RefiDiff: Refinement-Aware Diffusion for Efficient Missing Data Imputation

Md Atik Ahamed[1], **Qiang Ye**[2], **Qiang Cheng**[1,3*]
[1]Department of Computer Science
[2]Department of Mathematics
[3]Institute for Biomedical Informatics
University of Kentucky
`{atikahamed,qye3,qiang.cheng}@uky.edu`

Abstract

Missing values in high-dimensional, mixed-type datasets pose significant challenges for data imputation, particularly under Missing Not At Random (MNAR) mechanisms. Existing methods struggle to integrate local and global data characteristics, limiting performance in MNAR and high-dimensional settings. We propose an innovative framework, RefiDiff, combining local machine learning predictions with a novel Mamba-based denoising network capturing interrelationships among distant features and samples. Our approach leverages pre-refinement for initial warm-up imputations and post-refinement to polish results, enhancing stability and accuracy. By encoding mixed-type data into unified tokens, RefiDiff enables robust imputation without architectural or hyperparameter tuning. RefiDiff outperforms state-of-the-art (SOTA) methods across missing-value settings, excelling in MNAR with a 4x faster training time than SOTA DDPM-based approaches. Extensive evaluations on nine real-world datasets demonstrate its robustness, scalability, and effectiveness in handling complex missingness patterns.

1 Introduction

Missing values represent a pervasive challenge in digital datasets, stemming from sensor errors, data corruption, or various operational issues. Effective data imputation is crucial for ensuring robust analysis and modeling, particularly in high-dimensional and mixed-type datasets, such as those from the UCI Machine Learning Repository [4, 41, 5, 31, 11, 40, 48] and other sources [26, 36].

Numerous imputation methods have been proposed, including traditional statistical approaches, matrix completion techniques, deep generative models, diffusion-based methods, and hybrid frameworks. Despite these advances, significant challenges remain in handling complex missingness mechanisms: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). MNAR presents the most challenging scenario, as missingness depends on the missing values themselves, often requiring assumptions about the missingness mechanism or advanced modeling techniques[3, 28, 13]. Conversely, MCAR is the simplest case, with missingness independent of all data, making it more tractable than MAR or MNAR[46].

These missingness mechanisms present varying levels of complexity for imputation methods. Existing methods typically model relationships between missing and observed values from either a local perspective (using partial observations for each feature) or a global perspective (capturing dataset-wide patterns). However, they rarely integrate both approaches effectively, particularly in high-dimensional settings with mixed numerical and categorical data. These deficiencies lead to ineffective

[*] Corresponding author.

Preprint. Under review.

handling of the MNAR scenario, sensitivity to model hyperparameters, and poor generalizability of the models, particularly in the out-of-distribution cases.

To address these limitations and bridge the gap between local and global approaches, we propose an innovative framework, RefiDiff, to integrate local and global data characteristics for robust imputation. Locally, our approach uses machine learning predictions, adopting classifiers for categorical variables and regressors for numerical variables to model missing values based on partially observed values for each feature. This forms a progressive refinement process, with the warm-up stage for pre-refinement (see Figure 1), which provides high-quality initial imputations to improve stability during diffusion, and a polishing stage for post-refinement to further enhance the diffusion-imputed values. Globally, we introduce a novel Mamba-based denoising network for capturing long-range dependencies in high-dimensional tabular data [1, 45]. Mamba is a state-space model that scales linearly with sequence length while possessing selectivity and Transformer-like expressivity [15], ideal for capturing complex patterns efficiently for the imputation task. For mixed-type data, our local-to-global strategy encodes data examples with local noisy predictions into unified tokens, inspired by [1], which are subsequently cleansed by the Mamba-based denoising network. This design yields high-quality imputations without needing architectural or hyperparameter tuning, enhancing practicality for diverse datasets.

Our framework outperforms state-of-the-art (SOTA) methods, such as DIFFPUTER [51], across MCAR, MAR, and MNAR settings, excelling in the challenging MNAR scenario. By leveraging inter-relationships between observed and missing values, it efficiently captures dependencies in mixed-type data across various dimensions and examples. Compared to the computationally intensive TabDDPM denoising network [27] used in DIFFPUTER, our Mamba-based approach offers a lightweight footprint, significantly reducing training time to be 4x faster while maintaining accuracy. Extensive empirical evaluations and ablation studies on nine real-world datasets demonstrate the robustness and effectiveness of our method in addressing complex missingness patterns in complex data.

In summary, our contributions include:

- Development of an innovative framework that leverages progressive refinement to exploit the inter-relationship between observable entries and missing values for each feature while incorporating global statistical modeling of high-dimensional data;
- Introduction of a novel Mamba-based robust denoising network to capture long-range dependencies and exploit selectivity with minimal computational requirements compared to state-of-the-art diffusion models like DIFFPUTER;
- A user-friendly plug-and-play approach with default settings applicable across various datasets without requiring extensive architectural or hyperparameter tuning;
- Superior performance compared to state-of-the-art methods across MCAR, MAR, and MNAR settings, with particularly significant improvements in the challenging MNAR scenario;
- Comprehensive empirical validation through extensive experiments and ablation studies on nine real-world datasets, confirming the framework's robustness and scalability.

