Generative Modeling under Non-Monotone MAR Missingness via Approximate Wasserstein Gradient Flows

Gitte Kremling
Department of Mathematics
University of Hamburg
gitte.kremling@uni-hamburg.de

Jeffrey Näf
Research Institute for Statistics and Information Science
University of Geneva
jeffrey.naf@unige.ch

Johannes Lederer
Department of Mathematics
University of Hamburg
johannes.lederer@uni-hamburg.de

Abstract

The prevalence of missing values in data science poses a substantial risk to any further analyses. Despite a wealth of research, principled nonparametric methods to deal with general non-monotone missingness are still scarce. Instead, ad-hoc imputation methods are often used, for which it remains unclear whether the correct distribution can be recovered. In this paper, we propose FLOWGEM, a principled iterative method for generating a complete dataset from a dataset with values Missing at Random (MAR). Motivated by convergence results of the ignoring maximum likelihood estimator, our approach minimizes the expected Kullback–Leibler (KL) divergence between the observed data distribution and the distribution of the generated sample over different missingness patterns. To minimize the KL divergence, we employ a discretized particle evolution of the corresponding Wasserstein Gradient Flow, where the velocity field is approximated using a local linear estimator of the density ratio. This construction yields a data generation scheme that iteratively transports an initial particle ensemble toward the target distribution. Simulation studies and real-data benchmarks demonstrate that FLOWGEM achieves state-of-the-art performance across a range of settings, including the challenging case of non-monotone MAR mechanisms. Together, these results position FLOWGEM as a principled and practical alternative to existing imputation methods, and a decisive step towards closing the gap between theoretical rigor and empirical performance.

1 Introduction

Increasing volumes of data have made missing values an almost ubiquitous challenge across disciplines. Despite the steep upturn of research on the topic, there is still a substantial gap between theory and practice. This is particularly true in the challenging case of general non-monotone missingness illustrated in Figure 1: Instead of observing independently and identically distributed data directly, observations are masked according to an unknown missingness mechanism. While this missingness can be monotone, such as in the left example in Figure 1, complex non-monotone patterns of missingness are possible.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

$$\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ \text{NA} & x_{2,2} & x_{2,3} \\ \text{NA} & \text{NA} & x_{3,3} \end{pmatrix} \qquad \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & \text{NA} & x_{2,3} \\ \text{NA} & x_{3,2} & x_{3,3} \end{pmatrix} \qquad \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{1,2} & \text{NA} & x_{2,3} \\ \text{NA} & x_{3,2} & \text{NA} \end{pmatrix}$$

Figure 1: Three data matrices with missing values, each with three different patterns. The matrix on the left shows monotone missingness, while the middle and right show cases with non-monotone missingness. The middle matrix is the one used in the Example of Section 4.

Following Rubin's classical categorization [Rubin, 1976, Mealli and Rubin, 2015], the missing value mechanism can be divided into three categories of increasing complexity: Missing Completely at Random (MCAR), Missing at Random (MAR) and Missing Not at Random (MNAR). In this categorization, MCAR is generally understood to be too weak, while MNAR does not allow for identification in general [Molenberghs et al., 2008]. As such MAR is of special interest and has taken a prominent role in the applied and methodological literature on missing data. At the same time, MAR is a target of frequent misunderstandings, as evidenced by the number of papers simply discussing the condition [Seaman et al., 2013, Mealli and Rubin, 2015, Näf et al., 2026]. This, in addition to the scattered literature, might explain why there has been limited theoretical development for the case of non-monotone MAR. In particular, while Rubin [1976] introduced MAR to justify a parametric maximum likelihood approach that allows to model only the data distribution and not the missingness mechanism (thus "ignoring" the missingness), consistency and asymptotic normality of this approach was only shown as recently as Takai and Kano [2013]. Inspecting their approach and also later papers such as Frahm et al. [2020] reveals that these consistency results in the complex case of non-monotone general MAR are deeply connected with the minimization of the Kullback-Leibler (KL) divergence between observed and proposed distributions. In particular, consistency does not hold for general M-estimation under MAR. Using this connection in a Bayesian framework, Näf and Chérief-Abdellatif [2026] derived a minimax rate estimation result for nonparametric density estimation under non-monotone MAR missingness. Based on this, they developed a generative method that takes a dataset contaminated with missing values and returns a sample from the uncontaminated distribution. The generated sample then allows to calculate consistent estimates of any continuous functional of the density. However, as the method is based on Gaussian mixtures, its performance on complex real data is less than ideal, as the authors note themselves.

On the other hand, there is a wealth of empirically well-performing imputation methods that tackle (non-monotone) MAR missingness. Leaving the observed variables unchanged, these methods try to "fill in" the missing values. This has led to a belief that the goal of imputation is to find the best prediction of the missing values. However, as argued in van Buuren [2018], Näf et al. [2026] and others, the goal is really to obtain a generative method that recreates the original uncontaminated distribution, to be able to achieve unbiased estimates of functionals of interest in a second step. As such, both the generative method of Näf and Chérief-Abdellatif [2026] and the wealth of imputation methods arguably share the same goal of recreating the uncontaminated distribution. While there is interesting theory on parametric imputation [Wang and Robins, 1998, Liu et al., 2014, Zhu and Raghunathan, 2015, Guan and Yang, 2024], practitioners often resort to nonparametric imputation such as versions of MICE [van Buuren and Groothuis-Oudshoorn, 2011] and modern neural net and machine learning methods such as MIWAE [Mattei and Frellsen, 2019], GAIN [Yoon et al., 2018], or MIRI [Yu et al., 2025]. However, these nonparametric imputation methods lack theoretical grounding. In particular, it is not even clear whether they identify the right distribution under MAR in a population setting, that is, under perfect estimation, although important progress was made recently in Yu et al. [2025] as outlined in Section 2.3.

We attempt to close the gap between the imputation methods that perform empirically well but lack a theoretical underpinning and the generative method of Näf and Chérief-Abdellatif [2026] that is theoretically sound but empirically underperforming. For this, we follow the idea of the ignoring maximum likelihood estimator. Our main contributions are as follows:

- We show that, under MAR and a positivity condition, the true distribution emerges as the minimizer of the KL divergence on observed values averaged over all patterns.
- We adapt the approximate Wasserstein gradient flow approach of Liu et al. [2024] to develop a nonparametric sampling method, which we call Flow-based Generation for Missing

data (FLOWGEM), that generates a new sample from the uncontaminated distribution by minimizing the average KL divergence between the new distribution and the observed data.

