RefiDiff: Progressive Refinement Diffusion for Efficient Missing Data Imputation

Md Atik Ahamed, Qiang Ye, Qiang Cheng*

University of Kentucky
atikahamed@uky.edu, qye3@uky.edu, 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 efficiently capturing long-range dependencies among features and samples with low computational complexity. RefiDiff bridges the predictive and generative paradigms of imputation, leveraging 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, demonstrating strong performance in MNAR settings and superior out-of-sample generalization. Extensive evaluations on nine real-world datasets demonstrate its robustness, scalability, and effectiveness in handling complex missingness patterns.

Code — https://github.com/Atik-Ahamed/RefiDiff

1 Introduction

Missing values are a pervasive challenge in real-world datasets, arising from sensor failures, data corruption, or operational issues. Accurate imputation is essential for downstream analysis and modeling, particularly in high-dimensional, mixed-type datasets such as those from the UCI Machine Learning Repository (Kelly, Longjohn, and Nottingham 2023) and other sources (Koklu and Özkan 2020; Pace and Barry 1997).

A wide range of imputation methods has been developed, including statistical techniques, matrix completion, deep generative models, diffusion-based methods, and hybrid frameworks. These methods aim to address three canonical missingness mechanisms: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). Among these, MNAR is the most challenging, as the probability of missingness depends on the missing values themselves (Barnard and Meng 1999; Kyono et al. 2021; Géron 2022). In contrast, MCAR is the most tractable, with missingness independent of both observed and unobserved data (Van Buuren and Groothuis-Oudshoorn 2011).

Most imputation methods fall into two fundamental paradigms: *predictive* approaches, which estimate missing values through direct mappings from observed entries (e.g., via regressors or classifiers), and *generative* approaches, which treat imputation as a stochastic process by modeling the underlying data distribution (e.g., via diffusion models). Predictive methods are typically accurate and efficient but deterministic, often lacking uncertainty modeling. Generative models, by contrast, offer uncertainty-aware and diverse imputations but are computationally intensive and slower.

These paradigmatic differences are further reflected in their methodological perspectives: predictive models tend to adopt a *local* view, inferring missing values using partial observations per feature, while generative models take a *global* view, modeling the full data distribution. However, few methods effectively integrate these paradigms and perspectives, especially for high-dimensional, mixed-type data. This gap results in limited performance under MNAR settings, reduced robustness to distribution shifts, and sensitivity to hyperparameters. Despite their complementary strengths, predictive and generative approaches have rarely been unified into a general-purpose framework, representing a missed opportunity to jointly leverage deterministic precision and probabilistic robustness.

To address these challenges, we propose **RefiDiff**, an innovative framework that unifies predictive and generative paradigms through a progressive refinement process. RefiDiff begins with local imputations from machine learning regressors/classifiers, organized into a warm-up (pre-refinement) and polishing (post-refinement) sequence (Figure 1). The preliminary estimates are encoded as unified tokens and passed to a global generative stage powered by diffusion. For this, we introduce a Mamba-based denoising network (Gu and Dao 2023; Ahamed and Cheng 2024), a selective state-space model with linear complexity and Transformer-level expressivity. This design combines predictive efficiency with diffusion flexibility to model long-range dependencies in complex, mixed-type data.

*Corresponding author

RefiDiff outperforms state-of-the-art methods such as DIFFPUTER (Zhang et al. 2025) across all three missingness scenarios (MCAR, MAR, MNAR), with particularly strong performance under MNAR. Compared to DIFFPUTER's TabDDPM, our Mamba-based model reduces computational cost by $4\times$ while maintaining high accuracy. Extensive evaluations on nine real-world datasets confirm RefiDiff's robustness, scalability, and efficiency.

In summary, our main contributions are as follows:

- We propose a progressive refinement strategy that systematically unifies predictive and generative paradigms for robust and flexible imputation.
- We introduce a novel Mamba-based denoising model for capturing long-range dependencies efficiently in high-dimensional, mixed-type tabular data.
- RefiDiff is a plug-and-play solution, requiring no architectural or hyperparameter tuning across diverse datasets.
- It achieves state-of-the-art performance under MCAR, MAR, and MNAR settings, with notable gains in MNAR scenarios.

We validate RefiDiff through comprehensive experiments and ablation studies on nine real-world datasets.

