DOOBIMP: DIFFUSION IMPUTATION WITHOUT ARTIFICIAL MASKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Missing data are ubiquitous in real-world datasets, making reliable imputation essential for prediction and downstream analysis. Diffusion-based imputation methods typically learn conditional scores by artificially masking observed entries during training. This couples learning to a user-specified mask distribution, which may not match test-time missingness and can degrade imputation performance. We introduce DOOBIMP, a diffusion imputation framework that avoids artificial masking by decoupling test-time imputation from learning an unconditional diffusion backbone. Its two complementary components address conditioning and learning. DOOBCOND uses Doob's h-transform to characterize the ideal conditional reverse diffusion process and derives a practical approximation for conditioning on observed entries. DOOBFIT learns the unconditional backbone from incomplete training data by alternating between imputing missing entries with DOOBCOND and refitting the unconditional score. Across nine tabular datasets under three missingness mechanisms, DOOBIMP achieves the best overall performance over 17 baselines, attaining an average MAE rank of 1.19 and the highest aggregate categorical accuracy. Experiments on four time-series datasets further demonstrate robust performance under both pointwise and structured missingness. The resulting tabular imputations also support strong downstream classification performance. These results show that decoupling unconditional score learning from conditioning provides an effective and broadly applicable alternative to mask-based diffusion imputation.

1 INTRODUCTION

Missing data are ubiquitous in real-world datasets and can substantially degrade downstream analysis, prediction, and decision-making (Little & Rubin, 2019; Yoon et al., 2018; Cao et al., 2018). Imputation seeks to recover missing entries from observed information. Deep generative models offer a flexible approach to this problem by learning the underlying data distribution and generating missing entries conditional on the observed ones (Miao et al., 2021; Fortuin et al., 2020). Among them, diffusion models have emerged as expressive models of complex, high-dimensional distributions (Song et al., 2021b), and have achieved strong empirical performance for imputation across tabular and time-series data (Tashiro et al., 2021; Lopez Alcaraz & Strodthoff, 2023; Ouyang et al., 2023).

Because the ground-truth values of genuinely missing entries are unavailable, imputation models cannot be directly supervised at those locations. Many methods therefore create surrogate training targets by artificially masking observed entries and learning to reconstruct them (Choi et al., 2023; Du et al., 2024a; Tashiro et al., 2021). Diffusion-based imputers commonly follow this workflow by drawing artificial masks from a prescribed training distribution (Tashiro et al., 2021; Lopez Alcaraz & Strodthoff, 2023; Ye & Chen, 2025).

Nevertheless, the training-time artificial-mask distribution can differ from the test-time missingness distribution, creating *missingness distribution shift* (Qian et al., 2024; Ye & Chen, 2025). Changing which entries are observed changes the information available for imputation. A conditional score trained under one artificial-mask distribution can therefore perform poorly under a different test-time distribution, even with low training risk. Section 4.1 characterizes this effect theoretically and empirically. This raises a natural question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

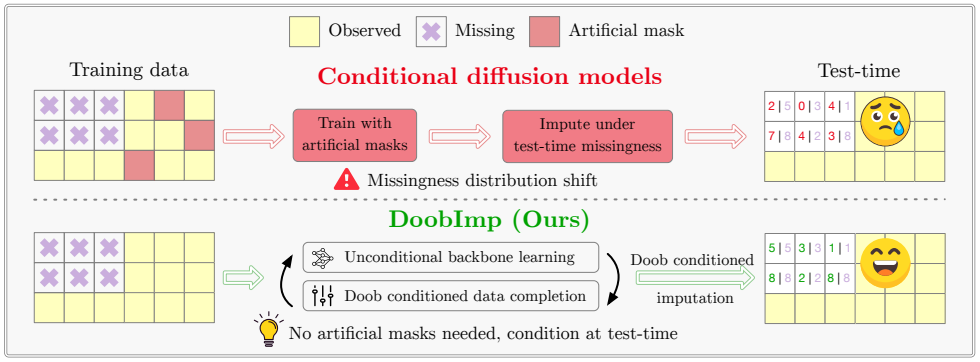


Figure 1: Overview of DOOBIMP. DOOBCOND performs test-time Doob conditioning and supplies the stochastic completions that DOOBFIT uses to iteratively learn the unconditional backbone from incomplete data.

Can we learn a diffusion imputer without artificial masking during training?

We answer this question with DOOBIMP, a framework that decouples test-time imputation from learning an unconditional diffusion backbone, as summarized in Figure 1. At its core, DOOBCOND turns unconditional backbones into imputers through inference-time conditioning, without updating their parameters. It uses Doob's h-transform (Rogers & Williams, 2000; Zhu et al., 2026) to characterize the conditional reverse process and derives a practical sampler for conditioning on observed entries. This procedure can be applied to an existing backbone. To learn the backbone from incomplete data without artificial masking, the complementary DOOBFIT procedure alternates between filling missing entries with DOOBCOND and refitting on the resulting completions (Zhang et al., 2025).

We summarize our contributions as follows.

- We introduce DOOBIMP, a diffusion imputation framework without artificial masking. Its core component, DOOBCOND, provides reusable inference-time conditioning for unconditional backbones through Doob's h-transform. The complementary DOOBFIT procedure uses DOOBCOND to learn the backbone from incomplete data.
- We formalize missingness distribution shift induced by training-time artificial masking and show how it can amplify imputation error despite low training conditional-score risk.
- We demonstrate strong tabular imputation performance, with DOOBIMP achieving an average MAE rank of 1.19 among 18 methods and strong downstream classification results. Our time-series experiments further show strong results under pointwise and structured missingness.

2 RELATED WORK

Diffusion models for data imputation. Tashiro et al. (2021) formulate probabilistic imputation as missing-data conditional score estimation. Subsequent methods incorporate state-space or spatiotemporal structure (Lopez Alcaraz & Strodthoff, 2023; Liu et al., 2023), enforce consistency with observed values (Wang et al., 2023; Zhou et al., 2024), and target partial blackouts or long-range dependencies (Islam et al., 2025; Xiao et al., 2025). These methods typically amortize conditional score learning over an artificial-mask distribution, so their performance can depend on how this distribution aligns with test-time missingness (Qian et al., 2024; Ye & Chen, 2025). A complementary line learns joint data distributions directly from incomplete or corrupted observations (Ouyang et al., 2023; Givens et al., 2025; Daras et al., 2023; Chen et al., 2024). Diffusion-based EM methods, including DiffPuter, instead alternate conditional completion and unconditional score learning (Zhang et al., 2025; Bai et al., 2024; Rozet et al., 2024).

Doob's h-transform as inference-time conditioning. Inference-time conditional diffusion is often implemented through projection or likelihood guidance, as in RePaint and diffusion posterior sampling (Lugmayr et al., 2022; Chung et al., 2023). Doob's h-transform instead gives a proba-

bilistic construction for conditioning Markov dynamics on terminal evidence (Rogers & Williams, 2000). Recent work uses this construction for learned and training-free conditioning in diffusion adaptation, editing, and constrained generation (Denker et al., 2024; Nguyen et al., 2025; Zhu et al., 2026; Chang et al., 2026; Guo et al., 2026; Wang et al., 2026). DOOBIMP brings this construction to imputation: DOOBCOND conditions an unconditional backbone on each realized incomplete sample, while DOOBFIT uses these conditional completions to learn the unconditional score from incomplete observations.

3 PRELIMINARIES

3.1 DIFFUSION MODEL BASICS

Let $X_0 \sim P_{\text{data}}$ with $X_0 \in \mathbb{R}^D$, where D denotes the number of coordinates, and let I_k denote the $k \times k$ identity matrix. Diffusion models learn to sample from P_{data} by estimating scores along coupled forward and reverse-time processes (Song et al., 2021b). We consider a linear stochastic differential equation (SDE) with prescribed scalar schedules $f(t)$ and $g(t) > 0$. The forward process is governed by

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}W_t, \qquad t \in [0,T], \quad X_0 \sim P_{\text{data}}, \tag{1}$$

where T is a terminal time and W_t is a Wiener process. Let p_t denote the marginal density of X_t, so p_0 is the density of P_{data}. The forward transition can be written as $X_t = \alpha_t X_0 + \sigma_t Z$, where $Z \sim \mathcal{N}(0, I_D)$ and α_t, σ_t are the scalar coefficients induced by f and g.

The reverse process generates samples by evolving the terminal marginal back to P_{data}. It satisfies

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

where $\overline{W}_t$ is a reverse-time Wiener process. This process is initialized from the forward terminal marginal and run from T to 0. In practice, an unconditional score network $s_\theta(x,t)$ approximates $\nabla_x \log p_t(x)$. This network serves as the unconditional backbone and models all variables jointly. With a tractable Gaussian terminal prior, it defines the learned reverse-time SDE

$$\mathrm{d}\widetilde{X}_t = \left[f(t)\widetilde{X}_t - g(t)^2 s_\theta(\widetilde{X}_t, t)\right]\mathrm{d}t + g(t)\mathrm{d}\overline{W}_t, \tag{3}$$

where $\widetilde{X}_T$ is drawn from this prior. We denote the path law of this learned reverse process by $\mathbb{P}_\theta$. Its endpoint law $P_{\theta,0}$ approximates P_{data}. Following Song et al. (2021b), the score network s_θ is trained by minimizing the denoising score matching loss

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\, X_0\sim P_{\text{data}}, X_t\sim p_{t|0}(\cdot|X_0)}\left[\lambda(t)\left\|s_\theta(X_t,t) - \nabla_{X_t}\log p_{t|0}(X_t \mid X_0)\right\|_2^2\right]. \tag{4}$$

Here $p_{t|0}(X_t \mid X_0) = \mathcal{N}(\alpha_t X_0, \sigma_t^2 I_D)$, and $\lambda(t) > 0$ is a weighting function. The Gaussian transition gives the target $\nabla_{X_t}\log p_{t|0}(X_t \mid X_0) = -(X_t - \alpha_t X_0)/\sigma_t^2$. Sampling then proceeds by discretizing (3) on a grid $T = t_1 > \cdots > t_{L+1} = 0$.

3.2 MISSING DATA IMPUTATION

In missing-data settings, only a subset of the coordinates of X_0 is observed (Little & Rubin, 2019). We model the observation pattern by a random missingness indicator $M \in \{0,1\}^D$, where $M_d = 1$ means that coordinate d of X_0 is observed, and $M_d = 0$ means that it is missing. For a realized pattern m, let $X_0^{\text{obs}} = X_{0,\{d:m_d=1\}}$ and $X_0^{\text{mis}} = X_{0,\{d:m_d=0\}}$, with dimensions $D_{\text{obs}} = \sum_{d=1}^D m_d$ and $D_{\text{mis}} = D - D_{\text{obs}}$, respectively. The joint law factorizes as $p_{X_0,M}(x_0,m) = p_0(x_0)q(m \mid x_0)$, where $q(m \mid x_0)$ is the missingness distribution. Under missing completely at random (MCAR), $q(m \mid x_0) = q(m)$; under missing at random (MAR), $q(m \mid x_0) = q(m \mid x_0^{\text{obs}})$; and under missing not at random (MNAR), $q(m \mid x_0)$ may depend on x_0^{mis} even conditional on x_0^{obs} (Rubin, 1976; Little & Rubin, 2019).

Given a fixed missingness pattern $M = m$ and the corresponding observed value x_0^{obs}, the imputation task is to generate samples from

$$P\left(X_0^{\text{mis}} \mid X_0^{\text{obs}} = x_0^{\text{obs}}, M = m\right). \tag{5}$$

4 METHODOLOGY

We first characterize the sensitivity of artificial-mask training to missingness distribution shift. We then introduce DOOBIMP, which separates unconditional score learning from sample-specific conditioning, thereby avoiding this source of shift.

4.1 ARTIFICIAL-MASK TRAINING IS SENSITIVE TO MISSINGNESS DISTRIBUTION SHIFT

Diffusion imputation is commonly formulated by training a conditional diffusion model (Tashiro et al., 2021), the conditional counterpart of the diffusion model introduced in Section 3.1. The model learns $s_\phi^{\mathrm{cond}}(X_t^{\mathrm{mis}}, X_0^{\mathrm{obs}}, M, t)$ to approximate the missing-data conditional score $\nabla_{X_t^{\mathrm{mis}}} \log p_t(X_t^{\mathrm{mis}} \mid X_0^{\mathrm{obs}}, M)$ and then runs reverse diffusion over the missing coordinates conditioned on (X_0^{obs}, M) (Lopez Alcaraz & Strodthoff, 2023; Liu et al., 2023). Appendix A gives the corresponding population objective.

Because truly missing entries provide no supervised targets, conditional diffusion methods typically hide additional observed entries during training, inducing an artificial-mask distribution q_{art} over hidden target patterns. Its mismatch with the test-time distribution of true missingness patterns defines *missingness distribution shift* and materially affects imputation accuracy (Ye & Chen, 2025).

Controlled simulation. To demonstrate this effect, we vary the test-time missingness distribution q_λ along a controlled path indexed by $\lambda \in [0, 1]$. We compare two models. *Fixed* is trained once with $q_{\mathrm{art}} = q_0$ and reused as the test distribution changes, so larger λ produces a larger training–test shift. In contrast, *Matched* is retrained with $q_{\mathrm{art}} = q_\lambda$ for each λ, avoiding this mismatch. Figure 2 shows that the excess risk of *Fixed* increases with the shift, whereas *Matched* remains stable. Here, excess risk measures error relative to the population conditional score; Appendix C.5 gives its formal definition and the experimental details.

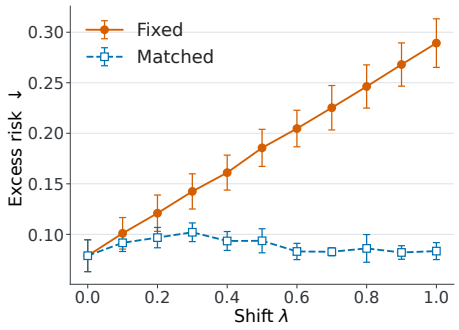


Figure 2: Conditional-score risk as missingness shift increases: *Fixed* degrades, while *Matched* remains stable.

Existing methods broaden artificial-mask coverage by varying target patterns or combining random, historical, scattered, and grouped masks (Liu et al., 2019; Choi et al., 2023; Tashiro et al., 2021; Ye & Chen, 2025). Although these strategies improve coverage, they remain sensitive to mismatch between the prescribed q_{art} and test-time missingness, as captured by the following proposition.

Proposition 1 (Sampling under missingness distribution shift (informal))**.** *Missingness distribution shift can amplify imputation error when masks that are rare during training become common at test time, even when q_{art} has full support and the training conditional-score risk is low.*

Appendix A.2 gives the formal statement and proof. Together, the proposition and controlled simulation connect missingness distribution shift to both score risk and posterior sampling. This motivates the separation of unconditional backbone learning and sample-specific conditioning in DOOBIMP.

DOOBIMP overview. DOOBFIT is the outer training procedure: it initializes an unconditional score from observed entries, repeatedly calls DOOBCOND to complete the training data, and refits the score on the resulting pseudo-complete data. At imputation time, DOOBCOND conditions the final score on each realized incomplete sample and returns its missing entries. Algorithms 1 and 2 summarize these operations; Sections 4.2 and 4.3 provide their derivation and implementation details.

4.2 DOOBCOND: CONDITIONAL IMPUTATION VIA DOOB'S h-TRANSFORM

Given an unconditional score s_θ as in Section 3.1, DOOBCOND adds sample-specific conditioning to target the distribution in (5). The backbone may be learned with DOOBFIT or trained on fully observed data; its parameters remain fixed during conditioning. We first derive the conditioning term through Doob's h-transform and then describe the approximation used in Algorithm 1.

Algorithm 1 DOOBCOND

Require: Unconditional score s_θ, observation (x_0^{obs}, m), grid $\{t_\ell\}_{\ell=1}^{L+1}$, hyperparameters η, τ, W, J, variance scales $\{v_i\}_{i=1}^{D}$, reverse solver

Initialize the missing block from the tractable Gaussian terminal prior and the observed block by (7);
for $\ell = 1, \dots, L$ **do**
  Compute $\widehat{x}_{\theta,0}$ and Λ_{t_ℓ} by (9)
  if $1 \le W \le \ell$ **then**
    Compute $\widehat{x}_{\theta,0}^{h,\mathrm{mis}}$ by (10)–(11); apply (12) J times and average when $J > 0$
  end if
  Apply the reverse solver to the missing block; refresh observed coordinates by (7);
end for
Restore the observed block to x_0^{obs}; **return** x_0^{mis}

Algorithm 2 DOOBFIT

Require: Incomplete training data $\{(X_{0,i}^{\mathrm{obs}}, M_i)\}_{i=1}^{n}$, stage budget K, completion budget S

Initialize $\theta^{(1)}$ by minimizing $\mathcal{L}^{(1)}$ in (13)
for $r = 2, \dots, K$ **do**
  for $i = 1, \dots, n$ **do**
    Independently draw $X_{0,i}^{\mathrm{mis},(\mathrm{r},\mathrm{j})} \leftarrow \mathrm{DOOBCOND}(s_{\theta^{(r-1)}}, X_{0,i}^{\mathrm{obs}}, M_i)$, $j = 1, \dots, S$
    Merge their coordinatewise average with $X_{0,i}^{\mathrm{obs}}$ according to M_i to form $X_{0,i}^{\mathrm{pseudo},(\mathrm{r})}$
  end for
  Update $\theta^{(r)}$ by minimizing (4) over $\{X_{0,i}^{\mathrm{pseudo},(\mathrm{r})}\}_{i=1}^{n}$
end for
return Unconditional score $s_{\theta^{(K)}}$

4.2.1 DOOB'S h-TRANSFORM FOR IMPUTATION

At imputation time, we observe x_0^{obs} under a realized pattern m. Under MCAR or MAR, we have $p(x_0^{\mathrm{mis}} \mid x_0^{\mathrm{obs}}, m) = p_0(x_0^{\mathrm{mis}} \mid x_0^{\mathrm{obs}})$. Thus, m only determines the coordinate partition, while the unconditional backbone determines the conditional law of the missing values. Following (5), the corresponding target under the current model is $P_{\theta,0}(\widetilde{X}_0^{\mathrm{mis}} \mid \widetilde{X}_0^{\mathrm{obs}} = x_0^{\mathrm{obs}})$. Because conditioning on this exact observed value induces a singular endpoint constraint at $t = 0$, we introduce a Gaussian smoothing scale. For $\eta > 0$, we define

$$w_\eta(\widetilde{x}_0) := \exp\left(-\frac{\|\widetilde{x}_0^{\mathrm{obs}} - x_0^{\mathrm{obs}}\|_2^2}{2\eta^2}\right), \quad P_{\theta,0}^{\eta}(\mathrm{d}\widetilde{x}_0) := \frac{w_\eta(\widetilde{x}_0) P_{\theta,0}(\mathrm{d}\widetilde{x}_0)}{\mathbb{E}_\theta[w_\eta(\widetilde{X}_0)]}.$$

Under the mild regularity conditions stated in Appendix B.1, the missing-coordinate marginal of $P_{\theta,0}^{\eta}$ converges as $\eta \to 0$ to $P_{\theta,0}(\widetilde{X}_0^{\mathrm{mis}} \mid \widetilde{X}_0^{\mathrm{obs}} = x_0^{\mathrm{obs}})$. Thus, $P_{\theta,0}^{\eta}$ is a smooth approximation to the imputation law implied by the learned unconditional backbone. We then define $h_\eta(x,t) := \mathbb{E}_\theta[w_\eta(\widetilde{X}_0) \mid \widetilde{X}_t = x]$ and the endpoint-tilted path measure $\mathbb{Q}_\eta$ by $\mathrm{d}\mathbb{Q}_\eta/\mathrm{d}\mathbb{P}_\theta = w_\eta(\widetilde{X}_0)/\mathbb{E}_\theta[w_\eta(\widetilde{X}_0)]$. Doob's h-transform gives the following result (Rogers & Williams, 2000).

Lemma 1. *Under standard regularity conditions, the reverse process $\widetilde{X}^h$ with path law $\mathbb{Q}_\eta$ satisfies*

$$\mathrm{d}\widetilde{X}_t^h = \left[f(t)\widetilde{X}_t^h - g(t)^2\left\{s_\theta(\widetilde{X}_t^h, t) + \nabla_{\widetilde{X}_t^h} \log h_\eta(\widetilde{X}_t^h, t)\right\}\right]\mathrm{d}t + g(t)\mathrm{d}\overline{W}_t. \tag{6}$$

Moreover, $\widetilde{X}_0^h \sim P_{\theta,0}^{\eta}$.

The proof is given in Appendix B.1. Lemma 1 reduces imputation to simulating the transformed reverse process in (6), whose only additional quantity beyond the unconditional score is $\nabla_x \log h_\eta(x,t)$. Under MNAR, the mask m can carry information about x_0^{mis}, so $p(x_0^{\mathrm{mis}} \mid x_0^{\mathrm{obs}}, m) = p_0(x_0^{\mathrm{mis}} \mid x_0^{\mathrm{obs}})$ need not hold. Exact conditioning on m would then require the h-function to account for the missingness mechanism. Without explicitly modeling the missingness mechanism, DOOBIMP uses the learned dependencies between observed and missing entries

through the conditioning term in (10) and achieves strong performance under MNAR (Section 5.1). Appendix B.4 discusses a possible mechanism-aware extension. We next develop a practical approximation to $\nabla_x \log h_\eta(x,t)$.

4.2.2 PRACTICAL DOOBCOND

In practice, each reverse step of DOOBCOND refreshes the observed coordinates, constructs a Tweedie estimate and diagonal posterior precision, evaluates the conditioning term through a vector–Jacobian product, and optionally applies same-noise-level refinement. Algorithm 1 summarizes the procedure.

For the observed coordinates, we refresh the state at each level using the forward transition

$$x_t^{\text{obs}} = \alpha_t x_0^{\text{obs}} + \sigma_t z_t^{\text{obs}}, \qquad z_t^{\text{obs}} \sim \mathcal{N}(0, I_{D_{\text{obs}}}), \tag{7}$$

following RePaint-style conditional sampling (Lugmayr et al., 2022). At level t_ℓ, merging this block with the current $x_{t_\ell}^{\text{mis}}$ according to m gives the state x_{t_ℓ} at which the quantities below are evaluated.

For the missing coordinates, consider the Gaussian reference case $P_{\theta,0} = P_{\text{data}} = \mathcal{N}(\mu, V)$. If $X_0 \mid X_t = x \sim \mathcal{N}(\mu_{0|t}(x), V_{0|t})$, Gaussian integration gives

$$\nabla_{x^{\text{mis}}} \log h_\eta(x,t) = \left[\partial \mu_{0|t}^{\text{obs}}(x)/\partial x^{\text{mis}}\right]^\top \left(V_{0|t}^{\text{obs,obs}} + \eta^2 I_{D_{\text{obs}}}\right)^{-1} \left(x_0^{\text{obs}} - \mu_{0|t}^{\text{obs}}(x)\right). \tag{8}$$

Appendix B.2 gives the derivation. This factorization combines the observed residual, the observed posterior precision, and the Jacobian that transfers it to the missing coordinates. It motivates our practical approximation: we replace the posterior mean with Tweedie's formula (Efron, 2011) and approximate the observed posterior precision by a diagonal matrix:

$$\widehat{x}_{\theta,0}(x,t) := (x + \sigma_t^2 s_\theta(x,t))/\alpha_t, \qquad (\Lambda_t)_{ii} := \left(\frac{v_i \sigma_t^2}{\alpha_t^2 v_i + \sigma_t^2} + \eta^2\right)^{-1}, \quad i : m_i = 1, \tag{9}$$

where v_i is a coordinatewise variance scale estimated from the training data, and η is the smoothing scale from h_η, which also stabilizes the approximate precision.

These substitutions define a plug-in $\widehat{h}_\eta$ whose missing-coordinate conditioning term is

$$\nabla_{x^{\text{mis}}} \log \widehat{h}_\eta(x,t) := \left[\partial \widehat{x}_{\theta,0}^{\text{obs}}(x,t)/\partial x^{\text{mis}}\right]^\top \Lambda_t \left(x_0^{\text{obs}} - \widehat{x}_{\theta,0}^{\text{obs}}(x,t)\right). \tag{10}$$

We evaluate this as a vector–Jacobian product through the frozen score network, keeping θ fixed while differentiating with respect to x. We retain η as the endpoint-smoothing scale and introduce separate controls for conditioning strength and duration: the temperature $\tau > 0$ controls the strength, and $W \in \{0, \dots, L\}$ specifies the active reverse levels. The resulting denoised estimate is

$$\widehat{x}_{\theta,0}^{h,\text{mis}}(x,t_\ell) := \widehat{x}_{\theta,0}^{\text{mis}}(x,t_\ell) + \mathbf{1}\{1 \le W \le \ell\} \frac{\sigma_{t_\ell}^2}{\alpha_{t_\ell} \tau^2} \nabla_{x^{\text{mis}}} \log \widehat{h}_\eta(x,t_\ell). \tag{11}$$

Same-noise-level refinement. Direct evaluation of h_η can require repeated reverse rollouts (Zhu et al., 2026), whereas (11) provides a single-pass conditioning approximation. To improve this approximation at active reverse levels, we apply $J \ge 0$ perturb-and-denoise steps at the same noise level while keeping the observed noisy coordinates fixed (Bengio et al., 2013; Wang et al., 2026). Starting from $\widehat{x}_0^{\text{mis},(0)} = \widehat{x}_{\theta,0}^{h,\text{mis}}(x_{t_\ell}, t_\ell)$, we iterate

$$x_{t_\ell}^{\text{mis},(j)} = \alpha_{t_\ell} \widehat{x}_0^{\text{mis},(j-1)} + \sigma_{t_\ell} \xi_j^{\text{mis}}, \quad \xi_j^{\text{mis}} \sim \mathcal{N}(0, I_{D_{\text{mis}}}); \quad \widehat{x}_0^{\text{mis},(j)} = \widehat{x}_{\theta,0}^{\text{mis}}(x_{t_\ell}^{(j)}, t_\ell), \tag{12}$$

for $j = 1, \dots, J$, where $x_{t_\ell}^{(j)}$ combines the perturbed missing block with the observed block. Conditioning is applied once before this loop; $J = 0$ skips refinement. At inactive levels, the reverse solver uses $(x_{t_\ell}^{\text{mis}}, \widehat{x}_{\theta,0}^{\text{mis}}(x_{t_\ell}, t_\ell))$. At active levels, it uses $(x_{t_\ell}^{\text{mis}}, \widehat{x}_{\theta,0}^{h,\text{mis}}(x_{t_\ell}, t_\ell))$ when $J = 0$, or $(x_{t_\ell}^{\text{mis},(J)}, J^{-1} \sum_{j=1}^{J} \widehat{x}_0^{\text{mis},(j)})$ when $J > 0$. The corresponding DDIM and EDM updates are given in Appendix B.3 (Song et al., 2021a; Karras et al., 2022).

Validation. Simulations in Appendix C show that DOOBCOND improves on the baseline without Doob conditioning across data dimensions and mitigates degradation under training–test missingness shift while reusing a single unconditional score.

4.3 DOOBFIT: LEARNING THE UNCONDITIONAL BACKBONE

DOOBFIT is summarized in Algorithm 2 and follows iterative diffusion methods (Zhang et al., 2025; Bai et al., 2024; Rozet et al., 2024; Hosseintabar et al., 2025). We index the trained score stages by $r = 1, \ldots, K$, where $r = 1$ is initialization and $r = 2, \ldots, K$ are subsequent refitting stages.

Following Ouyang et al. (2023), we zero-pad the missing entries as $X_0^{\mathrm{zf}} := M \odot X_0$ and $X_t^{\mathrm{zf}} := \alpha_t X_0^{\mathrm{zf}} + \sigma_t Z$, and train $\theta^{(1)}$ by minimizing

$$\mathcal{L}^{(1)}(\theta) = \mathbb{E}_{t, X_0, Z, M}\left[\lambda(t) \sum_{d=1}^{D} M_d \left| s_{\theta,d}(X_t^{\mathrm{zf}}, t) + (X_{t,d}^{\mathrm{zf}} - \alpha_t X_{0,d}^{\mathrm{zf}})/\sigma_t^2 \right|^2\right]. \tag{13}$$

This observed-entry objective initializes the unconditional backbone without artificial masking.

At each refitting stage $r = 2, \ldots, K$, DOOBCOND draws S independent completions for each of the n incomplete training instances under $s_{\theta^{(r-1)}}$. For $i = 1, \ldots, n$ and $j = 1, \ldots, S$, $X_{0,i}^{\mathrm{mis},(r,j)} \leftarrow \textsc{DoobCond}(s_{\theta^{(r-1)}}, X_{0,i}^{\mathrm{obs}}, M_i)$. We average these samples coordinatewise (Zhang et al., 2025): $\overline{X}_{0,i}^{\mathrm{mis},(r)} := \frac{1}{S}\sum_{j=1}^{S} X_{0,i}^{\mathrm{mis},(r,j)}$. Merging $X_{0,i}^{\mathrm{obs}}$ and $\overline{X}_{0,i}^{\mathrm{mis},(r)}$ in their corresponding coordinates according to M_i gives the pseudo-complete sample $X_{0,i}^{\mathrm{pseudo},(r)}$. DOOBFIT then refits the unconditional backbone $s_{\theta^{(r)}}$ by minimizing the denoising objective in (4) over these mean-completed training samples. In Appendix E, our ablation experiments on real-world data show that mean aggregation improves MAE and RMSE over alternative aggregation rules while maintaining comparable distributional accuracy. Together, DOOBCOND and DOOBFIT form the complete DOOBIMP framework evaluated in Section 5.