- We demonstrate that FLOWGEM is both theoretically sensible—under suitable assumptions, the distribution of the new sample converges to the true one—and empirically competitive, matching and exceeding the performance of state-of-the-art methods.

The new algorithm simply uses a local linear approximation and is thus fast to fit, while the gradient flow approach allows for great flexibility. All of this is possible with a minimal set of tuning parameters that can be chosen with heuristics that tend to work well for a wide range of datasets. In fact, our empirical section shows that our method matches or exceeds state-of-the-art methods on data ranging from 3-dimensional simulated data to real data with widely varying variances and up to 272 dimensions, all using the same setting.

Just as in Li et al. [2019], Ouyang et al. [2023], Näf and Chérief-Abdellatif [2026], our approach is motivated by the fact that data analysts are typically interested in aspects of the data distribution rather than the data themselves. Consequently, both a completed version of the original dataset, as targeted in imputation methods, as well as a newly generated complete dataset following the same distribution, as targeted in generative methods, serve the purpose of the subsequent analysis. Additionally, new samples do not expose the original data points, making it very useful for privacy-sensitive applications.

The paper proceeds as follows: We first dive into the background of the maximum likelihood estimator under MAR and an approximate Wasserstein gradient flow for complete data, including a further literature comparison, in Section 2. In Section 3, we derive our methodology, FLOWGEM, adapting the Wasserstein gradient flow framework to the missing data setting and establishing consistency of the velocity field approximation. Section 4 shows the impressive empirical performance of FLOWGEM in a challenging simulated example, designed to test imputation methods for their ability to deal with MAR, as well as a broad range of real datasets. Finally, in Section 5, we summarize our findings and outline directions for future research. Proofs, additional results, and technical details are provided in the Appendix. An open-source implementation of the proposed method is available as a Python package at `https://pypi.org/project/flowgem/`.

2 Background

This section establishes background on missing values, the ignoring maximum likelihood estimator, as well as the Wasserstein gradient flow approximation of Liu et al. [2024] with complete data.

We start by introducing the notation used throughout the paper. A table of key terminology is provided in Appendix A. Let $X \in \mathbb{R}^d$ denote a random vector with distribution $\mathbb{P}_X$ and density π, representing a complete observation of all feature variables, and $M \in \{0,1\}^d$ denote a missingness pattern with (discrete) distribution $\mathbb{P}_M$ and support $\mathcal{M} = \{m \in \{0,1\}^d \mid \mathbb{P}(M = m) > 0\}$. Given an independently and identically distributed sample $(X_i, M_i)_{i=1}^n$ of the joint distribution $\mathbb{P}_{X,M}$, in practice, we only observe the masked observations

$$X_{i,j}^* = \begin{cases} X_{i,j} & \text{for } M_{i,j} = 0 \\ \text{NA} & \text{for } M_{i,j} = 1 \end{cases}, \quad \text{with } i \in \{1,\dots,n\},\ j \in \{1,\dots,d\}.$$

For example, if the complete observation $X_1 = (X_{1,1}, X_{1,2}, X_{1,3})$ is associated to the missingness pattern $M_1 = (0,1,0)$, then we observe $X_1^* = (X_{1,1}, \text{NA}, X_{1,3})$. For $m \in \{0,1\}^d$, let $X^{(m)} = (X_j \mid m_j = 0,\ j \in \{1,\dots,d\})$ denote the sub-vector of X corresponding to the observed values according to m. In the previous example, we have $X_1^{(M_1)} = (X_{1,1}, X_{1,3})$. The distributions of $X^{(m)}$ and $X^{(m)} \mid M = m$ are denoted by $\mathbb{P}_X^{(m)}$ and $\mathbb{P}_{X|M=m}^{(m)}$ and their densities by $\pi^{(m)}$ and $\pi_m^{(m)}$, respectively. The goal of this paper is to generate a new sample $\{\tilde{X}_i\}_{i=1}^{\tilde{n}}$ of fully observed data points that (approximately) follow the true distribution $\mathbb{P}_X$.

Different assumptions can be made about the dependence between X and M. The missingness mechanism is called Missing Completely at Random (MCAR) if $\mathbb{P}_{X,M} = \mathbb{P}_X \otimes \mathbb{P}_M$, that is, if X and M are independent, and, on the other extreme, Missing Not at Random (MNAR) if the dependence structure between M and X can be arbitrary. If, on the other hand, it holds that $\mathbb{P}(M = m \mid X = x) = \mathbb{P}(M = m \mid X^{(m)} = x^{(m)})$, the mechanism is called Missing at

Random (MAR). We note that, here, m occurs on both sides of the probability statement, making the condition more complex than it may first seem. In particular, MAR does not mean that M is conditionally independent of X given $X^{(m)}$. We refer to the discussion in Mealli and Rubin [2015], Näf et al. [2026].

2.1 The ignoring maximum likelihood estimator

Under missing values, we essentially observe draws from $\mathbb{P}^{(m)}_{X|M=m}$ for different patterns m. MAR does not prohibit distribution shifts between patterns and can lead to arbitrary complex non-monotone patterns. This is one of the key reasons why results under general MAR are difficult to obtain.

Nonetheless, if a parametric model $\mathbb{P}_\theta$ is used to approximate the distribution $\mathbb{P}_X$, the parameter θ can be estimated via maximum likelihood under MAR. As shown in Golden et al. [2019] and discussed in Chérief-Abdellatif and Näf [2025, Theorem 2.2], the maximum likelihood estimator (MLE) θ_n^{ML} *ignoring the missing value mechanism* converges $\mathbb{P}_{X,M}$-a.s. to

$$\theta_\infty^{\text{ML}} = \operatorname*{arg\,min}_{\theta\in\Theta} \mathbb{E}_{M\sim\mathbb{P}_M}\left[\text{KL}\left(\mathbb{P}^{(M)}_{X|M} \,\|\, \mathbb{P}^{(M)}_\theta\right)\right] \tag{1}$$

as the sample size n goes to infinity, assuming that appropriate regularity conditions are satisfied. In other words, the estimator converges to a minimizer of the expected Kullback-Leibler (KL) divergence over all possible missingness patterns. Remarkably, when the model is well-specified, that is $\mathbb{P}_X = \mathbb{P}_{\theta^*}$ for some $\theta^* \in \Theta$, and the missingness mechanism is MAR, we recover $\theta_\infty^{\text{ML}} = \theta^*$. It turns out that this is deeply related to the KL divergence and does not hold for general M-estimators.

