section and appendix. The MIMIC-IV-ED dataset is cited and described in sufficient details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for polishing the writing of the paper, and does not impact the core method development. That is, the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.