5 EXPERIMENTS

We evaluate DOOBIMP on nine real-world tabular datasets from the DiffPuter benchmark (Zhang et al., 2025) and four time-series datasets from TSI-Bench (Du et al., 2024b). Tabular experiments use 30% MCAR, MAR, and MNAR missingness over ten seeds; time-series experiments cover point, subsequence, and block missingness over five seeds. For the main imputation benchmarks, models are trained on incomplete training partitions and evaluated on designated missing entries in held-out test data. Complete dataset descriptions, missingness construction, baseline configurations, hyperparameters, and per-dataset results appear in Appendices D–F.

5.1 TABULAR DATA IMPUTATION RESULTS

We compare DOOBIMP with 17 baselines, including DiffPuter, RefiDiff (Ahamed et al., 2026), ReMasker (Du et al., 2024a), MissDiff (Ouyang et al., 2023), TabCSDI (Zheng & Charoenphakdee, 2022), and HyperImpute (Jarrett et al., 2022). Table 1 summarizes the leading methods for numerical and categorical imputation. DOOBIMP achieves the lowest numerical errors and the highest categorical accuracy under all three mechanisms. Across the 27 dataset–mechanism settings, its average rank in mean absolute error (MAE) is 1.19, compared with 2.96 for second-ranked RefiDiff, and DOOBIMP ranks first in 23 settings. This lead across numerical and categorical variables supports the use of an unconditional backbone with sample-specific inference-time conditioning across different missingness mechanisms. Appendix D reports the complete method list, configurations, aggregate rankings, both in-sample (IS) and out-of-sample (OOS) metrics, and all per-dataset results.

Downstream classification measures whether the completed data preserve predictive structure. We fit a logistic-regression model to each completed training set using the same protocol. Panel (c) of Table 1 shows that DOOBIMP attains the highest aggregate multiclass macro-F1 under all three missingness mechanisms. This result extends the advantage of DOOBIMP beyond cellwise reconstruction to predictive structure in the completed data. Appendix D provides the binary AUROC results, complete method comparison, and dataset-level downstream tables.

5.2 TIME-SERIES DATA IMPUTATION RESULTS

We compare DOOBIMP with 22 baselines, including HELIX (Zhang et al., 2026), CSDI (Tashiro et al., 2021), ImputeFormer (Nie et al., 2024), SAITS (Du et al., 2023), and GP-VAE (Fortuin

Table 1: Aggregate tabular results over ten seeds. Panel (a) reports OOS numerical MAE and root mean squared error (RMSE), panel (b) reports OOS categorical accuracy, and panel (c) reports downstream multiclass macro-F1. Tabular entries report mean ± population standard deviation; panel (c) uses population-standard-deviation error bars. Average rank is based on OOS MAE across the 27 dataset–mechanism settings. Bold and underlining mark the best and second-best means.

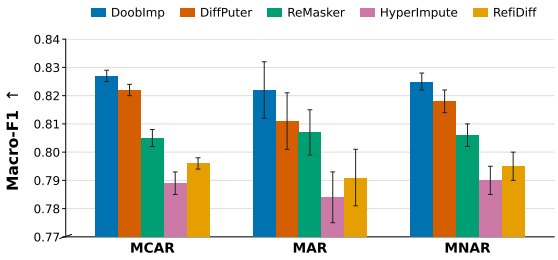
(a) Numerical imputation (↓)

	MAE			RMSE			
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR	Average rank
DOOBIMP	**0.293±0.001**	**0.312±0.011**	**0.307±0.006**	**0.593±0.005**	**0.630±0.019**	**0.628±0.017**	**1.185**
RefiDiff	0.334±0.001	0.349±0.011	0.343±0.006	0.649±0.005	0.685±0.018	0.679±0.020	2.963
DiffPuter	0.316±0.004	0.372±0.015	0.351±0.012	0.628±0.020	0.739±0.032	0.733±0.072	3.333
HyperImpute	0.357±0.005	0.429±0.020	0.388±0.012	1.536±1.057	1.459±1.465	1.508±1.008	5.444
EM	0.391±0.001	0.411±0.012	0.408±0.005	0.669±0.005	0.715±0.021	0.710±0.015	5.519
ReMasker	0.421±0.043	0.419±0.046	0.400±0.024	0.700±0.052	0.716±0.053	0.691±0.020	6.259
MissForest	0.430±0.004	0.457±0.013	0.447±0.008	0.744±0.008	0.812±0.016	0.786±0.019	8.111
TabCSDI	0.459±0.007	0.467±0.013	0.470±0.009	0.790±0.007	0.808±0.020	0.816±0.016	8.630

(b) Categorical accuracy (↑)

Method	MCAR	MAR	MNAR
DOOBIMP	**0.631±0.002**	**0.642±0.054**	**0.630±0.005**
ReMasker	0.621±0.015	0.627±0.048	0.624±0.014
DiffPuter	0.620±0.004	0.618±0.051	0.613±0.008
HyperImpute	0.607±0.015	0.609±0.057	0.614±0.014
RefiDiff	0.595±0.005	0.593±0.043	0.596±0.005
EM	0.582±0.002	0.592±0.044	0.583±0.003
MissDiff	0.576±0.003	0.587±0.046	0.577±0.004
MissForest	0.556±0.008	0.547±0.035	0.556±0.008

(c) Downstream macro-F1 (↑)



et al., 2020). Table 2 summarizes the ETT-h1 results. DOOBIMP achieves the lowest MAE in all four settings, spanning both point and structured missingness. Across the 13 dataset–missingness settings, DOOBIMP achieves the best or joint-best MAE in 10 settings and ranks among the top two methods in all 13. This lead over predictive and generative baselines shows that unconditional backbone learning with inference-time conditioning transfers effectively to structured sequences. Appendix F provides the full time-series results and experimental details.

Table 2: ETT-h1 MAE for the six leading predictive neural models and four deep generative models. Entries are mean ± sample standard deviation over five seeds; lower is better. Bold and underlining mark the best and second-best means.

Method	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%
Predictive neural models				
HELIX	0.128±0.005	0.189±0.012	0.372±0.015	0.489±0.014
SAITS	0.150±0.007	0.208±0.009	0.422±0.019	0.620±0.016
ImputeFormer	0.202±0.044	0.296±0.036	0.404±0.021	0.520±0.017
StemGNN	0.277±0.005	0.314±0.014	0.426±0.011	0.611±0.009
PatchTST	0.229±0.016	0.272±0.027	0.529±0.008	0.689±0.030
FreTS	0.264±0.020	0.294±0.022	0.505±0.014	0.661±0.019
Deep generative models				
DOOBIMP	**0.115±0.002**	**0.177±0.003**	**0.307±0.005**	**0.391±0.015**
CSDI	0.151±0.008	0.318±0.016	0.401±0.067	0.558±0.048
US-GAN	0.458±0.590	0.755±0.973	0.815±0.459	0.670±0.022
GP-VAE	0.329±0.017	0.414±0.013	0.651±0.077	0.806±0.055

5.3 ABLATION STUDIES

We vary the main algorithmic hyperparameters: the number of DOOBFIT stages K, the conditioning temperature and window (τ, W), and the refinement budget J. We also study score-learning quality, inference-time adaptation, and runtime on a subset of tabular datasets: Gesture, Letter, and News. Unless otherwise specified, all experiments use the tabular protocol and the reference DOOBIMP configuration. Figure 3 shows that increasing K yields rapid early gains followed by a stable trajectory. A broad temperature–window region outperforms the DiffPuter reference, indicat-

ing robustness to conditioning choices. Additional refinement steps lower MAE, with the first step capturing most of the gain, while DOOBIMP remains faster than DiffPuter across all tested values of J. At low missing rates, DOOBFIT achieves imputation MAE close to that of a reference trained on fully observed data. Appendix E provides further ablations of pseudo-sample aggregation and precision approximations.

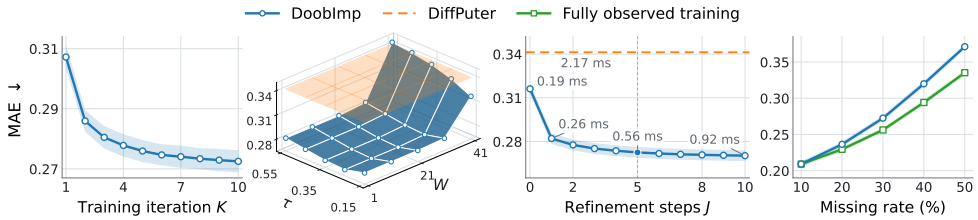


Figure 3: Ablations on three tabular datasets. From left to right: sensitivity to the number of DOOBFIT stages K, conditioning temperature and window (τ, W), and refinement budget J; and comparison with a reference trained on fully observed data across missing rates. Curves and bands show the mean ± population standard deviation over ten seeds after averaging MCAR, MAR and MNAR. Runtime annotations show five-seed means.

Inference-time adaptation. We evaluate adaptation across missingness mechanisms and rates by reusing trained DOOBIMP backbones with fixed conditioning parameters. We compare with DiffPuter, which also learns an unconditional backbone and supports this reuse.

In Table 3(a), MCAR-trained models achieve the lowest average MAE under every test mechanism for both methods, indicating that matching the training and test mechanisms is unnecessary in these experiments. This supports learning an unconditional score from incomplete observations and applying DOOBCOND at test time. DOOBIMP also maintains lower average MAE than DiffPuter across all nine training–test pairs. We further increase test missingness to 40% and 50% within each mechanism, reusing models trained at 30% missingness (Table 3(b)). DOOBIMP maintains lower MAE and RMSE at both rates, with a gradual increase in RMSE and low variability across seeds, whereas DiffPuter exhibits sharply increased RMSE and variability. Appendix E.3.2 provides detailed settings and complete adaptation results.

Table 3: Inference-time adaptation (mean ± std over ten seeds). Bold marks the lower error.

(a) Mechanism adaptation: MAE at 30%

		Test mechanism		
Train	Method	MCAR	MAR	MNAR
MCAR	DOOBIMP	**0.26±0.00**	**0.26±0.01**	**0.27±0.00**
	DiffPuter	0.29±0.01	0.29±0.01	0.30±0.01
MAR	DOOBIMP	**0.28±0.00**	**0.28±0.01**	**0.28±0.01**
	DiffPuter	0.33±0.00	0.34±0.01	0.34±0.01
MNAR	DOOBIMP	**0.27±0.00**	**0.27±0.01**	**0.28±0.01**
	DiffPuter	0.31±0.00	0.31±0.01	0.32±0.01

(b) Rate adaptation: trained at 30%

		Test missing rate		
Metric	Method	30%	40%	50%
MAE	DOOBIMP	**0.27±0.00**	**0.30±0.00**	**0.34±0.00**
	DiffPuter	0.32±0.00	0.36±0.03	0.39±0.04
RMSE	DOOBIMP	**0.52±0.01**	**0.57±0.01**	**0.62±0.01**
	DiffPuter	0.64±0.07	3.40±8.12	4.55±11.36

6 CONCLUSION AND DISCUSSION

We introduce DOOBIMP, a diffusion imputation framework that decouples unconditional backbone learning from test-time imputation. DOOBCOND conditions the unconditional score on each sample's observed entries, while DOOBFIT uses these conditional completions to learn from incomplete training data. This separation avoids training with artificial masks and allows the learned unconditional backbone to be reused across test-time missingness patterns. DOOBIMP demonstrates strong imputation performance on real-world tabular and time-series datasets under diverse missingness mechanisms, with tabular evaluations also showing strong downstream utility.

Looking ahead, DOOBCOND's ability to condition a fixed unconditional backbone opens a path to reusing pretrained diffusion foundation models for imputation. Exploring DOOBCOND as an inference-time plug-in for these models is a promising direction for future work.

AI USE STATEMENT

We used generative AI tools to assist with literature search, feedback on methodology and experiments, proof checking, code review, and manuscript drafting and editing. The authors reviewed all AI-assisted work and independently verified the mathematical claims, citations, code, experimental outputs, and final manuscript. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The appendix provides the complete proofs, algorithmic details, Gaussian validation protocols, model architecture, hyperparameter searches, data splits, missingness constructions, baselines, evaluation procedures, and complete time-series results needed to reproduce the reported findings. We also plan to release the code in the future.

REFERENCES

Md Atik Ahamed, Qiang Ye, and Qiang Cheng. RefiDiff: Progressive refinement diffusion for efficient missing data imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19551–19559, 2026. doi: 10.1609/aaai.v40i24.39034.

Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. An expectation-maximization algorithm for training clean diffusion models from corrupted observations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19447–19471, 2024. doi: 10.52202/079017-0613.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, volume 26, pp. 899–907. Curran Associates, Inc., 2013.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17766–17778, 2020.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/734e6bfcd358e25ac1db0a4241b95651-Paper.pdf.

Jinyuan Chang, Chenguang Duan, Yuling Jiao, Yi Xu, and Jerry Zhijian Yang. Inference-time alignment for diffusion models via variationally stable Doob's matching. *arXiv preprint arXiv:2601.06514*, 2026. doi: 10.48550/arXiv.2601.06514.

Zhichao Chen, Haoxuan Li, Fangyikang Wang, Odin Zhang, Hu Xu, Xiaoyu Jiang, Zhihuan Song, and Hao Wang. Rethinking the diffusion models for missing data imputation: A gradient flow perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3558.

Tae-Min Choi, Ji-Su Kang, and Jong-Hwan Kim. RDIS: Random drop imputation with self-training for incomplete time series data. *IEEE Access*, 11:100720–100728, 2023. doi: 10.1109/ACCESS.2023.3315343.

Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alexandros G. Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0015.

A. P. Dempster, N. M. Laird, and D. 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. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19636–19682, 2024. doi: 10.52202/079017-0620.

Tianyu Du, Luca Melis, and Ting Wang. ReMasker: Imputing tabular data with masked autoencoding. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=KI9NqjLVDT.

Wenjie Du, David Côté, and Yan Liu. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023. doi: 10.1016/j.eswa.2023.119619.

Wenjie Du, Jun Wang, Linglong Qian, Yiyuan Yang, Zina Ibrahim, Fanxing Liu, Zepu Wang, Haoxin Liu, Zhiyuan Zhao, Yingjie Zhou, Wenjia Wang, Kaize Ding, Yuxuan Liang, B. Aditya Prakash, and Qingsong Wen. TSI-Bench: Benchmarking time series imputation. *arXiv preprint arXiv:2406.12747*, 2024b.

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.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep probabilistic time series imputation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1651–1661. PMLR, 2020. URL https://proceedings.mlr.press/v108/fortuin20a.html.

Josh Givens, Song Liu, and Henry W. J. Reeve. Score matching with missing data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19523–19561, 2025. URL https://proceedings.mlr.press/v267/givens25a.html.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Zhengyi Guo, Wenpin Tang, and Renyuan Xu. Conditional diffusion guidance under hard constraint: A stochastic analysis approach. *arXiv preprint arXiv:2602.05533*, 2026. doi: 10.48550/arXiv.2602.05533.

Danial Hosseintabar, Fan Chen, Giannis Daras, Antonio Torralba, and Constantinos Daskalakis. DiffEM: Learning from corrupted data with diffusion models via expectation maximization. *arXiv preprint arXiv:2510.12691*, 2025. doi: 10.48550/arXiv.2510.12691.

Mohammad Rafid Ul Islam, Prasad Tadepalli, and Alan Fern. Self-attention-based diffusion model for time-series imputation in partial blackout scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17564–17572, 2025. doi: 10.1609/aaai.v39i17.33931.

Daniel Jarrett, Bogdan C. Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. HyperImpute: Generalized iterative imputation with automatic model selection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9916–9937, 2022. URL https://proceedings.mlr.press/v162/jarrett22a.html.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

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, pp. 26565–26577. Curran Associates, Inc., 2022.

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

Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 3rd edition, 2019.

Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. PriSTI: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1927–1939. IEEE, 2023. doi: 10.1109/ICDE55515.2023.00150.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. NAOMI: Non-autoregressive multiresolution sequence imputation. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/50c1f44e426560f3f2cdcb3e19e39903-Paper.pdf.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023.

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*, pp. 11461–11471, 2022. doi: 10.1109/CVPR52688.2022.01117.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4413–4423, 2019. URL https://proceedings.mlr.press/v97/mattei19a.html.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010. URL https://jmlr.org/papers/v11/mazumder10a.html.

Wang Miao, Lan Liu, Yilin Li, Eric J. Tchetgen Tchetgen, and Zhi Geng. Identification and semiparametric efficiency theory of nonignorable missing data with a shadow variable. *ACM/IMS Journal of Data Science*, 1(2):1–23, 2024. doi: 10.1145/3592389.

Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8983–8991, 2021. doi: 10.1609/aaai.v35i10.17086.

Toan Nguyen, Kien Do, Duc Kieu, and Thin Nguyen. h-Edit: Effective and flexible diffusion-based editing via Doob's h-transform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28490–28501, 2025.

Tong Nie, Guoyang Qin, Wei Ma, Yuewen Mei, and Jian Sun. ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2260–2271, 2024. doi: 10.1145/3637528.3671751.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Linglong Qian, Yiyuan Yang, Wenjie Du, Jun Wang, Richard Dobson, and Zina Ibrahim. Beyond random missingness: Clinically rethinking for healthcare time series imputation. *arXiv preprint arXiv:2405.17508*, 2024. doi: 10.48550/arXiv.2405.17508.

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*, pp. 14205–14214, 2020. doi: 10.1109/CVPR42600.2020.01421.

L. C. G. Rogers and David Williams. *Diffusions, Markov Processes, and Martingales*, volume 2. Cambridge University Press, 2000. doi: 10.1017/CBO9780511805141.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. Learning diffusion priors from observations by expectation maximization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87647–87682, 2024. doi: 10.52202/079017-2783.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581.

Gyuwon Sim, Sumin Lee, Heesun Bae, Byeonghu Na, Doyun Kwon, Ju-Hee Hwang, Jae-Young Lim, and Il-Chul Moon. Missing pattern recognized diffusion imputation model for missing not at random. *arXiv preprint arXiv:2605.25439*, 2026. doi: 10.48550/arXiv.2605.25439.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012. doi: 10.1093/bioinformatics/btr597.

BaoLuo Sun, Lan Liu, Wang Miao, Kathleen Wirth, James Robins, and Eric J. Tchetgen Tchetgen. Semiparametric estimation with data missing not at random using an instrumental variable. *Statistica Sinica*, 28(4):1965–1983, 2018. doi: 10.5705/ss.202016.0324.

Sabera J. Talukder, Yisong Yue, and Georgia Gkioxari. TOTEM: Tokenized time series embeddings for general time series analysis. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QlTLkH6xRC.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011. doi: 10.18637/jss.v045.i03.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1CLzLXSFNn.

Xu Wang, Hongbo Zhang, Pengkun Wang, Yudong Zhang, Binwu Wang, Zhengyang Zhou, and Yang Wang. An observed value consistent diffusion model for imputing missing values in multivariate time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2409–2418, 2023. doi: 10.1145/3580305.3599257.

Yuzhu Wang, Xi Ye, Duo Su, Yangyang Xu, and Jun Zhu. h-control: Training-free camera control via block-conditional Gibbs refinement. *arXiv preprint arXiv:2605.11871*, 2026. doi: 10.48550/arXiv.2605.11871.

Chunjing Xiao, Xue Jiang, Xianghe Du, Wei Yang, Wei Lu, Xiaomin Wang, and Kevin Chetty. Boundary-enhanced time series data imputation with long-term dependency diffusion models. *Knowledge-Based Systems*, 310:112917, 2025. doi: 10.1016/j.knosys.2024.112917.

Zeqi Ye and Minshuo Chen. Diffusion transformers for imputation: Statistical efficiency and uncertainty quantification. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4111.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5689–5698, 2018. URL https://proceedings.mlr.press/v80/yoon18a.html.

Tianxiang Zhan, Yuanpeng He, Yong Deng, Zhen Li, Wenjie Du, and Qingsong Wen. Time evidence fusion network: Multi-source view in long-term time series forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11220–11233, 2025. doi: 10.1109/TPAMI.2025.3596905.

Fengming Zhang, Wenjie Du, Huan Zhang, Ke Yu, and Shen Qu. HELIX: Hybrid encoding with learnable identity and cross-dimensional synthesis for time series imputation. *arXiv preprint arXiv:2605.02278*, 2026.

Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. DiffPuter: Empowering diffusion models for missing data imputation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3fl1SENSYO.

Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *arXiv preprint arXiv:2210.17128*, 2022.

Jianping Zhou, Junhao Li, Guanjie Zheng, Xinbing Wang, and Chenghu Zhou. MTSCI: A conditional diffusion model for multivariate time series consistent imputation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3474–3483, 2024. doi: 10.1145/3627673.3679532.

Qijie Zhu, Zeqi Ye, Han Liu, Zhaoran Wang, and Minshuo Chen. Training-free adaptation of diffusion models via Doob's h-transform. *arXiv preprint arXiv:2602.16198*, 2026. doi: 10.48550/arXiv.2602.16198.

A Conditional Diffusion under a Training-Time Artificial-Mask Distribution

A.1 Population Conditional Objective

We use the notation from Section 3.2. For each constructed training pair, M records which available entries remain observed, while X_0^{mis} consists of the artificially hidden entries and is therefore known during training. Entries that were already missing in the raw record do not contribute a supervised target. Applying the forward noising relation from Section 3.1 to the artificially hidden coordinates gives the conditional denoising score matching objective

$$\mathcal{L}_{\text{cond}}(\phi) = \mathbb{E}_{t,X_0,X_t,M}\left[\lambda(t)\left\|s_\phi^{\text{cond}}(X_t^{\text{mis}}, X_0^{\text{obs}}, M, t) - \nabla_{X_t^{\text{mis}}} \log p_{t|0}(X_t^{\text{mis}} \mid X_0, M)\right\|_2^2\right]. \tag{14}$$

For the Gaussian transition, the target on the artificially hidden coordinates is $\nabla_{X_t^{\text{mis}}} \log p_{t|0}(X_t^{\text{mis}} \mid X_0, M) = -(X_t^{\text{mis}} - \alpha_t X_0^{\text{mis}})/\sigma_t^2$. Thus, minimizing (14) trains s_ϕ^{cond} toward the missing-data conditional score required by (5), averaged over the training-time artificial-mask distribution used to construct the training pairs.

At a fixed diffusion time t, for a candidate missing-data conditional score s and a missingness distribution q, we define the fixed-time excess risk

$$\mathcal{E}_t(s;q) := \lambda(t)\,\mathbb{E}\left[\left\|s(X_t^{\text{mis}}, X_0^{\text{obs}}, M, t) - \nabla_{X_t^{\text{mis}}} \log p_t(X_t^{\text{mis}} \mid X_0^{\text{obs}}, M)\right\|_2^2\right], \tag{15}$$

where the expectation is over $M \sim q$, $X_0 \sim P_{\text{data}}$, and $Z \sim \mathcal{N}(0, I_D)$, with the observed and missing coordinates selected by M. By the conditional denoising-score identity, $\mathcal{E}_t(s;q)$ equals the fixed-t population version of (14), minus its minimum over candidate missing-data conditional scores. This subtraction removes the irreducible variance of the Gaussian denoising target.

A.2 Sampling under Missingness Distribution Shift

Fix a missing cardinality $k \in \{1, \dots, D-1\}$, restrict M to patterns satisfying $\sum_{d=1}^D (1 - M_d) = k$, and assume that M is independent of X_0. Consider the stationary variance-preserving diffusion

$$\mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t, \qquad X_0 \sim \mathcal{N}(0, I_D),$$

where $\beta(t) > 0$, $A_T := \frac{1}{2}\int_0^T \beta(t)\mathrm{d}t$, and $\alpha_T := e^{-A_T}$.

Using the reverse-SDE weighting $\lambda(t) = \beta(t)$ in (15), define the integrated risk by

$$\mathcal{E}_{\text{rev}}(s;q) := \int_0^T \mathcal{E}_t(s;q)\mathrm{d}t.$$

For $c = (x_0^{\text{obs}}, m)$, let $\widehat{P}_{0,s}^c$ be the reverse-SDE endpoint law under s, initialized from $\mathcal{N}(0, I_k)$ at time T. The true conditional endpoint is also $\mathcal{N}(0, I_k)$, so the aggregate sampling error is

$$\mathfrak{W}_2(s;q) := \left\{\mathbb{E}_{M\sim q, X_0}\left[W_2^2\left(\mathcal{N}(0, I_k), \widehat{P}_{0,s}^c\right)\right]\right\}^{1/2}.$$

Proposition 2 (Sampling under missingness distribution shift)**.** *Let q_{art} be any full-support training-time artificial mask distribution over the fixed-k patterns, and write*

$$q_{\min} := \min_{\sum_{d=1}^D (1-m_d)=k} q_{\text{art}}(m).$$

For every $\epsilon > 0$, there exist a full-support test-time missingness distribution q and a candidate missing-data conditional score s such that

$$\mathcal{E}_{\text{rev}}(s; q_{\text{art}}) = \frac{\epsilon}{2}, \qquad \mathcal{E}_{\text{rev}}(s;q) = \frac{3}{4q_{\min}}\mathcal{E}_{\text{rev}}(s; q_{\text{art}}).$$

Moreover,

$$\mathfrak{W}_2(s; q_{\text{art}}) = \frac{1-\alpha_T}{\sqrt{A_T}}\sqrt{\epsilon}, \qquad \mathfrak{W}_2(s;q) = \sqrt{\frac{3}{4q_{\min}}}\,\mathfrak{W}_2(s; q_{\text{art}}).$$

Proof. Choose a pattern $m_\star$ satisfying $q_{\mathrm{art}}(m_\star) = q_{\min}$. Since there are at least two fixed-k patterns, $q_{\min} < 1$. Define

$$q(m') = \begin{cases} \dfrac{3}{4}, & m' = m_\star, \\ \dfrac{q_{\mathrm{art}}(m')}{4(1-q_{\min})}, & m' \neq m_\star. \end{cases}$$

This is a full-support distribution. Under the stationary VP diffusion above, every marginal is standard Gaussian and the true missing-data conditional score is $-x_t^{\mathrm{mis}}$. Let $e_1 \in \mathbb{R}^k$ be the first coordinate vector and set

$$a := \sqrt{\frac{\epsilon}{4A_T q_{\min}}}, \qquad s(x_t^{\mathrm{mis}}, x_0^{\mathrm{obs}}, m', t) := -x_t^{\mathrm{mis}} + a e_1 \mathbf{1}\{m' = m_\star\}.$$

Because $\int_0^T \beta(t)\mathrm{d}t = 2A_T$, the score risk is

$$\mathcal{E}_{\mathrm{rev}}(s; q_{\mathrm{art}}) = 2A_T q_{\min} a^2 = \frac{\epsilon}{2},$$

whereas

$$\mathcal{E}_{\mathrm{rev}}(s; q) = 2A_T \frac{3}{4} a^2 = \frac{3\epsilon}{8q_{\min}}.$$

It remains to compute the sampling error. For the rare pattern, write the reverse process in increasing time as $Y_u = X_{T-u}^{\mathrm{mis}}$. Its first coordinate satisfies

$$\mathrm{d}Y_u = \left[-\frac{1}{2}\beta(T-u)Y_u + \beta(T-u)a\right]\mathrm{d}u + \sqrt{\beta(T-u)}\mathrm{d}B_u, \qquad Y_0 \sim \mathcal{N}(0,1).$$

The variance remains one, while the mean μ_u solves $\mathrm{d}\mu_u/\mathrm{d}u = -\frac{1}{2}\beta(T-u)\mu_u + \beta(T-u)a$ with $\mu_0 = 0$. Hence $\mu_T = 2a(1-\alpha_T)$. The other coordinates and all other patterns follow the exact reverse process. Therefore,

$$W_2^2\left(\mathcal{N}(0, I_k), \widehat{P}_{0,s}^c\right) = 4a^2(1-\alpha_T)^2$$

for $m_\star$, and it is zero otherwise. Averaging first under q_{art} and then under q gives

$$\mathfrak{W}_2(s; q_{\mathrm{art}}) = 2a(1-\alpha_T)\sqrt{q_{\min}} = \frac{1-\alpha_T}{\sqrt{A_T}}\sqrt{\epsilon},$$

$$\mathfrak{W}_2(s; q) = 2a(1-\alpha_T)\sqrt{\frac{3}{4}} = \sqrt{\frac{3}{4q_{\min}}}\,\mathfrak{W}_2(s; q_{\mathrm{art}}).$$

□

Discussion. Conditional-score error vanishes in the realizable population limit, but practical training observes only finitely many constructed mask–data pairs. Full support over mask vectors therefore does not ensure coverage of their joint configurations with deployment data. DOOBIMP uses each incomplete sample to refine a shared unconditional score and applies pattern-specific adaptation to the realized mask through DOOBCOND at inference time. Figures 2 and 8 provide complementary evidence at the score-risk and posterior-sampling levels.

B METHODOLOGICAL DERIVATIONS

This section collects the derivations underlying DOOBIMP. We first prove the Doob-conditioning result, then derive the practical Gaussian conditioning approximation and its DDIM and EDM discretizations, and finally discuss a possible mechanism-aware extension for MNAR.