2.2 Approximated Wasserstein gradient flow without missing values

In this subsection, we summarize the method proposed in Liu et al. [2024] using full data. Suppose we are given an independently and identically distributed sample of complete observations $\{X_i\}_{i=1}^n$ from the distribution $\mathbb{P}_X$ with density π and want to create a new sample $\{\tilde{X}_i\}_{i=1}^{\tilde{n}}$ following a distribution $\mathbb{P}_{\tilde{X}}$ with density ρ, that minimizes the f-divergence

$$D_f[\pi,\rho] = \int \rho(x) f(r(x))\,\mathrm{d}x \quad \text{with} \quad r(x) = \frac{\pi(x)}{\rho(x)}\,.$$

Note that, as a special case, $\text{KL}(\pi \,\|\, \rho) = D_f[\pi,\rho]$ with $f(r) = r\log(r)$. Liu et al. [2024] propose an iterative method to generate such a new sample $\tilde{X}$. More specifically, they use a Wasserstein Gradient Flow (WGF) for the functional $\mathcal{F}[\rho] = D_f[\pi,\rho]$ with an approximated velocity field. Notably, the X_i's are assumed to be fully observed here. Consequently, the corresponding method is not designed to handle missingness and thus needs to be modified to be applicable in the missing value setup, that we are interested in, as discussed in Section 3.

The WGF iteratively modifies the particles $\{\tilde{X}_{t,i}\}_{i=1}^{\tilde{n}}$, starting with $\{\tilde{X}_{0,i}\}_{i=1}^{\tilde{n}}$ drawn from some initial distribution ρ_0. Let ρ_t denote the distribution of $\tilde{X}_t$. Then, according to Liu et al. [2024, Theorem 2.1], briefly explained at the beginning of Section 3, the WGF of $\mathcal{F}[\rho] = D_f[\pi,\rho]$ characterizes the particle evolution via the ordinary differential equation (ODE)

$$\frac{\mathrm{d}\tilde{X}_t}{\mathrm{d}t} = \nabla(h\circ r_t)(\tilde{X}_t) \quad \text{with} \quad h(r) = rf'(r) - f(r) \text{ and } r_t(x) = \frac{\pi(x)}{\rho_t(x)}\,. \tag{2}$$

For the KL divergence, specifically, we have $h(r) = r$. The above ODE moves the density ρ_t of $\tilde{X}_t$ along a curve where increasing t results in decreasing $D_f[\pi,\rho_t]$, as desired. In practice, the ODE above is discretized using the forward Euler method, yielding

$$\tilde{X}_{t+1} = \tilde{X}_t + \eta\nabla(h\circ r_t)(\tilde{X}_t) \quad \text{with} \quad t\in\{0,1,\ldots,T-1\}\,,$$

where, by a slight abuse of notation, we use the index t for the discrete time steps as well. T and η denote the final time and step size, which both need to be chosen by the user.

As we do not have access to the densities π and ρ_t and thus also their ratio r_t, the velocity field $\nabla(h\circ r_t)$ needs to be estimated based on the samples $\{X_i\}_{i=1}^n \sim \pi$ and $\{\tilde{X}_{t,i}\}_{i=1}^{\tilde{n}} \sim \rho_t$ that we do have access to. For that, Liu et al. [2024] exploit the key observation that

$$h\circ r_t = \operatorname*{arg\,max}_{g:\mathbb{R}^d\to\mathbb{R}} \mathbb{E}_{x\sim\pi}\left[g(x)\right] - \mathbb{E}_{x_t\sim\rho_t}\left[\psi^*(g(x_t))\right]\,, \tag{3}$$

where ψ is chosen such that $\psi'(r) = rf'(r) - f(r)$ and ψ^* is the convex conjugate of ψ. For the KL divergence, specifically, we have $\psi(r) = r^2/2$ and $\psi^*(g) = g^2/2$. Based on (3), $h \circ r_t$ is approximated via a local linear estimator. For $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$, let $g_{w,b}(x) = w^T x + b$ be a linear approximation of g. Further, let $k_\sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a given kernel function. Then, locally, for x close to x^*, $(h \circ r_t)(x)$ is approximated by $g_{w_t(x^*),b_t(x^*)}(x)$, where

$$(w_t(x^*), b_t(x^*)) = \underset{w\in\mathbb{R}^d,\, b\in\mathbb{R}}{\arg\max} \frac{1}{n}\sum_{i=1}^{n} k_\sigma(X_i, x^*) g_{w,b}(X_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} k_\sigma(\tilde{X}_{t,i}, x^*)\psi^*(g_{w,b}(\tilde{X}_{t,i}))\,. \tag{4}$$

Note the three differences compared to the optimization target in (3): (i) The feasible set for g, consisting of all functions from $\mathbb{R}^d$ to $\mathbb{R}$, is restricted to the space of linear functions $\{g_{w,b} \mid w \in \mathbb{R}^d, b \in \mathbb{R}\}$, (ii) the expected values are replaced by their empirical counterparts based on the given samples from π and ρ_t, and (iii) the individual data points in the empirical means are weighted according to the kernel function k_σ localizing the approximation. Finally, $\nabla(h \circ r_t)(x^*)$ is approximated by $\nabla g_{w_t(x^*),b_t(x^*)}(x^*) = w_t(x^*)$, and the approximated (discretized) particle evolution becomes

$$\tilde{X}_{t+1} = \tilde{X}_t + \eta w_t(\tilde{X}_t) \quad \text{with} \quad t \in \{0, 1, \ldots, T-1\}$$

and $w_t(\tilde{X}_t)$ computed via the optimization in (4).

In the next section, we describe how the method can be modified to be applicable when the data points X_i are not fully observed but contaminated with missing values.

2.3 Further related literature

The idea of generating new values from observation with missing values itself is not entirely new. Besides Näf and Chérief-Abdellatif [2026], there are several recent papers developing this idea, including Li et al. [2019] and Ouyang et al. [2023]. However, they use GAN and diffusion-based approaches, respectively. Their approaches are thus different from our WGF approach, and in particular, both make use of neural networks for their computation, while our algorithm only uses a tuning-lean local linear approach.

As mentioned above, there is a wealth of imputation methods, which may also be seen as a way of generating complete data. Of particular interest are Chen et al. [2024] and Yu et al. [2025]. The former uses a WGF for imputation, and thus appears very close to our method at first glance. However, while we are able to use the elegant local linear approach of Liu et al. [2024] averaged over patterns, they are using a neural network and Denoising Score Matching, making their algorithm considerably more complex and prone to tuning parameters. Similarly, while Yu et al. [2025] also recognize the importance of minimizing the KL divergence through rectified flows, they need to iteratively decrease and then re-estimate the KL divergence between $\mathbb{P}_{\tilde{X},M}$ at time t and $\mathbb{P}_{\tilde{X}} \otimes \mathbb{P}_M$ at time $t-1$ and also use neural networks. We instead use the idea of the ignoring MLE to directly minimize the KL divergence between our generated data and the observed part of the real data.

