Latent Diffusion for Missing Data

Alberte Heering Estad[1] **Ignacio Peis**[1,2] **Jes Frellsen**[1,2]

[1]Technical University of Denmark
[2]Pioneer Centre for Artificial Intelligence

Abstract

Diffusion models have emerged as powerful generative approaches for missing-data imputation, yet most existing methods operate directly in data space and degrade when training data are heavily incomplete. We investigate whether shifting diffusion to a learned latent representation improves robustness under missing-completely-at-random (MCAR) corruption. To this end, we propose a two-stage framework: a robust VAE-based imputer first learns compact semantic features from incomplete observations, and a diffusion model is then trained in the resulting latent space. Across training missing rates, we perform a controlled comparison against pixel-space diffusion models under the same incomplete-data setting. The latent diffusion model maintains high sample quality and remains stable up to 50% missingness, while pixel-space diffusion degrades progressively as missingness increases. For downstream imputation, latent diffusion also achieves consistently better performance than pixel-space diffusion. These findings indicate that latent-space modeling mitigates artifact amplification from zero-imputed inputs and provides a more robust generative prior for incomplete-data learning. Overall, our results support latent diffusion as a strong and practically useful alternative to pixel-space diffusion for missing-data problems.

1. Introduction

Real-world datasets are often incomplete. Missing values arise from from varied sources including loss to follow-up in clinical studies, incomplete documentation in electronic health records, and practical constraints in data collection (Wells et al., 2013; Chang et al., 2021). The missing data mechanism is typically categorized as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) (Little and Rubin, 2002). Probabilistic generative models offer a principled approach to this problem, as they can learn the data distribution from incomplete observations and impute missing values by sampling from the conditional. Deep generative approaches based on VAEs (Nazabal et al., 2020; Peis et al., 2022; Mattei and Frellsen, 2019), GANs (Yoon et al., 2018; Li et al., 2019) and Flows (Richardson et al., 2020) have shown strong results for imputation. More recently, diffusion models have been applied to this task (Zhang et al., 2025; Givens et al., 2025; Ouyang et al., 2023; Zheng and Charoenphakdee, 2023; Tashiro et al., 2021), leveraging the connection between score-based generative modeling and conditional generation (Song et al., 2021). However, existing diffusion-based methods typically operate in the data space and often assume complete training data. The question of how the choice of either pixel or latent representation space affects both generation quality and imputation performance when training on incomplete data has, to the best of our knowledge, not been studied.

In this work, we present a systematic comparison between diffusion models for missing data in pixel space and our proposed method, denoted LDMiss, which performs diffusion in a latent space learned by a VAE for incomplete data. Both models are trained on MNIST under MCAR missingness

at varying rates. We evaluate both generative sample quality and imputation performance. Our results show that LDMiss retains higher sample quality as missingness increases and achieves superior imputation performance compared with the pixel-space DDPM.

2. Score-Based Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2021; Ho et al., 2020) generate data by progressively corrupting training samples with noise and then learning the reverse denoising process. Denoising Diffusion Probabilistic Models (DDPMs; Ho et al. (2020)) formulate this process as a discrete-time Markov chain. In contrast, score-based diffusion models (Song et al., 2021) define a continuous-time forward process $\{\mathbf{x}_t\}_{t=0}^{T}$ with $t \in [0, T]$ through the stochastic differential equation (SDE) $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$. The process starts from data $\mathbf{x}_0 \sim p_0$ and evolves toward a prior distribution $\mathbf{x}_T \sim p_T$ that contains no information about p_0.

The generative process is characterized by the corresponding reverse SDE: $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$. This reverse SDE depends only on the time-dependent gradient field $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, known as the score, of the transformed data distribution. The score can be estimated by training a time-dependent neural network $\mathbf{s}_\theta$ via the Denoising Score-Matching objective (Vincent, 2011):

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(\tau,1)} \left\{ \lambda_t \, \mathbb{E}_{\mathbf{x}_0\sim p_0} \mathbb{E}_{\mathbf{x}_t\sim p_{0t}(\mathbf{x}_t|\mathbf{x}_0)} \left[\parallel \mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t \mid \mathbf{x}_0) \parallel_2^2 \right] \right\} \tag{1}$$

Once $\mathbf{s}_\theta$ is obtained, the reverse SDE can be simulated using numerical methods such as Euler-Maruyama (Kloeden and Platen, 1992) to sample new data.