2 Related Work

The field of data imputation spans several methodological categories, which we review below.

Traditional methods rely on statistical and iterative approaches. The Expectation-Maximization (EM) algorithm [12, 7] provides maximum likelihood estimates of model parameters under a pre-specified model for imputing missing data. Multiple Imputation by Chained Equations (MICE) [46] uses regression models to impute values iteratively, while K-Nearest Neighbors (KNN) imputation [37] estimates missing values based on data point similarity. MissForest [44] employs random forests to handle mixed-type data non-parametrically. While these approaches are widely adopted for their interpretability and efficiency, they mainly leverage local relationships between observed values and often struggle with complex, high-dimensional datasets or non-linear patterns.

Matrix completion techniques offer another perspective on imputation, moving beyond traditional approaches. SoftImpute [16] uses low-rank matrix factorization to recover missing entries, optimizing via alternating least squares. This approach is particularly effective for datasets with underlying low-rank structures, such as those in recommender systems [29, 21]. However, these methods may

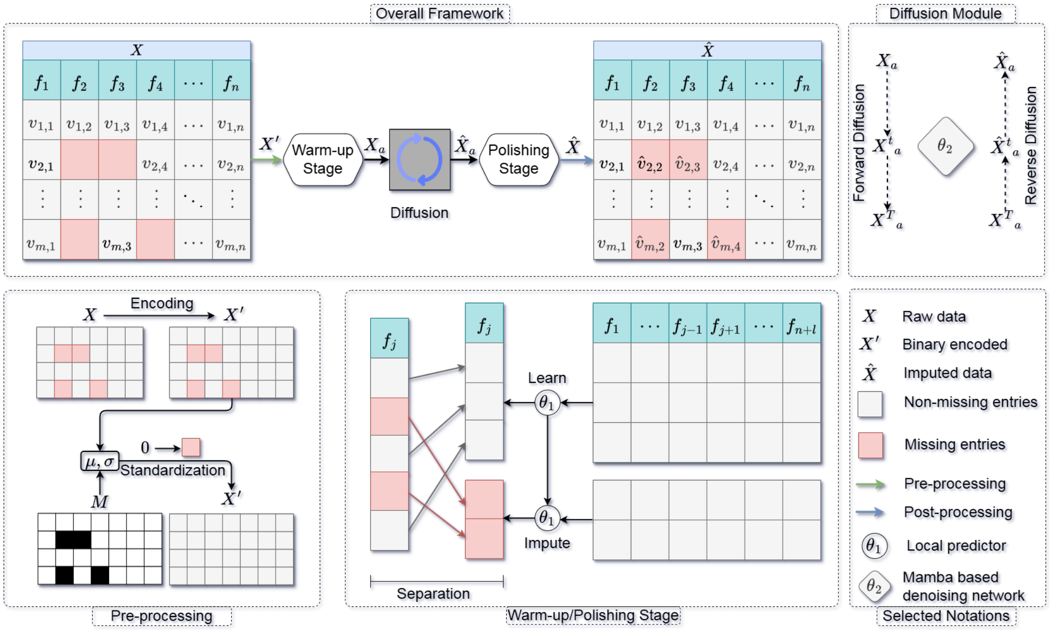


Figure 1: Overview of the proposed imputation framework. The process begins with a warm-up stage on pre-processed data, followed by a diffusion module that iteratively denoises the data. A polishing stage further enhances the imputations. Our designed denoiser θ_2 will be shown in Figure 2.

underperform when the missingness mechanism is complex or when the data do not conform to low-rank assumptions.

Deep learning advances have introduced more sophisticated imputation techniques. Variational Autoencoders (VAEs) [25] and their extensions, such as MIWAE [32] and MissVAE [34], model the data distribution to impute missing values, capturing complex patterns through latent representations. Generative Adversarial Networks (GANs) [14], as seen in GAIN [49], employ adversarial training to generate realistic imputations and have been adapted for tabular data. Normalizing flows [38], as explored in MCFlow [39], offer exact likelihood estimation via invertible mappings, which can improve modeling of complex global distributions but their impact on imputation accuracy is indirect and architecture-dependent. These generative models excel in capturing non-linear dependencies but may require significant computational resources and careful tuning [24].

Diffusion models have recently emerged as a promising paradigm for data imputation, inspired by their success in image generation [17, 22, 30]. TabDDPM [27] adapts denoising diffusion probabilistic models for tabular data, modeling data generation as a reverse diffusion process. Similarly, TabCSDI [53] and MissDiff [35] focus on handling missing values by conditioning diffusion processes on observed data. DIFFPUTER [51] integrates EM-driven diffusion for robust imputation, while DiffImpute [47] leverages denoising diffusion to impute tabular data with many iterations. ForestDiff [20] combines diffusion with gradient-boosted trees, offering a hybrid approach. These methods demonstrate strong performance in capturing complex global distributions but may face challenges with scalability and hyperparameter sensitivity [42, 43], and their iterative sampling nature may lead to slower inference times compared to autoregressive or flow-based models.

Causally-aware and graph-based methods, such as MIRACLE [28], incorporate causal relationships to improve imputation accuracy, particularly in datasets with structural dependencies. Graph-based approaches, including GRAPE [50] and IGRM [54], model data as graphs to leverage relational information for imputation. These methods are effective in structured datasets but may require domain-specific knowledge to define graph structures accurately, leading to limitations such as scalability with large graphs or sensitivity to graph structure specification.