One major motivation for data generation instead of imputation is to be able to obtain (identification) results under MAR, motivated by parametric theory. While there are interesting results in Yu et al. [2025] in this direction, they are also somewhat incomplete. In particular, it is unclear whether their optimization target, namely minimizing the mutual information between X and M, yields the true distribution under perfect estimation. In contrast, our approach is theoretically motivated by the parametric case, allowing us to more easily obtain theoretical results. For instance, we are able to show that our method can recover the true distribution under perfect estimation, under MAR and an additional positivity condition (see Proposition 1). Although this positivity condition is not discussed in Yu et al. [2025], it can be shown that it is necessary. The distribution recovered when it does not hold might still be one respecting MAR, but there is no guarantee that it is the correct one, even under perfect estimation. This is similarly true for the interesting results presented in Ouyang et al. [2023]. While they do not mention MAR, they show that under mild conditions on the missing value mechanism, their objective upper bounds the true negative likelihood for a given pattern. However, this again does not guarantee any identification, even under perfect estimation.

Finally, as noted above, we directly employ tools introduced in Liu et al. [2024] adapted to missing values. In fact, Liu et al. [2024] include an application example for missing values using the WGF to minimize $\mathrm{KL}(\mathbb{P}_{\tilde{X},M} \,\|\, \mathbb{P}_{\tilde{X}} \otimes \mathbb{P}_M)$. However, as pointed out in Yu et al. [2025], the WGF is not

applicable for such a target as the optimization variable $\tilde{X}$ appears in both densities, π and ρ. In a WGF, however, the distribution π is assumed to be fixed. Moreover, it is not clear whether this objective is reasonable under MAR. In Appendix A.5, we further discuss the relation of our method to Stein Variational Gradient Descent.

3 FLOWGEM

We now extend the approximate WGF framework of Liu et al. [2024] to the missing data setting. For that, we turn back to the missing value problem, where we observe the contaminated dataset $(X_i^*, M_i)_{i=1}^n$ with missing values $X_{i,j}^* = \text{NA}$ whenever $M_{i,j} = 1$. As motivated by the parametric case, described in Section 2.1, we want to generate a new complete sample $\{\tilde{X}_i\}_{i=1}^{\tilde{n}}$ in a way that the corresponding distribution $\mathbb{P}_{\tilde{X}}$ minimizes the expected KL divergence.

We start by providing the following population result, generalizing the case of the parametric MLE.

Proposition 1 (Population consistency of the KL minimizer)**.** *Assume that MAR holds and* $\mathbb{P}(M = 0 \mid X = x) > 0$ *for almost all* x*. Then* $\mathbb{P}_X$ *is the unique solution to*

$$\arg\min_{P_X} \mathbb{E}_{M\sim\mathbb{P}_M}\left[\mathrm{KL}\left(\mathbb{P}_{X|M}^{(M)} \,\|\, P_X^{(M)}\right)\right]. \tag{5}$$

We note that if either of the two conditions does not hold, i.e. if either $\mathbb{P}(M = 0 \mid X = x) = 0$ on a set of nonzero probability or the missingness mechanism is not MAR, (5) no longer holds. Both might thus be seen as necessary conditions.

Motivated by Proposition 1 and the consistency of the MLE, we now derive a WGF approach to approximately sample from $\mathbb{P}_X$. Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability measures on $\mathbb{R}^d$ with finite second moment, and let $\mathcal{F} : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ be a functional to be minimized. When ρ_t follows the WGF of $\mathcal{F}$, the corresponding probability flow ODE for particles $X_t \sim \rho_t$ takes the form

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = v_t(X_t) \quad \text{with} \quad v_t(x) = -\nabla\frac{\delta\mathcal{F}[\rho]}{\delta\rho}(x)\,.$$

Here, $\delta\mathcal{F}[\rho]/\delta\rho$ denotes the first L^2-variation of $\mathcal{F}$, i.e. the function that satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\rho_\varepsilon] = \int_{\mathbb{R}^d}\frac{\delta\mathcal{F}[\rho]}{\delta\rho}(x)\frac{\partial\rho_\varepsilon(x)}{\partial\varepsilon}\Big|_{\varepsilon=0}\mathrm{d}x$$

for any smooth $\rho_\varepsilon : (-\varepsilon_0, \varepsilon_0) \to \mathcal{P}_2(\mathbb{R}^d)$ with $\rho_0 = \rho$. In case $\mathcal{F}[\rho] = D_f[\pi, \rho]$ is an f-divergence, it can easily be verified that

$$\frac{\delta\mathcal{F}[\rho]}{\delta\rho}(x) = f\left(\frac{\pi(x)}{\rho(x)}\right) - f'\left(\frac{\pi(x)}{\rho(x)}\right)\frac{\pi(x)}{\rho(x)} = -(h\circ r)(x)\,,$$

which gives rise to the particle ODE (2). For our objective, we get the following result.

Lemma 2 (First variation of the objective)**.** *For* $\mathcal{F}[\rho] = \mathbb{E}_{M\sim\mathbb{P}_M}\left[D_f(\pi_M^{(M)}, \rho^{(M)})\right]$*, it holds that*

$$\frac{\delta\mathcal{F}[\rho]}{\delta\rho}(x) = -\mathbb{E}_{M\sim\mathbb{P}_M}\left[(h\circ r^{(M)})(x^{(M)})\right] \quad \textit{with} \quad r^{(M)} = \frac{\pi_M^{(M)}}{\rho^{(M)}}\,.$$

The proof can be found in Appendix A.7. It follows that, in the missing value setup, the particle ODE is given by

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = \nabla\mathbb{E}_{M\sim\mathbb{P}_M}\left[(h\circ r_t^{(M)})(X_t^{(M)})\right] \tag{6}$$

with $h(r) = r$ in the special case of the KL divergence. Next, we derive a result similar to the key observation (3) of Liu et al. [2024], that builds the backbone of our method.