B.1 PROOF OF LEMMA 1

Proof. Assume $\mathbb{E}_\theta[w_\eta(\widetilde{X}_0)] \in (0, \infty)$ and the standard regularity conditions for the change of measure. Introduce increasing reverse time $u = T - t$ and set $Y_u = \widetilde{X}_{T-u}$. Under $\mathbb{P}_\theta$,

$$\mathrm{d}Y_u = \left[-f(T-u)Y_u + g(T-u)^2 s_\theta(Y_u, T-u)\right]\mathrm{d}u + g(T-u)\mathrm{d}W_u. \tag{16}$$

Define $\bar{h}(y,u) := h_\eta(y, T-u)$. The Markov property makes $\bar{h}(y,u)$ space–time harmonic for the base generator, so under $\mathbb{Q}_\eta$, $Y_u^h := \widetilde{X}_{T-u}^h$ satisfies

$$\mathrm{d}Y_u^h = \Big[-f(T-u)Y_u^h + g(T-u)^2 \big\{ s_\theta(Y_u^h, T-u) + \nabla_y \log \bar{h}(Y_u^h, u) \big\} \Big] \mathrm{d}u + g(T-u)\mathrm{d}W_u^h. \tag{17}$$

Returning to decreasing time $t = T-u$ gives (6). For every bounded measurable φ, the definition of $\mathbb{Q}_\eta$ also gives

$$\mathbb{E}_{\mathbb{Q}_\eta}[\varphi(\widetilde{X}_0^h)] = \frac{\mathbb{E}_\theta[\varphi(\widetilde{X}_0) w_\eta(\widetilde{X}_0)]}{\mathbb{E}_\theta[w_\eta(\widetilde{X}_0)]} = \int \varphi(x) P_{\theta,0}^\eta(\mathrm{d}x), \tag{18}$$

which proves $\widetilde{X}_0^h \sim P_{\theta,0}^\eta$. □

Hard-constraint limit. Let $p_{\theta,0}$ denote the density of $P_{\theta,0}$. Assume it is continuous in the observed coordinates, is locally dominated in the remaining coordinates by an integrable function, and assigns positive marginal density to x_0^{obs}. The convergence statement preceding Lemma 1 follows from the Gaussian approximate-identity argument in the observed coordinates. In particular, for every bounded continuous function φ, the expectation of $\varphi(\widetilde{X}_0^{\mathrm{mis}})$ under $P_{\theta,0}^\eta$ is the ratio of its $w_\eta(\widetilde{X}_0)$-weighted integral under $P_{\theta,0}$ to $\mathbb{E}_\theta[w_\eta(\widetilde{X}_0)]$. Continuity at x_0^{obs}, together with the usual local domination for the approximate-identity limit, makes this ratio converge to

$$\frac{\int \varphi(\widetilde{x}_0^{\mathrm{mis}}) p_{\theta,0}(x_0^{\mathrm{obs}}, \widetilde{x}_0^{\mathrm{mis}}) \mathrm{d}\widetilde{x}_0^{\mathrm{mis}}}{\int p_{\theta,0}(x_0^{\mathrm{obs}}, \widetilde{x}_0^{\mathrm{mis}}) \mathrm{d}\widetilde{x}_0^{\mathrm{mis}}}.$$

The denominator is positive by assumption, and the ratio is the expectation under $P_{\theta,0}(\widetilde{X}_0^{\mathrm{mis}} \mid \widetilde{X}_0^{\mathrm{obs}} = x_0^{\mathrm{obs}})$.

B.2 Gaussian Derivation of Doob Conditioning

This subsection uses a Gaussian analysis model to derive (8) and motivate the approximations in (9). Suppose that the unconditional score is exact, $P_{\mathrm{data}} = \mathcal{N}(\mu, V)$, and the reverse process is initialized from the forward terminal marginal. Under these assumptions, the learned reverse process is the exact time reversal of the forward diffusion, $P_{\theta,0} = P_{\mathrm{data}}$, and the joint law relevant to h_η is equivalently obtained by forward noising $X_t = \alpha_t X_0 + \sigma_t Z$.

Specifically, suppose that

$$X_0 \sim \mathcal{N}(\mu, V), \qquad X_t = \alpha_t X_0 + \sigma_t Z, \qquad Z \sim \mathcal{N}(0, I_D).$$

Gaussian conditioning then gives

$$X_0 \mid X_t = x \sim \mathcal{N}\left(\mu_{0|t}(x), V_{0|t}\right),$$

where the posterior moments are

$$\mu_{0|t}(x) = \mu + \alpha_t V \left(\alpha_t^2 V + \sigma_t^2 I_D\right)^{-1} (x - \alpha_t \mu),$$

$$V_{0|t} = V - \alpha_t^2 V \left(\alpha_t^2 V + \sigma_t^2 I_D\right)^{-1} V.$$

In particular, $V_{0|t}$ does not depend on x.

Combining the observed-coordinate posterior with the Gaussian terminal weight w_η gives

$$h_\eta(x,t) \propto \mathcal{N}\left(x_0^{\mathrm{obs}}; \mu_{0|t}^{\mathrm{obs}}(x), V_{0|t}^{\mathrm{obs,obs}} + \eta^2 I_{D_{\mathrm{obs}}}\right).$$

Differentiating this expression gives (8). No derivative of the covariance appears because $V_{0|t}$ is independent of x.

The Gaussian score satisfies

$$\mu_{0|t}(x) = \frac{x + \sigma_t^2 \nabla_x \log p_t(x)}{\alpha_t}.$$

This Tweedie identity motivates the posterior-mean substitution in (9). When $s_\theta = \nabla_x \log p_t$ for the forward-noised data law, $\widehat{x}_{\theta,0}$ equals the corresponding posterior mean.

Finally, rather than estimating the complete observed covariance block at every reverse step, we approximate its diagonal using empirical marginal variances. For a scalar Gaussian prior with variance v_i, the clean posterior variance is $v_i \sigma_t^2 / (\alpha_t^2 v_i + \sigma_t^2)$. Applying the scalar Gaussian posterior variance coordinatewise gives Λ_t in (9). This approximation is applied only to the observed-coordinate precision; cross-coordinate dependence is retained through the Jacobian of the learned Tweedie estimate.

B.3 Diffusion Backward Discretizations for DoobCond

For compactness, we define the missing-coordinate state and denoised estimate used by the reverse solver as

$$\check{x}^{\mathrm{mis}}_{t_\ell} := \begin{cases} x^{\mathrm{mis},(J)}_{t_\ell}, & 1 \le W \le \ell,\ J > 0, \\ x^{\mathrm{mis}}_{t_\ell}, & \text{otherwise}, \end{cases} \qquad \check{x}^{\mathrm{mis}}_{0,\ell} := \begin{cases} J^{-1}\sum_{j=1}^{J} \widehat{x}^{\mathrm{mis},(j)}_0, & 1 \le W \le \ell,\ J > 0, \\ \widehat{x}^{h,\mathrm{mis}}_{\theta,0}(x_{t_\ell}, t_\ell), & \text{otherwise}. \end{cases}$$

DDIM. For a variance-preserving schedule with $\alpha_t^2 + \sigma_t^2 = 1$, we let $\zeta \in [0,1]$ denote the DDIM stochasticity coefficient and define

$$\rho_\ell := \zeta \frac{\sigma_{t_{\ell+1}}}{\sigma_{t_\ell}} \sqrt{1 - \frac{\alpha^2_{t_\ell}}{\alpha^2_{t_{\ell+1}}}}.$$

The DDIM update (Song et al., 2021a) is

$$x^{\mathrm{mis}}_{t_{\ell+1}} = \alpha_{t_{\ell+1}} \check{x}^{\mathrm{mis}}_{0,\ell} + \sqrt{\sigma^2_{t_{\ell+1}} - \rho^2_\ell} \frac{\check{x}^{\mathrm{mis}}_{t_\ell} - \alpha_{t_\ell} \check{x}^{\mathrm{mis}}_{0,\ell}}{\sigma_{t_\ell}} + \rho_\ell Z^{\mathrm{mis}}. \tag{19}$$

Here a fresh $Z^{\mathrm{mis}} \sim \mathcal{N}(0, I_{D_{\mathrm{mis}}})$ is drawn at each reverse step. The controlled Gaussian studies use the corresponding variance-preserving Euler discretization with $L = 200$ unless stated otherwise; $\zeta = 0$ gives deterministic DDIM.

EDM. For the variance-exploding specialization $\alpha_t = 1$, EDM (Karras et al., 2022) integrates the probability-flow ODE with σ_t as the independent variable. In this parameterization, the tractable Gaussian terminal prior is $\mathcal{N}(0, \sigma^2_{t_1} I_D)$. Its Euler proposal is

$$\bar{x}^{\mathrm{mis}}_{t_{\ell+1}} = \check{x}^{\mathrm{mis}}_{t_\ell} + \left(\sigma_{t_{\ell+1}} - \sigma_{t_\ell}\right) \frac{\check{x}^{\mathrm{mis}}_{t_\ell} - \check{x}^{\mathrm{mis}}_{0,\ell}}{\sigma_{t_\ell}}. \tag{20}$$

Within this multistage solver, we draw $x^{\mathrm{obs}}_{t_{\ell+1}}$ using (7) and pair it with the proposal to form $\bar{x}_{t_{\ell+1}}$; the subsequent observed-coordinate step in Algorithm 1 reuses this draw. If $\sigma_{t_{\ell+1}} > 0$, the second-order Heun correction is

$$x^{\mathrm{mis}}_{t_{\ell+1}} = \check{x}^{\mathrm{mis}}_{t_\ell} + \frac{\sigma_{t_{\ell+1}} - \sigma_{t_\ell}}{2} \left[\frac{\check{x}^{\mathrm{mis}}_{t_\ell} - \check{x}^{\mathrm{mis}}_{0,\ell}}{\sigma_{t_\ell}} + \frac{\bar{x}^{\mathrm{mis}}_{t_{\ell+1}} - \widehat{x}^{h,\mathrm{mis}}_{\theta,0}(\bar{x}_{t_{\ell+1}}, t_{\ell+1})}{\sigma_{t_{\ell+1}}} \right]. \tag{21}$$

At active steps $1 \le W \le \ell$, the endpoint readout in (21) is refined in the same way before evaluating the second slope. At $\sigma_{t_{\ell+1}} = 0$, EDM retains the Euler proposal to avoid division by zero. Both solvers use the same fixed (η, τ, W): the Doob conditioning has strength τ^{-2} when $1 \le W \le \ell$, and the unconditioned Tweedie estimate is used otherwise. This factor is already included in (11) and is not applied again inside either solver.

B.4 Further Discussion: Mechanism-Aware Conditioning under MNAR

Under MNAR, the missingness mechanism contributes information about the unobserved entries, and the posterior satisfies

$$p(x_0^{\mathrm{mis}} \mid x_0^{\mathrm{obs}}, m) \propto p_0(x_0^{\mathrm{obs}}, x_0^{\mathrm{mis}}) q(m \mid x_0^{\mathrm{obs}}, x_0^{\mathrm{mis}}).$$

If a model $q_\psi(m \mid x_0)$ of the missingness mechanism is available, one can develop a mechanism-aware extension using the hard target

$$P^{\mathrm{hard}}_{\theta,\psi,0}(\mathrm{d}\widetilde{x}_0) \propto \delta(\widetilde{x}_0^{\mathrm{obs}} - x_0^{\mathrm{obs}}) q_\psi(m \mid \widetilde{x}_0) P_{\theta,0}(\mathrm{d}\widetilde{x}_0).$$

Related diffusion-based treatments of MNAR likewise introduce an explicit model of the missingness mechanism; see, for example, Sim et al. (2026).

For the smoothed observed-value constraint, such an extension replaces w_η with

$$w_{\eta,\psi}(\widetilde{x}_0) := w_\eta(\widetilde{x}_0) q_\psi(m \mid \widetilde{x}_0),$$

and we define

$$h_{\eta,\psi}(x,t) := \mathbb{E}_\theta \left[w_{\eta,\psi}(\widetilde{X}_0) \mid \widetilde{X}_t = x \right].$$

Using the same clean reconstruction, one can obtain a practical mechanism-aware conditioning term by differentiating

$$-\frac{1}{2}\left(\widehat{x}^{\mathrm{obs}}_{\theta,0} - x^{\mathrm{obs}}_0\right)^\top \Lambda_t \left(\widehat{x}^{\mathrm{obs}}_{\theta,0} - x^{\mathrm{obs}}_0\right) + \log q_\psi\left(m \mid \widehat{x}_{\theta,0}\right)$$

with respect to x^{mis}. When q_ψ is available, this formulation can combine the unconditional score and the missingness-mechanism likelihood within the same Doob conditioning construction, providing a direct route to mechanism-aware MNAR imputation.

Relation to the evaluated method. The practical DOOBIMP evaluated in this paper does not explicitly model q_ψ, whose identification under MNAR generally relies on additional assumptions or auxiliary information, such as a valid shadow variable or instrument (Miao et al., 2024; Sun et al., 2018). Under the evaluated MNAR settings, Table 7 shows that DOOBIMP achieves the lowest aggregate MAE and RMSE among the compared methods. Explicitly modeling the missingness mechanism within DOOBIMP remains a direction for future work.

C CONTROLLED GAUSSIAN EXPERIMENTS

We use Gaussian models with analytic conditional distributions to evaluate the approximations underlying DOOBCOND. Subsection C.2 compares the ideal transform with the practical sampler; Subsection C.3 studies dimension scaling; Subsection C.4 isolates the diagonal posterior-precision approximation; and Subsection C.5 evaluates robustness to missingness distribution shift between the training-time artificial-mask distribution and the test-time missingness distribution. Subsection C.6 then examines sensitivity to endpoint smoothing. Together, these studies connect the conditioning identity in Lemma 1 to the finite-step approximation in (11).

C.1 EXPERIMENTAL SETTING

Data and conditional targets. Unless stated otherwise, $X_0 \sim \mathcal{N}(0, V)$. We use the first-order autoregressive (AR(1)) covariance family $V_{ij} = \rho^{|i-j|}$, where $\rho \in [0,1)$ is the adjacent-coordinate correlation and controls dependence strength. For a realized pattern m, let $o = \{d : m_d = 1\}$ and $u = \{d : m_d = 0\}$ denote the observed and missing coordinates. Given $x_{0,o}$, the hard conditional target is

$$X_{0,u} \mid X_{0,o} = x_{0,o} \sim \mathcal{N}(\mu_{u|o}, V_{u|o}), \quad \mu_{u|o} = V_{uo}V_{oo}^{-1}x_{0,o}, \quad V_{u|o} = V_{uu} - V_{uo}V_{oo}^{-1}V_{ou}. \tag{22}$$

The relaxed target induced by the Gaussian endpoint smoothing η has missing-coordinate moments

$$\mu^\eta_{u|o} = V_{uo}(V_{oo} + \eta^2 I_{D_{\mathrm{obs}}})^{-1}x_{0,o}, \qquad V^\eta_{u|o} = V_{uu} - V_{uo}(V_{oo} + \eta^2 I_{D_{\mathrm{obs}}})^{-1}V_{ou}. \tag{23}$$

Hence, the hard and relaxed targets and the exact Doob conditioning are available in closed form.

Missingness geometries. All missingness patterns have a fixed missing cardinality $k = |u|$ and are independent of X_0. Under the point-missingness distribution q_{point}, u is sampled uniformly from all k-coordinate subsets. Under the block-missingness distribution q_{block}, a start index is sampled uniformly from $\{1, \ldots, D-k+1\}$, and the corresponding k consecutive coordinates are missing.

Diffusion and score models. We use the variance-preserving SDE in (1) with $T = 1$, $f(t) = -\beta(t)/2$, $g(t) = \sqrt{\beta(t)}$, and $\beta(t) = 0.1 + 19.9t$, so that $X_t = \alpha_t X_0 + \sigma_t Z$, where $\alpha_t = \exp(-\frac{1}{2}\int_0^t \beta(r)\mathrm{d}r)$ and $\sigma_t^2 = 1 - \alpha_t^2$. Whenever a learned unconditional score is used, it is trained directly from fully observed Gaussian samples with $n_{\mathrm{train}} = 200D$ using a four-block residual multilayer perceptron of width 256, a 64-dimensional time embedding, and the objective in (4). The missingness-shift study additionally trains conditional scores under specified missingness distributions.

Table 4: Default controlled-Gaussian protocol. All Doob samplers use $(\tau, W, J) = (1, 1, 0)$. Each study uses five seeds and, unless stated otherwise, 50 independently generated conditions per seed and 1,024 posterior draws per condition. Individual subsections specify any deviations.

D	ρ	k	η	L	Seeds
40	0.6	10	0.1	200	5

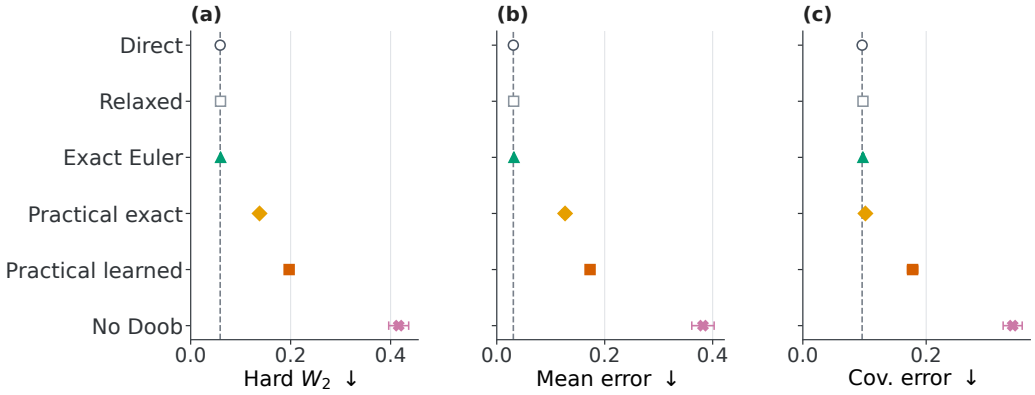


Figure 4: Conditional-sampling decomposition relative to the hard target. Panels (a)–(c) report $\widetilde{W}_2$, E_μ, and E_V, respectively. The labels *Direct* and *Relaxed* are direct hard and relaxed sampling; *Exact Euler* is exact h_η Euler–Maruyama sampling; *Practical exact* and *Practical learned* use the analytic and learned scores, respectively; and *No Doob* removes the conditioning from the learned-score variant. Exact Euler matches direct relaxed sampling at the finite-sample scale, while the practical conditioning remains markedly more accurate than no Doob. Error bars show the population standard deviation over five seeds.

Evaluation. For each condition, let $\widehat{\mu}$ and $\widehat{V}$ be the empirical moments of $B = 1{,}024$ posterior samples. Unless stated otherwise, all sampling metrics use the hard conditional in (22) as the reference:

$$E_\mu := \frac{\|\widehat{\mu} - \mu_{u|o}\|_2}{\sqrt{\mathrm{tr}(V_{u|o})}}, \qquad E_V := \frac{\|\widehat{V} - V_{u|o}\|_{\mathrm{F}}}{\|V_{u|o}\|_{\mathrm{F}}},$$

$$\widetilde{W}_2 := \frac{W_2\Big(\mathcal{N}(\widehat{\mu}, \widehat{V}), \mathcal{N}(\mu_{u|o}, V_{u|o})\Big)}{\sqrt{\mathrm{tr}(V_{u|o})}}.$$

Accordingly, the figure label *Hard* W_2 denotes $\widetilde{W}_2$, while *Mean error* and *Cov. error* denote E_μ and E_V, respectively. The label *Relaxed* W_2 uses the same normalized Gaussian W_2 with $(\mu^\eta_{u|o}, V^\eta_{u|o})$ as the reference and $\sqrt{\mathrm{tr}(V^\eta_{u|o})}$ as the normalizer. Direct hard-conditional sampling therefore measures the finite-B moment estimation floor, while direct relaxed-conditional sampling also reflects the endpoint relaxation. Conditions are averaged within each seed before computing the arithmetic mean and population standard deviation across five equally weighted seeds.

C.2 Accuracy of the Conditioning Approximation

Question and design. We ask whether the exact Doob transform recovers the analytic relaxed target and how much accuracy is retained by the approximations in practical DoobCond. *Direct hard* draws samples from the analytic hard conditional in (22), whereas *Direct relaxed* draws from the analytic relaxed conditional in (23); both use Cholesky sampling and serve as Monte Carlo references rather than diffusion samplers. *Exact* h_η *(Euler)* starts from the exact forward-noised relaxed law and applies the analytic unconditional score and full h_η conditioning in (6) with Euler–Maruyama discretization. *Practical (exact score)* and *Practical (learned score)* instead use the observed-coordinate refresh in (7) and the diagonal conditioning term in (11), with the analytic and learned unconditional scores, respectively. *No Doob (learned score)* uses the same learned score, terminal initialization, and observed-coordinate refresh as the practical learned-score sampler but omits h_η conditioning. All practical settings use $J = 0$.

Results. Figure 4 shows that exact Doob sampling is indistinguishable from direct relaxed sampling at the Monte Carlo scale, validating the conditioning identity. Relative to exact Doob, the practical exact-score variant primarily increases mean error. With a learned score, Doob conditioning more than halves *Hard* W_2 and substantially reduces covariance error relative to the matched no-Doob control. Thus, the practical approximation retains a clear conditioning benefit under both approximation and score-estimation error.

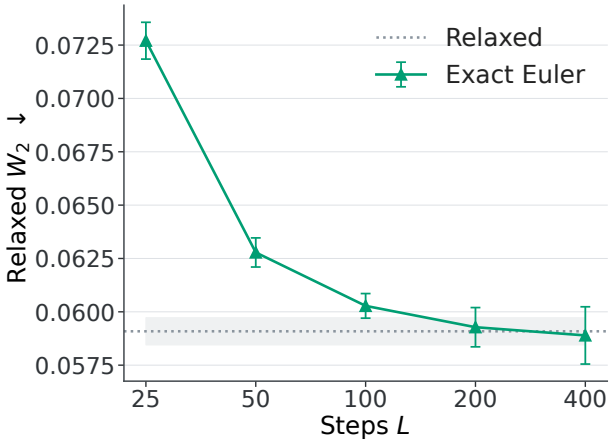


Figure 5: Reverse-time Euler discretization for the exact h_η sampler. The vertical axis is *Relaxed* W_2, the normalized Gaussian W_2 to the relaxed target; *Relaxed* is the direct-relaxed Monte Carlo reference, and *Exact Euler* is the exact h_η sampler with Euler–Maruyama. Performance reaches the reference scale by $L = 200$, with negligible improvement from doubling the number of steps. Error bars show the population standard deviation over five seeds.

Figure 5 shows that the sampling error plateaus by $L = 200$. We therefore use $L = 200$ in the remaining controlled studies.

C.3 Scaling with Data Dimension

Question and design. We next test whether the practical conditioning benefit persists as the data dimension increases using the five corresponding settings defined in Subsection C.2, excluding *Direct relaxed*. We vary $D \in \{10, 20, 40, 80, 160\}$, set $n_{\text{train}} = 200D$, and choose $|u| = \lceil 0.25D \rceil$ coordinates uniformly, using 30 conditions per seed at each dimension. The score-network width and all remaining training and sampling choices are fixed. The training sample size grows proportionally with dimension.

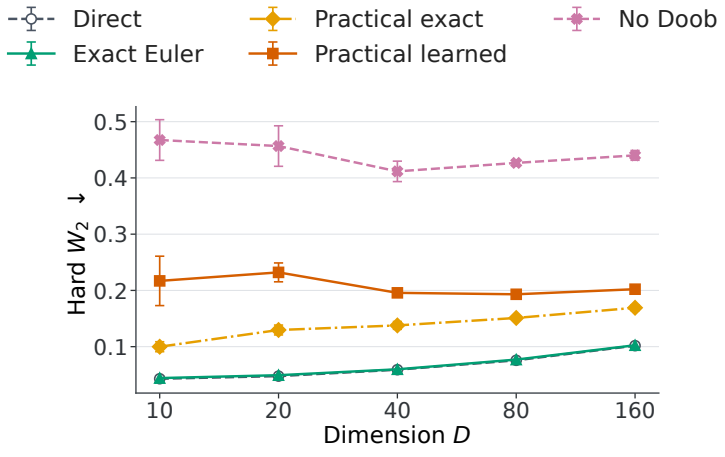


Figure 6: Dimension scaling under $n_{\text{train}} = 200D$, $k = \lceil 0.25D \rceil$, and a fixed-width score network. The vertical axis is *Hard* W_2, as defined in Subsection C.1. *Direct*, *Exact Euler*, *Practical exact*, *Practical learned*, and *No Doob* denote direct hard sampling, exact h_η Euler sampling, practical conditioning with the analytic score, practical conditioning with the learned score, and the learned-score no-Doob control, respectively. Exact Euler remains near the finite-sample reference, while practical DoobCond with a learned score remains stable and consistently outperforms no Doob. Error bars show the population standard deviation over five seeds.

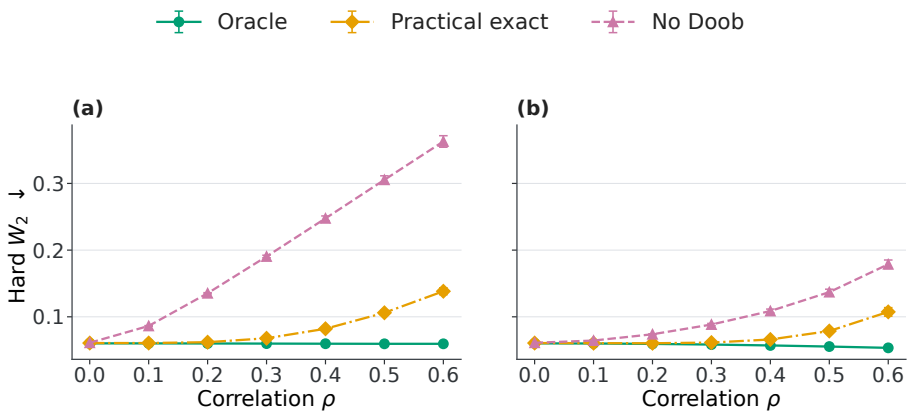


Figure 7: Effect of dependence strength and missingness geometry with the exact unconditional score. Panels (a) and (b) use uniformly selected point missingness and contiguous-block missingness, respectively; the vertical axis is *Hard* W_2. *Oracle* uses the full observed-posterior precision, *Practical exact* is the diagonal-precision practical sampler, and *No Doob* removes the conditioning. Practical exact improves over no Doob at the tested positive correlations, while its gap to oracle depends on the missingness geometry. Error bars show the population standard deviation over five seeds.

Results. Figure 6 shows that exact Doob sampling continues to track the finite-sample reference across the tested dimensions. Practical DOOBCOND with a learned score remains stable through $D = 160$ and consistently improves over the corresponding no-Doob sampler. The exact-score practical curve increases only gradually, indicating that the finite-step conditioning remains well behaved under the specified sample-to-dimension scaling rule.

C.4 DEPENDENCE STRENGTH AND MISSINGNESS GEOMETRY

Question and design. We isolate the diagonal observed-posterior precision in (9) by using the exact unconditional score throughout. We vary $\rho \in \{0, 0.1, \ldots, 0.6\}$ under the point and block missingness distributions defined in Subsection C.1. The *Oracle precision* replaces Λ_t with the full observed-posterior precision $(V_{0|t}^{\mathrm{obs,obs}} + \eta^2 I_{D_{\mathrm{obs}}})^{-1}$ within the same practical blockwise sampler. *Diagonal precision* uses Λ_t from (9), and *No Doob* omits conditioning. All three use the analytic unconditional score, the same terminal initialization, and the same observed-coordinate refresh. Hence, the oracle curve isolates precision-approximation error rather than representing the full-state exact Doob process.

Results. At $\rho = 0$, observed and missing coordinates are independent, and all three variants coincide at the Monte Carlo scale. As dependence increases, the diagonal approximation gradually departs from oracle precision but consistently outperforms no Doob for both missingness geometries through $\rho = 0.6$. The curves for point and block missingness show that the approximation error depends jointly on correlation strength and missing-coordinate geometry. These results support diagonal precision in the tested moderate-correlation regime while identifying structured precision as a potential extension under stronger dependence.

C.5 MISSINGNESS DISTRIBUTION SHIFT

Question and design. We test whether separating unconditional score learning from sample-specific conditioning reduces sensitivity to missingness distribution shift between the training-time artificial-mask distribution and the test-time missingness distribution. Let q_{block} and q_{point} be the missingness distributions defined in Subsection C.1. We fix $D = 40$, $\rho = 0.6$, and $k = 10$, and evaluate $q_\lambda := (0.99 - 0.98\lambda)q_{\mathrm{block}} + (0.01 + 0.98\lambda)q_{\mathrm{point}}$ for $\lambda \in \{0, 0.1, \ldots, 1\}$. Each primitive missingness distribution is represented by a fixed bank of 100 missingness patterns shared across methods, so λ changes their sampling frequencies over common empirical support. Here, q_λ is the test-time missingness distribution.

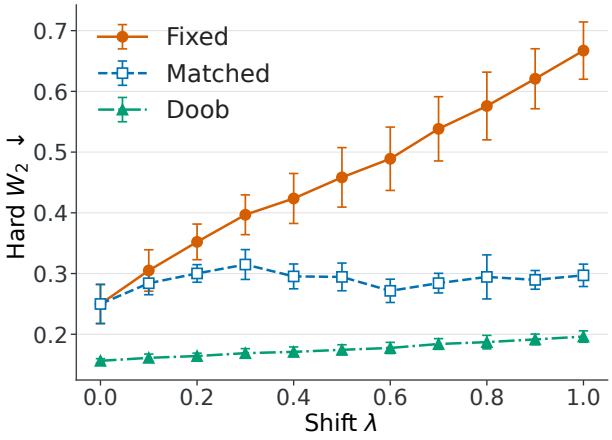


Figure 8: Posterior sampling under missingness distribution shift between the training-time artificial-mask distribution and the test-time missingness distribution. The vertical axis is *Hard* W_2. *Fixed* reuses a conditional score trained with $q_{\text{art}} = q_0$; *Matched* retrains with $q_{\text{art}} = q_\lambda$; and *Doob* is *Unconditional + Doob*, which reuses one unconditional score and conditions it at test time. Doob remains most accurate across the entire shift path without retraining for individual test-time missingness distributions. Error bars show the population standard deviation over five seeds.