Adaptive and hybrid frameworks aim to combine the strengths of multiple imputation strategies. HyperImpute [19] employs automatic model selection, e.g., via meta-learning, to choose the best imputation method for a given dataset, enhancing generalizability. ReMasker [10] utilizes masked autoencoding to impute tabular data, drawing inspiration from self-supervised learning paradigms. Optimal transport-based methods, such as MOT [33] and TDM [52], frame imputation as a distribution matching problem, offering robust solutions for heterogeneous data. These approaches are highly flexible but may introduce additional computational overhead.

In summary, these data imputation methods have advanced the modeling of complex missingness patterns in datasets such as those from the UCI Machine Learning Repository and real-world applications. However, most struggle to simultaneously capture both global and local structures of the observed and missing entries, as well as their interdependencies. To address this, we propose a novel framework that effectively models these relationships. Our approach achieves SOTA performance under MCAR, MAR, and particularly MNAR conditions, with a significantly more efficient architecture compared to diffusion-based models like DIFFPUTER.

3 Methodology

In this section, we describe each component of our framework step by step. The overall architecture, shown in Figure 1, consists of four major stages, each of which is discussed in this section.

Preliminaries. We consider a dataset as a matrix $X \in \mathbb{R}^{m \times n}$, where m denotes the number of samples and n the number of features. The missing entries in X are indicated by a binary mask $M \in \{0,1\}^{m \times n}$, where $M_{i,j} = 1$ signifies a missing value and $M_{i,j} = 0$ denotes an observed entry. The goal is to estimate the missing values $X_{i,j}$ for which $M_{i,j} = 1$, relying on the observed entries. The missingness pattern may follow one of three standard mechanisms: MCAR, MAR, or MNAR (see definitions in Appendix A). The dataset is then randomly partitioned into *in-sample* and *out-of-sample* subsets for robust generalization evaluation. Regardless of the missingness type or dataset partition, the core objective remains consistent: to robustly impute missing entries across MCAR, MAR, and MNAR mechanisms and generalize to out-of-sample data.

Pre-processing. The dataset $X \in \mathbb{R}^{m \times n}$ may contain numerical and categorical features. Categorical features are binary encoded, resulting in l additional binary columns, and concatenated with the numerical features to form the encoded input matrix $X' \in \mathbb{R}^{m \times (n+l)}$. The corresponding binary mask is also extended accordingly from $M \in \{0,1\}^{m \times n}$ to $M \in \{0,1\}^{m \times (n+l)}$ to account for the additional l columns. Each dataset is partitioned into In-sample and Out-of-sample subsets. The out-of-sample subset may exhibit different statistical properties, representing an out-of-distribution scenario. To ensure numerical stability and prevent data leakage, we standardize the observed values in the training set using feature-wise statistics (mean (μ) and standard deviation (σ)), computed only from the non-missing entries. The same statistics are applied to standardize the out-of-sample portion. The resulting standardized matrix and associated mask are then passed to subsequent modules for imputation. For missing entries, we simply put zero in those locations as placeholders to maintain matrix structure, which will be imputed in next stage. Therefore, at this stage, we find the data in standardized form and missing entries imputed by zeros.

Warm-up/Polishing Stage. Before applying the diffusion module, we perform a warm-up imputation to obtain an initial estimate of the missing values. This step mitigates the difficulty of learning from heavily corrupted inputs or out-of-sample distributions.

With the standardized binary encoded X' with the extended mask M, we adopt a column-wise imputation strategy. For each feature column f_j, we partition the data into observed and missing entries using M, as shown in Figure 1. The observed values in column f_j are treated as targets, while the remaining columns $\{f_1, \ldots, f_{j-1}, f_{j+1}, \ldots, f_{n+l}\}$ are used as input features. We construct input–output training pairs using only the rows where f_j is observed, and train a predictive model $\theta_1^{(j)}$ for this feature. We adopt a lightweight model (e.g., XGBoost, or CatBoost) with default objective function and hyperparameters for numerical and categorical features. While a particular model $\theta_1^{(j)}$ is trained per column, we retain only the current one at each step, ensuring memory efficiency. The trained model $\theta_1^{(j)}$ is then used to estimate the missing values in f_j, as identified by the mask M and treated as testing labels. This procedure is applied sequentially across all feature columns. In contrast to traditional approaches like EM and MICE or recent SOTA models like DIFFPUTER that rely on iterative reimputation, our approach performs a single pass over the features, improving computational efficiency. The same column-wise strategy is reused during the polishing stage, where it is employed once more to refine the outputs produced by the diffusion module. The polishing stage uses the same column-wise strategy to the diffusion output $\hat{X}_a$, further refining the imputed values.

The output of our warm-up/polishing stage has desirable properties as shown in Theorem 1 (see the proof in Appendix B). Let $d = n + l$. For the warm-up stage, we regard X' as Z, and for the polishing stage, we regard diffusion-processed $\hat{X}_a$ as Z, thus allowing Theorem 1 to characterize both stages. We use $Z_{k,\setminus j}$ to denote the k-th row of Z excluding the j-th column.