Lemma 3 (Variational form for $h \circ r$)**.** *For each missingness pattern* $m \in \mathcal{M}$*, it holds that*

$$h\circ r^{(m)} = \arg\max_{g_m:\,\mathbb{R}^{d_m}\to\mathbb{R}}\left\{\mathbb{E}_{X^{(m)}\sim\pi_m^{(m)}}\left[g_m(X^{(m)})\right] - \mathbb{E}_{X^{(m)}\sim\rho^{(m)}}\left[\psi^*(g_m(X^{(m)}))\right]\right\}, \tag{7}$$

where $d_m = d - \sum_{j=1}^d m_j$ *denotes the number of observed values under missingness pattern* m*.*

The proof is again deferred to the appendix. Similar to Liu et al. [2024], we estimate the maximizer in (7) by using a local linear approximation for g_m, replacing the expectations by their empirical counterparts and weighting the individual summands with a kernel function. In our specific case, we let $\mathcal{M}_n = \{m \in \mathcal{M} \mid \exists i \in \{1,\dots,n\} \text{ s.t. } M_i = m\}$ denote the set of observed patterns. For each $m \in \mathcal{M}_n$, we define $\{X_{m,i}^{(m)}\}_{i=1}^{n_m} = \{X_i^{*(m)} \mid M_i = m,\ i \in \{1,\dots,n\}\}$ to be the subset of observations with missingness pattern m. This can be interpreted as a sample from the conditional distribution $\mathbb{P}_{X|M=m}^{(m)}$ with density $\pi_m^{(m)}$. Note that the sample size n_m might differ for each missingness pattern m, summing up to n in total. Moreover, for each $m \in \mathcal{M}$, $\{\tilde{X}_{t,i}^{(m)}\}_{i=1}^{\tilde{n}}$ is a sample from $\mathbb{P}_{\tilde{X}_t}^{(m)}$ with density $\rho_t^{(m)}$. A local linear approximation of the maximizer $(h \circ r_t^{(m)})(x)$ for x close to x^* is then given by $g_{w,b}(x) = w^T x + b$ with the optimal choice of (w, b) defined as

$$
\begin{aligned}
(w_{t,m}^{(m)}(x^*), b_{t,m}(x^*)) = \underset{w\in\mathbb{R}^{d_m}, b\in\mathbb{R}}{\arg\max} \frac{1}{n_m} \sum_{i=1}^{n_m} k_\sigma(X_{m,i}^{(m)}, x^{*(m)}) g_{w,b}(X_{m,i}^{(m)}) \\
- \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma(\tilde{X}_{t,i}^{(m)}, x^{*(m)}) \psi^*(g_{w,b}(\tilde{X}_{t,i}^{(m)}))\,. \qquad (8)
\end{aligned}
$$

For the KL divergence, we use $\psi^*(g) = g^2/2$. Consequently, the gradient $\nabla(h \circ r_t^{(m)})(x^*)$ can be estimated by $\nabla g_{w_{t,m}(x^*), b_{t,m}(x^*)}(x^*) = w_{t,m}(x^*)$, whereby $w_{t,m}(x^*)$ is the same as $w_{t,m}^{(m)}(x^*)$ for any index j with $m_j = 0$ and zero otherwise. Using this approximation in (6) and replacing the expectation over M by its empirical counterpart, we finally get the approximated discretized WGF particle ODE

$$
\tilde{X}_{t+1} = \tilde{X}_t + \eta \sum_{m\in\mathcal{M}_n} \frac{n_m}{n} w_{t,m}(\tilde{X}_t) \quad \text{with} \quad t \in \{0, 1, \dots, T-1\}\,, \qquad (9)
$$

starting with $\tilde{X}_0$ drawn from some initial distribution ρ_0, which, together with the step size η and the number of iterations T, needs to be chosen by the user.

Next, we provide a first result on the convergence properties of the proposed iterative method. Let

$$
v_t(x) \coloneqq \sum_m \mathbb{P}(M = m) \nabla\Big(h \circ r_t^{(m)}\Big)(x) \quad \text{and} \quad \tilde{v}_t(x) \coloneqq \sum_m \frac{n_m}{n} w_{t,m}(x)
$$

denote the true velocity field corresponding to $\mathcal{F}$ (see Lemma 2) and the approximated velocity field driving the effective particle updates (9), respectively. Further, let $\bar{X}_{t+1} = \tilde{X}_t + \eta v_t(\tilde{X}_t)$ and $\tilde{X}_{t+1} = \tilde{X}_t + \eta \tilde{v}_t(\tilde{X}_t)$ denote the discretized particle evolutions according to v_t and $\tilde{v}_t$ with densities $\bar{\rho}_t$ and $\tilde{\rho}_t$, respectively. Note that both iterations, in particular also the exact one $\bar{X}_t$, are based on the *approximate* previous iteration $\tilde{X}_t$. We suppose the following assumptions.

Assumption 1. *For all $t \in \{0, \dots, T\}$, it holds that*

(a) there exists an optimal coupling γ_t of $\tilde{\rho}_t$ and π (in the sense that this coupling yields $\mathcal{W}_2(\tilde{\rho}_t, \pi)$) such that $\int \langle x - y, v_t(x)\rangle \,\mathrm{d}\gamma_t(x, y) \le -\lambda \mathcal{W}_2^2(\tilde{\rho}_t, \pi)$;

(b) $||v_t||_{L^2(\tilde{\rho}_t)} \le L\mathcal{W}_2(\tilde{\rho}_t, \pi)$.

Assumption 2. *With high probability, it holds that $\sup_t ||\tilde{v}_t - v_t||_{L^2(\tilde{\rho}_t)} \le \varepsilon$, where ε may depend on n, σ, d, and further properties of π and the missingness mechanism.*

Theorem 4 (Non-asymptotic convergence)**.** *Suppose Assumption 1 and 2 hold. If $\eta \le \frac{\lambda}{L^2}$, then it holds with high probability that*

$$
\mathcal{W}_2(\tilde{\rho}_T, \pi) \le e^{-\frac{1}{2}\lambda T \eta} \mathcal{W}_2(\rho_0, \pi) + \frac{2}{\lambda}\varepsilon\,.
$$

The theorem gives a non-asymptotic error bound for the distribution of the generated complete sample. It remains open to identify concrete conditions on the true data distribution π and the missing value mechanism $\mathbb{P}(M = m|X)$ that guarantee the assumptions to be satisfied. Assumption 1 holds, for example, if $\mathcal{F}$ is strongly geodesically convex and has a Lipschitz Wasserstein gradient, mirroring the classic assumptions in optimization. Assumption 2, on the other hand, imposes a uniform bound on the approximation error. The following proposition gives a preliminary result in that direction.