Song et al. (2021) show that the DDPM perturbation kernels converge to the Variance-Preserving (VP) SDE: $\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}$, with noise schedule $\beta(t) = \beta_{min} + t(\beta_{max} - \beta_{min})$, when time is continuous. In this study, we use the VP SDE and its corresponding reverse SDE to model the forward and reverse processes, respectively. The linearity of the drift and diffusion coefficients causes the transition kernel $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$ to become Gaussian, resulting in a standard Gaussian prior $\mathbf{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$. By reparameterization of $\mathbf{x}_t \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$, we obtain the simplified objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(\tau,1), \mathbf{x}_0\sim p_0, \epsilon\sim\mathcal{N}(0,\boldsymbol{I})} \left[\lambda_t \left\| \mathbf{s}_\theta(\mathbf{x}_t, t) + \frac{\epsilon}{\sigma_t} \right\|_2^2 \right] \tag{2}$$

3. Latent Diffusion Models

Latent diffusion models (LDMs, Rombach et al., 2022) compress the original images to a latent space of lower dimension, in which the diffusion process is then carried out. This is more computationally efficient and allows the model to train on the most important semantic bits of the data. An LDM consists of an encoder $\mathcal{E}$, which encodes the data $\mathbf{x}$ into a latent representation $\mathbf{z} = \mathcal{E}(\mathbf{x})$, and a decoder $\mathcal{D}$, that uses the latent to reconstruct the data $\hat{\mathbf{x}} = \mathcal{D}(\mathbf{z}) = \mathcal{D}(\mathcal{E}(\mathbf{x}))$.

In this study, we use a β-VAE (Higgins et al., 2017) with $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ and $q_{\mathcal{E}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu_{\mathcal{E}}, \sigma_{\mathcal{E}}^2)$, trained with the Evidence Lower Bound (ELBO), with KL regularization weighted by $\beta = 10^{-6}$ to ensure high-fidelity reconstructions. Subsequently, the diffusion model is trained on latent representations $\mathbf{z} = \mathcal{E}(\mathbf{x})$ using the objective in Equation 2, with $\mathbf{z}$ replacing $\mathbf{x}$.

4. Missing Data Framework

Given dataset $\boldsymbol{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ with p features, we define $i \in \{1, \ldots, n\}$ as the index of the data point $\mathbf{x}_i$, and $j \in \{1, \ldots, p\}$ the index of the feature $\mathbf{x}_{ij}$. In the case of MNIST, $\mathbf{x}_i$ is a flattened

image, and $\mathbf{x}_{ij}$ is a pixel in image $\mathbf{x}_i$. When working with missing data, we split each data point into missing and observed features $\mathbf{x}_i = \{\mathbf{x}_i^{\text{obs}}, \mathbf{x}_i^{\text{miss}}\}$. The missing features of each data point $\mathbf{x}_i$ are defined according to a binary mask vector $\mathbf{m}_i \in \{0,1\}^p$, such that $\mathbf{m}_{ij} = 1$, if $\mathbf{x}_{ij}$ is observed and $\mathbf{m}_{ij} = 0$, if it is missing. In this study, we assume that the data is *missing-completely-at-random*.

4.1. Training with Missing Data

For the diffusion modelling in pixel space, we employ two recent alternatives. First, we adopt the MissDiff approach (Ouyang et al., 2023): after zero-imputing the data, the loss is only computed on observed dimensions by factoring out the missing dimensions and normalizing by the number of observed dimensions:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(\tau,1),\mathbf{x}_0\sim p_0,\epsilon\sim\mathcal{N}(0,\boldsymbol{I}),\mathbf{m}}\left[\lambda_t \frac{||\mathbf{m}\odot(\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p_{0t}(\mathbf{x}_t|\mathbf{x}_0))||^2}{\sum_j \mathbf{m}_j}\right] \tag{3}$$

Second, we use DiffPuter (Zhang et al., 2025), an EM-based iterative algorithm that imputes the missing values in the E-step and retrains the diffusion model in the M-step.

For our LDMiss, we follow Mattei and Frellsen (2019) and zero-impute missing entries. The autoencoder is then trained using a β-VAE ELBO, where the reconstruction term is evaluated only on the observed data and the KL divergence term is weighted by β_{KL}. The diffusion process in latent space has the convenient property that missing dimensions factor out under the conditional independence assumption of our diagonal Gaussian encoder (derivation in appendix B). We therefore simply use the latent version of the objective given in Equation (2) during latent diffusion.