Theorem 1. *Let $Z \in \mathbb{R}^{m\times d}$ be the standardized and binary-encoded data matrix, and $M \in \{0,1\}^{m\times d}$ be the binary mask, where $M_{i,j} = 0$ indicates that the entry $Z_{i,j}$ is observed (non-missing) and $M_{i,j} = 1$ indicates that it is missing. For each feature index $j \in \{1,\ldots,d\}$, let $\theta_1^{(j)} : \mathbb{R}^{d-1} \to \mathbb{R}$ be a predictive function trained using the observed pairs $\mathcal{D}_j := \big\{(Z_{k,\setminus j}, Z_{k,j}) \mid M_{k,j} = 0, \text{for any } k \le m\big\}$. Define the imputed matrix $\hat{Z} \in \mathbb{R}^{m\times d}$ entry-wise as:*

$$\hat{Z}_{i,j} = \begin{cases} Z_{i,j}, & \text{if } M_{i,j} = 0, \\ \theta_1^{(j)}(Z_{i,\setminus j}), & \text{if } M_{i,j} = 1. \end{cases}$$

Then, the following properties hold:

1. ***Non-overwriting:** For all (i,j) such that $M_{i,j} = 0$, we have $\hat{Z}_{i,j} = Z_{i,j}$.*
2. ***Well-defined mapping:** For all (i,j) such that $M_{i,j} = 1$, the imputed value $\hat{Z}_{i,j}$ is a measurable function of $Z_{i,\setminus j}$ with respect to the model $\theta_1^{(j)}$ trained on $\mathcal{D}_j$.*
3. ***One-pass process:** Each column is processed once, and each missing entry is imputed a single time. The full imputation completes in one pass over the feature set.*

Diffusion Module. After the warm-up stage, we utilize the refined dataset denoted by X_a as input to the diffusion module, as depicted in the overall scheme in Figure 1. The primary objective of this module is to progressively denoise corrupted inputs while exactly preserving observed entries, thereby enabling robust imputation under complex and high-dimensional missing patterns. Unlike the warm-up and polishing stages, which operate from a local perspective by modeling each feature based on its partially observed context, the diffusion module provides a global modeling of the data. This allows the model to leverage dependencies across all features and samples simultaneously, capturing richer structural patterns that are crucial for accurate imputation.

During the forward diffusion process, at each time step $t = 1,\ldots,T$, Gaussian noise is incrementally added to the input, following a monotonically increasing power-law noise schedule, generating a sequence of progressively corrupted inputs $\{X_a^t\}_{t=1}^T$. Further details on the diffusion process are provided in Appendix C. The denoising network θ_2 is trained to reverse this process by predicting and removing the injected noise. Importantly, θ_2 is conditioned on all the observed entries indicated by the known M to ensure that only the missing values are updated, while the observed entries remain unaltered during the reverse diffusion.

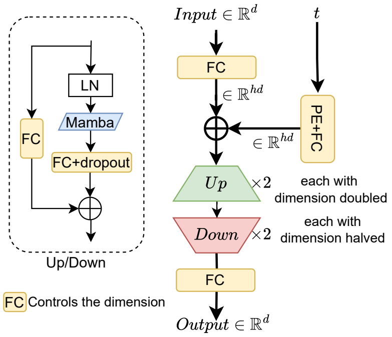


Figure 2: Illustration of the denoising network θ_2, featuring a diamond-shaped structure with Mamba-based residual Up/Down blocks.

To facilitate efficient denoising, we introduce a lightweight yet expressive denoising network module θ_2, illustrated in Figure 2. It has a diamond-shaped structure composed of two upsampling and two downsampling blocks, enabling symmetric multi-scale feature transformations. Each block contains residual blocks based on Mamba [15], a recent state-space model known for its ability to model long-range dependencies with selectivity property and high efficiency compared to transformers. Each fully connected layer (FC) in a block controls the upsampling (for Up block) or downsampling (for Down block) with a factor of 2. These blocks are wrapped with layer normalization (LN) and residually connected to enhance training stability.

Table 1: Performance comparison across all methods and settings for numerical columns. We report MAE and RMSE for in-sample and out-of-sample imputation under three missingness mechanisms: MNAR, MCAR, and MAR. Lower values indicate better performance. The best and second-best average ranks are highlighted in **bold** and underline, respectively.