The *Fixed* conditional score uses $q_{\text{art}} = q_0$ as its training-time artificial-mask distribution and is reused for every q_λ. The *Matched* conditional score is retrained with $q_{\text{art}} = q_\lambda$. The *Unconditional + Doob* method trains one unconditional score per seed without a training-time artificial mask distribution, reuses it for all λ, and applies DOOBCOND for sample-specific conditioning. For posterior sampling, the two conditional-score methods start from the exact forward-noised hard conditional at T and evolve only the missing coordinates by Euler–Maruyama. *Unconditional + Doob* instead starts the missing block from the standard terminal Gaussian, refreshes the observed coordinates using (7), and uses the learned unconditional score with the diagonal conditioning term in (11).

For the conditional-score comparison, we report the time-averaged excess risk

$$\overline{\mathcal{E}}(s;q) := \mathbb{E}_{t\sim\text{Unif}[0.001,1]}\left[\mathcal{E}_t(s;q)\right], \qquad \lambda(t) = \sigma_t^2,$$

where $\mathcal{E}_t$ is defined in (15). This is the expected squared deviation from the analytic Gaussian conditional score, summed over missing coordinates after removing the irreducible denoising variance; its Monte Carlo estimate uses 10,000 examples per seed. Because an unconditional score estimates a different score object, Figure 2 compares only *Fixed* and *Matched*; posterior *Hard* W_2 compares all three methods.

Results. The fixed conditional model's excess risk increases monotonically with shift severity, whereas matched retraining remains stable. The contrast with matched retraining points to training–test missingness shift as a substantial source of the degradation. The posterior-sampling comparison in Figure 8 shows the same pattern: the fixed conditional sampler degrades sharply, matched retraining remains comparatively stable, and *Unconditional + Doob* is most accurate at every shift level while requiring no λ-specific retraining. Thus, unconditional score learning followed by inference-time conditioning substantially mitigates the missingness distribution shift between the training-time artificial mask distribution and the test-time missingness distribution studied in Section 4.1.

C.6 SENSITIVITY TO ENDPOINT SMOOTHING

Question and design. We examine how the relaxed endpoint constraint affects both sampling accuracy and fidelity to the hard conditional target. We vary $\eta \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ while retaining the default setting in Table 4. The three panels report *Relaxed* W_2, the analytic smoothing bias between the relaxed and hard targets, and *Hard* W_2, respectively. *Direct relaxed* samples from (23). *Exact transformed path* starts from the exact transformed terminal law and samples every reverse Gaussian transition analytically, avoiding confounding small η with uniform-step Euler resolution. The two practical samplers are defined in Subsection C.2, and the learned-score no-Doob reference is independent of η. Within each seed, all η values share the same conditions

and sampling streams. The middle-panel smoothing bias is the normalized Gaussian W_2 between the relaxed and hard conditional targets, $W_2\left(\mathcal{N}(\mu^{\eta}_{u|o}, V^{\eta}_{u|o}), \mathcal{N}(\mu_{u|o}, V_{u|o})\right) / \sqrt{\mathrm{tr}(V_{u|o})}$.

Results. Figure 9 shows that small endpoint smoothing preserves the hard conditional target, whereas large η introduces substantial smoothing bias. We therefore use a small but nonvanishing η in subsequent implementations, balancing fidelity to the hard constraint against numerical stability.

Together, the controlled studies validate the Gaussian conditioning identity, quantify the approximation introduced by practical DOOBCOND, and characterize its tested operating regime. The conditioning remains effective with learned scores, stable under the dimension-scaling protocol, and beneficial under moderate dependence, while conditioning an unconditional backbone substantially reduces sensitivity to missingness distribution shift between the training-time artificial-mask distribution and the test-time missingness distribution. A small but nonvanishing endpoint smoothing scale further avoids excessive smoothing bias while preserving numerical stability. The real-data experiments complement this mechanism-level analysis by evaluating the complete framework on practical imputation tasks.

D TABULAR EXPERIMENTAL DETAILS

D.1 PROTOCOL AND METRICS

We evaluate nine tabular datasets under 30% MCAR, MAR, and MNAR missingness over ten seeds, using the fixed train–test splits and missingness laws of DiffPuter (Zhang et al., 2025). For each setting, the training and test partitions receive missingness generated from the same distribution, so the training inputs are already incomplete. Several competitors additionally mask observed entries within these incomplete training inputs to construct their training targets. Numerical features are standardized with training-partition statistics, categorical features use the DiffPuter binary encoding, and the prediction target is not imputed.

Numerical MAE and RMSE are micro-averages over numerical cells designated as missing on the shared normalized scale. Categorical accuracy is computed after decoding each categorical value designated as missing and does not include numerical columns. Stochastic imputers return the mean of 20 samples. We report the mean and population standard deviation over seeds. Following DiffPuter, DOOBIMP raw-imputation results use the final DOOBFIT training iteration by default. In-sample (IS) evaluation uses designated missing entries in the training set, whose ground-truth

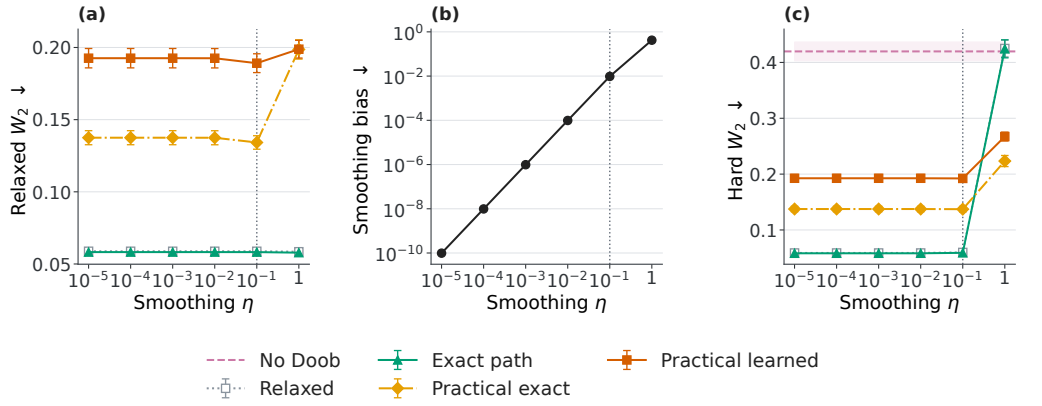


Figure 9: Sensitivity to endpoint smoothing. Panels (a)–(c) report *Relaxed* W_2, smoothing bias, and *Hard* W_2, respectively. *Relaxed* is direct relaxed sampling; *Exact path* is the analytic exact transformed path; *Practical exact* and *Practical learned* use the analytic and learned scores, respectively; and *No Doob* is the η-independent learned-score control without conditioning. The dotted line marks the default $\eta = 0.1$. Large η causes smoothing bias to dominate hard-target error. Error bars show the population standard deviation over five seeds.

values are retained for evaluation. Out-of-sample (OOS) evaluation uses designated missing entries in the held-out test set.

For downstream classification, we fit logistic regression with $C = 1$ and at most 5,000 iterations to each completed training set. We report AUROC for the binary Adult, Default, Magic, and Shoppers tasks, and macro-F1 for the multiclass Bean and Letter tasks.

D.2 Hyperparameter Setting

Following the DiffPuter protocol (Zhang et al., 2025), DoobImp uses one architecture and imputation configuration across all nine tabular datasets and three missingness mechanisms. Table 5 gives the complete setting, including the trained score stages, η, τ, W, J, and sampling budgets.

We use the implementations and critical hyperparameters reported by DiffPuter for its bundled baselines (Zhang et al., 2025), and the official RefiDiff implementation (Ahamed et al., 2026). These baselines include ReMasker (Du et al., 2024a), MissDiff (Ouyang et al., 2023), TabCSDI (Zheng & Charoenphakdee, 2022), MCFlow (Richardson et al., 2020), MIWAE (Mattei & Frellsen, 2019), and HyperImpute (Jarrett et al., 2022). Additional baselines include EM (Dempster et al., 1977), SoftImpute (Mazumder et al., 2010), GAIN (Yoon et al., 2018), MissForest (Stekhoven & Bühlmann, 2012), MICE (van Buuren & Groothuis-Oudshoorn, 2011), and MIRACLE (Kyono et al., 2021). All methods use the same missingness patterns, train-fitted normalization, and OOS evaluation; stochastic imputers use a 20-sample mean readout.

Table 5: Critical settings for learned and generative methods.

Method	Critical settings
DoobImp	DiffPuter MLP/EDM backbone; 10 trained score stages with fresh initialization; 10,001 maximum epochs; Adam 10^{-4}; batch 4,096; width 1,024; patience 500. The first stage uses the zero-padded observed-entry MissDiff loss. Imputation uses $\eta = 0.01$, $\tau = 0.25$, $W = 20$, $J = 5$, 50 reverse steps, and 20 samples. Clean-space conditioning updates are norm-clipped at 5 for numerical coordinates and 1 for categorical coordinates.
DiffPuter	10 fresh M-steps; 10,001 maximum epochs; Adam 10^{-4}; batch 4,096; width 1,024; scheduler factor 0.9 with patience 50; early-stop patience 500; 50 reverse steps; 20 samples.
RefiDiff	Official CustomDenoiser/Mamba refinement model; half-scale input; width 32; batch 8,192; 10,001 maximum epochs; Adam 10^{-4}; scheduler factor 0.9 with patience 40; early-stop patience 500; 50 reverse steps; 20 samples.
ReMasker	Embedding width 32; encoder/decoder depths 6/4; 4 heads; input missingness ratio 0.5; 600 epochs; batch 64; learning rate 10^{-3}; 40 warmup epochs; weight decay 0.05.
MissDiff	Official MissDiff-SDE configuration; 10,001 maximum epochs; patience 1,000; batch 4,096; width 1,024; Adam 10^{-4}; scheduler factor 0.9 with patience 20; 50 reverse steps; 20 samples.
TabCSDI	100 epochs; batch 256; learning rate 5×10^{-4}; 4 diffusion layers; 128 channels; 4 heads; embedding width 128; 100 quadratic-schedule diffusion steps; 20 samples.
MCFlow	3 flow layers; 300 epochs; batch 64; learning rate 10^{-4}; dropout-imputation rate 0.2.
MIWAE	500 epochs; batch 256; one-dimensional latent state and one hidden layer; Adam 10^{-3}; 20 training importance samples; importance-weighted mean of 20 imputation samples.
HyperImpute	HyperImputePlugin defaults for candidate generation and model selection.

Table 6: Critical settings for additional baselines.

Method	Critical settings
EM	Gaussian EM with at most 500 iterations and convergence tolerance 10^{-8}.
SoftImpute	At most 1,000 iterations; tolerance 10^{-5}; maximum rank 2; three shrinkage candidates.
GAIN	Batch 128; 100 epochs; hint rate 0.8; reconstruction weight 10.
MissForest	Iterative imputation with a 10-tree random forest, depth 3, bootstrap sampling, and at most 100 iterations.
MICE	Iterative imputation with posterior sampling, at most 100 iterations, and tolerance 0.001.
MIRACLE	Official MIRACLE plugin defaults used by DiffPuter.
Mean	Per-column arithmetic mean fitted on the training partition.
Median	Per-column median fitted on the training partition.
Most-Frequent	Per-column mode fitted on the training partition.

D.3 Complete Results

We compare DoobImp with DiffPuter (Zhang et al., 2025), RefiDiff (Ahamed et al., 2026), ReMasker (Du et al., 2024a), MissDiff (Ouyang et al., 2023), TabCSDI (Zheng & Charoenphakdee, 2022), MCFlow (Richardson et al., 2020), MIWAE (Mattei & Frellsen, 2019), HyperImpute (Jarrett et al., 2022), EM (Dempster et al., 1977), SoftImpute (Mazumder et al., 2010), GAIN (Yoon et al.,

2018), MissForest (Stekhoven & Bühlmann, 2012), MICE (van Buuren & Groothuis-Oudshoorn, 2011), and MIRACLE (Kyono et al., 2021), together with mean, median, and most-frequent imputation.

D.3.1 Aggregate Raw Imputation Results

Table 7: Aggregate OOS numerical imputation performance (↓). The MAE and RMSE columns report mean ± population standard deviation; the final column reports average rank based on OOS MAE across the 27 dataset–mechanism settings. Best mean is bold; second-best mean is underlined.

	MAE			RMSE			
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR	Average rank
DoobImp	**0.293±0.001**	**0.312±0.011**	**0.307±0.006**	**0.593±0.005**	**0.630±0.019**	**0.628±0.017**	**1.185**
RefiDiff	0.334±0.001	0.349±0.011	0.343±0.006	0.649±0.005	0.685±0.018	0.679±0.020	2.963
DiffPuter	0.316±0.004	0.372±0.015	0.351±0.012	0.628±0.020	0.739±0.032	0.733±0.072	3.333
HyperImpute	0.357±0.005	0.429±0.020	0.388±0.012	1.536±1.057	1.459±1.465	1.508±1.008	5.444
EM	0.391±0.001	0.411±0.012	0.408±0.005	0.669±0.005	0.715±0.021	0.710±0.015	5.519
ReMasker	0.421±0.043	0.419±0.046	0.400±0.024	0.700±0.052	0.716±0.053	0.691±0.020	6.259
MissForest	0.430±0.004	0.457±0.013	0.447±0.008	0.744±0.008	0.812±0.016	0.786±0.019	8.111
TabCSDI	0.459±0.007	0.467±0.013	0.470±0.009	0.790±0.007	0.808±0.020	0.816±0.016	8.630
MCFlow	0.483±0.004	0.510±0.017	0.506±0.009	0.809±0.006	0.857±0.029	0.856±0.024	9.519
Median	0.609±0.001	0.620±0.018	0.634±0.013	1.018±0.005	1.040±0.030	1.062±0.029	10.778
MissDiff	0.545±0.046	0.608±0.132	0.628±0.085	4.698±5.309	8.034±8.387	12.208±23.564	11.481
MIRACLE	0.558±0.016	0.646±0.029	0.570±0.010	1.041±0.039	1.126±0.054	1.070±0.030	11.704
SoftImpute	0.533±0.003	0.559±0.021	0.558±0.011	0.860±0.011	0.916±0.045	0.922±0.030	11.778
MIWAE	0.635±0.002	0.646±0.018	0.659±0.012	1.021±0.005	1.044±0.031	1.064±0.028	12.963
Mean	0.657±0.001	0.674±0.009	0.682±0.009	0.985±0.005	1.017±0.029	1.032±0.026	14.111
MICE	0.663±0.002	0.686±0.041	0.667±0.003	0.994±0.078	4.075±8.421	1.246±0.830	14.704
Most-Frequent	0.869±0.024	0.888±0.051	0.888±0.021	1.387±0.041	1.414±0.072	1.412±0.031	15.889
GAIN	1.032±0.037	1.026±0.091	1.057±0.024	1.450±0.081	1.454±0.151	1.515±0.052	16.630

Table 8: Aggregate OOS categorical accuracy (↑). Best mean is bold; second-best mean is underlined.

Method	MCAR	MAR	MNAR
DoobImp	**0.631±0.002**	**0.642±0.054**	**0.630±0.005**
RefiDiff	0.595±0.005	0.593±0.043	0.596±0.005
DiffPuter	0.620±0.004	0.618±0.051	0.613±0.008
EM	0.582±0.002	0.592±0.044	0.583±0.003
ReMasker	0.621±0.015	0.627±0.048	0.624±0.014
HyperImpute	0.607±0.015	0.609±0.057	0.614±0.014
MissForest	0.556±0.008	0.547±0.035	0.556±0.008
TabCSDI	0.490±0.005	0.492±0.022	0.492±0.006
MCFlow	0.536±0.003	0.546±0.026	0.536±0.006
SoftImpute	0.436±0.001	0.432±0.015	0.435±0.009
Median	0.438±0.001	0.444±0.017	0.441±0.004
MIRACLE	0.546±0.005	0.529±0.036	0.546±0.006
MissDiff	0.576±0.003	0.587±0.046	0.577±0.004
Mean	0.438±0.001	0.444±0.017	0.441±0.004
MIWAE	0.428±0.002	0.428±0.018	0.429±0.002
MICE	0.455±0.002	0.460±0.026	0.457±0.003
GAIN	0.444±0.002	0.448±0.018	0.449±0.003
Most-Frequent	0.438±0.001	0.444±0.017	0.441±0.004

D.3.2 AGGREGATE DOWNSTREAM RESULTS

Table 9: Aggregate downstream performance (↑). Binary datasets report AUROC and multiclass datasets report macro-F1. Entries are mean ± population standard deviation. Best mean is bold; second-best mean is underlined.

	Binary AUROC			Multiclass macro-F1		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	<u>0.818±0.002</u>	<u>0.814±0.006</u>	<u>0.817±0.003</u>	**0.827±0.002**	**0.822±0.010**	**0.825±0.003**
RefiDiff	0.814±0.002	0.809±0.007	0.813±0.004	0.796±0.002	0.791±0.010	0.795±0.005
DiffPuter	**0.819±0.002**	**0.816±0.006**	**0.819±0.004**	<u>0.822±0.002</u>	<u>0.811±0.010</u>	<u>0.818±0.004</u>
EM	0.812±0.002	0.807±0.008	0.810±0.006	0.754±0.003	0.762±0.010	0.757±0.004
ReMasker	0.814±0.005	0.813±0.007	0.816±0.003	0.805±0.003	0.807±0.008	0.806±0.004
HyperImpute	0.815±0.002	0.810±0.008	0.814±0.005	0.789±0.004	0.784±0.009	0.790±0.005
MissForest	0.807±0.004	0.802±0.008	0.808±0.005	0.731±0.003	0.747±0.011	0.739±0.006
TabCSDI	0.541±0.016	0.551±0.013	0.546±0.017	0.135±0.033	0.156±0.039	0.146±0.036
MCFlow	0.764±0.003	0.764±0.007	0.765±0.008	0.654±0.011	0.666±0.017	0.663±0.017
SoftImpute	0.801±0.002	0.798±0.010	0.797±0.006	0.726±0.003	0.747±0.010	0.733±0.006
Median	0.797±0.002	0.800±0.010	0.797±0.004	0.688±0.003	0.732±0.009	0.704±0.006
MIRACLE	0.794±0.004	0.784±0.014	0.792±0.004	0.725±0.012	0.711±0.026	0.721±0.012
MissDiff	0.809±0.002	0.808±0.007	0.809±0.006	0.758±0.006	0.776±0.015	0.764±0.005
Mean	0.803±0.002	0.804±0.009	0.803±0.004	0.692±0.002	0.734±0.010	0.707±0.007
MIWAE	0.795±0.002	0.798±0.011	0.795±0.004	0.678±0.002	0.726±0.011	0.695±0.007
MICE	0.779±0.003	0.781±0.013	0.777±0.005	0.692±0.004	0.712±0.016	0.696±0.005
GAIN	0.789±0.002	0.797±0.012	0.790±0.003	0.682±0.004	0.732±0.010	0.699±0.009
Most-Frequent	0.784±0.002	0.791±0.013	0.786±0.004	0.665±0.004	0.728±0.009	0.687±0.006

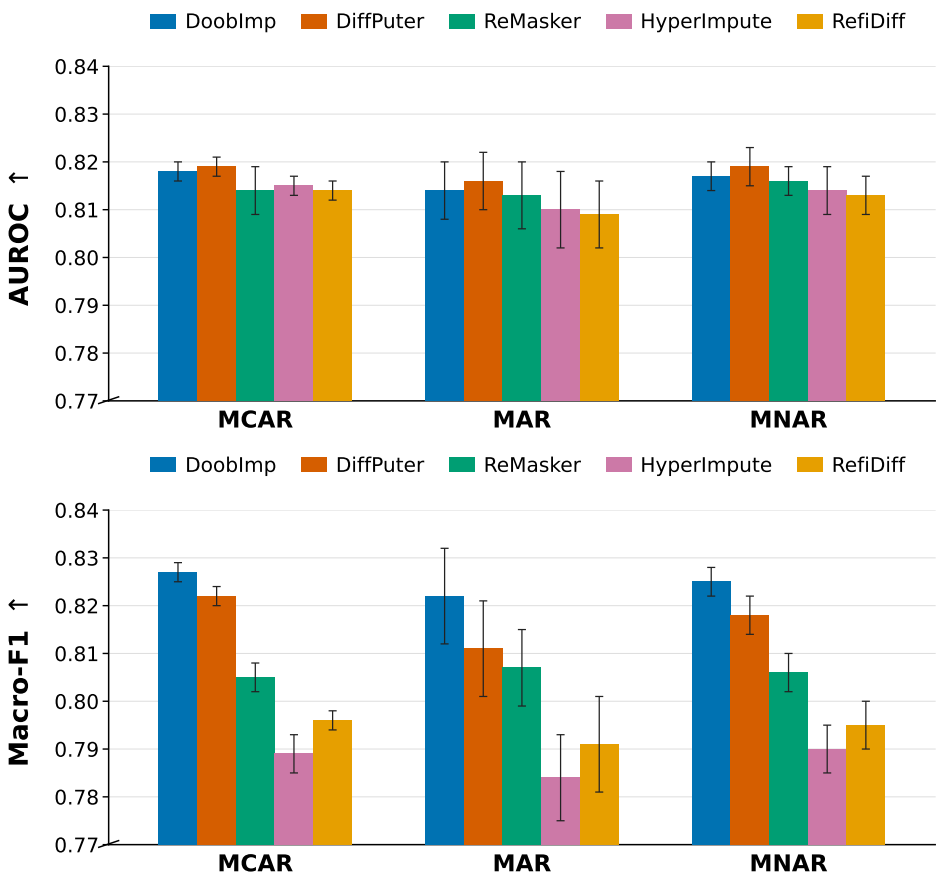


Figure 10: Top-five downstream performance (↑). The upper panel reports binary AUROC and the lower panel reports multiclass macro-F1. Error bars show population standard deviations over ten seeds.

D.3.3 Complete Numerical Imputation Results

All entries are mean ± population standard deviation across seeds.

Table 10: IS MAE and RMSE on Bean (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	0.112±0.004	<u>0.138±0.027</u>	<u>0.116±0.006</u>	0.317±0.005	0.352±0.052	0.318±0.016
RefiDiff	0.117±0.002	0.149±0.028	0.123±0.006	0.335±0.007	0.387±0.057	0.341±0.017
DiffPuter	0.117±0.003	0.207±0.043	0.174±0.026	0.326±0.008	0.566±0.117	0.515±0.100
EM	**0.106±0.001**	**0.123±0.020**	**0.108±0.005**	**0.283±0.004**	**0.308±0.049**	**0.287±0.015**
ReMasker	0.133±0.009	0.152±0.026	0.139±0.010	0.315±0.003	<u>0.344±0.051</u>	<u>0.317±0.019</u>
HyperImpute	0.114±0.003	0.311±0.174	0.167±0.112	<u>0.293±0.007</u>	0.489±0.166	0.359±0.109
MissForest	0.255±0.005	0.268±0.021	0.258±0.008	0.433±0.010	0.471±0.044	0.435±0.018
TabCSDI	0.208±0.011	0.225±0.025	0.207±0.012	0.418±0.012	0.443±0.043	0.412±0.019
MCFlow	0.227±0.006	0.285±0.041	0.255±0.017	0.429±0.010	0.539±0.067	0.485±0.035
SoftImpute	0.280±0.001	0.298±0.025	0.297±0.012	0.476±0.004	0.513±0.043	0.499±0.022
Median	0.721±0.002	0.786±0.047	0.778±0.043	1.022±0.004	1.098±0.090	1.106±0.086
MIRACLE	<u>0.111±0.004</u>	0.207±0.034	0.149±0.026	0.306±0.009	0.562±0.095	0.439±0.087
MissDiff	0.456±0.013	0.283±0.031	0.457±0.076	0.844±0.222	0.643±0.160	0.783±0.139
Mean	0.744±0.002	0.812±0.029	0.801±0.030	1.001±0.003	1.089±0.070	1.089±0.071
MIWAE	0.747±0.003	0.804±0.041	0.799±0.039	1.038±0.004	1.107±0.081	1.115±0.079
MICE	0.174±0.003	0.177±0.026	0.171±0.005	0.413±0.005	0.424±0.066	0.413±0.009
GAIN	0.850±0.022	0.945±0.077	0.898±0.045	1.114±0.036	1.234±0.086	1.182±0.059
Most-Frequent	1.291±0.073	1.325±0.174	1.334±0.101	1.696±0.082	1.781±0.280	1.754±0.105

Table 11: OOS MAE and RMSE on Bean (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	<u>0.111±0.004</u>	<u>0.137±0.027</u>	<u>0.115±0.006</u>	0.306±0.008	0.339±0.056	0.310±0.019
RefiDiff	0.122±0.002	0.158±0.030	0.131±0.007	0.335±0.006	0.395±0.063	0.349±0.018
DiffPuter	0.117±0.004	0.210±0.045	0.177±0.028	0.316±0.010	0.572±0.127	0.523±0.107
EM	**0.106±0.001**	**0.124±0.021**	**0.109±0.006**	**0.280±0.003**	**0.306±0.054**	**0.287±0.016**
ReMasker	0.133±0.008	0.152±0.027	0.139±0.009	0.307±0.006	<u>0.335±0.059</u>	<u>0.310±0.018</u>
HyperImpute	0.120±0.004	0.220±0.100	0.190±0.113	<u>0.302±0.006</u>	0.422±0.081	0.380±0.120
MissForest	0.255±0.004	0.269±0.024	0.258±0.007	0.427±0.010	0.463±0.048	0.427±0.016
TabCSDI	0.206±0.011	0.225±0.026	0.206±0.012	0.404±0.011	0.429±0.044	0.401±0.014
MCFlow	0.227±0.007	0.286±0.042	0.257±0.017	0.418±0.010	0.529±0.075	0.481±0.035
SoftImpute	0.276±0.002	0.297±0.028	0.294±0.013	0.458±0.003	0.497±0.046	0.487±0.022
Median	0.719±0.004	0.784±0.053	0.778±0.046	1.019±0.007	1.094±0.100	1.108±0.091
MIRACLE	0.141±0.003	0.201±0.033	0.174±0.028	0.328±0.011	0.455±0.072	0.411±0.082
MissDiff	0.451±0.013	0.279±0.034	0.453±0.075	0.710±0.080	0.488±0.058	0.688±0.155
Mean	0.738±0.004	0.808±0.034	0.798±0.034	0.994±0.006	1.083±0.080	1.090±0.078
MIWAE	0.744±0.004	0.801±0.046	0.798±0.041	1.032±0.007	1.100±0.091	1.115±0.084
MICE	0.174±0.002	0.177±0.025	0.172±0.006	0.415±0.004	0.419±0.067	0.413±0.011
GAIN	0.847±0.020	0.940±0.074	0.897±0.045	1.112±0.033	1.231±0.087	1.183±0.064
Most-Frequent	1.296±0.076	1.330±0.176	1.341±0.104	1.696±0.085	1.782±0.284	1.759±0.111

Table 12: IS MAE and RMSE on California (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	0.322±0.002	0.366±0.064	0.345±0.024	0.576±0.010	0.636±0.080	0.613±0.042
RefiDiff	**0.315±0.003**	**0.325±0.068**	**0.334±0.017**	0.568±0.012	**0.583±0.079**	0.597±0.032
DiffPuter	0.353±0.005	0.407±0.061	0.391±0.037	0.634±0.104	0.717±0.081	0.680±0.085
EM	0.415±0.002	0.455±0.060	0.439±0.021	0.647±0.010	0.704±0.062	0.684±0.036
ReMasker	0.345±0.029	0.354±0.050	0.356±0.023	**0.565±0.033**	0.584±0.062	**0.583±0.036**
HyperImpute	0.321±0.011	0.436±0.091	0.341±0.014	0.581±0.011	0.696±0.085	0.615±0.029
MissForest	0.429±0.007	0.451±0.063	0.460±0.031	0.679±0.013	0.723±0.080	0.723±0.056
TabCSDI	0.417±0.005	0.458±0.063	0.436±0.022	0.674±0.011	0.726±0.070	0.701±0.037
MCFlow	0.593±0.017	0.631±0.053	0.599±0.033	0.882±0.025	0.931±0.089	0.901±0.060
SoftImpute	0.522±0.004	0.581±0.067	0.555±0.026	0.810±0.007	0.885±0.122	0.862±0.040
Median	0.719±0.004	0.730±0.054	0.738±0.035	1.061±0.011	1.063±0.076	1.082±0.062
MIRACLE	0.478±0.011	0.523±0.058	0.480±0.028	0.847±0.028	0.875±0.094	0.852±0.063
MissDiff	0.483±0.028	0.908±0.694	0.577±0.114	2.888±1.817	40.405±74.683	7.257±7.206
Mean	0.752±0.003	0.769±0.038	0.773±0.024	1.008±0.011	1.034±0.056	1.034±0.053
MIWAE	0.750±0.003	0.761±0.047	0.768±0.027	1.048±0.010	1.065±0.068	1.070±0.058
MICE	0.618±0.006	0.656±0.059	0.635±0.017	0.911±0.008	0.958±0.065	0.937±0.025
GAIN	2.563±0.367	2.598±0.708	2.762±0.261	3.596±0.757	3.713±1.176	4.065±0.486
Most-Frequent	0.925±0.173	0.888±0.106	0.885±0.034	1.411±0.312	1.313±0.135	1.324±0.051