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 (García-Laencina, Sancho-Gómez, and Figueiras-Vidal 2010; Dempster, Laird, and Rubin 1977) provides maximum likelihood estimates of model parameters under a pre-specified model for imputing missing data. Multiple Imputation by Chained Equations (MICE) (Van Buuren and Groothuis-Oudshoorn 2011) uses regression models to impute values iteratively, while K-Nearest Neighbors (KNN) imputation (Pujianto et al. 2019) estimates missing values based on data point similarity. MissForest (Stekhoven and Bühlmann 2012) 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 (Hastie et al. 2015) 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 (Luenberger 1997; Kang, Peng, and Cheng 2016). However, these methods may 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) (Kingma and Welling 2013) and their extensions, such as MIWAE (Mattei and Frellsen 2019) and MissVAE (Nazabal et al. 2020), model the data distribution to impute missing values, capturing complex patterns through latent representations. Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), as seen in GAIN (Yoon, Jordon, and Schaar 2018), employ adversarial training to generate realistic imputations and have been adapted for tabular data. Normalizing flows (Rezende and Mohamed 2015), as explored in MCFlow (Richardson et al. 2020), 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 (Kingma and Ba 2015).

Diffusion models have recently emerged as a promising paradigm for data imputation, inspired by their success in image generation (Ho, Jain, and Abbeel 2020; Karras et al. 2022a; Lugmayr et al. 2022). TabDDPM (Kotelnikov et al. 2023) adapts denoising diffusion probabilistic models for tabular data, modeling data generation as a reverse diffusion process. Similarly, TabCSDI (Zheng and Charoenphakdee 2022) and MissDiff (Ouyang et al. 2023) focus on handling missing values by conditioning diffusion processes on observed data. DIFFPUTER (Zhang et al. 2025) integrates EM-driven diffusion for robust imputation, while DiffImpute (Wen et al. 2024) leverages denoising diffusion to impute tabular data with many iterations. ForestDiff (Jolicoeur-Martineau, Fatras, and Kachman 2024) 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 (Song et al. 2021a,b), 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 (Kyono et al. 2021), incorporate causal relationships to improve imputation accuracy, particularly in datasets with structural dependencies. Graph-based approaches, including GRAPE (You et al. 2020) and IGRM (Zhong, Gui, and Ye 2023), 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 (Jarrett et al. 2022) employs automatic model selection, e.g., via meta-learning, to choose the best imputation method for a given dataset, enhancing generalizability. ReMasker (Du, Melis, and Wang 2024) utilizes masked autoencoding to impute tabular data, drawing inspiration from self-supervised learning paradigms. Optimal transport-based methods, such as MOT (Muzellec et al. 2020) and TDM (Zhao et al. 2023), frame imputation as a distribution matching problem, offering robust solutions for heterogeneous data. These approaches are flexible but computationally demanding.