5. Self-Guided Missing Data Imputation

Song et al. (2021) perform imputation by sampling noisy observations from the known forward process $p_t(\mathbf{x}_t^{\text{miss}}|\mathbf{x}_0^{\text{obs}}) \approx p_t(\mathbf{x}_t^{\text{miss}}|\hat{\mathbf{x}}_t^{\text{obs}})$, where $\hat{\mathbf{x}}_t^{\text{obs}}$ is a random sample from $p_t(\mathbf{x}_t^{\text{obs}}|\mathbf{x}_0^{\text{obs}})$. At each iteration, they replace the observed dimensions with their noisy counterparts and update the missing dimensions using the score network. We refer to this as the *replacement* method. However, this method is not viable for the LDMiss, because we have no notion of missing dimensions in latent space.

An alternative for imputing missing values with unconditional diffusion models is to condition the generative process on the observed dimensions. Specifically, we replace the score in Equation (2) with the conditional score $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0^{\text{obs}})$. By Bayes' rule, this conditional score decomposes into the unconditional score and a guidance term:

$$\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_0^{\text{obs}}) = \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_0^{\text{obs}}|\mathbf{x}_t) \tag{4}$$

However, since $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_0^{\text{obs}}|\mathbf{x}_t)$ is intractable, we approximate it using Tweedie's formula (Efron, 2011) to obtain the posterior mean estimate $\hat{\mathbf{x}}_0^{\text{obs}}(\mathbf{x}_t)$. Given that our model is Gaussian, the guidance term can be expressed as

$$\log p_t(\mathbf{x}_0^{\text{obs}}|\hat{\mathbf{x}}_0^{\text{obs}}(\mathbf{x}_t)) \propto -\frac{1}{2\sigma^2}\|\mathbf{x}_0^{\text{obs}} - \hat{\mathbf{x}}_0^{\text{obs}}(\mathbf{x}_t)\|^2, \tag{5}$$

where we use a standard Gaussian prior $\mathbf{x}_T \sim \mathcal{N}(0,\boldsymbol{I})$. For the LDM, we decode the latent posterior mean estimate and compute gradients in pixel-space:

$$\log p_t(\mathbf{x}_0^{\text{obs}}|\hat{\mathbf{z}}_0^{\text{obs}}(\mathbf{z}_t)) \propto -\frac{1}{2\sigma^2}\|\mathbf{x}_0^{\text{obs}} - \mathcal{D}(\hat{\mathbf{z}}_0^{\text{obs}}(\mathbf{z}_t))\|^2, \tag{6}$$

with a standard Gaussian prior $\mathbf{z}_T \sim \mathcal{N}(0,\boldsymbol{I})$. We refer to this as the self-guidance method, because the model uses its own predictive capabilities to guide the sampling process. Some approaches use a weight or a schedule on the guidance term (Dhariwal and Nichol, 2021). However, we obtain the best results when scaling the guidance term to have equal magnitude to that of the unconditional score in Equation 4. This ensures equal strength guidance at all time steps.

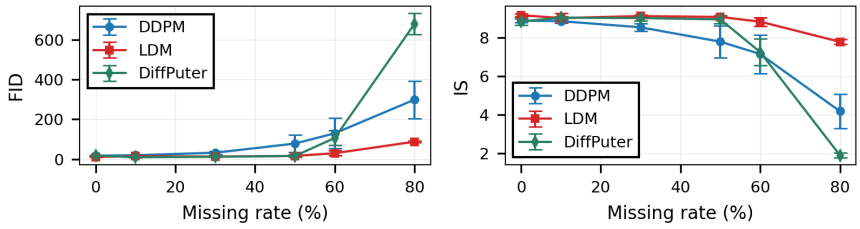

Figure 1. Sample quality (FID, IS) versus training missing rate. Metrics averaged over three seeds.

6. Experiments

In this section, we report experiments on the MNIST dataset (Deng, 2012). For simplicity, we refer to the score-based DDPM described in section 2 as DDPM. The full experimental setup is reported in appendix A, and the code for reproducing our experiments is accessible at this location.

6.1. Sample Quality

Figure 1 compares the generative capabilities of each model across training missing rates. The LDMiss retains better FID and IS than DDPM across all missing rates, demonstrating greater robustness to incomplete training data. Notably, the sample quality of LDMiss and Diffputer remains largely stable up to 50% missing data, whereas DDPM degrades continuously. In high-missingness regimes, LDMiss remains more stable than the alternatives. The LDMiss also exhibits lower variance across seeds, suggesting more consistent performance.