Method	**MNAR**				**MCAR**				**MAR**				Rank (↓)
	In-Sample		Out-of-Sample		In-Sample		Out-of-Sample		In-Sample		Out-of-Sample		
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	
EM	42.42	82.30	46.13	107.00	38.70	67.42	40.93	89.71	42.57	82.63	43.72	91.08	5.42
MIWAE	68.21	121.21	65.99	110.34	62.55	102.88	62.59	101.61	68.24	122.50	65.85	112.69	10.08
GAIN	75.84	126.83	73.26	117.78	67.02	104.05	66.38	104.74	82.85	138.06	77.00	126.87	11.75
SoftImpute	59.82	103.66	59.68	99.27	52.74	84.01	53.95	87.48	60.80	104.45	59.30	99.16	7.92
MICE	69.01	124.49	70.83	614.63	65.66	95.08	65.60	94.31	69.47	108.12	73.17	1062.95	10.67
MIRACLE	54.54	110.86	55.86	110.21	48.47	89.45	51.24	99.10	59.48	117.85	64.17	116.85	8.67
KNN	55.95	108.99	51.30	92.93	46.42	82.48	46.39	80.90	43.22	88.43	46.45	87.82	6.42
MissForest	46.50	89.51	46.13	83.89	43.01	75.27	43.05	74.18	48.21	94.38	46.05	86.80	5.58
HyperImpute	37.95	79.61	38.45	104.61	34.51	65.22	36.38	158.27	38.62	80.81	38.89	111.65	4.67
DIFFPUTER	37.27	86.86	34.54	72.73	31.72	63.49	31.39	61.85	39.15	90.95	35.32	76.09	2.67
ReMasker	39.66	80.23	39.52	74.14	35.84	65.19	35.84	64.15	38.39	78.82	38.06	74.90	3.00
Ours	34.49	78.83	34.38	70.12	31.41	63.16	32.20	63.11	34.52	78.22	34.43	73.82	**1.17**

The network begins by projecting the input to a hidden representation of dimension hd, where a learnable noise embedding based on timestep t is added using sinusoidal positional encoding (PE), followed by a feed-forward FC layer. The upsampling path expands the feature space progressively, allowing the network to capture rich contextual information globally. The downsampling path then compresses the representation to reconstruct clean signals, ensuring alignment with the original input dimensions. Each upsampling/downsampling block maps from one fixed dimension to another, with transitions denoted by the element-wise addition operator $\oplus$ for clarity.

We train θ_2 using the EDM loss [23], which scales the denoising objective based on the injected noise level. For each input, a noise scale S is sampled, used to perturb the data, and then weighted to balance the learning across noise levels. At inference, we perform N stochastic sampling runs per input using θ_2 and aggregate the results via averaging. This ensembling improves the robustness and stability of the imputation $\hat{X}_a$ at this stage. Overall, the diffusion module serves as a global distribution-aware reconstruction engine that complements the local structure-informed warm-up stage, achieving coherence between observed and imputed entries.

Post-processing. The post-processing step follows a procedure complementary to pre-processing, applied in reverse order. Given its conceptual similarity to pre-processing, we omit a separate diagram for brevity. After diffusion and the polishing stage, the imputed matrix $\hat{X}_a$ is de-standardized using the pre-computed mean μ and standard deviation σ from the pre-processing step. Finally, categorical variables are decoded into their original formats, producing the final imputed matrix $\hat{X}$, which retains all observed entries from X, with only missing entries imputed.

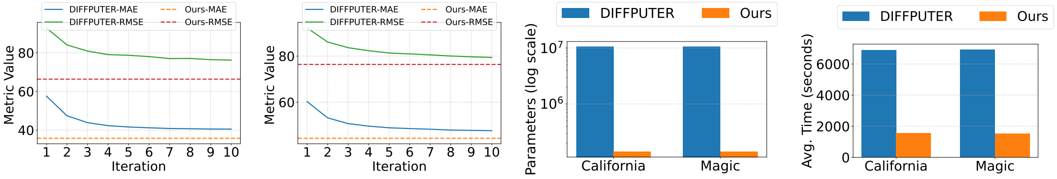
4 Experiments

In this section, we evaluate the performance of our proposed framework against existing imputation methods on nine benchmark datasets, following the experimental setup used in the recent SOTA method DIFFPUTER [51]. We report results under three standard missingness mechanisms: MNAR, MCAR, and MAR, consistent with established protocols [51, 10]. While prior work, such as [10, 51] does not consistently report results for all combinations of datasets and missingness mechanisms, we provide a complete evaluation across all three types to ensure a comprehensive assessment. To ensure fairness, all methods, including ours, are tested using identical 10 randomly generated masks across datasets. For baseline models, we adopt the original papers' official implementations and apply their recommended hyperparameter settings.

Datasets and Evaluation Metrics. We evaluate all methods on nine real-world benchmark datasets [51]. Of these, five datasets (California, Magic, Bean, Gesture, and Letter) consist exclusively of numerical features, while the remaining four (Default, News, Adult, and Shoppers) contain a mix of numerical and categorical features. For numerical attributes, we assess performance using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). For categorical attributes, we report classification accuracy. Detailed descriptions and statistics are provided in Appendix D.

Table 2: Performance comparison across all methods and settings for categorical columns. We report accuracy for in-sample and out-of-sample imputation under three missingness mechanisms: MNAR, MCAR, and MAR. Higher values indicate better performance. The best and second-best average ranks are highlighted in **bold** and underline, respectively.