Proposition 5 (Velocity field approximation error)**.** *Under Assumptions 7 – 10 in Appendix A.6 and if $\sigma \leq 1$, it holds that, for all $t \in \{0, 1, \ldots, T-1\}$ and $x^* \in \mathcal{X}$,*

$$\|\tilde{v}_t(x^*) - v_t(x^*)\|_2 \leq \mathcal{O}\left(\sqrt{\frac{|\mathcal{M}|}{n\sigma^d}} + \sigma^2\right), \tag{10}$$

where $|\mathcal{M}|$ denotes the cardinality of $\mathcal{M}$, that is, the number of patterns with positive probability.

The proof and the the precise assumptions are deferred to the appendix, and we refer to Liu et al. [2024] for the constants hidden in the $\mathcal{O}$-notation. Consequently, the estimation error vanishes as n tends to infinity and σ goes to zero, provided $n\sigma^d$ diverges. Moreover, the bound in (10) illuminates the dependence of the estimation error on the cardinality of $\mathcal{M}$: The more missingness patterns have nonzero probability, the worse the constant in the rate gets, with $|\mathcal{M}| = 2^d$ in the worst case. It is also instructive to note that $|\mathcal{M}|$ could be replaced by $|\mathcal{M}_n| \leq |\mathcal{M}|$, the actual number of patterns drawn for a given n. This illustrates that the rate is not dependent on the minimum number of observations across patterns, but rather on the overall number of patterns relative to n. In particular, having a small number of observations n_m for some patterns m will not diminish the rate as long as $|\mathcal{M}_n|$ is small relative to n. Although the result holds for all points x^* and time steps t, the coefficients in the upper bound on the right-hand side of (10) may well depend on their values. Establishing a uniform bound in the form of Assumption 2 remains an interesting direction for future work. We also note that, thanks to the MAR assumption and Lemma 2, it is possible to focus only on the observed part of a given pattern m. This is what makes the approximation result possible, even under complex non-monotone missingness, where each $X \mid M = m$ may follow a different distribution.

As for the implementation of the algorithm, Liu et al. [2024] propose to use gradient descent for the optimization in (4) or, in our case, (8). However, since we are specifically interested in the KL divergence, where $\psi^*(g) = g^2/2$ has a linear derivative $(\psi^*)'(d) = d$, the optimization problem reduces to a system of linear equations that can be solved directly. The advantage of using a direct numerical solver for this system of linear equations instead of a gradient descent is threefold: (i) It results in a higher accuracy of the solution, (ii) it is computationally more efficient and (iii) it reduces the number of hyperparameters, as, for gradient descent, the user needs to choose the step size and the number of steps (or another termination condition). Derivations of the linear approach, as well as further details on the implementation are given in Appendix A.2. This appendix also includes pseudo-code in Algorithm 1.

Finally, we note that our method has only three hyperparameters: the time step size η, number of time steps T, and bandwidth σ. Importantly, the choice of T and η does not involve a trade-off. In principle, smaller values of η and larger values of T should always improve performance, with the only practical limitation being the available computational resources. By contrast, σ must be selected more carefully, as both excessively large and small values can degrade the results. Nevertheless, as shown in the empirical section, a simple heuristic choice of σ already yields strong performance.

4 Empirical results

In this section, we demonstrate that FLOWGEM can match or even exceed state-of-the-art imputation methods in MAR examples. As competitors we include MICE [van Buuren and Groothuis-Oudshoorn, 2011], GAIN [Yoon et al., 2018], Hyperimpute [Jarrett et al., 2022], MIRI [Yu et al., 2025], the Bayesian approach of Näf and Chérief-Abdellatif [2026], MissDiff [Ouyang et al., 2023] and NewImp [Chen et al., 2024]. We compare these methods on a simulated example, originally devised to test imputations under MAR, as well as ten real datasets. In all experiments, we use the RBF kernel as k_σ, take $T = 1000$, $\eta = 0.01$, and σ following the median heuristic of Gretton et al. 2012 based on the initial sample $\tilde{X}_0$. A sensitivity analysis for the parameters is provided in Appendix A.4. The code for replication of the experiments is available at `https://github.com/gkremling/FLOWGEM-experiments`.

4.1 Simulation study

The following example, inspired by Näf et al. [2026, Example 6], is designed to test imputation methods for their handling of MAR. For $d = 3$, we take X to have uniform marginals on $[0, 1]$. Using

a copula, we induce dependence between X_1, X_2, while X_3 is independent of both. We then define the following missingness mechanism: For $(m_1, m_2, m_3) = ((0,0,0),(0,1,0),(1,0,0))$, let

$$\begin{aligned}\mathbb{P}(M = m_1 \mid X = x) &= (x_1 + x_2)/3\,;\\ \mathbb{P}(M = m_2 \mid X = x) &= (2 - x_1)/3\,;\\ \mathbb{P}(M = m_3 \mid X = x) &= (1 - x_2)/3\,.\end{aligned}$$

This mechanism clearly meets MAR and, moreover, $\mathbb{P}(M = 0 \mid X = x) > 0$ for all x, as required by Proposition 1. Nonetheless, as outlined in Näf et al. [2026], this is a rather complex non-monotone mechanism, designed to be difficult for imputation and estimation. The challenge formulated in Näf et al. [2026], Näf [2026] was to estimate the 0.1 quantile of X_1. We note that any other base distribution $\mathbb{P}_X$ could be used for this experiment. In Appendix A.3, we perform the same experiment using a multivariate Gaussian distribution instead. Moreover, we show results for regression as another downstream task there.

We simulate the above data generating process for a sample size of $n = 2000$ and $B = 20$ replications and attempt to generate a sample from $\mathbb{P}_X$ with our FLOWGEM and the aforementioned imputation methods. For our method and MIRI, we initialize the particle evolution at $\tilde{X}_0 = X^*$ with missing values replaced by samples from the observed values in the same column. Figure 2 shows the result for both the quantile estimation of X_1 (right) as well as the energy distance [Székely, 2003] between an independent sample and the recreated distribution (left). We can see that FLOWGEM meets the challenge remarkably well in both metrics, far surpassing GAIN, Hyperimpute, MIRI, the Bayesian approach, MissDiff and NewImp. The only method coming close is MICE, which tends to empirically work exceptionally well in continuous MAR settings, but it is still less accurate than FLOWGEM. In particular, our approach is arguably the method that estimates the true quantile most accurately, a rather impressive feat given the difficulty of the MAR mechanism and the strength of the competitors. Figure 5 in the Appendix also confirms this visually, showing that FLOWGEM recreates the distribution most accurately. Though this is merely a toy example, one would be hard-pressed to find a method that performs this well.