In summary, these data imputation methods have ad-

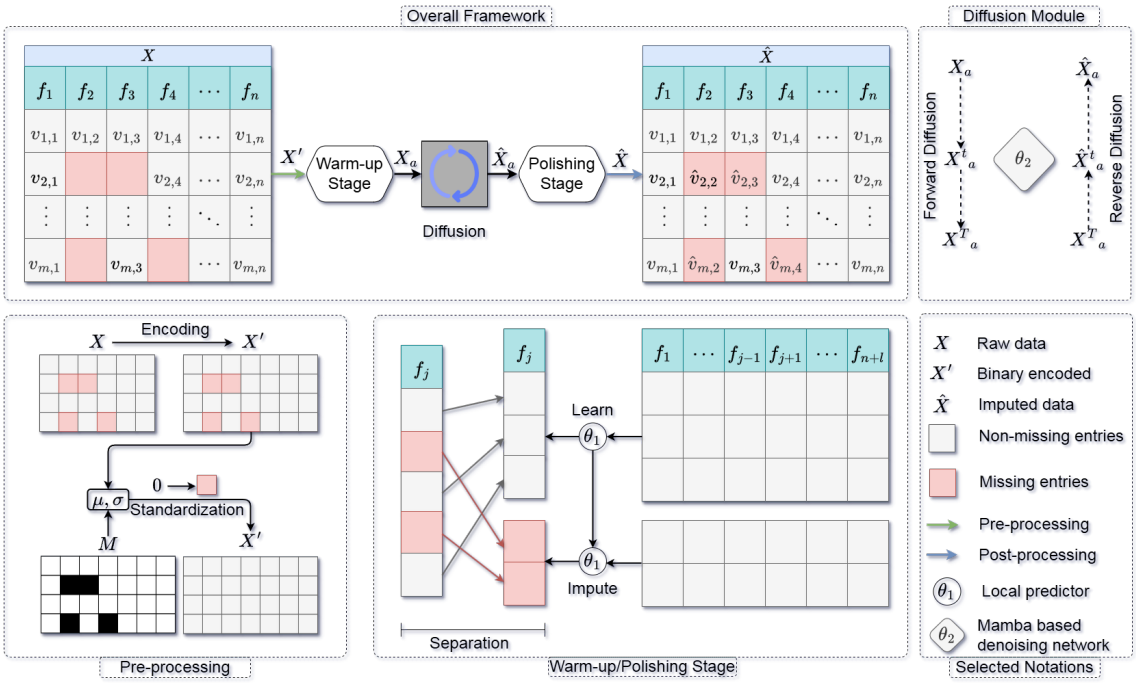


Figure 1: Overview of the proposed imputation framework. The process begins with a warm-up stage on pre-processed data, followed by a diffusion module that iteratively denoises the data. A polishing stage further enhances the imputations. Our designed denoiser θ_2 will be shown in Figure 2.

vanced 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: Pre-processing, Warm-up Refinement, Diffusion Imputation, and Post-processing and Polishing, which we detail below.

Preliminaries. We consider a dataset represented as a matrix $X \in \mathbb{R}^{m\times n}$, where m is the number of samples and n is the number of features. Missing entries in X are indicated by a binary mask $M \in \{0,1\}^{m\times n}$, where $M_{i,j} = 1$ means $X_{i,j}$ is missing and $M_{i,j} = 0$ means it is observed. Our goal is to estimate all missing values $X_{i,j}$ for which $M_{i,j} = 1$, using only the observed portions of the data. The missingness pattern can follow any of the standard mechanisms: MCAR, MAR, or MNAR (details are provided in Appendix A (Ahamed, Ye, and Cheng 2025)), and we consider both in-sample and out-of-sample scenarios when evaluating our method. Regardless of mechanism or data split, the core objective remains the same: to robustly impute missing entries across all three missingness settings and generalize well to unseen (out-of-sample) data.

Pre-processing. Real-world datasets can contain both numerical and categorical features. We first apply a type-preserving encoding and normalization pipeline to prepare the data for downstream modeling. Categorical features are binary-encoded, yielding l additional binary indicator columns. These are concatenated with the n numerical features to form an expanded matrix $X' \in \mathbb{R}^{m\times d}$ with $d = n+l$ columns. The mask M is expanded correspondingly to d columns (marking the new binary features as missing wherever the original categorical entry was missing). Next, we standardize each numerical feature using its mean μ and standard deviation σ computed from the *non-missing training entries only* (to avoid leakage of test information or missingness bias). The same μ and σ are later used to standardize that feature in the out-of-sample set. All observed values in X' are thus transformed to have zero mean and unit variance (approximately), while missing entries are temporarily filled with zeros as neutral placeholders. After pre-processing, we get a standardized matrix with normalized observed entries and zero-filled missing entries awaiting imputation.

Warm-up Refinement. Before invoking the diffusion model, we perform a warm-up imputation that provides initial guesses for all missing values in X'. The idea is to leverage fast, local models to ease the subsequent global modeling task, by filling in plausible values; especially in heavy-

MNAR or out-of-distribution cases, we reduce the burden on the diffusion model, which would otherwise have to start from arbitrary initializations (e.g., zero-filled vectors). We adopt a simple column-wise regression strategy inspired by methods like MICE, but done in a single pass. Concretely, for each feature column f_j (where $j = 1, \ldots, d$), we use the current partially imputed data to train a lightweight predictive model $\theta_1^{(j)}$ (e.g., a default XGBoost or CatBoost regressor/classifier). The model $\theta_1^{(j)} : \mathbb{R}^{d-1} \to \mathbb{R}$ is trained to predict feature f_j from all other features: we take all samples i where f_j is observed ($M_{i,j} = 0$) and use $(X'_{i,\setminus j}, X'_{i,j})$ as input-output pairs to fit $\theta_1^{(j)}$. Once trained, we apply $\theta_1^{(j)}$ to estimate the missing entries in column j, i.e. for each i with $M_{i,j} = 1$ we set $(X_a)_{i,j} := \theta_1^{(j)}(X'_{i,\setminus j})$. We then discard $\theta_1^{(j)}$ and move to the next column. This process sweeps through the feature set $j = 1$ to d exactly once. Unlike iterative methods (EM, MICE) or recent models like DIFFPUTER that perform multiple passes, our warm-up produces a complete imputed dataset X_a in *one pass*, greatly improving efficiency.