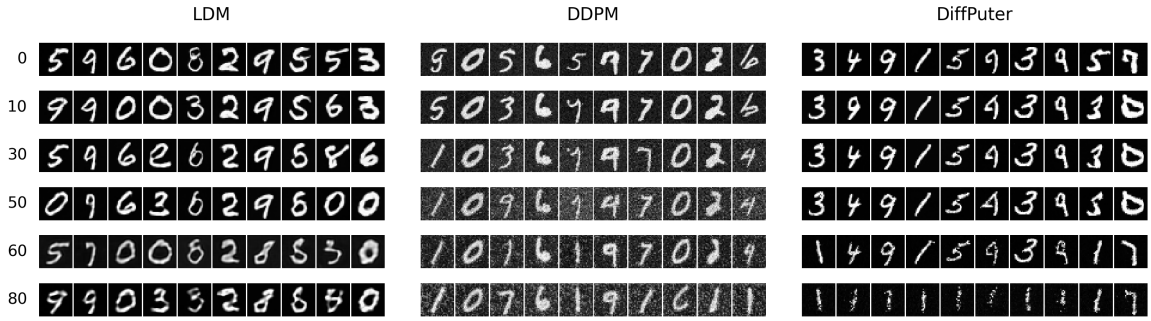

Figure 2. Generated samples across training missing rates.

Figure 2 provides a qualitative comparison. The LDMiss remains coherent up to 50% missingness and blurs only at higher rates, while DDPM becomes grainy beyond 10% because its pixel-space score network directly processes many zero-imputed values. In contrast, LDMiss diffuses in compressed latent features, reducing zero-imputation artifacts. DiffPuter is sharp at low to moderate missingness, but in the 60–80% regime its diversity and fidelity drop.

6.2. Imputation Quality

Figure 3 compares imputation performance across several methods: self-guidance for LDMiss and DDPM, DDPM replacement, and autoencoder reconstruction. All methods are evaluated at a 50% test missing rate. The LDMiss remains robust, maintaining strong imputation performance even when trained with 80% missing data. DDPM degrades as training missingness increases; within DDPM, guidance outperforms replacement at low missing rates but deteriorates faster at high missingness because it relies fully on a degraded score network. DiffPuter, designed for tabular

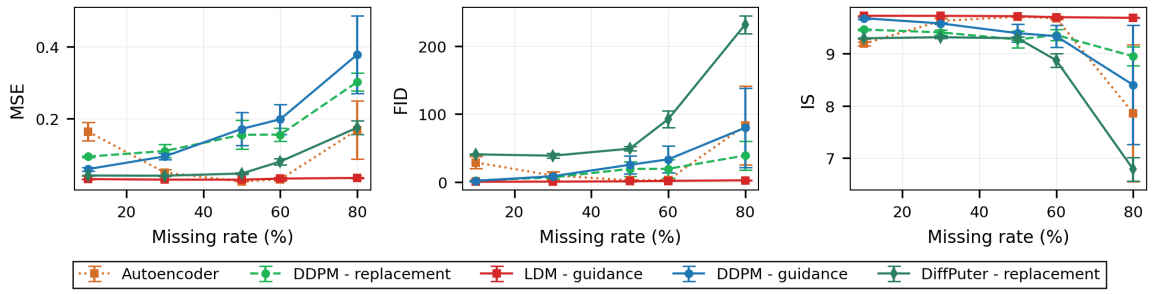

Figure 3. Imputation metrics (MSE, FID, IS) on 10,000 samples at 50% test missing rate versus training missing rate, averaged over three seeds.

data imputation, is competitive in terms of MSE at low to moderate training missing rates, but at 80% missingness its reconstructions lose detail and become less consistent than LDMiss. Longer EM routines could improve DiffPuter at high missingness, albeit with a substantial increase in computational cost. The autoencoder alone reaches MSE comparable to LDM guidance around 50–60% training missing rates, but its fixed reconstruction mapping generalizes poorly outside that regime. Overall, LDMiss provides the most reliable imputation quality across all missing rates.

Figure 4 confirms these findings qualitatively. The LDMiss imputations remain accurate across all missing rates, while DDPM imputations exhibit the same graininess observed in sample generation.