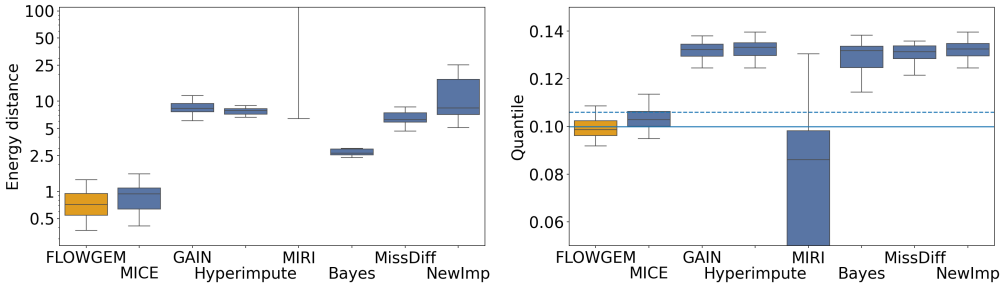


Figure 2: Standardized Energy distance in log-scale (left) and quantile estimate (right) for the simulated example with uniform distribution and $n = 2000$, repeated over $B = 20$ times. The solid blue line on the right indicates the true quantile value of the uncontaminated distribution, while the dashed line shows the population quantile when missing values are ignored. Our method, FLOWGEM, clearly outperforms its competitors, minimizing the energy distance and estimating the quantile most accurately. We note that MICE also performs well, but unlike FLOWGEM, it lacks theoretical justification. Some values are cut off for readability: the energy distance for MIRI ranges from 6 to $4 \cdot 10^{148}$ with a median of $2 \cdot 10^{46}$, and its quantile estimates reach as low as -2.95.

4.2 Real examples

In this section, we consider ten real datasets of varying size and dimension with simulated missingness. The datasets are taken from various sources, such as the UCI machine learning repository [Lichman, 2013]. Table 3 in the appendix gives a detailed account of the source of each dataset. We randomly split each dataset in two parts and generate MAR missing values for the first part, using random patterns and the ampute function of Schouten et al. [2018]. This has become a standard way of introducing MAR and leads to around 40–50% of missing values. For details on the missing value generation and a sensitivity analysis under different generation setups and missing ratios, we refer to Appendix A.3 and A.4, respectively. Given this dataset with missing values, we approximately

generate the full data either using FLOWGEM or one of the competing methods and then compare the result to the second (independent) sample using the standardized energy distance. Note that both the samples drawn from the ground-truth complete data and the generated/imputed samples are in the fully observed space. For a definition of the energy distance and results for other metrics, we refer to Appendix A.3. Table 1 shows the result. Crucially, in all ten datasets, our method displays the lowest or second lowest energy value, sometimes by a large margin compared to the other methods. This is despite the large range in number of observations and dimensions, ranging from very small ($n = 215$, $d = 6$) to very large ($n = 4901$, $d = 272$).

Table 1: Standardized energy distance between generated and held-out samples across ten real-world datasets (lower is better). Mean and standard deviation over ten runs are reported, except for dataset scm1d, where only a single run was performed due to long runtimes. The results are normalized per dataset: each value is divided by the best (lowest) value in its row. As a result, the best method in each row has a value of 1.00 and every other entry indicates the factor by which its error exceeds the best. For each dataset, the two best results are marked in bold. Our method is consistently either best or second best with only a slight increase in error. The results for MIRI and NewImp are omitted because their error exceeds 100 times the best on (nearly) all datasets. The Bayesian approach is omitted because it exceeded the runtime limit on several datasets. Full results for all methods are given in Appendix A.3.

Dataset	FLOWGEM	MICE	GAIN	Hyperimpute	MissDiff
scm1d	**1.03** (—-)	**1.00** (—-)	> 100	1.22 (—-)	17.91 (—-)
scm20d	**1.00** (0.02)	**1.23** (0.02)	> 100	1.73 (0.00)	14.28 (1.50)
pumadyn32nm	**1.09** (0.03)	**1.00** (0.08)	72.87 (33.39)	9.78 (0.69)	20.51 (1.90)
parkinsons	**1.00** (0.31)	**1.54** (0.06)	68.40 (15.58)	1.57 (0.00)	3.00 (0.70)
allergens	**1.41** (0.77)	6.91 (0.15)	7.15 (8.08)	**1.00** (0.22)	3.80 (0.97)
concrete	**1.00** (0.07)	3.76 (0.20)	13.49 (4.14)	**2.11** (0.01)	3.43 (0.46)
stock	**1.00** (0.07)	**1.09** (0.13)	49.94 (30.05)	1.74 (0.04)	1.84 (0.60)
forest	**1.00** (0.03)	1.13 (0.13)	1.16 (0.16)	**1.03** (0.02)	1.10 (0.12)
housing	**1.03** (0.10)	**1.00** (0.08)	20.48 (5.71)	2.93 (0.15)	1.50 (0.20)
windspeed	**1.03** (0.02)	1.05 (0.10)	1.19 (0.10)	**1.00** (0.00)	1.39 (0.18)

5 Conclusion

In this paper, we introduced FLOWGEM, a principled sample generator under MAR based on an approximate Wasserstein gradient flow. The method is grounded in a deep theoretical justification: the target functional, minimizing the average KL divergence over patterns, recovers the true distribution at the population level under MAR (Proposition 1). Further, as an important first step in ensuring theoretical validity in the finite-sample regime, a non-asymptotic error bound for the distribution of the generated sample can be derived under suitable assumptions (Theorem 4), and the approximation error of the local linear velocity field estimator can be shown to vanish asymptotically (Proposition 5). Empirically, FLOWGEM achieves impressive performance in MAR examples that rivals and often far surpasses modern imputation methods (Figure 2 and Table 1), without relying on computationally expensive or difficult-to-tune neural networks. Notably, for all experiments, ranging from simulated data with $d = 3$ to real data with widely varying scales up to $d = 272$, the same values of η and T, with a simple heuristic for σ, were used. In summary, FLOWGEM shows that theoretical rigor and strong empirical performance can go hand in hand, a combination that remains rare among existing imputation methods.

This work opens several interesting directions of further research. First, while our principled approach already admits some theoretical analysis, establishing concrete assumptions on the true data distribution and missingenss mechanism under which Theorem 4 holds is an important next step. Second, our algorithm is already fast for dimensions $d < 100$ but might require further improvements for higher dimensions. Third, extending FLOWGEM to categorical or mixed-type data is a natural and practically important next step.

Acknowledgments and Disclosure of Funding

J.L. is grateful for partial funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project numbers 543964668 (SPP2298) and 520388526 (TRR391).