Method	**MNAR**		**MCAR**		**MAR**		Rank (↓)
	In-Sample	Out-of-Sample	In-Sample	Out-of-Sample	In-Sample	Out-of-Sample	
EM	58.15	57.77	58.26	58.08	54.77	57.81	4.67
MIWAE	41.74	42.06	41.91	42.07	40.57	41.99	12.00
GAIN	48.31	46.98	48.86	47.62	44.99	47.21	9.33
SoftImpute	47.86	47.11	47.25	47.35	45.67	46.84	9.67
MICE	45.52	45.53	45.41	45.54	43.33	45.84	11.00
MIRACLE	54.97	54.57	55.22	54.46	49.84	51.43	7.33
KNN	53.77	54.09	54.14	54.01	53.37	55.65	7.50
MissForest	55.34	55.25	55.76	55.69	52.25	57.55	6.00
HyperImpute	59.69	58.96	60.14	58.59	54.42	57.04	4.50
DIFFPUTER	60.07	60.49	60.26	60.36	57.24	60.85	3.00
ReMasker	63.01	62.92	63.25	63.04	59.88	64.43	1.83
Ours	63.19	63.08	63.56	63.20	60.05	64.35	**1.17**



(a) California

(b) Magic

(c) Parameters

(d) Run time

Figure 3: Comparison between DIFFPUTER and our method on two datasets (California and Magic) under the MNAR setting. (a) and (b) show in-sample MAE and RMSE over iterations. (c) compares denoising network parameter counts. (d) presents average runtime over 10 random masks.

Data Processing. Each dataset is preprocessed by consolidating all features, both numerical and categorical, into a unified data matrix. Standardization is applied to the data as described in Section 3 Pre-processing. 70% is used as in-sample 30% is used as out-of-sample. A missing rate of 30% is introduced using pre-generated binary masks corresponding to the three missingness mechanisms: MNAR, MCAR, and MAR. Separate masks are applied for in-sample and out-of-sample evaluations to ensure unbiased assessment. Categorical features are encoded as fixed-length binary vectors, where each category is mapped to its binary representation and zero-padded to ensure uniform width across all entries. Missing values are initialized to zero by masking their positions in the input matrix during the pre-processing stage, ensuring that our method has no access to the ground truth for these entries during training. For evaluation, we follow the protocol used in DIFFPUTER, where only the categorical columns are de-standardized and decoded for computing classification accuracy, while the numerical columns remain in standardized form for MAE and RMSE calculations to ensure fair comparison. Full de-standardization is straightforward if required.

Baselines and Implementation. For benchmarking, we compare our model, RefiDiff with default structures and hyperparameters, against a carefully selected group of eleven baselines, representing classical and SOTA models. The baselines include EM [12], MIWAE [32], GAIN [49], SoftImpute [16], MICE [46], MIRACLE [28], KNN, MissForest [44], HyperImpute [19], DIFFPUTER [51], and ReMasker [10], which are representative of traditional iterative, matrix completion-based imputation techniques, deep learning-based methods, adversarial learning, diffusion methods. For each baseline, we adopt their implementations and configure them according to their recommended settings. All baseline methods and ours are consistently evaluated using the same masking patterns, evaluation metrics, and in-sample/out-of-sample settings to ensure a fair comparison. We report more details about the implementation, hyperparameters of the baselines, and ours in Appendix E.

Result Analysis. We report quantitative performance across all methods under three missingness mechanisms using both in-sample and out-of-sample evaluation metrics. Table 1 and Table 2 represent the average results for each setting and for each metric across all datasets and ten random masks for each dataset. Detailed results for each dataset are presented in Appendix F.

Table 1 presents the MAE and RMSE for (shown using base 10^{-2} for better readability, following DIFFPUTER) numerical feature imputation, while Table 2 summarizes accuracy scores for categorical feature recovery in percentage. Our model consistently achieves the lowest or second-lowest error across nearly all scenarios, demonstrating robust performance in both in-sample and out-of-sample settings. In particular, our method achieves the best average rank of 1.17 in both tables, clearly outperforming strong baselines such as DIFFPUTER, ReMasker, and HyperImpute.

In Table 1, our method consistently achieves lower MAE and RMSE across different missingness types, highlighting its superior ability to model complex data distributions and reconstruct continuous features effectively. These results indicate that our model generalizes well to the imputation task on different data and maintains strong imputation performance across both numerical and categorical features. Particularly in the challenging scenario for the MNAR case, our method substantially outperforms the SOTA baselines. The consistent superiority in both rank and raw performance underscores the effectiveness and scalability of our design choices, including the denoising and refinement stages. We further validate our results with statistical significance tests in Appendix G inspired by [8, 2, 18].

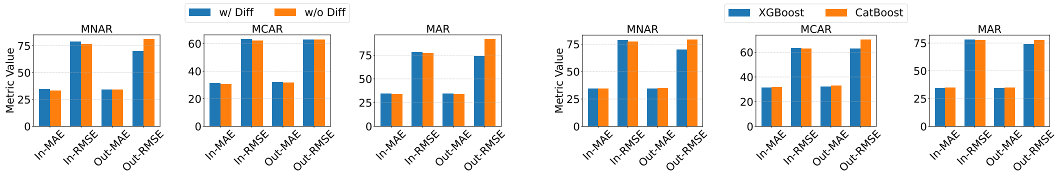
Figure 3 provides a comparison between our method and DIFFPUTER in terms of convergence behavior, model complexity, and runtime efficiency. Figure 3a and Figure 3b show the in-sample MAE and RMSE across iterations for the California and Magic datasets under the MNAR setting. While DIFFPUTER progressively improves performance over multiple iterations, our method achieves strong results without iteration, highlighting its ability to produce high-quality imputations without requiring iterations.