Table 13: OOS MAE and RMSE on California (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.321±0.003**	0.363±0.065	**0.343±0.022**	0.570±0.009	0.627±0.081	0.613±0.041
RefiDiff	0.326±0.003	**0.340±0.072**	0.344±0.018	0.578±0.008	0.605±0.085	0.621±0.037
DiffPuter	0.351±0.004	0.405±0.061	0.390±0.037	0.611±0.029	0.728±0.105	0.681±0.089
EM	0.415±0.003	0.453±0.062	0.438±0.020	0.640±0.005	0.696±0.064	0.681±0.037
ReMasker	0.343±0.029	0.352±0.050	0.352±0.023	**0.556±0.028**	**0.578±0.062**	**0.582±0.040**
HyperImpute	0.324±0.010	0.443±0.074	0.347±0.013	0.584±0.013	0.713±0.077	0.627±0.030
MissForest	0.429±0.006	0.449±0.062	0.461±0.035	0.673±0.011	0.718±0.079	0.731±0.064
TabCSDI	0.419±0.005	0.458±0.063	0.437±0.021	0.673±0.010	0.723±0.070	0.709±0.038
MCFlow	0.593±0.020	0.629±0.053	0.601±0.033	0.880±0.027	0.930±0.088	0.912±0.061
SoftImpute	0.515±0.004	0.560±0.057	0.540±0.017	0.781±0.007	0.834±0.083	0.829±0.029
Median	0.717±0.005	0.730±0.055	0.739±0.033	1.056±0.016	1.063±0.080	1.094±0.064
MIRACLE	0.492±0.021	0.523±0.043	0.486±0.028	0.806±0.040	0.829±0.043	0.823±0.041
MissDiff	0.513±0.124	0.980±0.885	0.545±0.123	4.618±8.609	43.334±78.966	3.507±5.653
Mean	0.755±0.004	0.770±0.038	0.777±0.022	1.007±0.015	1.035±0.059	1.049±0.054
MIWAE	0.750±0.005	0.762±0.049	0.772±0.027	1.046±0.016	1.066±0.072	1.084±0.060
MICE	0.620±0.005	0.656±0.061	0.638±0.018	0.910±0.007	0.957±0.067	0.941±0.026
GAIN	2.569±0.368	2.603±0.712	2.764±0.268	3.606±0.759	3.721±1.178	4.070±0.490
Most-Frequent	0.916±0.173	0.884±0.106	0.884±0.035	1.401±0.317	1.309±0.135	1.332±0.058

Table 14: IS MAE and RMSE on Gesture (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.262±0.003**	**0.279±0.014**	**0.289±0.012**	**0.587±0.012**	**0.622±0.028**	**0.656±0.026**
RefiDiff	0.295±0.004	0.299±0.019	0.317±0.017	0.671±0.017	0.692±0.036	0.732±0.035
DiffPuter	0.323±0.003	0.383±0.020	0.378±0.019	0.757±0.021	0.884±0.029	0.879±0.041
EM	0.341±0.003	0.358±0.020	0.369±0.016	0.669±0.009	0.710±0.040	0.736±0.033
ReMasker	0.342±0.022	0.344±0.015	0.346±0.016	0.675±0.023	0.705±0.024	0.712±0.022
HyperImpute	0.365±0.014	0.374±0.024	0.378±0.019	0.707±0.018	0.723±0.042	0.743±0.032
MissForest	0.363±0.003	0.367±0.014	0.387±0.017	0.735±0.012	0.763±0.028	0.803±0.039
TabCSDI	0.621±0.060	0.634±0.073	0.644±0.060	1.046±0.057	1.065±0.066	1.094±0.058
MCFlow	0.490±0.004	0.517±0.018	0.526±0.018	0.919±0.013	0.973±0.028	0.991±0.030
SoftImpute	0.508±0.004	0.551±0.035	0.552±0.020	0.941±0.010	1.011±0.055	1.020±0.033
Median	0.519±0.003	0.530±0.021	0.550±0.017	1.011±0.010	1.040±0.035	1.079±0.030
MIRACLE	0.704±0.044	0.683±0.043	0.692±0.057	1.209±0.050	1.159±0.047	1.211±0.066
MissDiff	0.904±0.225	1.492±1.376	1.950±1.870	9.720±5.341	40.226±55.878	52.746±93.697
Mean	0.540±0.003	0.556±0.019	0.568±0.014	0.993±0.010	1.022±0.029	1.058±0.027
MIWAE	0.550±0.003	0.561±0.018	0.579±0.015	1.016±0.011	1.044±0.032	1.082±0.029
MICE	0.652±0.001	0.656±0.011	0.663±0.009	0.933±0.005	0.954±0.026	0.973±0.023
GAIN	0.674±0.011	0.656±0.015	0.678±0.019	1.085±0.007	1.082±0.014	1.123±0.020
Most-Frequent	0.812±0.082	0.874±0.136	0.833±0.073	1.448±0.116	1.517±0.179	1.475±0.109

Table 15: OOS MAE and RMSE on Gesture (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.256±0.003**	**0.269±0.013**	**0.277±0.011**	**0.549±0.014**	**0.571±0.027**	**0.596±0.028**
RefiDiff	0.306±0.005	0.304±0.017	0.320±0.014	0.658±0.019	0.658±0.030	0.685±0.030
DiffPuter	0.292±0.006	0.344±0.015	0.334±0.015	0.703±0.155	0.763±0.034	0.757±0.065
EM	0.336±0.003	0.349±0.018	0.358±0.012	0.641±0.012	0.664±0.033	0.688±0.026
ReMasker	0.337±0.022	0.334±0.014	0.334±0.015	0.645±0.023	0.652±0.025	0.654±0.026
HyperImpute	0.372±0.006	0.383±0.019	0.381±0.020	0.688±0.016	0.699±0.031	0.712±0.029
MissForest	0.355±0.003	0.353±0.013	0.373±0.014	0.700±0.013	0.701±0.023	0.735±0.031
TabCSDI	0.614±0.057	0.621±0.067	0.631±0.057	1.020±0.051	1.023±0.058	1.047±0.052
MCFlow	0.483±0.005	0.506±0.017	0.513±0.015	0.888±0.012	0.925±0.018	0.938±0.023
SoftImpute	0.506±0.005	0.536±0.035	0.542±0.017	0.906±0.012	0.952±0.044	0.968±0.024
Median	0.509±0.004	0.516±0.019	0.534±0.014	0.975±0.013	0.989±0.028	1.022±0.026
MIRACLE	0.567±0.013	0.602±0.050	0.611±0.035	1.124±0.023	1.129±0.073	1.184±0.064
MissDiff	0.954±0.363	1.141±1.005	1.192±0.380	10.168±7.037	14.615±24.491	17.293±14.721
Mean	0.530±0.004	0.541±0.017	0.553±0.011	0.957±0.013	0.972±0.023	1.002±0.022
MIWAE	0.540±0.004	0.547±0.016	0.564±0.013	0.980±0.013	0.994±0.026	1.026±0.025
MICE	0.650±0.003	0.648±0.014	0.655±0.007	0.918±0.007	0.921±0.022	0.941±0.019
GAIN	0.664±0.011	0.643±0.017	0.664±0.020	1.053±0.013	1.036±0.017	1.071±0.026
Most-Frequent	0.805±0.080	0.867±0.140	0.821±0.073	1.423±0.116	1.488±0.187	1.436±0.113

Table 16: IS MAE and RMSE on Letter (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.274±0.003**	**0.302±0.015**	**0.285±0.008**	**0.428±0.006**	**0.471±0.025**	**0.447±0.014**
RefiDiff	0.353±0.002	0.373±0.015	0.355±0.007	0.523±0.003	0.550±0.024	0.524±0.013
DiffPuter	0.301±0.003	0.391±0.014	0.337±0.006	0.467±0.005	0.587±0.016	0.520±0.010
EM	0.548±0.002	0.568±0.021	0.556±0.008	0.744±0.002	0.771±0.025	0.756±0.012
ReMasker	0.363±0.004	0.373±0.007	0.365±0.007	0.507±0.007	0.523±0.010	0.510±0.011
HyperImpute	0.374±0.008	0.587±0.102	0.447±0.016	0.543±0.014	0.802±0.132	0.628±0.020
MissForest	0.586±0.003	0.602±0.023	0.590±0.012	0.805±0.004	0.828±0.027	0.809±0.014
TabCSDI	0.585±0.007	0.594±0.019	0.587±0.013	0.793±0.010	0.809±0.024	0.797±0.018
MCFlow	0.612±0.005	0.639±0.024	0.616±0.012	0.820±0.006	0.852±0.027	0.825±0.015
SoftImpute	0.646±0.001	0.666±0.021	0.652±0.006	0.862±0.002	0.883±0.024	0.869±0.009
Median	0.749±0.002	0.760±0.018	0.757±0.015	1.012±0.002	1.030±0.019	1.026±0.018
MIRACLE	0.634±0.047	0.748±0.081	0.684±0.044	0.953±0.076	1.080±0.116	1.036±0.060
MissDiff	0.473±0.024	0.465±0.028	0.471±0.016	0.650±0.031	0.652±0.040	0.652±0.022
Mean	0.767±0.002	0.779±0.017	0.776±0.012	0.999±0.002	1.013±0.016	1.010±0.014
MIWAE	0.791±0.002	0.801±0.015	0.798±0.012	1.027±0.002	1.040±0.015	1.037±0.013
MICE	0.807±0.002	0.825±0.031	0.812±0.004	1.053±0.003	1.075±0.035	1.060±0.006
GAIN	0.786±0.002	0.789±0.017	0.792±0.015	1.014±0.002	1.021±0.015	1.022±0.017
Most-Frequent	0.767±0.002	0.799±0.054	0.790±0.026	1.052±0.002	1.094±0.067	1.083±0.033

Table 17: OOS MAE and RMSE on Letter (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.273±0.003**	**0.301±0.014**	**0.283±0.008**	**0.426±0.005**	**0.469±0.024**	**0.444±0.013**
RefiDiff	0.376±0.003	0.399±0.014	0.377±0.006	0.552±0.005	0.583±0.022	0.553±0.011
DiffPuter	0.289±0.002	0.372±0.015	0.320±0.007	0.453±0.005	0.564±0.018	0.497±0.011
EM	0.553±0.003	0.575±0.020	0.562±0.009	0.752±0.004	0.781±0.023	0.765±0.014
ReMasker	0.364±0.004	0.374±0.007	0.366±0.007	0.509±0.007	0.526±0.009	0.511±0.011
HyperImpute	0.414±0.004	0.567±0.118	0.478±0.044	0.593±0.007	0.783±0.151	0.669±0.054
MissForest	0.590±0.003	0.608±0.020	0.594±0.013	0.811±0.004	0.837±0.024	0.815±0.017
TabCSDI	0.588±0.007	0.599±0.018	0.591±0.013	0.798±0.009	0.812±0.024	0.801±0.015
MCFlow	0.616±0.005	0.645±0.024	0.622±0.013	0.826±0.006	0.859±0.027	0.832±0.017
SoftImpute	0.649±0.003	0.669±0.021	0.654±0.005	0.868±0.003	0.888±0.024	0.873±0.011
Median	0.754±0.004	0.766±0.019	0.763±0.015	1.017±0.005	1.036±0.020	1.030±0.018
MIRACLE	0.704±0.065	0.794±0.051	0.703±0.057	1.063±0.105	1.139±0.070	1.062±0.096
MissDiff	0.475±0.024	0.467±0.029	0.473±0.016	0.653±0.030	0.655±0.040	0.655±0.021
Mean	0.771±0.004	0.785±0.018	0.780±0.012	1.004±0.005	1.019±0.018	1.014±0.014
MIWAE	0.794±0.003	0.805±0.017	0.802±0.012	1.030±0.004	1.044±0.017	1.039±0.014
MICE	0.811±0.006	0.831±0.030	0.815±0.006	1.058±0.007	1.084±0.034	1.065±0.007
GAIN	0.791±0.003	0.795±0.017	0.797±0.014	1.019±0.004	1.027±0.017	1.027±0.017
Most-Frequent	0.773±0.004	0.804±0.054	0.795±0.026	1.056±0.005	1.099±0.068	1.086±0.034

Table 18: IS MAE and RMSE on Magic (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.410±0.001**	**0.459±0.044**	**0.435±0.025**	**0.673±0.006**	**0.736±0.076**	**0.727±0.056**
RefiDiff	0.427±0.002	0.476±0.046	0.445±0.027	0.712±0.005	0.784±0.083	0.761±0.067
DiffPuter	0.446±0.003	0.554±0.043	0.500±0.031	0.705±0.003	0.831±0.071	0.791±0.060
EM	0.493±0.001	0.550±0.045	0.527±0.031	0.745±0.003	0.845±0.080	0.822±0.059
ReMasker	0.782±0.402	0.658±0.192	0.559±0.132	1.064±0.476	0.928±0.228	0.823±0.162
HyperImpute	0.423±0.003	0.502±0.066	0.492±0.062	0.697±0.010	0.801±0.110	0.781±0.075
MissForest	0.502±0.007	0.535±0.035	0.522±0.029	0.765±0.008	0.832±0.072	0.814±0.055
TabCSDI	0.501±0.003	0.539±0.040	0.524±0.024	0.772±0.009	0.815±0.059	0.819±0.045
MCFlow	0.521±0.005	0.563±0.042	0.551±0.031	0.817±0.010	0.874±0.073	0.880±0.065
SoftImpute	0.578±0.002	0.673±0.077	0.641±0.055	0.847±0.004	1.016±0.112	0.970±0.083
Median	0.723±0.002	0.753±0.047	0.758±0.037	1.023±0.002	1.044±0.052	1.070±0.053
MIRACLE	0.435±0.003	0.683±0.034	0.499±0.037	0.715±0.014	1.119±0.054	0.871±0.094
MissDiff	0.517±0.014	0.555±0.042	0.563±0.044	0.848±0.040	1.031±0.326	1.106±0.429
Mean	0.745±0.002	0.780±0.040	0.783±0.037	0.999±0.002	1.032±0.056	1.051±0.053
MIWAE	0.750±0.003	0.782±0.046	0.785±0.037	1.031±0.003	1.055±0.055	1.079±0.053
MICE	0.759±0.003	0.807±0.040	0.771±0.019	1.055±0.004	1.129±0.055	1.089±0.034
GAIN	1.277±0.066	1.190±0.179	1.286±0.112	1.642±0.067	1.529±0.281	1.636±0.139
Most-Frequent	0.942±0.035	1.014±0.134	0.999±0.086	1.292±0.040	1.365±0.157	1.358±0.094

Table 19: OOS MAE and RMSE on Magic (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.409±0.003**	**0.455±0.044**	**0.430±0.026**	**0.669±0.010**	**0.726±0.075**	**0.717±0.057**
RefiDiff	0.436±0.003	0.484±0.044	0.453±0.029	0.722±0.008	0.789±0.081	0.765±0.061
DiffPuter	0.441±0.003	0.547±0.044	0.490±0.032	0.697±0.009	0.820±0.071	0.777±0.061
EM	0.493±0.003	0.543±0.043	0.522±0.032	0.741±0.007	0.831±0.079	0.811±0.060
ReMasker	0.782±0.404	0.656±0.193	0.556±0.134	1.061±0.480	0.923±0.228	0.816±0.162
HyperImpute	0.434±0.007	0.551±0.076	0.492±0.033	0.709±0.009	0.854±0.091	0.798±0.060
MissForest	0.503±0.007	0.533±0.035	0.518±0.031	0.763±0.007	0.826±0.073	0.805±0.060
TabCSDI	0.501±0.004	0.535±0.040	0.520±0.025	0.763±0.008	0.802±0.061	0.805±0.047
MCFlow	0.523±0.003	0.559±0.042	0.548±0.032	0.813±0.008	0.864±0.074	0.871±0.067
SoftImpute	0.571±0.004	0.645±0.065	0.624±0.049	0.832±0.012	0.959±0.091	0.935±0.074
Median	0.722±0.004	0.749±0.048	0.755±0.038	1.019±0.007	1.037±0.056	1.064±0.055
MIRACLE	0.440±0.004	0.651±0.079	0.500±0.035	0.709±0.014	1.084±0.143	0.878±0.084
MissDiff	0.511±0.015	0.547±0.042	0.564±0.061	0.800±0.047	0.985±0.247	1.313±1.022
Mean	0.742±0.004	0.776±0.041	0.779±0.039	0.994±0.005	1.024±0.060	1.045±0.056
MIWAE	0.746±0.003	0.777±0.047	0.781±0.039	1.025±0.005	1.046±0.059	1.072±0.055
MICE	0.755±0.005	0.804±0.041	0.765±0.019	1.048±0.008	1.122±0.056	1.080±0.035
GAIN	1.275±0.061	1.184±0.177	1.284±0.115	1.641±0.062	1.524±0.279	1.636±0.143
Most-Frequent	0.944±0.035	1.013±0.134	0.999±0.086	1.292±0.042	1.362±0.159	1.356±0.095

Table 20: IS MAE and RMSE on Adult (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.482±0.004**	**0.478±0.072**	**0.497±0.041**	0.928±0.010	0.977±0.116	0.969±0.111
RefiDiff	0.527±0.004	0.523±0.064	0.521±0.036	0.952±0.010	0.996±0.101	0.969±0.106
DiffPuter	0.493±0.003	0.510±0.071	0.515±0.033	0.907±0.013	0.972±0.104	0.955±0.107
EM	0.560±0.003	0.559±0.059	0.577±0.030	0.934±0.011	1.002±0.126	0.999±0.129
ReMasker	0.527±0.017	0.522±0.052	0.524±0.040	**0.900±0.016**	**0.949±0.101**	**0.927±0.111**
HyperImpute	0.538±0.009	0.578±0.051	0.549±0.023	0.928±0.016	1.043±0.079	0.973±0.095
MissForest	0.612±0.031	0.712±0.107	0.617±0.050	1.053±0.064	1.283±0.185	1.070±0.106
TabCSDI	0.572±0.015	0.552±0.082	0.589±0.043	0.984±0.010	1.010±0.116	1.020±0.100
MCFlow	0.544±0.012	0.545±0.081	0.563±0.038	0.955±0.008	1.005±0.120	1.000±0.106
SoftImpute	0.717±0.002	0.717±0.126	0.741±0.070	1.087±0.006	1.197±0.205	1.219±0.252
Median	0.539±0.004	0.511±0.086	0.560±0.042	1.005±0.011	1.034±0.122	1.047±0.106
MIRACLE	0.583±0.012	0.924±0.165	0.601±0.047	1.026±0.026	1.555±0.143	1.117±0.113
MissDiff	0.619±0.038	0.604±0.049	0.613±0.031	1.037±0.091	1.048±0.102	1.057±0.118
Mean	0.598±0.003	0.580±0.059	0.617±0.030	0.997±0.011	1.032±0.118	1.043±0.104
MIWAE	0.568±0.004	0.541±0.097	0.590±0.042	1.020±0.011	1.048±0.121	1.063±0.104
MICE	0.980±0.004	0.971±0.063	0.979±0.023	1.322±0.007	1.360±0.027	1.347±0.047
GAIN	0.720±0.005	0.692±0.039	0.720±0.027	1.070±0.008	1.078±0.090	1.091±0.087
Most-Frequent	0.575±0.017	0.548±0.096	0.596±0.053	1.065±0.025	1.089±0.130	1.105±0.111

Table 21: OOS MAE and RMSE on Adult (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.486±0.002**	**0.481±0.070**	**0.502±0.045**	0.956±0.017	0.991±0.124	0.984±0.124
RefiDiff	0.548±0.008	0.543±0.067	0.543±0.040	1.005±0.021	1.030±0.102	1.003±0.116
DiffPuter	0.493±0.005	0.509±0.070	0.515±0.037	0.935±0.019	0.983±0.114	0.968±0.123
EM	0.563±0.003	0.561±0.059	0.581±0.033	0.960±0.018	1.002±0.124	0.998±0.119
ReMasker	0.532±0.016	0.524±0.052	0.527±0.044	**0.928±0.022**	**0.958±0.113**	**0.938±0.127**
HyperImpute	0.557±0.008	0.607±0.067	0.569±0.031	0.970±0.023	1.075±0.079	1.008±0.106
MissForest	0.617±0.031	0.713±0.103	0.621±0.051	1.079±0.057	1.289±0.186	1.084±0.110
TabCSDI	0.578±0.015	0.555±0.081	0.593±0.045	1.012±0.018	1.022±0.124	1.033±0.112
MCFlow	0.551±0.014	0.548±0.080	0.570±0.042	0.987±0.020	1.018±0.127	1.018±0.119
SoftImpute	0.733±0.011	0.726±0.131	0.759±0.072	1.116±0.009	1.201±0.178	1.228±0.205
Median	0.546±0.004	0.518±0.086	0.566±0.046	1.034±0.017	1.049±0.130	1.063±0.118
MIRACLE	0.842±0.127	1.137±0.209	0.831±0.098	1.418±0.191	1.825±0.269	1.377±0.118
MissDiff	0.622±0.036	0.609±0.048	0.618±0.034	1.059±0.080	1.063±0.106	1.077±0.133
Mean	0.605±0.004	0.586±0.058	0.622±0.032	1.026±0.017	1.047±0.126	1.058±0.116
MIWAE	0.574±0.004	0.546±0.096	0.596±0.044	1.049±0.017	1.064±0.129	1.079±0.116
MICE	0.987±0.006	0.972±0.061	0.981±0.019	1.343±0.019	1.366±0.036	1.356±0.062
GAIN	0.724±0.008	0.698±0.036	0.725±0.030	1.094±0.016	1.092±0.098	1.105±0.099
Most-Frequent	0.580±0.016	0.554±0.096	0.602±0.056	1.091±0.028	1.103±0.138	1.120±0.121

Table 22: IS MAE and RMSE on Default (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.206±0.002**	**0.227±0.020**	**0.225±0.019**	**0.655±0.022**	**0.711±0.078**	**0.739±0.084**
RefiDiff	0.235±0.003	0.260±0.028	0.253±0.021	0.682±0.025	0.789±0.113	0.757±0.087
DiffPuter	0.228±0.005	0.293±0.042	0.277±0.032	0.692±0.019	0.864±0.076	0.854±0.081
EM	0.279±0.002	0.305±0.026	0.306±0.019	0.694±0.017	0.777±0.074	0.779±0.069
ReMasker	0.502±0.157	0.529±0.210	0.529±0.152	0.898±0.127	1.006±0.251	0.989±0.140
HyperImpute	0.262±0.007	0.407±0.066	0.333±0.044	0.666±0.024	0.814±0.052	0.784±0.075
MissForest	0.327±0.010	0.387±0.048	0.367±0.039	0.806±0.025	0.993±0.139	0.945±0.129
TabCSDI	0.295±0.004	0.304±0.018	0.311±0.018	0.787±0.020	0.816±0.074	0.851±0.078
MCFlow	0.435±0.013	0.496±0.053	0.492±0.035	0.857±0.021	0.950±0.075	0.968±0.064
SoftImpute	0.340±0.003	0.406±0.064	0.397±0.038	0.798±0.017	0.883±0.068	0.899±0.063
Median	0.481±0.003	0.504±0.050	0.521±0.051	1.042±0.015	1.077±0.085	1.135±0.087
MIRACLE	0.379±0.028	0.447±0.054	0.408±0.034	0.992±0.034	1.062±0.098	1.065±0.095
MissDiff	0.534±0.176	0.602±0.366	0.840±0.571	16.288±17.444	18.960±18.829	35.101±30.363
Mean	0.548±0.002	0.578±0.028	0.586±0.031	1.001±0.015	1.060±0.069	1.103±0.076
MIWAE	0.497±0.003	0.522±0.046	0.538±0.047	1.041±0.015	1.081±0.082	1.136±0.085
MICE	0.595±0.005	0.580±0.031	0.580±0.012	0.958±0.011	1.008±0.077	0.994±0.037
GAIN	0.867±0.025	0.872±0.075	0.838±0.032	1.300±0.024	1.304±0.102	1.305±0.043
Most-Frequent	0.549±0.003	0.555±0.059	0.584±0.056	1.152±0.014	1.171±0.085	1.237±0.088

Table 23: OOS MAE and RMSE on Default (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.201±0.003**	**0.221±0.020**	**0.219±0.018**	**0.586±0.027**	**0.638±0.076**	**0.662±0.072**
RefiDiff	0.239±0.004	0.261±0.026	0.256±0.022	0.634±0.024	0.701±0.078	0.712±0.086
DiffPuter	0.221±0.006	0.284±0.043	0.267±0.031	0.615±0.026	0.796±0.101	0.778±0.090
EM	0.274±0.003	0.299±0.027	0.301±0.019	0.633±0.026	0.714±0.073	0.725±0.060
ReMasker	0.501±0.157	0.524±0.209	0.524±0.150	0.849±0.145	0.945±0.262	0.934±0.131
HyperImpute	0.274±0.008	0.349±0.085	0.332±0.071	0.639±0.021	0.732±0.087	0.733±0.095
MissForest	0.324±0.011	0.383±0.050	0.362±0.040	0.741±0.029	0.929±0.138	0.886±0.134
TabCSDI	0.291±0.005	0.299±0.019	0.306±0.017	0.724±0.023	0.746±0.062	0.787±0.060
MCFlow	0.434±0.014	0.494±0.054	0.489±0.035	0.807±0.019	0.893±0.082	0.914±0.060
SoftImpute	0.331±0.002	0.363±0.040	0.360±0.023	0.734±0.023	0.773±0.056	0.805±0.055
Median	0.483±0.004	0.504±0.051	0.523±0.051	0.994±0.018	1.020±0.101	1.083±0.097
MIRACLE	0.506±0.032	0.502±0.048	0.492±0.038	1.167±0.056	1.096±0.073	1.143±0.077
MissDiff	0.404±0.065	0.478±0.280	0.560±0.317	2.367±1.700	6.667±10.765	8.469±10.994
Mean	0.549±0.003	0.577±0.029	0.586±0.031	0.951±0.018	1.003±0.082	1.049±0.084
MIWAE	0.498±0.004	0.522±0.047	0.539±0.047	0.994±0.017	1.024±0.098	1.084±0.095
MICE	0.590±0.007	0.574±0.029	0.575±0.011	0.918±0.020	0.956±0.067	0.945±0.026
GAIN	0.864±0.026	0.871±0.076	0.839±0.033	1.263±0.038	1.260±0.120	1.269±0.055
Most-Frequent	0.549±0.005	0.554±0.060	0.584±0.057	1.110±0.017	1.119±0.101	1.189±0.098

Table 24: IS MAE and RMSE on News (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.262±0.001**	**0.265±0.009**	**0.268±0.006**	**0.530±0.010**	**0.543±0.024**	**0.561±0.032**
RefiDiff	0.283±0.001	0.282±0.015	0.286±0.004	0.571±0.012	0.586±0.019	0.597±0.034
DiffPuter	0.307±0.044	0.331±0.018	0.309±0.012	0.609±0.054	0.688±0.079	0.760±0.359
EM	0.379±0.001	0.384±0.011	0.387±0.004	0.631±0.008	0.655±0.034	0.659±0.027
ReMasker	0.380±0.119	0.422±0.166	0.372±0.105	0.697±0.146	0.744±0.194	0.703±0.130
HyperImpute	0.267±0.002	0.273±0.025	0.293±0.037	0.565±0.017	0.571±0.033	0.608±0.045
MissForest	0.389±0.002	0.389±0.018	0.395±0.007	0.687±0.011	0.699±0.019	0.715±0.034
TabCSDI	0.406±0.005	0.398±0.011	0.408±0.004	0.744±0.011	0.741±0.023	0.763±0.023
MCFlow	0.433±0.006	0.437±0.025	0.446±0.023	0.771±0.012	0.785±0.049	0.811±0.051
SoftImpute	0.620±0.003	0.639±0.018	0.636±0.012	0.913±0.008	0.972±0.052	0.974±0.057
Median	0.590±0.001	0.586±0.011	0.593±0.006	1.044±0.008	1.054±0.017	1.060±0.020
MIRACLE	0.384±0.005	0.400±0.014	0.394±0.013	0.761±0.013	0.785±0.045	0.799±0.035
MissDiff	0.438±0.031	0.435±0.030	0.994±1.618	6.545±8.335	2.473±2.479	89.950±258.785
Mean	0.651±0.001	0.651±0.010	0.653±0.004	0.995±0.008	1.004±0.018	1.011±0.021
MIWAE	0.617±0.001	0.613±0.012	0.619±0.006	1.034±0.008	1.044±0.017	1.050±0.021
MICE	0.622±0.001	0.617±0.015	0.622±0.007	0.896±0.004	0.905±0.025	0.912±0.013
GAIN	0.812±0.009	0.785±0.042	0.802±0.027	1.162±0.011	1.136±0.080	1.171±0.041
Most-Frequent	1.225±0.017	1.248±0.115	1.233±0.024	1.944±0.015	1.982±0.148	1.965±0.032

Table 25: OOS MAE and RMSE on News (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.262±0.001**	**0.265±0.010**	**0.267±0.007**	**0.520±0.011**	**0.532±0.020**	**0.555±0.028**
RefiDiff	0.290±0.001	0.289±0.015	0.294±0.006	0.574±0.014	0.595±0.022	0.624±0.075
DiffPuter	0.298±0.033	0.318±0.017	0.302±0.012	0.570±0.045	0.632±0.027	0.832±0.428
EM	0.378±0.001	0.381±0.011	0.384±0.004	0.621±0.008	0.636±0.023	0.645±0.021
ReMasker	0.377±0.118	0.419±0.165	0.370±0.104	0.672±0.147	0.729±0.197	0.691±0.125
HyperImpute	0.325±0.044	0.330±0.095	0.304±0.042	8.539±9.494	7.011±13.261	7.816±9.133
MissForest	0.388±0.002	0.388±0.019	0.393±0.007	0.675±0.008	0.691±0.024	0.705±0.024
TabCSDI	0.405±0.005	0.397±0.011	0.406±0.004	0.724±0.015	0.726±0.024	0.751±0.018
MCFlow	0.432±0.006	0.435±0.025	0.445±0.023	0.750±0.014	0.769±0.052	0.797±0.053
SoftImpute	0.671±0.021	0.683±0.075	0.680±0.035	1.163±0.108	1.190±0.341	1.228±0.203
Median	0.582±0.002	0.578±0.011	0.584±0.006	1.018±0.009	1.031±0.021	1.039±0.019
MIRACLE	0.561±0.040	0.600±0.068	0.576±0.047	1.370±0.475	1.136±0.232	1.387±0.263
MissDiff	0.510±0.262	0.425±0.021	0.729±0.856	21.039±50.370	2.557±3.009	75.570±217.406
Mean	0.644±0.002	0.643±0.010	0.646±0.005	0.969±0.010	0.982±0.022	0.991±0.019
MIWAE	0.610±0.002	0.605±0.012	0.612±0.006	1.009±0.010	1.022±0.022	1.030±0.019
MICE	0.623±0.004	0.746±0.350	0.636±0.050	1.267±0.686	28.737±75.806	3.382±7.438
GAIN	0.804±0.011	0.777±0.042	0.794±0.028	1.140±0.019	1.116±0.077	1.153±0.043
Most-Frequent	1.223±0.013	1.245±0.116	1.230±0.024	1.936±0.009	1.975±0.151	1.959±0.034

Table 26: IS MAE and RMSE on Shoppers (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.316±0.004**	**0.309±0.044**	**0.321±0.037**	**0.779±0.024**	**0.783±0.095**	**0.797±0.088**
RefiDiff	0.360±0.006	0.354±0.053	0.364±0.032	0.805±0.023	0.812±0.097	0.817±0.080
DiffPuter	0.350±0.005	0.354±0.055	0.363±0.037	0.797±0.021	0.822±0.109	0.825±0.097
EM	0.404±0.004	0.404±0.058	0.413±0.031	0.779±0.024	0.814±0.147	0.801±0.087
ReMasker	0.416±0.016	0.427±0.048	0.427±0.039	0.801±0.016	0.808±0.082	0.813±0.076
HyperImpute	0.371±0.007	0.372±0.054	0.380±0.036	0.805±0.014	0.827±0.124	0.832±0.098
MissForest	0.412±0.030	0.418±0.063	0.442±0.039	0.852±0.058	0.867±0.104	0.916±0.083
TabCSDI	0.533±0.005	0.512±0.054	0.537±0.033	1.036±0.020	1.010±0.103	1.044±0.086
MCFlow	0.489±0.016	0.486±0.062	0.501±0.035	0.952±0.021	0.941±0.115	0.968±0.090
SoftImpute	0.544±0.003	0.565±0.049	0.576±0.030	0.922±0.018	1.007±0.083	1.000±0.059
Median	0.448±0.004	0.432±0.063	0.459±0.045	1.070±0.020	1.058±0.113	1.088±0.096
MIRACLE	0.807±0.066	0.843±0.087	0.787±0.068	1.469±0.092	1.522±0.119	1.445±0.114
MissDiff	0.498±0.034	0.605±0.175	0.556±0.130	3.217±1.664	5.164±4.464	3.153±2.591
Mean	0.583±0.002	0.582±0.023	0.593±0.020	1.011±0.020	1.005±0.093	1.029±0.084
MIWAE	0.460±0.004	0.442±0.059	0.469±0.042	1.066±0.020	1.048±0.108	1.081±0.093
MICE	0.758±0.006	0.764±0.041	0.759±0.009	1.089±0.014	1.123±0.091	1.104±0.042
GAIN	0.745±0.014	0.718±0.039	0.744±0.035	1.154±0.013	1.101±0.080	1.159±0.068
Most-Frequent	0.736±0.006	0.735±0.181	0.733±0.048	1.504±0.015	1.499±0.243	1.500±0.072

Table 27: OOS MAE and RMSE on Shoppers (↓). Best mean is bold; second-best mean is underlined.