Properties of One-Pass Imputation: This refinement has three key properties: (i) *Non-overwriting*: all originally observed entries remain unchanged in X_a; (ii) *Well-defined mapping*: each missing entry is imputed by a deterministic function of that sample's other features (through the learned model for that column); and (iii) *One-pass completion*: each feature is processed once and each missing value is filled exactly one time, so the procedure terminates after a single sweep. These outcomes are guaranteed by construction, and we formalize them in Proposition 2 (Appendix B (Ahamed, Ye, and Cheng 2025)) and provide a proof there for completeness. In summary, the warm-up stage yields an initial imputed matrix X_a that preserves all observed data and provides reasonable first-fill values for all missing entries.

Diffusion Module. After warm-up, we feed the refined data X_a into a generative diffusion model to *globally refine* the imputation. The diffusion module treats the entire dataset (or each data sample) as input and performs a denoising diffusion process conditioned on the mask M. The core idea is to progressively replace the initial guesses in X_a with more accurate values by sampling from a model of the joint data distribution, all while exactly preserving the observed entries. We adopt a continuous-time diffusion process: starting from $X_0 = X_a$, we gradually add Gaussian noise to corrupt X_0 into a noisy version X_t (for t increasing to some T), according to a noise schedule $\sigma(t)$ (details in Appendix C (Ahamed, Ye, and Cheng 2025)). A denoising network θ_2 is trained to reverse this process, i.e., to predict the instantaneous noise at each step t and push X_t back toward the clean X_0. Crucially, θ_2 is given access to the mask M and is designed to leave observed components untouched: at each denoising step, we clamp X_t at the observed coordinates to equal X_a, and update only the missing coordinates. This way, the diffusion never alters ground-truth data and focuses its denoising only on unknown parts. We use a lightweight yet expressive architecture for θ_2, illustrated in Figure 2.

It has a symmetric "diamond" structure with two residual

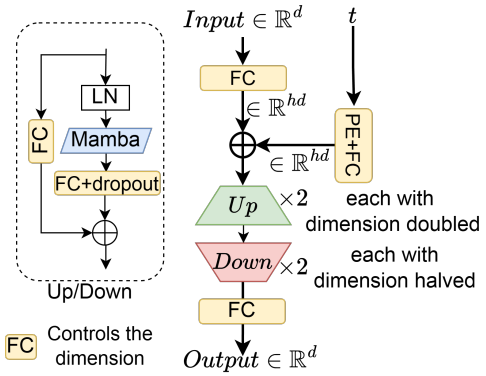


Figure 2: Denoising network θ_2, featuring a diamond-shaped structure with Mamba-based residual Up/Down blocks.

up-sampling blocks followed by two down-sampling blocks (each block built around the Mamba state-space layer to capture long-range dependencies efficiently). This design gives Transformer-like expressivity with linear complexity, enabling us to model high-dimensional data (many features and samples) without quadratic cost. Each block applies dimension doubling or halving (for up/down steps) with skip connections, fully connected layers (FC), positional encoding (PE), layer normalization, and dropout, yielding a robust multi-scale denoiser.

During training, we optimize θ_2 using the EDM loss (Karras et al. 2022b). Specifically, the loss is of the form:

$$\mathcal{L}_{\text{SM}}(\theta_2) = \mathbb{E}_{X_0,\,\varepsilon,\,t}\big[\|\theta_2(X_t, t, M) - \nabla_{X_t} \log p(X_t \,|\, X_0)\|_2^2\big].$$

This loss weights the denoising objective according to noise magnitude (so the model learns to handle all corruption levels), thus handling noise-adaptivity better than standard score-matching loss. At inference, we run the reverse diffusion: initialize X_T as Gaussian noise (with observed entries still clamped to X_a), and iteratively denoise from $t = T$ down to $t = 0$ using θ_2. We perform N stochastic runs of this reverse process for each input and average the results to obtain a final imputed output $\hat{X}_a$. This ensembling over multiple diffusion trajectories improves the robustness and stability of the imputation.

We adopt the Variance Exploding (VE) SDE formulation for our diffusion process, as it provides higher noise levels at earlier timesteps, which empirically leads to improved sample diversity and better exploration of the missing-value space. VE SDE is particularly advantageous in imputation tasks under MNAR conditions, where the model benefits from starting from a more randomized state to avoid overfitting to local patterns. Unlike VP SDEs (e.g., DDPM), which gradually corrupt data with bounded noise, VE allows a more aggressive corruption and reconstruction process, which improves robustness in our experiments. We found this choice to yield more stable and diverse imputations.