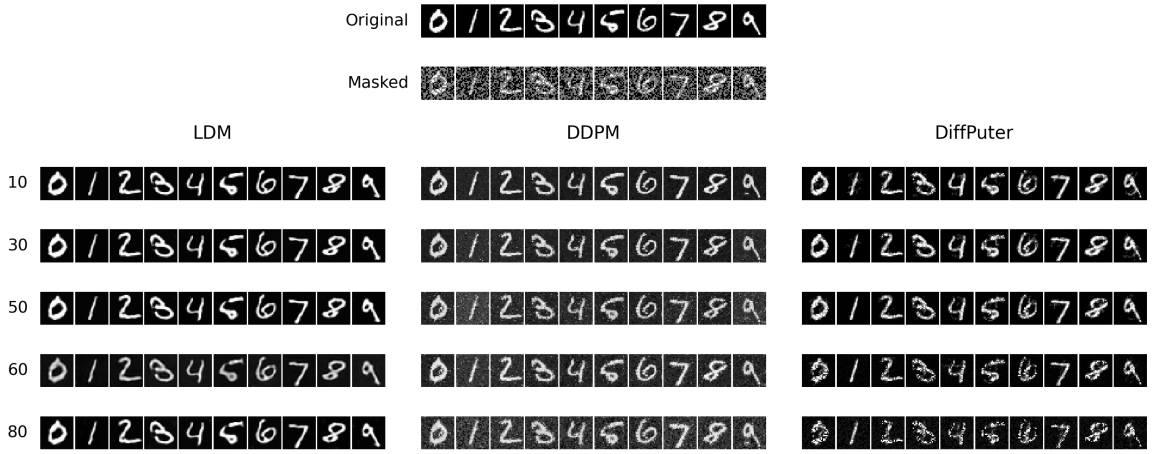

Figure 4. Imputed digits at 50% test missing rate across training missing rates. Left: LDM guidance. Center: DDPM replacement. Right: DiffPuter Replacement.

7. Conclusion

We compared pixel-space and latent-space score-based diffusion models trained on incomplete data and found that the LDMiss is consistently more robust to training missingness, preserving sample quality up to 50% missing data and outperforming DDPM on imputation across all training missing rates. A likely reason is that DDPM operates directly in pixel space, where zero-imputed values degrade score estimates, while latent diffusion filters many of these artifacts through semantic compression before denoising. This study is limited to MNIST under MCAR with zero-imputation, and architectural differences may also contribute to the observed gap. Future work should test more complex datasets, MAR/MNAR mechanisms, and alternative imputation functions to assess how well these advantages scale.

Acknowledgements

This research was supported by the Villum Foundation through the Synergy project number 50091, by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (MLLS, grant no. NNF20OC0062606), and by the Independent Research Fund Denmark (grant no. 5334-00122B and 5334-00076B). Jes Frellsen was further supported by funding from the Reinholdt W. Jorck og Hustrus Fond. Ignacio Peis acknowledges support by Danish Data Science Academy, which is funded by the Novo Nordisk Foundation (NNF21SA0069429). We thank Pierre-Alexandre Mattei, Federico Bergamin and Hugo Senetaire for valuable and insightful discussions.

References

Gary J Chang, Yi Mu, et al. Prevalence of missing data in the National Cancer Database and association with overall survival. *JAMA Network Open*, 4(3):e211793, 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181. URL https://doi.org/10.1198/jasa.2011.tm11181. PMID: 22505788.

Josh Givens, Song Liu, and Henry Reeve. Score matching with missing data. In *Forty-second International Conference on Machine Learning*, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer Berlin, Heidelberg, 1992. URL https://doi-org.proxy.findit.cvt.dk/10.1007/978-3-662-12616-5.

Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., 2nd edition, 2002.

Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data, 2019. URL https://arxiv.org/abs/1812.02633.

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

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.

Ignacio Peis, Chao Ma, and José Miguel Hernández-Lobato. Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo. *Advances in Neural Information Processing Systems*, 35:35839–35851, 2022.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL https://arxiv.org/abs/1606.03498.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. *eGEMs*, 1(3), 2013.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets, 2018. URL https://arxiv.org/abs/1806.02920.

Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. Diffputer: Empowering diffusion models for missing data imputation, 2025. URL https://arxiv.org/abs/2405.20690.

Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data, 2023. URL https://arxiv.org/abs/2210.17128.

Appendix

A. Experimental Setup

A.1. Architecture

A.1.1. Score Network

We use a DDPM++ architecture based on Ho et al. (2020) and Song et al. (2021). The network is a U-Net with sinusoidal timestep embeddings, group normalization, and rescaled skip connections. We trained on MNIST with the configuration reported in table 1.

Table 1. Model Architectures.