Figure 3c presents the number of trainable parameters in the denoising networks. Our designed denoising network uses significantly fewer parameters compared to DIFFPUTER, demonstrating that it is not only effective but also computationally efficient in terms of memory footprint.

Figure 3d compares the average runtime over 10 random masks. Since DIFFPUTER requires multiple iterations, its total runtime is substantially higher. In contrast, our method achieves better performance (around four times faster) with considerably lower computational cost. To ensure a fair comparison, both our method and DIFFPUTER were conducted on the same computational environment, using a single node equipped with one NVIDIA V100 GPU, 8 CPU cores, and 32 GB of RAM. These results demonstrate that our design is not only accurate but also efficient. Moreover, we have performed variation analysis to verify the stability of our model (see Appendix H), demonstrating RefiDiff's strong generalization ability.

5 Ablation Study and Sensitivity Analysis

To better understand the contribution and behavior of individual components within our framework, we conduct both ablation studies and sensitivity analyses. The ablation studies aim to isolate the effects of critical architectural choices, specifically the inclusion of the diffusion module and the choice of regression model, on imputation performance. In parallel, the sensitivity analysis investigates how our method responds to variations in the denoising network architecture and the number of sampling trials during the reverse diffusion process. Together, these evaluations offer insights into the robustness, computational efficiency, and generalization ability of our framework.



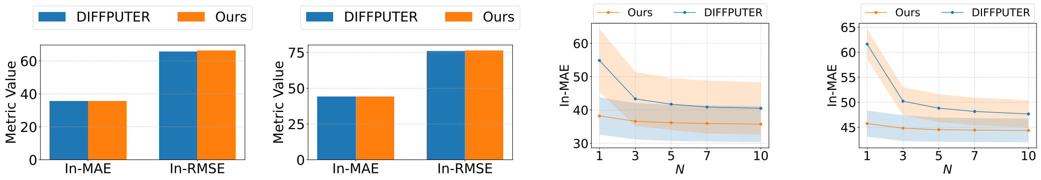
(a) With and without diffusion

(b) Ablation on regressor choice

Figure 4: Ablation experiments of our proposed framework in various settings and components.

Effectiveness of the Diffusion Module. To evaluate the impact of the diffusion module within our framework, we conduct an ablation study comparing the full model ("w/ Diff") to a reduced variant that includes only the warm-up and polishing stages ("w/o Diff"). As shown in Figure 4a, we report in-sample and out-of-sample MAE and RMSE under all three missingness mechanisms: MCAR, MAR, and MNAR. While the "w/o Diff" model achieves slightly lower in-sample MAE in the MCAR setting, it performs notably worse in out-of-sample RMSE, especially under MAR and MNAR. Specifically, the out-of-sample RMSE increases from 73.82 to 91.80 under MAR and from 70.12 to 81.07 under MNAR when diffusion is removed. These results highlight that the diffusion module plays a vital role in improving generalization and capturing complex feature interactions, particularly under more difficult missingness scenarios.

Ablation on Regressor Choice. The warm-up and polishing stages in our framework rely on a regression model to estimate missing numerical values. To evaluate the sensitivity of our framework to the choice of regressor, we compare XGBoost [6], which serves as the default in our main experiments, with CatBoost [9] as an alternative. As shown in Figure 4b, both regressors perform competitively across all missingness types and evaluation metrics. While CatBoost yields slightly better in-sample RMSE under the MCAR setting, XGBoost consistently achieves lower out-of-sample MAE and RMSE, particularly in the more challenging MAR and MNAR scenarios. These results justify our use of XGBoost as the default regressor, as it provides a favorable balance between accuracy and robustness. Importantly, both variants outperform existing SOTA baselines, reinforcing the generalizability and effectiveness of our overall framework.



(a) California (b) Magic (c) California (d) Magic

Figure 5: Sensitivity analysis of our framework. Subfigures (a, b) evaluate denoising networks, while (c, d) assess the impact of sampling trials N during reverse diffusion.

Sensitivity Analysis. We evaluate RefiDiff's sensitivity with respect to two primary factors: (1) the choice of denoising network and (2) the number of sampling trials N in the reverse diffusion process.

First, we compare our Mamba-based denoising module θ_2 against DIFFPUTER's larger Transformer-based denoiser (TabDDPM [27]). As shown in Figures 5a and 5b, our framework maintains comparable MAE and RMSE scores across datasets (within 2% difference), demonstrating robustness to denoiser architecture changes. Notably, our denoiser achieves these results with significantly fewer parameters (Figure 3c), offering substantial computational efficiency benefits.

Second, we examine the effect of varying the number of diffusion sampling trials N. While DIFFPUTER reported N to be critical for reducing stochastic variance, our method remains remarkably stable even with minimal sampling. As shown in Figures 5c and 5d, our approach maintains consistent performance with as few as 1 or 3 trials, whereas DIFFPUTER exhibits performance drops of up to 20% with reduced sampling. This stability eliminates the need for computationally expensive ensembling during inference.

Additional experiments in Appendix J further demonstrate our framework's versatility. When integrated into DIFFPUTER as a replacement denoiser, our θ_2 yields comparable performance, underscoring its plug-and-play capability. We also analyze the relationship between imputation performance and the Mamba block's hidden dimension size and its selective modeling of dependencies, providing insights into optimal model configuration and the mechanism behind its effectiveness.