To provide a theoretical foundation for the validity of our masked diffusion-based imputation, we establish a condi-

tional sampling guarantee under ideal training assumptions: The RefiDiff reverse process recovers the true conditional distribution of the missing values given the observed ones. This result and proof can be found in Appendix D (Ahamed, Ye, and Cheng 2025).

To further strengthen the theoretical foundation of RefiDiff, we provide a quantitative bound on its imputation accuracy. Specifically, we analyze how closely the reverse diffusion process recovers the true conditional distribution $p(\mathbf{x}^{\text{mis}} \mid \mathbf{x}^{\text{obs}})$, in terms of the quality of the learned denoising score function, the discretization of the diffusion process, and the number of stochastic samples used. This result is formalized in the proposition below:

Proposition 1 (Approximation Bound for RefiDiff)**.** *Under mild regularity conditions, the KL divergence between the RefiDiff-imputed distribution $\hat{p}(\mathbf{x}^{mis} \mid \mathbf{x}^{obs})$ and the true conditional distribution is bounded by:*

$$\mathrm{KL}\Big(\hat{p}(\mathbf{x}^{mis} \,|\, \mathbf{x}^{obs}) \,\Big\|\, p(\mathbf{x}^{mis} \,|\, \mathbf{x}^{obs})\Big) \leq C_1 T\, \varepsilon_\theta^2 + C_2\, \delta t + C_3\, \tfrac{1}{N},$$

where ε_θ is the error of the learned score function, δt is the diffusion step size, N is the number of reverse diffusion trajectories averaged, and C_1, C_2, C_3 are constants independent of the data.

The full version of this proposition and its brief proof are provided in Appendix E (Ahamed, Ye, and Cheng 2025). This bound shows that RefiDiff's imputation converges to the Bayes-optimal conditional mean as the score model improves, the diffusion steps are refined, and sufficient averaging is performed. Compared to prior qualitative results, this quantitative guarantee helps justify our design choices, including the use of ensembling, careful time discretization, and the progressive refinement pipeline. This bound also motivates our specific design strategies, including multi-trajectory averaging and a progressive denoising approach.

Post-processing and Polishing. Finally, we apply a brief post-processing and polishing step. After the polishing stage, we invert the earlier standardization and encoding: the entries of $\hat{X}$ are de-standardized by multiplying by σ and adding μ for each feature, and binary-encoded categorical columns are decoded back to the original category labels. This yields the final imputed data matrix $\hat{X}$ in the original feature space. As an additional polishing, optionally, we perform a final sweep of the lightweight column-wise models $\theta_1^{(j)}$ on $\hat{X}_a$ to refine any residual discrepancies introduced during diffusion. This polishing may help because diffusion may introduce slight distributional noise, a quick predictive pass can adjust the estimates to better conform to typical feature distributions. This polishing re-uses the same one-pass procedure described earlier, now applied to $\hat{X}_a$. The polishing ensures that $\hat{X}$ retains all originally observed entries (by construction) and that all imputed values are as reasonable and consistent as possible feature-wise. After post-processing, $\hat{X}$ is ready for downstream use. It contains no missing entries, preserves information from X, and reflects a blend of local predictive accuracy and global generative coherence.

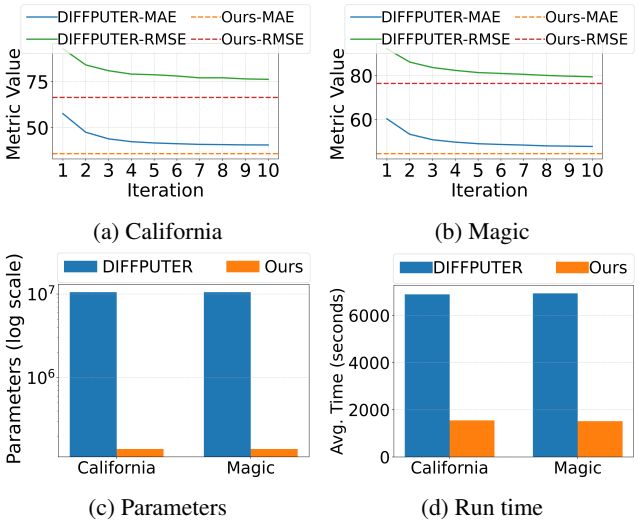


Figure 3: Comparison between DIFFPUTER and our method on two datasets (California and Magic) under the MNAR setting. (a) and (b) show in-sample MAE and RMSE over iterations. (c) compares denoising network parameter counts. (d) presents average runtime over 10 random masks.