	MAE			RMSE		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.316±0.006**	**0.314±0.040**	**0.325±0.034**	0.751±0.017	**0.774±0.084**	**0.775±0.053**
RefiDiff	0.367±0.007	0.367±0.052	0.373±0.031	0.784±0.015	0.811±0.088	0.798±0.055
DiffPuter	0.344±0.008	0.354±0.050	0.362±0.034	**0.747±0.025**	0.794±0.079	0.785±0.062
EM	0.404±0.006	0.411±0.055	0.419±0.031	0.752±0.016	0.801±0.123	0.788±0.065
ReMasker	0.416±0.019	0.434±0.043	0.433±0.037	0.770±0.018	0.802±0.058	0.786±0.054
HyperImpute	0.396±0.014	0.407±0.051	0.397±0.041	0.799±0.024	0.843±0.096	0.829±0.079
MissForest	0.410±0.029	0.421±0.059	0.443±0.038	0.824±0.053	0.855±0.091	0.887±0.062
TabCSDI	0.532±0.010	0.516±0.050	0.538±0.029	0.993±0.017	0.993±0.083	1.008±0.059
MCFlow	0.490±0.015	0.491±0.055	0.506±0.031	0.912±0.021	0.926±0.086	0.937±0.060
SoftImpute	0.545±0.005	0.552±0.042	0.567±0.027	0.884±0.012	0.949±0.064	0.943±0.043
Median	0.447±0.007	0.438±0.057	0.463±0.040	1.028±0.015	1.043±0.094	1.056±0.070
MIRACLE	0.771±0.039	0.809±0.056	0.753±0.068	1.384±0.049	1.441±0.075	1.364±0.144
MissDiff	0.464±0.017	0.545±0.138	0.517±0.102	0.866±0.037	1.944±1.087	1.305±0.727
Mean	0.579±0.006	0.583±0.021	0.593±0.017	0.965±0.013	0.985±0.071	0.992±0.055
MIWAE	0.458±0.007	0.447±0.054	0.473±0.037	1.024±0.015	1.032±0.088	1.048±0.066
MICE	0.758±0.010	0.766±0.040	0.762±0.013	1.071±0.015	1.112±0.077	1.093±0.024
GAIN	0.747±0.017	0.721±0.034	0.745±0.033	1.121±0.022	1.084±0.053	1.127±0.045
Most-Frequent	0.739±0.010	0.740±0.176	0.737±0.043	1.476±0.014	1.487±0.231	1.474±0.052

D.3.4 COMPLETE CATEGORICAL IMPUTATION RESULTS

News MAR uses nine seeds because one seed contains no eligible missing categorical cell; all other entries use ten seeds.

Table 28: Categorical accuracy on Adult (↑). Best mean is bold; second-best mean is underlined.

	IS			OOS		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	0.711±0.002	0.714±0.054	0.711±0.006	0.713±0.002	0.715±0.054	0.712±0.008
RefiDiff	0.677±0.007	0.670±0.062	0.677±0.008	0.679±0.005	0.669±0.064	0.681±0.008
DiffPuter	0.707±0.002	0.699±0.053	0.704±0.007	0.711±0.002	0.701±0.053	0.705±0.009
EM	0.631±0.003	0.623±0.030	0.631±0.007	0.631±0.003	0.624±0.033	0.632±0.008
ReMasker	**0.720±0.021**	**0.732±0.054**	**0.722±0.023**	**0.721±0.022**	**0.731±0.054**	**0.722±0.024**
HyperImpute	0.708±0.012	0.696±0.058	0.706±0.018	0.703±0.014	0.686±0.050	0.710±0.010
MissForest	0.664±0.016	0.638±0.061	0.652±0.019	0.663±0.015	0.638±0.060	0.649±0.019
TabCSDI	0.601±0.003	0.597±0.032	0.603±0.006	0.604±0.004	0.599±0.032	0.605±0.007
MCFlow	0.607±0.003	0.603±0.034	0.607±0.007	0.607±0.004	0.603±0.035	0.605±0.006
SoftImpute	0.533±0.002	0.513±0.038	0.521±0.036	0.533±0.004	0.514±0.039	0.521±0.035
Median	0.536±0.002	0.547±0.044	0.539±0.008	0.539±0.003	0.549±0.045	0.541±0.008
MIRACLE	0.657±0.007	0.622±0.041	0.660±0.010	0.636±0.013	0.605±0.041	0.640±0.012
MissDiff	0.626±0.008	0.629±0.050	0.630±0.015	0.627±0.010	0.630±0.050	0.630±0.016
Mean	0.536±0.002	0.547±0.044	0.539±0.008	0.539±0.003	0.549±0.045	0.541±0.008
MIWAE	0.533±0.002	0.539±0.043	0.534±0.005	0.536±0.003	0.542±0.044	0.537±0.006
MICE	0.498±0.001	0.488±0.033	0.498±0.004	0.499±0.003	0.489±0.035	0.499±0.006
GAIN	0.559±0.003	0.558±0.041	0.566±0.011	0.562±0.004	0.560±0.043	0.568±0.013
Most-Frequent	0.536±0.002	0.547±0.044	0.539±0.008	0.539±0.003	0.549±0.045	0.541±0.008

Table 29: Categorical accuracy on Default (↑). Best mean is bold; second-best mean is underlined.

	IS			OOS		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.755±0.001**	0.749±0.026	0.755±0.005	**0.756±0.002**	0.750±0.027	0.758±0.004
RefiDiff	0.715±0.009	0.694±0.032	0.716±0.011	0.716±0.008	0.694±0.033	0.720±0.009
DiffPuter	0.743±0.002	0.712±0.028	0.729±0.005	0.745±0.003	0.716±0.030	0.731±0.006
EM	0.711±0.002	0.709±0.020	0.712±0.006	0.711±0.001	0.710±0.019	0.713±0.005
ReMasker	0.740±0.024	**0.752±0.032**	0.747±0.019	0.739±0.024	**0.753±0.032**	0.748±0.020
HyperImpute	0.754±0.015	0.724±0.043	**0.761±0.006**	0.753±0.009	0.741±0.038	**0.759±0.007**
MissForest	0.685±0.007	0.675±0.025	0.689±0.014	0.686±0.008	0.678±0.027	0.692±0.013
TabCSDI	0.666±0.010	0.670±0.032	0.668±0.009	0.668±0.008	0.675±0.032	0.673±0.007
MCFlow	0.685±0.005	0.689±0.021	0.686±0.010	0.685±0.004	0.692±0.021	0.687±0.011
SoftImpute	0.540±0.002	0.539±0.019	0.546±0.015	0.546±0.002	0.549±0.019	0.554±0.014
Median	0.532±0.002	0.534±0.026	0.538±0.022	0.536±0.002	0.540±0.026	0.544±0.020
MIRACLE	0.697±0.007	0.682±0.021	0.691±0.009	0.666±0.004	0.659±0.028	0.667±0.011
MissDiff	0.708±0.007	0.706±0.024	0.709±0.011	0.709±0.007	0.707±0.023	0.712±0.012
Mean	0.532±0.002	0.534±0.026	0.538±0.022	0.536±0.002	0.540±0.026	0.544±0.020
MIWAE	0.501±0.001	0.490±0.020	0.500±0.015	0.504±0.002	0.493±0.020	0.504±0.012
MICE	0.578±0.002	0.576±0.012	0.580±0.006	0.581±0.003	0.581±0.012	0.583±0.006
GAIN	0.533±0.002	0.537±0.026	0.541±0.022	0.537±0.002	0.542±0.027	0.547±0.021
Most-Frequent	0.532±0.002	0.534±0.026	0.538±0.022	0.536±0.002	0.540±0.026	0.544±0.020

Table 30: Categorical accuracy on News (↑). Best mean is bold; second-best mean is underlined.

	IS			OOS		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	**0.472±0.003**	**0.526±0.223**	**0.470±0.012**	**0.470±0.003**	**0.523±0.220**	**0.467±0.014**
RefiDiff	0.429±0.010	0.455±0.165	0.428±0.013	0.418±0.010	0.447±0.164	0.417±0.011
DiffPuter	0.454±0.016	0.500±0.201	0.453±0.021	0.457±0.011	0.502±0.202	0.452±0.020
EM	0.414±0.002	0.466±0.193	0.414±0.010	0.412±0.004	0.462±0.188	0.413±0.009
ReMasker	0.439±0.042	0.442±0.194	0.443±0.042	0.438±0.040	0.439±0.191	0.439±0.043
HyperImpute	0.432±0.019	0.471±0.175	0.441±0.016	0.414±0.013	0.453±0.175	0.421±0.022
MissForest	0.351±0.018	0.389±0.133	0.350±0.018	0.347±0.015	0.386±0.133	0.349±0.014
TabCSDI	0.199±0.013	0.200±0.027	0.199±0.014	0.194±0.014	0.196±0.028	0.196±0.016
MCFlow	0.318±0.013	0.344±0.101	0.315±0.017	0.314±0.013	0.339±0.104	0.313±0.018
SoftImpute	0.173±0.003	0.174±0.004	0.173±0.003	0.172±0.004	0.173±0.005	0.172±0.004
Median	0.173±0.003	0.174±0.004	0.173±0.003	0.172±0.004	0.173±0.005	0.172±0.004
MIRACLE	0.416±0.006	0.431±0.168	0.414±0.007	0.368±0.016	0.367±0.139	0.360±0.016
MissDiff	0.427±0.004	0.477±0.194	0.428±0.009	0.427±0.006	0.475±0.191	0.426±0.009
Mean	0.173±0.003	0.174±0.004	0.173±0.003	0.172±0.004	0.173±0.005	0.172±0.004
MIWAE	0.176±0.003	0.175±0.003	0.175±0.002	0.177±0.004	0.174±0.005	0.176±0.004
MICE	0.302±0.003	0.335±0.115	0.302±0.007	0.303±0.006	0.330±0.117	0.302±0.007
GAIN	0.173±0.003	0.174±0.005	0.173±0.003	0.173±0.004	0.174±0.004	0.172±0.004
Most-Frequent	0.173±0.003	0.174±0.004	0.173±0.003	0.172±0.004	0.173±0.005	0.172±0.004

Table 31: Categorical accuracy on Shoppers (↑). Best mean is bold; second-best mean is underlined.

	IS			OOS		
Method	MCAR	MAR	MNAR	MCAR	MAR	MNAR
DOOBIMP	0.584±0.001	0.582±0.034	0.584±0.005	0.585±0.004	0.581±0.035	0.583±0.006
RefiDiff	0.565±0.005	0.559±0.033	0.567±0.007	0.565±0.006	0.561±0.033	0.566±0.009
DiffPuter	0.561±0.005	0.554±0.039	0.556±0.005	0.566±0.006	0.558±0.036	0.561±0.008
EM	0.571±0.002	0.575±0.038	0.573±0.006	0.572±0.004	0.575±0.038	0.573±0.006
ReMasker	**0.585±0.004**	**0.585±0.035**	**0.589±0.006**	**0.587±0.005**	**0.584±0.035**	**0.587±0.007**
HyperImpute	0.581±0.013	0.582±0.034	0.585±0.011	0.557±0.048	0.561±0.063	0.566±0.052
MissForest	0.526±0.011	0.490±0.036	0.534±0.017	0.527±0.012	0.492±0.039	0.532±0.020
TabCSDI	0.491±0.003	0.502±0.039	0.494±0.008	0.493±0.004	0.501±0.041	0.494±0.007
MCFlow	0.535±0.005	0.550±0.042	0.539±0.007	0.536±0.005	0.549±0.044	0.540±0.006
SoftImpute	0.491±0.002	0.493±0.044	0.490±0.004	0.494±0.006	0.494±0.045	0.491±0.007
Median	0.507±0.002	0.518±0.041	0.510±0.007	0.505±0.004	0.515±0.042	0.508±0.006
MIRACLE	0.506±0.007	0.501±0.040	0.509±0.011	0.514±0.006	0.492±0.040	0.517±0.012
MissDiff	0.538±0.010	0.536±0.042	0.542±0.007	0.541±0.006	0.537±0.043	0.542±0.006
Mean	0.507±0.002	0.518±0.041	0.510±0.007	0.505±0.004	0.515±0.042	0.508±0.006
MIWAE	0.495±0.003	0.506±0.040	0.498±0.007	0.496±0.004	0.507±0.039	0.500±0.006
MICE	0.438±0.003	0.441±0.039	0.441±0.005	0.440±0.003	0.442±0.041	0.443±0.005
GAIN	0.505±0.003	0.517±0.042	0.509±0.007	0.504±0.003	0.515±0.042	0.507±0.006
Most-Frequent	0.507±0.002	0.518±0.041	0.510±0.007	0.505±0.004	0.515±0.042	0.508±0.006

D.3.5 COMPLETE DOWNSTREAM CLASSIFICATION RESULTS

All entries are mean ± population standard deviation over ten seeds.

Table 32: Downstream classification under MCAR missingness (↑). Binary datasets report AUROC and multiclass datasets report macro-F1. Entries are mean ± population standard deviation over ten seeds. Best mean is bold; second-best mean is underlined.

	Binary AUROC				Multiclass macro-F1	
Method	Adult	Default	Magic	Shoppers	Bean	Letter
DoobImp	0.879±0.002	<u>0.734±0.003</u>	<u>0.820±0.004</u>	0.838±0.006	0.931±0.002	**0.723±0.005**
RefiDiff	0.875±0.002	0.731±0.003	0.816±0.006	0.832±0.007	0.929±0.001	0.662±0.004
DiffPuter	<u>0.881±0.002</u>	**0.735±0.003**	**0.821±0.005**	0.840±0.007	0.930±0.002	<u>0.715±0.004</u>
EM	0.870±0.002	0.729±0.003	0.802±0.005	<u>0.846±0.005</u>	**0.934±0.002**	0.573±0.006
ReMasker	**0.884±0.002**	0.729±0.005	0.796±0.019	**0.847±0.006**	0.932±0.001	0.679±0.006
HyperImpute	0.877±0.002	0.729±0.004	0.819±0.005	0.835±0.008	<u>0.932±0.002</u>	0.647±0.007
MissForest	0.869±0.004	0.731±0.004	0.795±0.006	0.831±0.011	0.916±0.002	0.546±0.005
TabCSDI	0.572±0.063	0.526±0.009	0.533±0.019	0.532±0.015	0.211±0.054	0.059±0.020
MCFlow	0.840±0.004	0.719±0.003	0.793±0.010	0.705±0.006	0.892±0.010	0.417±0.016
SoftImpute	0.846±0.002	0.722±0.004	0.794±0.005	0.840±0.006	0.922±0.003	0.530±0.006
Median	0.850±0.002	0.729±0.002	0.780±0.006	0.829±0.007	0.876±0.005	0.501±0.006
MIRACLE	0.859±0.005	0.723±0.004	0.816±0.004	0.779±0.008	0.927±0.002	0.523±0.025
MissDiff	0.865±0.005	0.725±0.004	0.810±0.003	0.836±0.009	0.907±0.004	0.610±0.013
Mean	0.851±0.002	0.728±0.003	0.792±0.004	0.842±0.005	0.880±0.005	0.505±0.006
MIWAE	0.846±0.002	0.721±0.003	0.785±0.005	0.827±0.008	0.868±0.003	0.487±0.005
MICE	0.833±0.003	0.717±0.003	0.769±0.005	0.797±0.009	0.925±0.002	0.459±0.006
GAIN	0.844±0.003	0.720±0.003	0.771±0.004	0.820±0.005	0.868±0.004	0.496±0.007
Most-Frequent	0.847±0.002	0.727±0.002	0.739±0.010	0.823±0.008	0.840±0.007	0.491±0.007

Table 33: Downstream classification under MAR missingness (↑). Binary datasets report AUROC and multiclass datasets report macro-F1. Entries are mean ± population standard deviation over ten seeds. Best mean is bold; second-best mean is underlined.

	Binary AUROC				Multiclass macro-F1	
Method	Adult	Default	Magic	Shoppers	Bean	Letter
DoobImp	0.879±0.007	**0.737±0.010**	**0.812±0.013**	0.830±0.025	0.928±0.004	**0.717±0.017**
RefiDiff	0.876±0.008	0.734±0.011	0.804±0.015	0.822±0.028	0.926±0.004	0.657±0.019
DiffPuter	<u>0.880±0.007</u>	<u>0.737±0.009</u>	<u>0.810±0.015</u>	<u>0.835±0.024</u>	0.928±0.003	<u>0.694±0.018</u>
EM	0.873±0.010	0.731±0.011	0.788±0.028	0.835±0.023	**0.932±0.002**	0.593±0.019
ReMasker	**0.883±0.007**	0.736±0.009	0.796±0.025	**0.838±0.023**	0.929±0.004	0.685±0.015
HyperImpute	0.873±0.009	0.732±0.011	0.808±0.014	0.825±0.026	<u>0.930±0.003</u>	0.639±0.017
MissForest	0.869±0.010	0.732±0.011	0.781±0.025	0.825±0.028	0.921±0.005	0.573±0.020
TabCSDI	0.583±0.064	0.537±0.015	0.537±0.018	0.545±0.026	0.246±0.065	0.065±0.022
MCFlow	0.847±0.012	0.724±0.011	0.780±0.031	0.706±0.017	0.895±0.007	0.438±0.031
SoftImpute	0.860±0.016	0.725±0.014	0.775±0.031	0.830±0.026	0.923±0.004	0.570±0.020
Median	0.863±0.019	0.733±0.010	0.779±0.025	0.824±0.029	0.913±0.009	0.552±0.017
MIRACLE	0.858±0.013	0.727±0.010	0.778±0.021	0.774±0.050	0.923±0.010	0.499±0.048
MissDiff	0.871±0.011	0.732±0.011	0.800±0.020	0.830±0.027	0.922±0.005	0.630±0.028
Mean	0.864±0.018	0.733±0.010	0.787±0.024	0.834±0.024	0.914±0.008	0.555±0.017
MIWAE	0.860±0.021	0.727±0.012	0.782±0.025	0.823±0.029	0.912±0.009	0.540±0.019
MICE	0.852±0.016	0.723±0.013	0.754±0.037	0.796±0.042	0.928±0.003	0.496±0.030
GAIN	0.860±0.017	0.728±0.011	0.778±0.032	0.821±0.030	0.913±0.008	0.551±0.018
Most-Frequent	0.861±0.020	0.732±0.010	0.752±0.034	0.821±0.032	0.909±0.009	0.547±0.015

Table 34: Downstream classification under MNAR missingness (↑). Binary datasets report AUROC and multiclass datasets report macro-F1. Entries are mean ± population standard deviation over ten seeds. Best mean is bold; second-best mean is underlined.

	Binary AUROC				Multiclass macro-F1	
Method	Adult	Default	Magic	Shoppers	Bean	Letter
DoobImp	0.877±0.005	0.735±0.004	**0.821±0.006**	0.836±0.009	0.930±0.003	**0.720±0.005**
RefiDiff	0.873±0.006	0.732±0.004	0.816±0.007	0.829±0.007	0.929±0.003	0.662±0.009
DiffPuter	0.879±0.004	**0.735±0.006**	0.820±0.007	0.841±0.009	0.929±0.003	0.707±0.007
EM	0.868±0.006	0.729±0.005	0.800±0.012	0.842±0.010	**0.933±0.002**	0.581±0.007
ReMasker	**0.882±0.004**	0.732±0.005	0.807±0.012	**0.844±0.008**	0.931±0.002	0.682±0.007
HyperImpute	0.875±0.005	0.730±0.006	0.819±0.007	0.832±0.010	0.931±0.002	0.650±0.008
MissForest	0.869±0.005	0.729±0.004	0.798±0.007	0.836±0.012	0.919±0.003	0.559±0.011
TabCSDI	0.570±0.069	0.536±0.027	0.540±0.017	0.539±0.026	0.230±0.060	0.063±0.019
MCFlow	0.837±0.006	0.720±0.005	0.795±0.018	0.707±0.012	0.893±0.011	0.433±0.026
SoftImpute	0.845±0.008	0.718±0.004	0.786±0.013	0.837±0.010	0.923±0.005	0.544±0.010
Median	0.849±0.008	0.728±0.003	0.782±0.008	0.827±0.011	0.891±0.006	0.518±0.009
MIRACLE	0.857±0.006	0.722±0.006	0.813±0.006	0.775±0.013	0.926±0.003	0.516±0.023
MissDiff	0.866±0.005	0.725±0.005	0.810±0.009	0.833±0.012	0.913±0.004	0.615±0.009
Mean	0.849±0.008	0.728±0.003	0.793±0.006	0.841±0.010	0.892±0.006	0.522±0.009
MIWAE	0.844±0.008	0.721±0.004	0.788±0.007	0.828±0.011	0.885±0.007	0.506±0.009
MICE	0.832±0.005	0.716±0.003	0.766±0.009	0.793±0.012	0.926±0.003	0.466±0.008
GAIN	0.845±0.005	0.720±0.003	0.776±0.010	0.821±0.006	0.882±0.007	0.517±0.013
Most-Frequent	0.845±0.010	0.727±0.003	0.747±0.011	0.823±0.013	0.863±0.009	0.511±0.012

E Tabular Ablation Studies

We use Gesture, Letter, and News as three representative datasets from the nine-dataset tabular benchmark in Appendix D. Unless stated otherwise, every ablation retains the same train–test splits, train-fitted normalization, missingness construction, model architecture, optimization, and OOS evaluation protocol as the main tabular experiments. We evaluate 30% MCAR, MAR, and MNAR missingness over ten seeds. Within each seed, an Average result weights the three mechanisms equally and then weights the three datasets equally; figures and tables report the mean and population standard deviation across seeds.

The reference configuration uses $K = 10$ trained score stages, $S = 20$ training completions, $R = 20$ test-time samples with mean readout, $\tau = 0.25$, $W = 20$, $J = 5$, and coordinatewise residual precision. It also retains the endpoint smoothing and clipping settings in Table 5. Each experiment changes only its named factor.

E.1 DoobFit Training-Iteration Analysis

E.1.1 Evolution across DoobFit training iterations

Following MissDiff (Ouyang et al., 2023), DoobFit uses zero padding to initialize its first score stage. Figure 11 shows consistent empirical improvement across the ten trained score stages: MAE and RMSE decrease sharply during the first iterations on every dataset and then improve more gradually. Section E.3.1 and Figure 15 further show that DoobImp approaches the imputation performance of a reference trained on fully observed data.

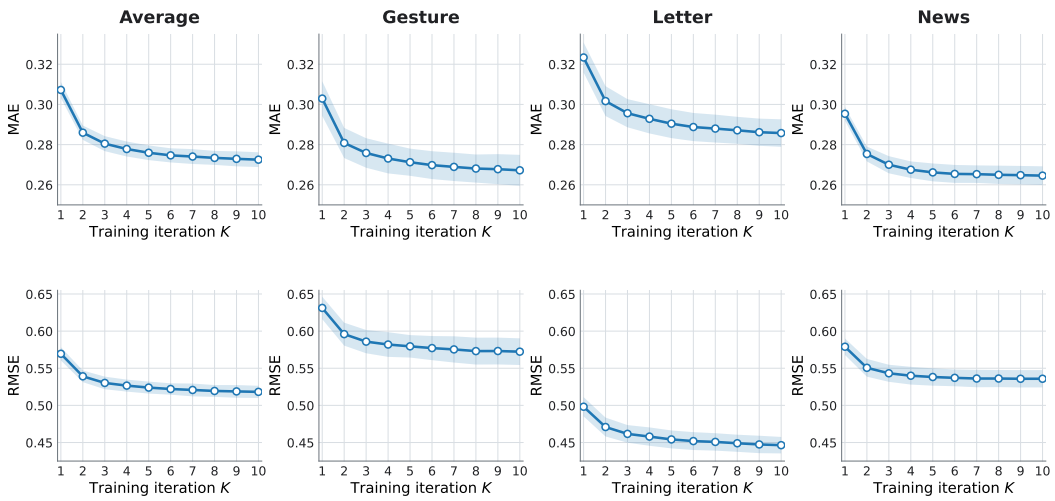


Figure 11: DOOBIMP imputation quality across trained score stages $K = 1, \ldots, 10$. Rows report MAE and RMSE; columns report the equal-weight Average and the three representative datasets. Curves and bands show the mean and population standard deviation over ten seeds after averaging MCAR, MAR, and MNAR equally within each seed. Lower is better.

E.1.2 TRAINING COMPLETION BUDGET

The completion stage averages S stochastic completions before constructing the pseudo-complete data used by the next score-refitting stage. Figure 12 varies $S \in \{1, 5, 10, 20\}$ while fixing the test-time readout at $R = 20$. Most of the aggregate improvement is already realized by $S = 5$, after which both metrics are stable. We use $S = 20$ in the main experiments to match the sampling budget of DiffPuter. Table 36 further compares point errors and distributional accuracy across these completion budgets on Letter.

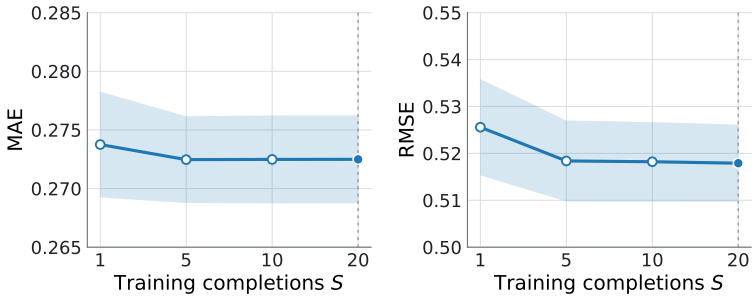


Figure 12: Average MAE and RMSE as the number S of training completions varies. Test-time sampling remains fixed at $R = 20$. Curves and bands show the mean and population standard deviation over ten seeds, with mechanisms and datasets averaged equally within each seed. Lower is better.