6 Conclusion

This paper presents RefiDiff, a framework for robust data imputation in high-dimensional, mixed-type datasets with complex missingness. RefiDiff bridges gaps in existing methods by integrating local and global data characteristics via a progressive pre- and post-refinement strategy. Locally, it uses machine learning predictions, while globally, a Mamba-based denoising network captures feature and sample dependencies [1]. Pre-refinement generates initial imputations, refined post-hoc for accuracy and stability, enabling tuning-free imputation across diverse datasets. By encoding mixed-type data into unified tokens, RefiDiff ensures robust performance. Evaluations on nine real-world datasets show RefiDiff outperforms state-of-the-art methods in MCAR, MAR, and MNAR scenarios, excelling in MNAR with 4x faster training than the DIFFPUTER approach. Its efficiency and user-friendly design make RefiDiff ideal for practical applications. While binary encoding ensures compatibility with continuous diffusion, future work could explore native treatments of categorical variables to improve semantic fidelity. We also plan to extend RefiDiff to streaming data, adaptive refinement for sparse datasets, and applications in domains like healthcare and finance.

References

[1] Md Atik Ahamed and Qiang Cheng. MambaTab: A plug-and-play model for learning tabular data. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 369–375. IEEE, 2024.

[2] Md Atik Ahamed and Qiang Cheng. TSCMamba: Mamba meets multi-view learning for time series classification. *Information Fusion*, 120:103079, 2025.

[3] John Barnard and Xiao-Li Meng. Applications of multiple imputation in medical studies: from AIDS to NHANES. *Statistical Methods in Medical Research*, 8(1):17–36, 1999.

[4] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[5] R. Bock. MAGIC Gamma Telescope. UCI Machine Learning Repository, 2004. DOI: https://doi.org/10.24432/C52C8B.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] 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.

[8] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

[9] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

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

[11] Kelwin Fernandes, Pedro Vinagre, Paulo Cortez, and Pedro Sernadela. Online News Popularity. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5NS3V.

[12] 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.

[13] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2022.

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014.

[15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[16] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020.

[18] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[19] Daniel Jarrett, Bogdan C Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. In *International Conference on Machine Learning*, pages 9916–9937. PMLR, 2022.

[20] 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.

[21] Zhao Kang, Chong Peng, and Qiang Cheng. Top-n recommender system via matrix completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 26565–26577, 2022.

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577, 2022.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] Murat Koklu and Ilker Ali Özkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.*, 174:105507, 2020.

[27] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[28] Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. Miracle: Causally-aware imputation via learning missing data mechanisms. *Advances in Neural Information Processing Systems*, 34:23806–23817, 2021.

[29] David G Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1997.

[30] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[31] Renata Madeo, Priscilla Wagner, and Sarajane Peres. Gesture phase segmentation, 2013. DOI: https://doi.org/10.24432/C5Z32C.

[32] 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.

[33] Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, pages 7130–7140. PMLR, 2020.

[34] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[35] 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.

[36] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[37] Utomo Pujianto, Aji Prasetya Wibawa, Muhammad Iqbal Akbar, et al. K-nearest neighbor (k-nn) based missing data imputation. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, pages 83–88. IEEE, 2019.

[38] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[39] Trevor W Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A Bernal. Mcflow: Monte carlo flow models for data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14205–14214, 2020.

[40] C. Sakar and Yomi Kastro. Online Shoppers Purchasing Intention Dataset. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5F88Q.

[41] David Slate. Letter Recognition. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5ZP40.

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

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

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

[45] Anton Frederik Thielmann, Manish Kumar, Christoph Weisser, Arik Reuter, Benjamin Säfken, and Soheila Samiee. Mambular: A sequential model for tabular deep learning. *arXiv preprint arXiv:2408.06291*, 2024.

[46] Stef Van Buuren and Karin Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45:1–67, 2011.

[47] Yizhu Wen, Kai Yi, Jing Ke, and Yiqing Shen. DiffImpute: Tabular data imputation with denoising diffusion probabilistic model. *arXiv preprint arXiv:2403.13863*, 2024.

[48] I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.

[49] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, pages 5689–5698. PMLR, 2018.

[50] Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems*, 33:19075–19087, 2020.

[51] Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. Diffputer: An EM-driven diffusion model for missing data imputation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] He Zhao, Ke Sun, Amir Dezfouli, and Edwin V Bonilla. Transformed distribution matching for missing value imputation. In *International Conference on Machine Learning*, pages 42159–42186. PMLR, 2023.

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

[54] Jiajun Zhong, Ning Gui, and Weiwei Ye. Data imputation with iterative graph reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11399–11407, 2023.

APPENDIX

A Missingness Mechanisms

The appendices will be available soon.

B Proof of Theorem 1

The appendices will be available soon.

C Diffusion Details

The appendices will be available soon.

D Datasets Details

The appendices will be available soon.

E Implementation Details and Hyperparameters

The appendices will be available soon.

F Detailed results

The appendices will be available soon.

G Statistical Analysis

The appendices will be available soon.

H Variation analysis

The appendices will be available soon.

I Robustness of Our Designed Denoiser.

The appendices will be available soon.

J Additional Ablation Studies

The appendices will be available soon.