Collectively, these stages (pre-processing, warm-up refinement, diffusion-based global refinement, and post-processing) form a theoretically grounded and practically robust imputation pipeline for diverse missingness patterns.

4 Experiments

In this section, we evaluate our framework against existing imputation methods on nine benchmark datasets, following the setup of DIFFPUTER (Zhang et al. 2025). Results are reported under three missingness mechanisms: MNAR, MCAR, and MAR, adhering to established protocols (Zhang et al. 2025; Du, Melis, and Wang 2024). Unlike prior work (Du, Melis, and Wang 2024; Zhang et al. 2025) that omits some dataset-mechanism combinations, we conduct a complete evaluation for all three types. To ensure fairness, all methods are tested with identical 10 random masks. For baselines, we use official implementations with recommended hyperparameters. Performance is measured only on missing entries, as observed values remain unchanged.

Datasets and Evaluation Metrics. We evaluate all methods on nine real datasets (Zhang et al. 2025): five with only numerical features (California, Magic, Bean, Gesture, Letter) and four with mixed features (Default, News, Adult, Shoppers). Performance is measured by MAE and RMSE for numerical attributes and accuracy for categorical ones. Dataset details are in Appendix F (Ahamed, Ye, and Cheng 2025).

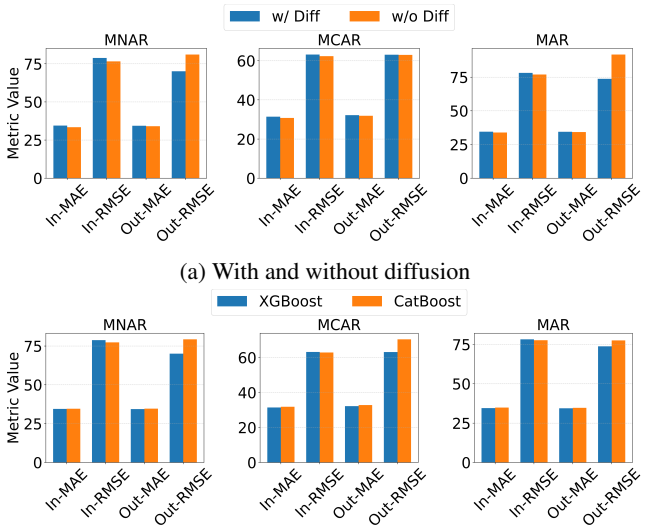
Data Processing. All features (numerical and categorical) are merged into a unified data matrix and standardized (Section 3 Pre-processing). We use 70% of the data as in-sample and 30% as out-of-sample, introducing 30% missingness via binary masks for MNAR, MCAR, and MAR. Separate masks are used for in- and out-of-sample evaluations. Categorical features are binary-encoded and zero-

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	<u>2.67</u>
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**

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 <u>underline</u>, respectively.

Method	MNAR		MCAR		MAR		Rank
	IS	OOS	IS	OOS	IS	OOS	↓
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
SI	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
MF	55.34	55.25	55.76	55.69	52.25	57.55	6.00
HI	59.69	58.96	60.14	58.59	54.42	57.04	4.50
DP	60.07	60.49	60.26	60.36	57.24	60.85	3.00
RM	63.01	62.92	63.25	63.04	59.88	64.43	<u>1.83</u>
Ours	63.19	63.08	63.56	63.20	60.05	64.35	**1.17**

Table 2: Performance comparison across all methods and settings for categorical columns. We report accuracy for in-sample (IS) and out-of-sample (OOS) imputation under all missingness mechanisms. Higher values indicate better performance. The best and second-best average ranks are highlighted in **bold** and <u>underline</u>, respectively.



(a) With and without diffusion

(b) Ablation on regressor choice

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

padded for uniform width, while missing values are zero-initialized and masked to prevent access to ground truth during training. Following DIFFPUTER, categorical columns are de-standardized and decoded for accuracy, while numerical columns remain standardized for MAE and RMSE. Full de-standardization can be applied if needed.