E.1.3 POSTERIOR-COMPLETION AGGREGATION

Algorithm 2 uses the coordinatewise mean of S posterior completions as one pseudo-complete sample for score refitting. We test this choice on Letter with 30% MCAR and MAR missingness, $S = 20$, and ten seeds. All three aggregation rules receive the same number of optimizer updates. Let m_i equal one on observed coordinates, let x_i denote the incomplete row, and let $x_i^{(j)}$ be posterior completion j. Define

$$\mathcal{C}_i(z) := m_i \odot x_i + (1 - m_i) \odot z, \quad \widetilde{x}_i^{\text{mean}} := \mathcal{C}_i\left(S^{-1}\sum_{j=1}^{S} x_i^{(j)}\right),$$

$$A_i \sim \text{Unif}\{1, \ldots, S\}, \quad \widetilde{x}_i^{\text{one}} := \mathcal{C}_i\left(x_i^{(A_i)}\right),$$

$$\widetilde{\mathcal{D}}_{\text{multi}} := \left\{\mathcal{C}_i\left(x_i^{(j)}\right) : i = 1, \ldots, n,\ j = 1, \ldots, S\right\}.$$

The random-one rule keeps one complete vector for each row, whereas the retained-multi rule cycles through all completed vectors without increasing the optimizer-update budget. Table 35 shows that the coordinatewise mean attains the lowest MAE and RMSE under both mechanisms, supporting the mean aggregation used by DOOBFIT.

Table 35: Letter MAE and RMSE for posterior-completion aggregation. Entries are mean ± population standard deviation over ten seeds; lower is better.

	MAE			RMSE		
Aggregation	MCAR	MAR	Average	MCAR	MAR	Average
Coordinatewise mean	**0.273±0.003**	**0.301±0.015**	**0.287±0.008**	**0.428±0.006**	**0.470±0.025**	**0.449±0.014**
Random-one vector	0.276±0.003	0.307±0.017	0.292±0.009	0.440±0.006	0.489±0.033	0.465±0.018
Retained multi-sample	0.276±0.004	0.308±0.018	0.292±0.010	0.440±0.007	0.491±0.034	0.466±0.019

Point and distributional accuracy. We further evaluate the final checkpoints on Letter with 30% MCAR, MAR, and MNAR missingness over five paired seeds. Each method generates $R = 20$ independent completions per test row using the same missingness mask for each seed and mechanism. MAE and RMSE score their coordinatewise mean, while the fair continuous ranked probability score (CRPS) evaluates the individual draws. For a held-out numerical value y and draws $z_1, \ldots, z_R$, we compute

$$\text{CRPS}_{\text{fair}}(z_{1:R}, y) := \frac{1}{R}\sum_{r=1}^{R}|z_r - y| - \frac{1}{2R(R-1)}\sum_{r\neq s}|z_r - z_s|.$$

All three metrics average over originally missing numerical test cells on the shared train-normalized scale, with equal weight across seeds.

Table 36 compares completion budgets and aggregation rules alongside DiffPuter and TabCSDI. The completion-budget models use early stopping, with the default $S = 20$ checkpoint taken from the main experiment. For the aggregation comparison, we train three $S = 20$ variants with fixed optimizer-update counts at each stage, matched to the default run for each seed. The default and matched-update mean rows thus use separate training runs with early stopping and fixed updates, respectively. DiffPuter and TabCSDI use their final checkpoints and standard samplers.

Increasing S from 1 to 20 reduces MAE by 1.2%–2.3% and RMSE by 3.0%–5.1% across the three mechanisms, while mean fair CRPS changes by less than 2%. Under matched optimizer-update budgets, coordinatewise mean achieves lower MAE and RMSE than both random-one and retained-multi aggregation under every mechanism. Paired 95% Student-t intervals support the MAE and RMSE improvements over retained-multi aggregation in all three mechanisms. Mean fair CRPS differs by at most 1.2% across these aggregation comparisons, with coordinatewise mean attaining the lowest value in each mechanism.

The default DOOBIMP also outperforms DiffPuter and TabCSDI in MAE and RMSE under all three mechanisms. Its fair CRPS is 16.6% and 5.9% lower than DiffPuter's under MAR and MNAR, respectively, and close under MCAR (2.6% higher). Compared with TabCSDI, DOOBIMP reduces fair CRPS by 37.1%–45.3% across all three mechanisms. These results show that multiple training completions and coordinatewise-mean aggregation improve point imputation with little change in fair CRPS.

Table 36: Point and distributional imputation accuracy on Letter at 30% missingness. Entries are mean ± population standard deviation over five paired seeds; lower is better. All methods use $R = 20$ test draws. The completion-budget rows use mean aggregation with standard early stopping; the three aggregation-control rows use $S = 20$ and matched optimizer-update budgets. Bold marks the lowest unrounded mean in each metric and mechanism across all rows.

Model / training aggregation	Fair CRPS	MAE	RMSE
MCAR			
DOOBIMP, $S = 1$	0.227±0.003	0.277±0.004	0.441±0.007
DOOBIMP, $S = 5$	0.227±0.004	0.274±0.004	0.430±0.008
DOOBIMP, $S = 10$	0.226±0.003	0.273±0.004	0.429±0.007
DOOBIMP, $S = 20$ (default)	0.226±0.002	**0.273±0.004**	**0.428±0.007**
Mean (matched updates)	0.227±0.003	0.274±0.004	0.430±0.008
Random-one (matched updates)	0.228±0.003	0.277±0.004	0.442±0.007
Retained-multi (matched updates)	0.228±0.003	0.277±0.005	0.442±0.009
DiffPuter, $S = 20$	**0.221±0.003**	0.290±0.004	0.455±0.006
TabCSDI	0.413±0.006	0.590±0.007	0.798±0.009
MAR			
DOOBIMP, $S = 1$	0.269±0.011	0.311±0.015	0.500±0.031
DOOBIMP, $S = 5$	0.267±0.010	0.305±0.013	0.481±0.021
DOOBIMP, $S = 10$	0.266±0.010	0.305±0.013	0.480±0.021
DOOBIMP, $S = 20$ (default)	**0.264±0.010**	**0.304±0.013**	**0.475±0.020**
Mean (matched updates)	0.266±0.010	0.304±0.013	0.478±0.020
Random-one (matched updates)	0.268±0.012	0.310±0.016	0.499±0.031
Retained-multi (matched updates)	0.269±0.012	0.311±0.016	0.499±0.030
DiffPuter, $S = 20$	0.317±0.012	0.379±0.015	0.575±0.016
TabCSDI	0.420±0.008	0.599±0.013	0.814±0.018
MNAR			
DOOBIMP, $S = 1$	0.242±0.006	0.289±0.006	0.464±0.013
DOOBIMP, $S = 5$	0.240±0.005	0.285±0.006	0.449±0.010
DOOBIMP, $S = 10$	0.241±0.005	0.285±0.006	0.449±0.010
DOOBIMP, $S = 20$ (default)	0.241±0.004	0.285±0.005	0.449±0.008
Mean (matched updates)	**0.240±0.005**	**0.284±0.006**	**0.448±0.010**
Random-one (matched updates)	0.241±0.005	0.288±0.006	0.463±0.010
Retained-multi (matched updates)	0.243±0.006	0.290±0.006	0.466±0.011
DiffPuter, $S = 20$	0.256±0.006	0.324±0.007	0.503±0.012
TabCSDI	0.416±0.009	0.593±0.012	0.804±0.016

E.2 Practical Design

E.2.1 Temperature and conditioning window

We evaluate the full Cartesian grid $\tau \in \{0.15, 0.25, 0.35, 0.45, 0.55\}$ and $W \in \{1, 11, 21, 31, 41\}$, where conditioning is active at reverse indices $\ell \geq W$. Figure 13 shows a broad low-error region for moderate temperature and sufficiently long conditioning. The shortest tested active interval is consistently weaker, while extending an already long interval produces little additional improvement. This behavior shows that conditioning is stable across a broad neighborhood of temperature and window choices.

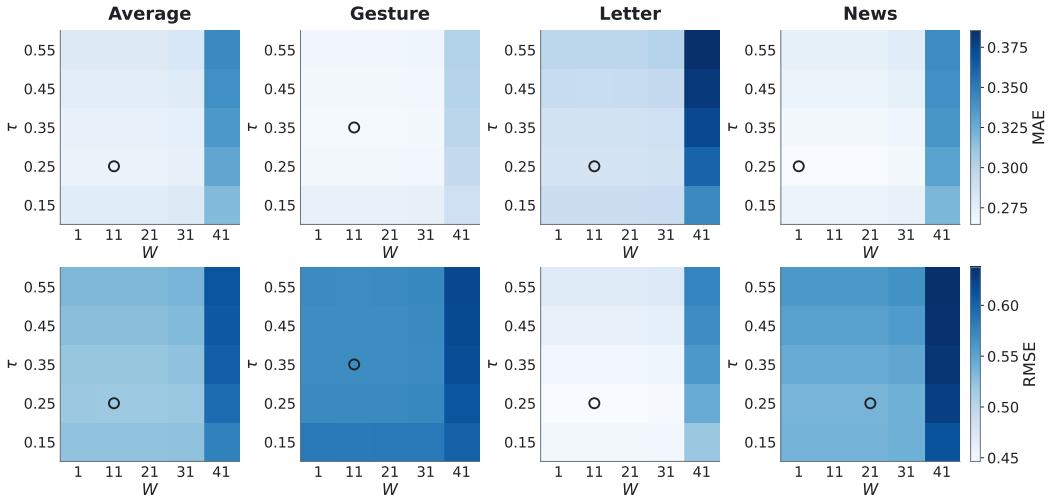


Figure 13: MAE and RMSE over the temperature–window grid. Rows report MAE and RMSE; columns report the equal-weight Average and each representative dataset. Each cell is the ten-seed mean after averaging MCAR, MAR, and MNAR equally within seed; outlined cells attain the lowest mean in their panel. Lower is better.

E.2.2 Same-noise-level refinement

Figure 14 separates observed-coordinate projection, Doob conditioning without refinement ($J = 0$), and $J = 1, \ldots, 10$ same-noise-level refinement steps. Conditioning already improves substantially over projection alone, and the first refinement step provides the largest additional gain on every dataset. Later steps yield diminishing improvements, making $J = 5$ a practical balance between accuracy and computation.

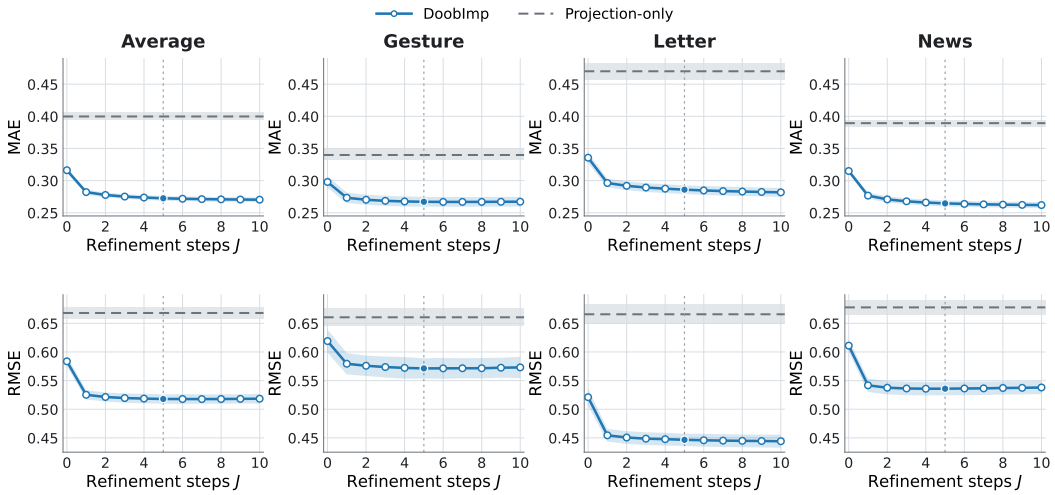


Figure 14: MAE and RMSE versus the same-noise-level refinement budget J. Rows report MAE and RMSE; columns report the equal-weight Average and each representative dataset. DOOBIMP curves and projection-only reference bands show the mean and population standard deviation over ten seeds after averaging mechanisms equally within each seed. Lower is better.

E.2.3 RESIDUAL PRECISION

The practical conditioning in (10) can either use temperature scaling alone or additionally normalize the observed residual by the coordinatewise posterior-variance approximation in (9). Table 37 shows that diagonal precision reduces both errors in the Average and on every representative dataset. The result supports adapting the residual scale across coordinates and diffusion levels.

Table 37: MAE and RMSE for temperature-only and diagonal residual precision. Entries are mean ± population standard deviation over ten seeds; lower is better.

	MAE		RMSE	
Dataset	Temperature-only	Diagonal	Temperature-only	Diagonal
Average	0.291±0.004	**0.272±0.004**	0.548±0.008	**0.518±0.008**
Gesture	0.273±0.007	**0.267±0.008**	0.575±0.016	**0.571±0.018**
Letter	0.316±0.009	**0.286±0.007**	0.496±0.013	**0.447±0.011**
News	0.284±0.004	**0.265±0.005**	0.574±0.012	**0.536±0.011**

E.3 UNCONDITIONAL BACKBONE LEARNING AND INFERENCE-TIME ADAPTATION

E.3.1 COMPARISON WITH TRAINING ON FULLY OBSERVED DATA

For reference, we train an unconditional score network for 1,000 epochs on fully observed training data without stochastic completion. Each resulting score is conditioned at inference time using the same DOOBCOND sampler as DOOBIMP. For DOOBIMP, the training and test missing rates are matched; the reference unconditional backbone is reused across all test rates. Figure 15 shows that DOOBIMP closely tracks the imputation performance of this reference at low missing rates. The gap widens as incomplete training data provide less information for DOOBFIT to learn the unconditional score. The reference uses a fixed training budget, while the DOOBFIT refitting stages use early stopping.

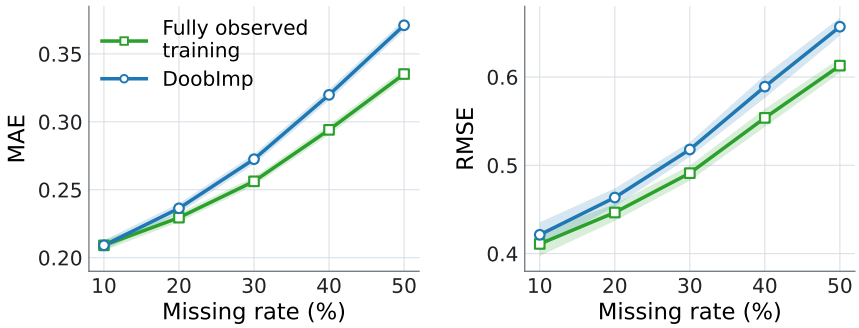


Figure 15: Average MAE and RMSE across test missing rates for DOOBIMP and a reference trained on fully observed data. DOOBIMP uses matched training and test missing rates. Both methods use the same inference-time conditioning. Curves and bands show the mean and population standard deviation over ten seeds after averaging the three mechanisms and representative datasets equally within each seed. Lower is better.

E.3.2 ADAPTATION ACROSS MISSINGNESS MECHANISMS AND RATES

We reuse final models while changing the test missingness law, with all conditioning parameters fixed. The mechanism comparison in Table 3(a) holds both missing rates at 30%. Each method uses one model per dataset, training mechanism, and seed across all test mechanisms. Both use mean readout over 20 samples; DiffPuter retains its standard sampling protocol. Matched results are reused, and the other conditions require inference only. For each mechanism pair, we average the three datasets equally within seed before computing the ten-seed mean and population standard deviation.

Table 38 reports the MAE results in Table 3(a) to three decimal places. Table 39 reports the corresponding RMSE. DOOBIMP also has lower average RMSE in every mechanism pair. Table 40 fixes the training missing rate at 30% and varies the test rate within each mechanism, additionally averaging mechanisms equally within seed. DOOBIMP achieves lower MAE and RMSE at every test rate. At higher test missing rates, DiffPuter exhibits sharply increased RMSE and variability, whereas DOOBIMP degrades smoothly, indicating more stable adaptation to increased missingness.

Table 38: Average MAE for the mechanism-adaptation settings in Table 3(a). Entries are mean ± population standard deviation over ten seeds; bold marks the better method per pair.

Train	Method	Test MCAR	Test MAR	Test MNAR
MCAR	DOOBIMP	**0.263±0.001**	**0.262±0.005**	**0.268±0.005**
	DiffPuter	0.293±0.012	0.293±0.014	0.302±0.014
MAR	DOOBIMP	**0.278±0.003**	**0.278±0.006**	**0.284±0.006**
	DiffPuter	0.329±0.004	0.345±0.007	0.345±0.007
MNAR	DOOBIMP	**0.270±0.002**	**0.270±0.006**	**0.276±0.006**
	DiffPuter	0.306±0.003	0.314±0.007	0.319±0.007

Table 39: Average RMSE for the mechanism-adaptation settings in Table 3(a). Entries are mean ± population standard deviation over ten seeds; bold marks the better method per pair.

Train	Method	Test MCAR	Test MAR	Test MNAR
MCAR	DOOBIMP	**0.498±0.006**	**0.499±0.015**	**0.519±0.013**
	DiffPuter	0.575±0.053	0.658±0.142	0.921±0.918
MAR	DOOBIMP	**0.524±0.008**	**0.524±0.014**	**0.542±0.012**
	DiffPuter	0.667±0.135	0.653±0.014	0.656±0.012
MNAR	DOOBIMP	**0.512±0.008**	**0.514±0.013**	**0.531±0.013**
	DiffPuter	0.619±0.053	0.613±0.016	0.695±0.157

Table 40: Test-rate adaptation from models trained at 30% missingness. Average MAE and RMSE are reported as mean ± population standard deviation over ten seeds, averaging mechanisms and datasets equally within seed. Bold marks the lower error.

Metric	Method	10%	20%	30%	40%	50%
MAE	DOOBIMP	**0.243±0.006**	**0.254±0.004**	**0.272±0.004**	**0.304±0.004**	**0.341±0.004**
	DiffPuter	0.308±0.008	0.306±0.005	0.319±0.005	0.355±0.035	0.393±0.043
RMSE	DOOBIMP	**0.474±0.014**	**0.488±0.010**	**0.518±0.008**	**0.570±0.011**	**0.623±0.008**
	DiffPuter	0.636±0.055	0.615±0.048	0.641±0.071	3.402±8.122	4.546±11.363

Finally, Table 41 crosses training and test missing rates for DOOBIMP, keeping each mechanism matched. Training with the least missing data gives the lowest MAE at every test rate. This pattern supports learning an unconditional score from more informative training observations and conditioning it on each realized test pattern without retraining.

Table 41: Average DOOBIMP MAE across training and test missing rates with matched mechanisms. Entries are mean ± population standard deviation over ten seeds after averaging mechanisms and datasets equally within seed. Bold marks the lowest error in each test-rate column.

Train / Test	10%	20%	30%	40%	50%
10%	**0.209±0.004**	**0.226±0.003**	**0.251±0.003**	**0.289±0.004**	**0.330±0.004**
20%	0.223±0.005	0.236±0.004	0.259±0.003	0.294±0.004	0.334±0.004
30%	0.243±0.006	0.254±0.004	0.272±0.004	0.304±0.004	0.341±0.004
40%	0.259±0.005	0.271±0.004	0.290±0.003	0.320±0.004	0.352±0.004
50%	0.293±0.005	0.303±0.004	0.319±0.003	0.344±0.004	0.371±0.004

E.4 RUNTIME ANALYSIS

Runtime is measured on Letter with 30% MCAR missingness over five seeds 2026–2030. All methods run sequentially on a single NVIDIA RTX A6000, so timed regions do not overlap. These measurements use the model architecture, batch shape, masks, and numerical precision of the tabular protocol.

E.4.1 INFERENCE TIME

For each seed, we perform one warm-up and three timed repeats and retain the median repeat. Table 42 reports milliseconds per imputed row per posterior sample. Inference time increases smoothly with the number of refinement steps, while DOOBIMP remains faster than DiffPuter at every tested J.

Table 42: Sequential inference time on one RTX A6000. Entries are mean ± population standard deviation over five seeds in milliseconds per imputed row per posterior sample. Lower is better.

Method	Inference time
Projection-only	0.122±0.002
DOOBIMP, $J = 0$	0.189±0.001
DOOBIMP, $J = 1$	0.262±0.003
DOOBIMP, $J = 2$	0.336±0.003
DOOBIMP, $J = 3$	0.409±0.003
DOOBIMP, $J = 4$	0.482±0.004
DOOBIMP, $J = 5$	0.555±0.005
DOOBIMP, $J = 6$	0.629±0.006
DOOBIMP, $J = 7$	0.702±0.006
DOOBIMP, $J = 8$	0.776±0.007
DOOBIMP, $J = 9$	0.849±0.008
DOOBIMP, $J = 10$	0.923±0.009
DiffPuter	2.171±0.016

E.4.2 TRAINING TIME

We compare ten DOOBFIT training iterations with the ten corresponding DiffPuter training iterations. Both methods use fresh score initialization at each refitting stage, up to 10,001 epochs with early-stop patience 500, and $S = 20$ training completions. DOOBIMP uses $J = 5$ in its completion stage. Each entry in Table 43 measures one complete training iteration. After the first iteration, DOOBIMP is faster at every stage, and its average training time per iteration is lower than DiffPuter's.

Table 43: Sequential training time on one RTX A6000. Entries are mean ± population standard deviation over five seeds in seconds for one complete training iteration. Lower is better.

Training iter	DOOBIMP	DiffPuter
1	1073.11±263.59	1067.23±49.14
2	569.62±105.45	997.98±57.12
3	520.10±66.98	998.16±61.02
4	515.71±51.53	965.98±37.23
5	466.63±38.67	951.06±35.91
6	505.96±71.04	962.44±48.33
7	543.11±78.17	982.16±78.40
8	560.15±28.96	997.20±32.98
9	547.11±52.85	964.91±58.03
10	469.13±51.79	974.65±39.38
Average	**577.06±34.78**	986.18±16.66

F TIME-SERIES EXPERIMENTAL DETAILS

F.1 DATASETS AND MISSINGNESS SETTINGS

We follow the data processing and evaluation protocol of the Time-Series Imputation Benchmark (TSI-Bench) (Du et al., 2024b), as subsequently adopted by Hybrid Encoding with Learnable Identity and Cross-dimensional Synthesis (HELIX) (Zhang et al., 2026). Table 44 summarizes the four evaluated datasets. ETT-h1 denotes the first hourly Electricity Transformer Temperature subset. We partition ETT-h1 and BeijingAir chronologically, partition ItalyAir in chronological proportions of 60%/20%/20%, and use the official stratified PhysioNet2012 split. All series are standardized using statistics fitted on the training partition and then applied unchanged to validation and test data.

ETT-h1 contains electricity-transformer load and oil-temperature measurements from July 2016 through June 2018; the first 14 months form training data and the following two five-month periods form validation and test data. BeijingAir contains 11 hourly variables from each of 12 monitoring sites between March 2013 and February 2017; its chronological split contains 28, 10, and 10 months. ItalyAir contains hourly sensor and reference-analyzer measurements from March 2004 through February 2005. We form nonoverlapping windows of lengths 48, 24, and 12 for ETT-h1, BeijingAir, and ItalyAir, respectively. PhysioNet2012 uses subset A of the 2012 PhysioNet/Computing in Cardiology Challenge, with each sample representing the first 48 intensive-care hours. We use a fixed split stratified by in-hospital mortality.

Table 44: Time-series datasets. The training, validation, and test partitions contain disjoint samples.

Dataset	Samples	Variables	Length	Interval	Train	Validation/Test
ETT-h1	358	7	48	1 hour	212	75/71
BeijingAir	1,458	132	24	1 hour	851	304/303
ItalyAir	774	13	12	1 hour	466	154/154
PhysioNet2012	3,997	35	48	1 hour	2,557	640/800

We evaluate four induced-missingness settings spanning three pattern types: point, subsequence, and block. Point-10% and Point-50% independently hide 10% and 50% of the originally observed coordinates, respectively. Subsequence-50% hides a contiguous temporal interval within a randomly selected sample–feature series, and Block-50% hides rectangles spanning consecutive time steps and adjacent features. The latter two names indicate their nominal missing rates; overlap and natural missingness can make the realized rate differ slightly. Table 45 gives the exact TSI-Bench con-

struction parameters. The block factor is the sampling rate supplied to the official block generator; together with the stated block size, it produces the nominal 50% rate. For every dataset–missingness setting, we independently apply the same missingness distribution to the training, validation, and test partitions using a fixed random seed for missingness generation. The training partition is therefore incomplete under the designated pattern rather than observed in full. This partition-level missingness is distinct from additional artificial masking used internally to construct training targets. On these already incomplete training inputs, we directly follow each baseline's artificial-mask construction: HELIX hides 20% of observed entries under MCAR, while CSDI uses random target masking (Zhang et al., 2026; Tashiro et al., 2021). In contrast, DOOBIMP introduces no additional artificial masking during training. For PhysioNet2012, we evaluate the Point-10% setting used in the benchmark.

Table 45: Induced missingness settings applied independently to the training, validation, and test partitions. The first and second block-size entries give the numbers of consecutive time steps and adjacent features, respectively. A dash indicates that the setting is not evaluated.

Dataset	Point rates	Subsequence length	Block size	Block factor
ETT-h1	10%, 50%	36	6×6	0.36
BeijingAir	10%, 50%	18	6×6	0.0055
ItalyAir	10%, 50%	8	4×4	0.07
PhysioNet2012	10%	–	–	–

F.2 EVALUATION PROTOCOL AND BASELINES

The training, validation, and test partitions have disjoint roles. For each dataset–missingness setting, we train the full DOOBIMP pipeline on the corresponding incomplete training partition: initialization uses only its observed entries, the completion stages fill its missing entries with DOOBCOND, and the refitting stages learn unconditional scores from the resulting completions. Validation mean squared error selects the network and training configuration. Within each dataset, the selected DOOBCOND configuration is shared across all missingness patterns; validation also selects the DOOBIMP iterate for each seed. The selected model then imputes the test partition once. All tables report the mean and sample standard deviation over five seeds.

We evaluate only entries that are originally observed and then artificially hidden; naturally missing entries never enter the metric. Mean absolute error (MAE) averages absolute reconstruction error over these entries, while mean squared error (MSE) averages squared reconstruction error over the same set. Both metrics are computed after applying the training-set standardization, and lower values are better.

Baseline values and tuned configurations follow the HELIX benchmark for naive and predictive methods (Zhang et al., 2026) and TSI-Bench for the generative baselines CSDI, US-GAN, and GP-VAE (Du et al., 2024b; Tashiro et al., 2021; Miao et al., 2021; Fortuin et al., 2020). LOCF denotes last observation carried forward. For DOOBIMP, Tables 46– 48 report the selected configurations and complete search space.

F.3 IMPLEMENTATION AND HYPERPARAMETERS

Score-network architecture. We adapt the Diffusion Transformer (DiT) backbone with adaLN-Zero from Peebles & Xie (2023). Each time point is one token containing all variables, with linear time-series input and output maps replacing image-patch maps. The network receives the noisy state over all variables and diffusion time, but not the missingness indicator M.

For a network input $z \in \mathbb{R}^{P \times F}$, let P denote the window length, F the number of variables, d_h the hidden width, N_{blk} the number of Transformer blocks, and N_{head} the number of attention heads. The quantities d_h, N_{blk}, and N_{head} correspond, respectively, to the Width, Layers, and Heads entries

in Table 46. At diffusion time t, the network is

$$\begin{aligned} e_t &= \mathrm{MLP}_{\mathrm{noise}}\big(\mathrm{SinEmb}\big(\tfrac{1}{4}\log\sigma_t\big)\big) \in \mathbb{R}^{d_h}, \\ H^{(0)} &= \mathrm{Linear}_{\mathrm{in}}(z) + E_{\mathrm{pos}} \in \mathbb{R}^{P\times d_h}, \\ H^{(b)} &= \mathrm{AdaLN}^{(b)}(H^{(b-1)}, e_t), \qquad b = 1,\dots,N_{\mathrm{blk}}, \\ f_\theta(z,t) &= \mathrm{Linear}_{\mathrm{out}}\Big(\mathrm{AdaLN}_{\mathrm{out}}(H^{(N_{\mathrm{blk}})}, e_t)\Big) \in \mathbb{R}^{P\times F}. \end{aligned}$$

Here σ_t is the forward noise scale, $\mathrm{SinEmb} : \mathbb{R} \to \mathbb{R}^{d_h}$ is the standard sinusoidal embedding, $\mathrm{MLP}_{\mathrm{noise}}$ is a two-layer multilayer perceptron with a Gaussian error linear unit (GELU), $\mathrm{Linear}_{\mathrm{in}} : \mathbb{R}^F \to \mathbb{R}^{d_h}$ is applied to every time point, and $E_{\mathrm{pos}} \in \mathbb{R}^{P\times d_h}$ is a learned temporal positional embedding. Each hidden state $H^{(b)}$ has shape $P \times d_h$. The operator $\mathrm{AdaLN}^{(b)}$ is the b-th adaLN-Zero DiT block with multi-head self-attention over the P tokens, N_{head} attention heads, and a GELU multilayer perceptron of hidden width $4d_h$. Adaptive layer normalization uses e_t to modulate normalized activations, thereby conditioning each block and the output layer on diffusion time; we otherwise use the DiT construction directly (Peebles & Xie, 2023). Attention and multilayer-perceptron dropout are both 0.1, and the modulation and output maps are zero-initialized. Finally, $\mathrm{AdaLN}_{\mathrm{out}}$ is the output adaptive layer normalization, and $\mathrm{Linear}_{\mathrm{out}} : \mathbb{R}^{d_h} \to \mathbb{R}^F$ is applied tokenwise.