Hyperparameter	Pixel-space	Latent-space
Base channels	48	64
Channel multipliers	1, 2, 4	1, 2
Residual blocks per resolution	3	2
Attention resolutions	None	7
Dropout	~ 0.12	~ 0.08
Group normalization groups	16	16

For DiffPuter, we used the architecture specified by Zhang et al. (2025) in their Appendix D.4.

A.1.2. Autoencoder

For latent diffusion, we train a KL-regularized VAE following Rombach et al. (2022). The encoder and decoder use convolutional residual blocks with channel multipliers [1, 2, 4], downsampling 28×28 images to 7×7 latent representations with 2 channels: $1 \times 28 \times 28 \to 2 \times 7 \times 7 = 784 \to 98$, resulting in a compression factor of $785/98 = 8$. We use a very small KL weight ($\beta_{\text{KL}} = 10^{-6}$) to ensure high reconstruction quality. The autoencoder is trained with MSE reconstruction loss on data with MCAR missing mechanism.

A.1.3. Classifier

For evaluation, we train a CNN classifier on MNIST with two convolutional layers (32 and 64 channels, 3×3 kernels) followed by max pooling, and two fully connected layers (64 hidden units). The classifier achieves 99.2% test accuracy.

A.2. Training

All models are trained with Adam optimizer using default hyperparameters. We use batch size 256 and train for 50 epochs. Learning rates were determined via hyperparameter sweep to be $\eta_{\text{DDPM}} \approx 1.96 \cdot 10^{-5}$ and $\eta_{\text{LDM}} \approx 9.05 \cdot 10^{-5}$. Each model has been trained for three different seeds: 42, 43, and 44.

A.2.1. Noise Schedule

We use the VP SDE with linear β-schedule:

$$\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min}), \qquad t \in [\tau, 1] \tag{7}$$

with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20.0$, following Song et al. (2021). We discretize with $T = 1000$ timesteps and use $\tau = 10^{-3}$ as the minimum time for numerical stability.

A.2.2. Diffusion in Latent Space

When traing diffusion models in latents space, we scale the latent space according to Rombach et al. (2022).

A.2.3. Missing Data

We train models on MNIST with MCAR missing mechanism at rates {0.0, 0.1, 0.3, 0.5, 0.6, 0.8}. Missing pixels are set to zero during training (zero imputation).

A.3. Generation Quality Metrics

We evaluate sample quality using IS (Salimans et al., 2016) and FID (Heusel et al., 2018). IS measures both diversity and clarity of generated images, reflecting how easily they can be classified as distinct digits, with 10 being the highest score. FID measures the similarity between the distributions of generated and real images, capturing how much generated samples resemble the MNIST test set. For imputation, we use FID and IS to assess the quality of imputed images, and additionally MSE between imputed and original images to measure reconstruction accuracy.

A.4. Compute

All DDPM experiments have been performed on a NVIDIA Titan V GPU, and all LDM experiments on a NVIDIA GeForce RTX 4070 Ti SUPER GPU.

B. Missing Dimensions Factorization

$$
\begin{aligned}
p(\mathbf{x}^{\text{obs}}|\mathbf{z}) &= \int p(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}}|\mathbf{z})\, \mathrm{d}\mathbf{x}^{\text{miss}} && (8)\\
&= \int \prod_i p(\mathbf{x}_i^{\text{obs}}, \mathbf{x}_i^{\text{miss}}|\mathbf{z})\, \mathrm{d}\mathbf{x}^{\text{miss}} && (9)\\
&= \prod_i p(\mathbf{x}_i^{\text{obs}}|\mathbf{z}) \cdot \int \prod_i p(\mathbf{x}_i^{\text{miss}}|\mathbf{z})\, \mathrm{d}\mathbf{x}^{\text{miss}} && (10)\\
&= \prod_i p(\mathbf{x}_i^{\text{obs}}|\mathbf{z}) \cdot \prod_i \underbrace{\int p(\mathbf{x}_i^{\text{miss}}|\mathbf{z})\, \mathrm{d}\mathbf{x}_i^{\text{miss}}}_{=1} && (11)\\
&= \prod_i p(\mathbf{x}_i^{\text{obs}}|\mathbf{z}) && (12)\\
&= p(\mathbf{x}^{\text{obs}}|\mathbf{z}) && (13)
\end{aligned}
$$

In Equation (10), we assume conditional independence between observed and missing dimensions given $\mathbf{z}$, which holds because we assume a diagonal Gaussian distribution for our encoder $\mathcal{E}$.