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 (García-Laencina, Sancho-Gómez, and Figueiras-Vidal 2010), MIWAE (Mattei and Frellsen 2019), GAIN (Yoon, Jordon, and Schaar 2018), SoftImpute (SI) (Hastie et al. 2015), MICE (Van Buuren and Groothuis-Oudshoorn 2011), MIRACLE (Miracle) (Kyono et al. 2021), KNN, MissForest (MF) (Stekhoven and Bühlmann 2012), HyperImpute (HI) (Jarrett et al. 2022), DIFFPUTER (DP) (Zhang et al. 2025), and ReMasker (RM) (Du, Melis, and Wang 2024), which are representative of traditional iterative, matrix completion-based imputation techniques, deep learning-based methods, adversarial learning, diffusion methods. We use official implementations and recommended settings for all baselines, evaluating them under the same masks, metrics, and data splits for fairness. Implementation and hyperparameter details are in Appendix G (Ahamed, Ye, and Cheng 2025).

Result Analysis. We evaluate all methods under three missingness mechanisms using in- and out-of-sample metrics. Tables 1 and 2 show averaged results across datasets and masks, with detailed results in Appendix K (Ahamed, Ye, and Cheng 2025).

Table 1 reports MAE/RMSE (scaled by 10^{-2} for better readability) for numerical imputation, and Table 2 shows categorical accuracy. RefiDiff achieves the best or second-best performance in most cases, with an average rank of 1.17, outperforming SOTA baselines like DIFFPUTER, ReMasker, and HyperImpute.

In Table 1, our method achieves consistently lower MAE and RMSE across all missingness types, demonstrating strong generalization for both numerical and categorical features. It notably surpasses SOTA baselines, particularly in the challenging MNAR scenario, due to our effective denoising and refinement stages. Our model shows statistically significant improvement over baseline models. We validate our results with statistical significance tests in Appendix H (Ahamed, Ye, and Cheng 2025).

Figure 3 compares our method with DIFFPUTER in terms of convergence, complexity, and runtime. Figures 3a-3b show in-sample MAE and RMSE for California and Magic datasets under MNAR. While DIFFPUTER improves gradually over iterations, our method achieves strong results without iterations, showing its ability to produce high-quality imputations efficiently. Figure 3c shows that our denoising network uses far fewer parameters than DIFFPUTER, showing both effectiveness and memory efficiency.

Figure 3d compares average runtime over 10 random masks. DIFFPUTER's iterative process makes it much slower, while our method is about four times faster with lower computational cost. Both methods were tested on the same setup (NVIDIA V100 GPU, 8 CPU cores, 32 GB RAM). These results confirm that our design is both accurate and efficient, with variation analysis in Appendix I (Ahamed, Ye, and Cheng 2025), further verifying RefiDiff's stability and generalization.

5 Ablation Study and Sensitivity Analysis

To analyze the contributions of individual components, we perform ablation and sensitivity studies. The ablation examines the impact of key design choices, such as the diffusion module and regression model, on imputation performance. Sensitivity analysis explores the effect of varying the denoising network architecture and the number of sampling trials in reverse diffusion. Together, these evaluations reveal the framework's robustness, efficiency, and generalization.

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 RefiDiff 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 (Chen and Guestrin 2016), which serves as the default in our main experiments, with CatBoost (Dorogush, Ershov, and Gulin 2018) 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.

Additional experiments in Appendix J (Ahamed, Ye, and Cheng 2025) further demonstrate RefiDiff's versatility. When integrated into DIFFPUTER as a replacement denoiser (Figure 8 in Appendix J (Ahamed, Ye, and Cheng 2025)), 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 (Figures 9 and 10 in Appendix J (Ahamed, Ye, and Cheng 2025)), 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 (Ahamed and Cheng 2024). 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.

Acknowledgments

This work was supported in part by the NSF under Grants IIS 2327113 and ITE 2433190; and the NIH under Grants P30AG072946. We would like to thank the NSF support for AI research resources with NAIRR240219, Jetstream2, PSC, and the University of Kentucky Center for Computational Sciences and Information Technology Services Research Computing for their support and use of the LCC.

References

Ahamed, M. A.; and Cheng, Q. 2024. MambaTab: A plug-and-play model for learning tabular data. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 369–375. IEEE.

Ahamed, M. A.; Ye, Q.; and Cheng, Q. 2025. RefiDiff: Progressive Refinement Diffusion for Efficient Missing Data Imputation. *arXiv preprint arXiv:2505.14451*.

Barnard, J.; and Meng, X.-L. 1999. Applications of multiple imputation in medical studies: from AIDS to NHANES. *Statistical Methods in Medical Research*, 8(1): 17–36.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (methodological)*, 39(1): 1–22.

Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Du, T.; Melis, L.; and Wang, T. 2024. ReMasker: Imputing Tabular Data with Masked Autoencoding. In *The Twelfth International Conference on Learning Representations*.

García-Laencina, P. J.; Sancho-Gómez, J.-L.; and Figueiras-Vidal, A. R. 2010. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19: 263–282.

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

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2672–2680.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hastie, T.; Mazumder, R.; Lee, J. D.; and Zadeh, R. 2015. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1): 3367–3402.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 6840–6851.

Jarrett, D.; Cebere, B. C.; Liu, T.; Curth, A.; and van der Schaar, M. 2022. Hyperimpute: Generalized iterative imputation with automatic model selection. In *International Conference on Machine Learning*, 9916–9937. PMLR.

Jolicoeur-Martineau, A.; Fatras, K.; and Kachman, T. 2024. Generating and Imputing Tabular Data via Diffusion and Flow-based Gradient-Boosted Trees. In *International Conference on Artificial Intelligence and Statistics*, 1288–1296. PMLR.

Kang, Z.; Peng, C.; and Cheng, Q. 2016. Top-n recommender system via matrix completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022a. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 26565–26577.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022b. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 26565–26577.

Kelly, M.; Longjohn, R.; and Nottingham, K. 2023. The UCI machine learning repository.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Koklu, M.; and Özkan, I. A. 2020. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.*, 174: 105507.

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

Kyono, T.; Zhang, Y.; Bellot, A.; and van der Schaar, M. 2021. Miracle: Causally-aware imputation via learning missing data mechanisms. *Advances in Neural Information Processing Systems*, 34: 23806–23817.

Luenberger, D. G. 1997. *Optimization by Vector Space Methods*. John Wiley & Sons.

Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Van Gool, L. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11461–11471.

Mattei, P.-A.; and Frellsen, J. 2019. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, 4413–4423. PMLR.

Muzellec, B.; Josse, J.; Boyer, C.; and Cuturi, M. 2020. Missing data imputation using optimal transport. In *International Conference on Machine Learning*, 7130–7140. PMLR.

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

Ouyang, Y.; Xie, L.; Li, C.; and Cheng, G. 2023. Missdiff: Training diffusion models on tabular data with missing values. *arXiv preprint arXiv:2307.00467*.

Pace, R. K.; and Barry, R. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297.

Pujianto, U.; Wibawa, A. P.; Akbar, M. I.; et al. 2019. K-nearest neighbor (k-NN) based missing data imputation. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, 83–88. IEEE.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 1530–1538. PMLR.

Richardson, T. W.; Wu, W.; Lin, L.; Xu, B.; and Bernal, E. A. 2020. Mcflow: Monte carlo flow models for data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14205–14214.

Song, Y.; Durkan, C.; Murray, I.; and Ermon, S. 2021a. Maximum Likelihood Training of Score-Based Diffusion Models. In *Advances in Neural Information Processing Systems*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021b. Score-Based Generative Modeling through Stochastic Differential Equations. In *The Ninth International Conference on Learning Representations*.

Stekhoven, D. J.; and Bühlmann, P. 2012. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1): 112–118.

Van Buuren, S.; and Groothuis-Oudshoorn, K. 2011. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45: 1–67.

Wen, Y.; Yi, K.; Ke, J.; and Shen, Y. 2024. DiffImpute: Tabular Data Imputation With Denoising Diffusion Probabilistic Model. *arXiv preprint arXiv:2403.13863*.

Yoon, J.; Jordon, J.; and Schaar, M. 2018. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, 5689–5698. PMLR.

You, J.; Ma, X.; Ding, Y.; Kochenderfer, M. J.; and Leskovec, J. 2020. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems*, 33: 19075–19087.

Zhang, H.; Fang, L.; Wu, Q.; and Yu, P. S. 2025. DiffPuter: An EM-Driven Diffusion Model for Missing Data Imputation. In *The Thirteenth International Conference on Learning Representations*.

Zhao, H.; Sun, K.; Dezfouli, A.; and Bonilla, E. V. 2023. Transformed distribution matching for missing value imputation. In *International Conference on Machine Learning*, 42159–42186. PMLR.

Zheng, S.; and Charoenphakdee, N. 2022. Diffusion models for missing value imputation in tabular data. In *NeurIPS 2022 First Table Representation Workshop*.

Zhong, J.; Gui, N.; and Ye, W. 2023. Data imputation with iterative graph reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11399–11407.