Optimization and sampling. We use the variance-exploding EDM schedule and solver described in Appendix B.3. Training draws continuous noise from a log-normal distribution with log-space mean -1.2, log-space standard deviation 1.2, and sampled noise levels clipped to $[0.002, 80]$. We optimize with AdamW using learning rate 10^{-3}, zero weight decay, and no early stopping, and use batch size 32 for both training and imputation.

For sampling, we instantiate this schedule on a Karras grid with $L = 50$ steps, minimum and maximum noise levels 0.002 and 80 and schedule exponent 7. The observed-coordinate transition uses its marginal distribution, the variance scales in Algorithm 1 are one after standardization, and the final observed coordinates are projected exactly to x_0^{obs}. We set $\eta = 0.01$. At each completion stage, we average $S = 20$ stochastic completions per training example to form the pseudo-complete sample used for refitting. At evaluation, we independently draw 20 posterior samples and use their mean as the imputation. When $J > 0$, the practical implementation averages the J denoised estimates in (12), consistently with the reverse update in Section 4.2.

Tables 46 and 47 give every selected dataset-specific setting. Each stage trains a freshly initialized network: stage (1) is initialization, and stages (2,. . . ,K) are DOOBFIT refitting stages.

For numerical stability, we bound the conditioning update in (11) samplewise:

$$\begin{aligned} &\frac{\sigma_{t_\ell}^2}{\alpha_{t_\ell}\tau^2}\nabla_{x^{\mathrm{mis}}}\log\widehat{h}_\eta(x,t_\ell) \\ &\quad\leftarrow \min\left\{1, \frac{c}{\left\|\frac{\sigma_{t_\ell}^2}{\alpha_{t_\ell}\tau^2}\nabla_{x^{\mathrm{mis}}}\log\widehat{h}_\eta(x,t_\ell)\right\|_2}\right\}\frac{\sigma_{t_\ell}^2}{\alpha_{t_\ell}\tau^2}\nabla_{x^{\mathrm{mis}}}\log\widehat{h}_\eta(x,t_\ell), \end{aligned}$$

where $c > 0$ is the update bound and the multiplicative factor is one when the norm vanishes. Under the reverse-grid indexing of Algorithm 1, conditioning is active exactly when $1 \le W \le \ell$.

Table 46: Selected architecture and training hyperparameters. Initial and refitting epochs refer to each freshly initialized model. The value K counts all trained stages: stage 1 is initialization, and stages $2,\dots,K$ are DOOBFIT refitting stages.

Dataset	Width	Layers	Heads	Parameters	Initial epochs	K	Refitting epochs
ETT-h1	128	6	4	1,855,111	500	2	500
BeijingAir	320	4	4	7,895,492	333	3	333
ItalyAir	96	4	4	710,317	200	5	200
PhysioNet2012	128	4	4	1,268,643	200	5	200

Table 47: Selected DOOBCOND hyperparameters. All settings are shared across missingness patterns within a dataset.

Dataset	τ	W	J	c
ETT-h1	0.30	15	1	20
BeijingAir	0.35	15	1	30
ItalyAir	0.25	15	1	20
PhysioNet2012	0.30	20	3	20

Table 48: Complete hyperparameter search space for our time-series experiments. Each epoch-allocation tuple specifies initial-model epochs, the total number K of trained stages, and epochs per refitting stage. Entries spanning all dataset columns are shared across datasets, and widths are instantiated only when divisible by the selected number of attention heads.

Parameter	**ETT-h1**	**BeijingAir**	**ItalyAir**	**PhysioNet2012**
Model Architecture				
Width	$\{96, 128, 160\}$	$\{192, 256, 320\}$	$\{64, 96, 128\}$	$\{64, 96, 128\}$
Transformer layers	$\{2, 4, 6\}$	$\{2, 4, 6\}$	$\{2, 4, 6\}$	$\{2, 4, 6\}$
Attention heads	$\{2, 4, 6\}$	$\{2, 4, 6\}$	$\{2, 4, 6\}$	$\{2, 4, 6\}$
Training Configuration				
Epoch allocation	$\{(500, 2, 500), (333, 3, 333), (200, 5, 200)\}$			
Doob Conditioning				
Temperature τ	$\{0.25, 0.30, 0.35\}$			
Activation index W	$\{15, 20, 25\}$			
Refinement steps J	$\{1, 3, 5\}$			
Update bound c	$\{10, 20, 30\}$			

F.4 COMPLETE RESULTS

We report complete MAE and MSE results for every evaluated dataset–missingness setting. The best mean in each column is bold and the second best is underlined. A dash indicates that the source did not report the result. All MAE and MSE values are shown to three decimal places; emphasis is determined from the means before rounding.

The predictive baselines are HELIX (Zhang et al., 2026), ImputeFormer (Nie et al., 2024), SAITS (Du et al., 2023), Nonstationary Transformer (Liu et al., 2022), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), TEFN (Zhan et al., 2025), TimeMixer (Wang et al., 2024), TimeMixer++ (Wang et al., 2025), ModernTCN (Luo & Wang, 2024), StemGNN (Cao et al., 2020), TOTEM (Talukder et al., 2024), FreTS (Yi et al., 2023), Time-LLM (Jin et al., 2024), and MOMENT (Goswami et al., 2024). The generative baselines are CSDI (Tashiro et al., 2021), USGAN (Miao et al., 2021), and GP-VAE (Fortuin et al., 2020).

F.4.1 RESULTS ON BEIJINGAIR

Table 49: Complete mean absolute error (MAE) on BeijingAir. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined. The table also reports trainable parameter counts; a dash denotes an unavailable value.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%	Parameters
Naive baselines					
Linear interpolation	0.112	0.165	0.285	0.358	–
LOCF	0.188	0.264	0.400	0.482	–
Median imputation	0.681	0.677	0.682	0.676	–
Mean imputation	0.721	0.708	0.714	0.711	–
Predictive neural baselines					
HELIX	**0.073±0.004**	**0.102±0.005**	0.131±0.005	0.166±0.009	4.6M
ImputeFormer	0.095±0.005	0.122±0.002	0.158±0.004	0.322±0.009	3.4M
SAITS	0.151±0.001	0.190±0.003	0.207±0.005	0.217±0.005	7.2M
Nonstationary Transformer	0.178±0.003	0.201±0.002	0.275±0.003	0.331±0.009	7.0M
PatchTST	0.181±0.002	0.214±0.005	0.273±0.002	0.301±0.003	30.3M
iTransformer	0.124±0.004	0.160±0.003	0.349±0.061	0.626±0.010	8.3M
TEFN	0.141±0.004	0.198±0.001	0.315±0.001	0.353±0.001	37.5K
TimeMixer	0.278±0.007	0.293±0.010	0.300±0.007	0.311±0.003	168.7K
TimeMixer++	0.532±0.061	0.615±0.020	0.734±0.010	0.738±0.010	33.5M
ModernTCN	0.257±0.012	0.371±0.017	0.492±0.020	0.590±0.014	14.1M
StemGNN	0.170±0.004	0.185±0.003	0.233±0.007	0.270±0.007	1.1M
TOTEM	0.322±0.206	0.452±0.013	0.682±0.007	0.725±0.015	98.9K
FreTS	0.193±0.006	0.217±0.005	0.260±0.016	0.294±0.002	909.9K
Time-LLM	0.211±0.005	0.223±0.013	0.469±0.055	0.607±0.024	31.8M
MOMENT	0.607±0.650	1.383±1.247	0.854±0.039	0.799±0.221	109.8M
Deep generative models					
DOOBIMP (ours)	0.077±0.001	0.111±0.000	**0.131±0.000**	**0.142±0.000**	7.9M
CSDI	0.102±0.010	0.144±0.007	0.181±0.025	0.211±0.031	244.8K
US-GAN	0.137±0.002	0.192±0.001	0.216±0.001	0.233±0.002	6.1M
GP-VAE	0.240±0.006	0.258±0.004	0.294±0.009	0.321±0.005	1.0M

Table 50: Complete mean squared error (MSE) on BeijingAir. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%
Naive baselines				
Linear interpolation	0.114	0.231	0.418	0.588
LOCF	0.243	0.429	0.715	0.832
Median imputation	1.093	1.143	1.175	1.155
Mean imputation	1.035	1.078	1.114	1.102
Predictive neural baselines				
HELIX	0.151±0.009	0.172±0.010	0.204±0.008	<u>0.233±0.010</u>
ImputeFormer	<u>0.095±0.021</u>	<u>0.156±0.017</u>	<u>0.203±0.010</u>	0.470±0.013
SAITS	0.129±0.008	0.180±0.007	0.217±0.009	0.237±0.014
Nonstationary Transformer	0.227±0.004	0.252±0.004	0.365±0.004	0.451±0.021
PatchTST	0.138±0.004	0.235±0.005	0.339±0.003	0.383±0.003
iTransformer	0.218±0.004	0.255±0.004	0.472±0.080	0.914±0.009
TEFN	0.194±0.015	0.252±0.003	0.430±0.003	0.517±0.002
TimeMixer	0.381±0.008	0.386±0.007	0.404±0.005	0.427±0.003
TimeMixer++	0.664±0.112	0.829±0.033	1.116±0.015	1.122±0.023
ModernTCN	0.297±0.013	0.437±0.024	0.633±0.033	0.821±0.024
StemGNN	0.272±0.009	0.292±0.004	0.348±0.007	0.393±0.006
TOTEM	0.461±0.353	0.570±0.024	1.008±0.006	1.106±0.038
FreTS	0.179±0.014	0.260±0.012	0.320±0.019	0.384±0.003
Time-LLM	0.232±0.005	0.272±0.010	0.625±0.082	0.893±0.039
MOMENT	2.981±5.308	6.680±8.829	1.549±0.141	1.446±0.509
Deep generative models				
DOOBIMP (ours)	**0.061±0.001**	**0.109±0.001**	**0.157±0.002**	**0.146±0.003**
CSDI	0.507±0.235	0.472±0.155	0.399±0.172	0.512±0.323
US-GAN	0.100±0.006	0.187±0.005	0.233±0.006	0.269±0.010
GP-VAE	0.187±0.007	0.234±0.008	0.307±0.021	0.343±0.013

F.4.2 RESULTS ON ETT-H1

Table 51: Complete mean absolute error (MAE) on ETT-h1. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined. The table also reports trainable parameter counts; a dash denotes an unavailable value.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%	Parameters
Naive baselines					
Linear interpolation	0.197	0.267	0.527	0.722	–
LOCF	0.315	0.425	0.721	0.809	–
Median imputation	0.710	0.708	0.712	0.784	–
Mean imputation	0.737	0.738	0.720	0.773	–
Predictive neural baselines					
HELIX	<u>0.128±0.005</u>	<u>0.189±0.012</u>	<u>0.372±0.015</u>	<u>0.489±0.014</u>	803.5K
ImputeFormer	0.202±0.044	0.296±0.036	0.404±0.021	0.520±0.017	124.0K
SAITS	0.150±0.007	0.208±0.009	0.422±0.019	0.620±0.016	88.2M
Nonstationary Transformer	0.301±0.009	0.358±0.007	0.483±0.007	0.586±0.009	589.9K
PatchTST	0.229±0.016	0.272±0.027	0.529±0.008	0.689±0.030	72.2K
iTransformer	0.269±0.003	0.339±0.003	0.494±0.004	0.740±0.014	23.7M
TEFN	0.307±0.003	0.475±0.019	0.576±0.015	0.672±0.015	985
TimeMixer	0.578±0.231	0.699±0.121	0.768±0.039	0.877±0.011	11.1K
TimeMixer++	0.333±0.007	0.378±0.006	0.515±0.008	0.665±0.008	11.2M
ModernTCN	0.298±0.011	0.435±0.012	0.593±0.010	0.771±0.005	175.7K
StemGNN	0.277±0.005	0.314±0.014	0.426±0.011	0.611±0.009	1.6M
TOTEM	0.368±0.195	0.526±0.026	0.741±0.009	0.868±0.002	23.8K
FreTS	0.264±0.020	0.294±0.022	0.505±0.014	0.661±0.019	465.3K
Time-LLM	0.290±0.001	0.417±0.030	0.614±0.004	0.775±0.004	31.1M
MOMENT	0.550±0.145	0.757±0.129	0.838±0.013	0.924±0.026	109.7M
Deep generative models					
DOOBIMP (ours)	**0.115±0.002**	**0.177±0.003**	**0.307±0.005**	**0.391±0.015**	1.9M
CSDI	0.151±0.008	0.318±0.016	0.401±0.067	0.558±0.048	1.2M
US-GAN	0.458±0.590	0.755±0.973	0.815±0.459	0.670±0.022	3.8M
GP-VAE	0.329±0.017	0.414±0.013	0.651±0.077	0.806±0.055	384.8K

Table 52: Complete mean squared error (MSE) on ETT-h1. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%
Naive baselines				
Linear interpolation	0.106	0.178	0.699	1.322
LOCF	0.277	0.491	1.317	1.619
Median imputation	1.044	1.022	1.079	1.337
Mean imputation	0.990	0.971	0.973	1.194
Predictive neural baselines				
HELIX	<u>0.049±0.004</u>	<u>0.094±0.009</u>	<u>0.320±0.030</u>	<u>0.659±0.029</u>
ImputeFormer	0.100±0.039	0.199±0.047	0.352±0.029	0.712±0.037
SAITS	0.054±0.004	0.102±0.007	0.375±0.033	0.851±0.055
Nonstationary Transformer	0.190±0.015	0.268±0.009	0.478±0.011	0.817±0.027
PatchTST	0.115±0.009	0.152±0.024	0.544±0.025	1.024±0.106
iTransformer	0.153±0.003	0.224±0.005	0.461±0.006	1.105±0.019
TEFN	0.231±0.005	0.509±0.047	0.718±0.046	1.173±0.076
TimeMixer	0.684±0.428	0.909±0.252	1.061±0.086	1.396±0.029
TimeMixer++	0.215±0.005	0.266±0.006	0.499±0.012	0.870±0.026
ModernTCN	0.182±0.012	0.362±0.019	0.671±0.020	1.143±0.009
StemGNN	0.155±0.006	0.191±0.015	0.358±0.018	0.815±0.027
TOTEM	0.313±0.305	0.525±0.048	0.976±0.022	1.364±0.006
FreTS	0.140±0.019	0.169±0.022	0.507±0.030	0.944±0.056
Time-LLM	0.168±0.002	0.321±0.039	0.681±0.018	1.180±0.004
MOMENT	0.584±0.322	0.997±0.322	1.258±0.070	1.524±0.097
Deep generative models				
DOOBIMP (ours)	**0.039±0.003**	**0.085±0.003**	**0.253±0.008**	**0.392±0.025**
CSDI	0.057±0.005	0.207±0.011	0.340±0.100	0.783±0.127
US-GAN	1.405±2.683	2.119±3.955	1.549±1.788	0.901±0.059
GP-VAE	0.202±0.017	0.301±0.011	0.812±0.169	1.265±0.147

F.4.3 IMPUTATION EXAMPLES ON ETT-H1

Figures 16 and 17 show five ETT-h1 imputation examples for each missingness setting. Observed values are gray, held-out values are red, and the predictions of DOOBIMP, HELIX, and CSDI are shown in dashed teal, solid amber, and dotted purple, respectively. The shaded regions are empirical 90% intervals from 20 samples for DOOBIMP and CSDI. Predictions and intervals are displayed only over held-out spans and are anchored to the neighboring observed values.

HELIX is deterministic and produces only point imputations, whereas CSDI and DOOBIMP also quantify uncertainty through empirical 90% intervals. In these examples, the CSDI intervals are often overly diffuse, and the point imputations from both CSDI and HELIX deviate more from the held-out values. DOOBIMP follows the held-out trajectories more closely while providing tighter intervals. This qualitative behavior agrees with Tables 51 and 52, where DOOBIMP achieves the lowest MAE and MSE across all four ETT-h1 missingness settings.

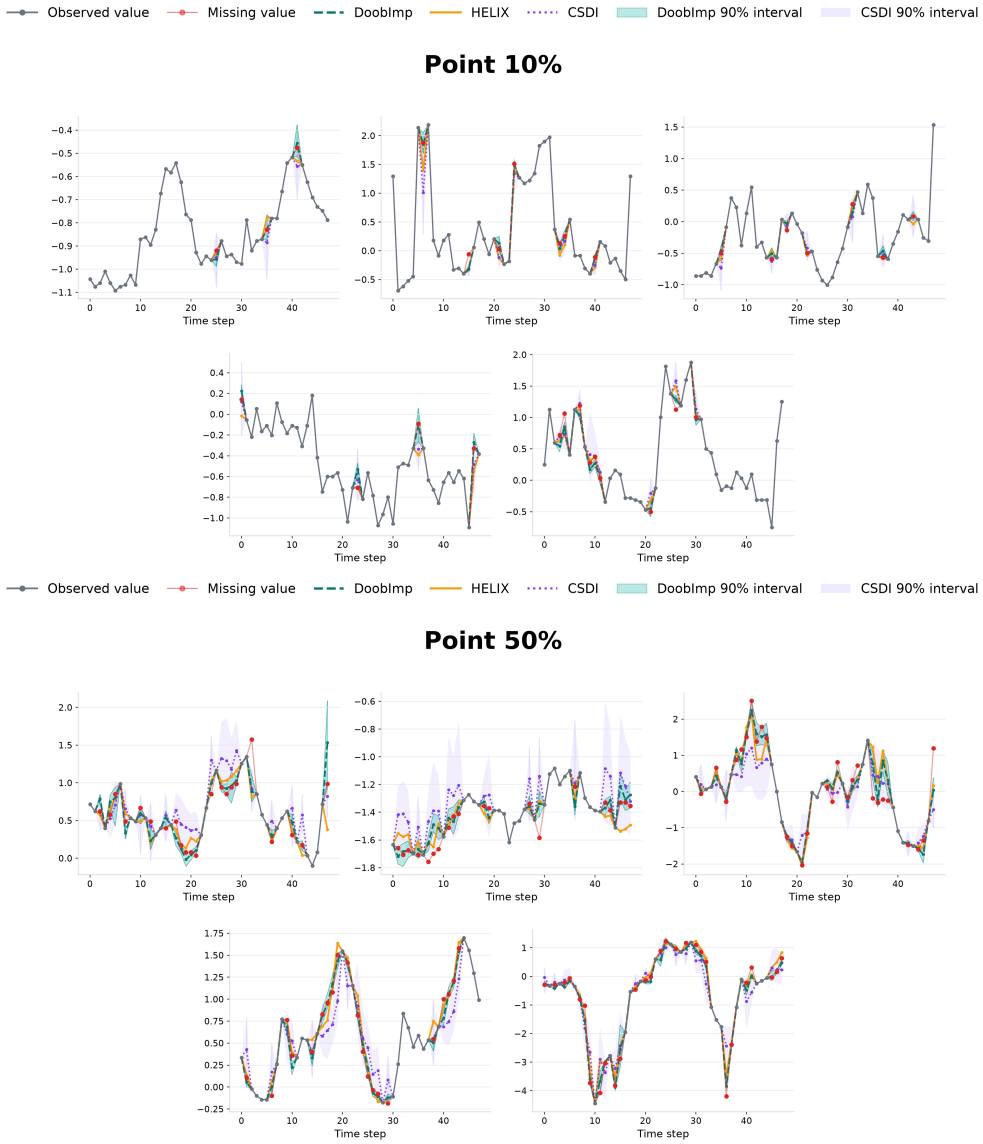


Figure 16: ETT-h1 imputation examples under point missingness at rates 10% (top) and 50% (bottom). Each panel shows one sample-feature trajectory.

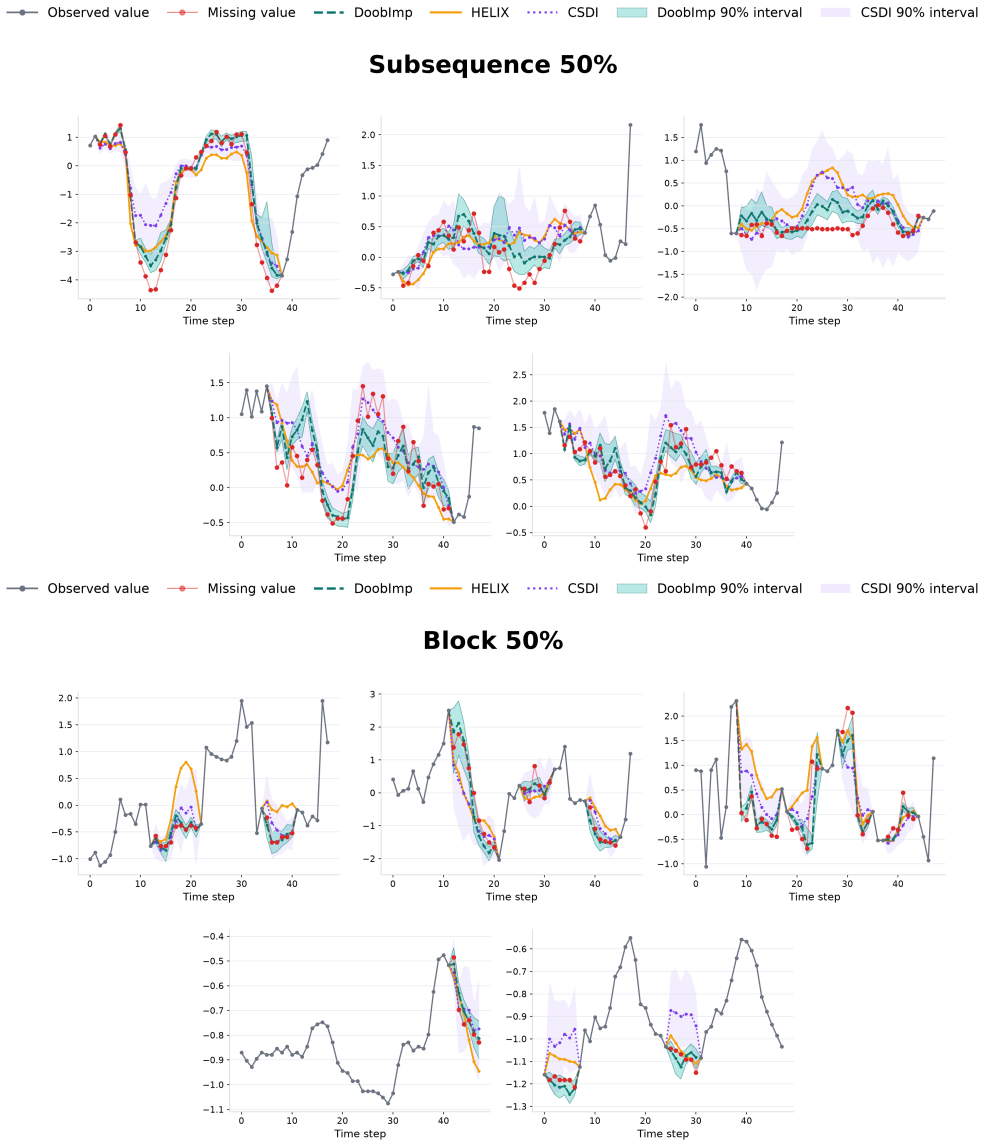


Figure 17: ETT-h1 imputation examples under 50% subsequence missingness (top) and 50% block missingness (bottom). Each panel shows one sample-feature trajectory.

F.4.4 RESULTS ON ITALYAIR

Table 53: Complete mean absolute error (MAE) on ItalyAir. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined. The table also reports trainable parameter counts; a dash denotes an unavailable value.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%	Parameters
Naive baselines					
Linear interpolation	0.135	0.214	0.377	0.329	–
LOCF	0.233	0.346	0.493	0.528	–
Median imputation	0.518	0.533	0.589	0.549	–
Mean imputation	0.574	0.588	0.625	0.612	–
Predictive neural baselines					
HELIX	<u>0.122±0.010</u>	<u>0.203±0.009</u>	<u>0.332±0.011</u>	<u>0.297±0.015</u>	77.3K
ImputeFormer	0.192±0.017	0.282±0.015	0.397±0.019	0.356±0.009	149.1K
SAITS	0.173±0.004	0.276±0.008	0.413±0.015	0.396±0.006	16.6M
Nonstationary Transformer	0.230±0.004	0.281±0.004	0.417±0.009	0.381±0.008	8.4M
PatchTST	0.271±0.018	0.340±0.019	0.514±0.015	0.496±0.006	5.1M
iTransformer	0.220±0.010	0.319±0.008	0.504±0.012	0.556±0.017	18.9M
TEFN	0.142±0.001	0.267±0.004	0.421±0.001	0.391±0.004	551
TimeMixer	0.580±0.157	0.664±0.105	0.797±0.044	0.751±0.032	9.6K
TimeMixer++	0.615±0.054	0.642±0.018	0.733±0.020	0.730±0.018	4.2M
ModernTCN	0.348±0.015	0.461±0.016	0.614±0.014	0.599±0.008	58.1K
StemGNN	0.277±0.028	0.301±0.008	0.448±0.010	0.425±0.018	211.6K
TOTEM	0.677±0.108	0.698±0.060	0.815±0.021	0.757±0.028	27.9K
FreTS	0.279±0.018	0.320±0.010	0.498±0.017	0.447±0.008	668.3K
Time-LLM	0.287±0.008	0.436±0.016	0.682±0.005	0.625±0.004	31.1M
MOMENT	–	–	–	–	–
Deep generative models					
DOOBIMP (ours)	**0.077±0.001**	**0.127±0.002**	**0.253±0.006**	**0.185±0.007**	710.3K
CSDI	0.539±0.418	0.958±0.551	0.675±0.358	0.477±0.115	933.2K
US-GAN	0.264±0.012	0.357±0.009	0.484±0.007	0.446±0.025	3.9M
GP-VAE	0.369±0.012	0.453±0.014	0.556±0.012	0.544±0.018	130.6K

Table 54: Complete mean squared error (MSE) on ItalyAir. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each setting is bold and the second best is underlined.

Model	Point – 10%	Point – 50%	Block – 50%	Subsequence – 50%
Naive baselines				
Linear interpolation	0.106	0.252	0.478	0.407
LOCF	0.220	0.511	0.736	0.900
Median imputation	1.045	1.116	1.192	1.151
Mean imputation	1.034	1.096	1.163	1.130
Predictive neural baselines				
HELIX	0.066±0.015	0.149±0.012	0.322±0.027	0.301±0.024
ImputeFormer	0.140±0.016	0.253±0.026	0.444±0.045	0.378±0.036
SAITS	0.088±0.004	0.204±0.009	0.406±0.031	0.373±0.018
Nonstationary Transformer	0.184±0.011	0.289±0.008	0.534±0.027	0.472±0.011
PatchTST	0.188±0.021	0.302±0.021	0.600±0.020	0.615±0.005
iTransformer	0.173±0.009	0.322±0.009	0.652±0.027	0.712±0.046
TEFN	0.115±0.003	0.285±0.006	0.550±0.003	0.483±0.010
TimeMixer	0.945±0.361	1.208±0.279	1.551±0.126	1.431±0.090
TimeMixer++	1.002±0.091	1.120±0.082	1.390±0.068	1.311±0.036
ModernTCN	0.368±0.029	0.605±0.037	0.961±0.041	0.940±0.025
StemGNN	0.215±0.032	0.262±0.008	0.478±0.017	0.455±0.023
TOTEM	1.202±0.445	1.258±0.257	1.614±0.052	1.464±0.055
FreTS	0.189±0.017	0.281±0.012	0.592±0.040	0.504±0.023
Time-LLM	0.282±0.021	0.533±0.021	1.173±0.014	1.028±0.025
MOMENT	–	–	–	–
Deep generative models				
DOOBIMP (ours)	**0.033±0.001**	**0.077±0.001**	**0.243±0.017**	**0.120±0.004**
CSDI	23.503±35.685	29.266±31.183	7.791±11.133	2.273±1.532
US-GAN	0.144±0.011	0.278±0.011	0.451±0.008	0.406±0.031
GP-VAE	0.314±0.016	0.495±0.022	0.649±0.044	0.661±0.038

F.4.5 RESULTS ON PHYSIONET2012

Table 55: Complete mean absolute error (MAE) and mean squared error (MSE) on PhysioNet2012 Point-10%. Values are the mean ± sample standard deviation over five seeds; lower is better. The best mean in each metric is bold and the second best is underlined. The Parameters column reports trainable parameter counts; a dash denotes an unavailable value.

Model	MAE	MSE	Parameters
Naive baselines			
Linear interpolation	0.366	0.425	–
LOCF	0.449	0.604	–
Median imputation	0.690	1.000	–
Mean imputation	0.708	0.967	–
Predictive neural baselines			
HELIX	**0.215±0.002**	**0.206±0.003**	454.9K
ImputeFormer	0.240±0.009	<u>0.229±0.008</u>	1.4M
SAITS	0.247±0.010	0.229±0.009	44.3M
Nonstationary Transformer	0.316±0.002	0.300±0.003	1.6M
PatchTST	0.282±0.010	0.274±0.015	644.1K
iTransformer	0.373±0.003	0.360±0.005	6.9M
TEFN	0.364±0.001	0.376±0.002	3.1K
TimeMixer	0.429±0.003	0.450±0.006	73.4K
TimeMixer++	0.416±0.002	0.410±0.003	44.7M
ModernTCN	0.499±0.006	0.573±0.016	850.2K
StemGNN	0.391±0.034	0.392±0.048	6.5M
TOTEM	0.474±0.012	0.519±0.017	86.6K
FreTS	0.317±0.012	0.314±0.016	1.9M
Time-LLM	0.457±0.003	0.474±0.006	31.1M
MOMENT	0.600±0.133	0.752±0.269	35.4M
Deep generative models			
DOOBIMP (ours)	<u>0.222±0.001</u>	0.231±0.002	1.3M
CSDI	0.252±0.004	0.313±0.039	1.5M
US-GAN	0.310±0.003	0.318±0.005	4.3M
GP-VAE	0.445±0.006	0.499±0.011	501.7K