References

B. Bischl, G. Casalicchio, T. Das, M. Feurer, S. Fischer, P. Gijsbers, S. Mukherjee, A. C. Müller, L. Németh, L. Oala, L. Purucker, S. Ravi, J. N. van Rijn, P. Singh, J. Vanschoren, J. van der Velde, and M. Wever. Openml: Insights from 10 years and more than a thousand papers. *Patterns*, 6(7), 2025. doi: 10.1016/j.patter.2025.101317.

Z. Chen, H. Li, F. Wang, O. Zhang, H. Xu, X. Jiang, Z. Song, and H. Wang. Rethinking the diffusion models for missing data imputation: A gradient flow perspective. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 112050–112103, 2024.

B.-E. Chérief-Abdellatif and J. Näf. Parametric MMD estimation with missing values: Robustness to missingness and data model misspecification. *arXiv preprint arXiv:2503.00448*, 2025.

G. Frahm, K. Nordhausen, and H. Oja. M-estimation with incomplete and dependent multivariate data. *Journal of Multivariate Analysis*, 176:104569, 2020.

R. M. Golden, S. S. Henley, H. White, and T. M. Kashner. Consequences of model misspecification for maximum likelihood estimation with missing data. *Econometrics*, 7(3), 2019.

A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, pages 1205–1213, 2012.

K. Grzesiak, C. Muller, J. Josse, and J. Näf. Do we need dozens of methods for real world missing value imputation? *arXiv preprint arXiv:2511.04833*, 2025.

Q. Guan and S. Yang. A unified inference framework for multiple imputation using martingales. *Statistica Sinica*, 34:1649–1673, 2024.

D. Jarrett, B. C. Cebere, T. Liu, A. Curth, and M. van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. In *39th International Conference on Machine Learning*, pages 9916–9937. PMLR, 2022.

O. Kovalenko. A note on the Kolmogorov-type inequalities for more than three norms. *arXiv preprint arXiv:2603.02044*, 2026.

S. C.-X. Li, B. Jiang, and B. M. Marlin. MisGAN: Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019.

M. Lichman. UCI machine learning repository, 2013. URL `https://archive.ics.uci.edu/`.

J. Liu, A. Gelman, J. Hill, Y.-S. Su, and J. Kropko. On the stationary distribution of iterative imputations. *Biometrika*, 101(1):155–173, 2014.

Q. Liu. Stein variational gradient descent as gradient flow. *Advances in Neural Information Processing Systems*, 2017.

S. Liu, J. Yu, J. Simons, M. Yi, and M. Beaumont. Minimizing f-divergences by interpolating velocity fields. In *International Conference on Machine Learning*, pages 32308–32331. PMLR, 2024.

P.-A. Mattei and J. Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 4413–4423, 2019.

F. Mealli and D. B. Rubin. Clarifying missing at random and related definitions, and implications when coupled with exchangeability. *Biometrika*, 102(4):995–1000, 2015.

G. Molenberghs, C. Beunckens, C. Sotto, and M. G. Kenward. Every missingness not at random model has a missingness at random counterpart with equal fit. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 70(2):371–388, 2008.

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

J. Näf and B.-E. Chérief-Abdellatif. Asymptotics of nonparametric estimation under general non-monotone MAR missingness: A Bayesian approach. *arXiv preprint arXiv:2603.23449*, 2026.

J. Näf. A practical guide to modern imputation. *arXiv preprint arXiv:2601.14796*, 2026.

J. Näf, E. Scornet, and J. Josse. What is a good imputation under MAR missingness? *arXiv preprint arXiv:2403.19196*, 2026.

Y. Ouyang, L. Xie, C. Li, and G. Cheng. MissDiff: Training diffusion models on tabular data with missing values. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

R. M. Schouten, P. Lugtig, and G. Vink. Generating missing values for simulation purposes: a multivariate amputation procedure. *Journal of Statistical Computation and Simulation*, 88(15): 2909–2930, 2018.

S. Seaman, J. Galati, D. Jackson, and J. Carlin. What is meant by "Missing at Random"? *Statistical Science*, 28(2):257–268, 2013.

Société Française d'Allergologie and AllergoBioNet. Allergen chip challenge database, 2024. URL `https://www.data.gouv.fr/fr/datasets/allergen-chip-challenge/`.

G. J. Székely. E-statistics: the energy of statistical samples. Technical Report 05, Bowling Green State University, Department of Mathematics and Statistics, 2003. URL `http://pages.stat.wisc.edu/~wahba/stat860public/pdf4/Energy/JSPI5102.pdf`.

K. Takai and Y. Kano. Asymptotic inference with incomplete data. *Communications in Statistics - Theory and Methods*, 42(17):3174–3190, 2013.

S. van Buuren. *Flexible Imputation of Missing Data. Second Edition*. Chapman & Hall/CRC Press, 2018.

S. van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011.

S. van Buuren, K. Groothuis-Oudshoorn, G. Vink, R. Schouten, A. Robitzsch, P. Rockenschaub, L. Doove, S. Jolani, M. Moreno-Betancur, I. White, P. Gaffert, F. Meinfelder, B. Gray, V. Arel-Bundock, M. Cai, T. Volker, E. Costantini, C. van Lissa, H. Oberman, and S. Wade. *mice: Multivariate Imputation by Chained Equations*, 2023. R package version 3.16.0.

vanderschaarlab. HyperImpute: Software package, 2022. URL `https://github.com/vanderschaarlab/hyperimpute`.

N. Wang and J. M. Robins. Large-sample theory for parametric multiple imputation procedures. *Biometrika*, 85(4):935–948, 1998.

I.-C. Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

J. Yoon, J. Jordon, and M. van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5689–5698, 2018.

J. Yu, Q. Ying, L. Wang, Z. Jiang, and S. Liu. Missing data imputation by reducing mutual information with rectified flows. In *Proceedings of the 39th Conference on Neural Information Processing Systems*, 2025.

J. Zhu and T. E. Raghunathan. Convergence properties of a sequential regression multiple imputation algorithm. *Journal of the American Statistical Association*, 110(511):1112–1124, 2015.

A Appendix

A.1 Terminology table

For better readability, we provide a summary of our notation in Table 2.

Table 2: Notation summary.

Symbol	Description
X	Complete observation in $\mathbb{R}^d$
π	Density of complete observation X
M	Missingness pattern in $\{0,1\}^d$ ($M=0$: observed, $M=1$: missing)
$\mathcal{M}$	Support of missingness distribution (patterns with positive probability)
X^*	Incomplete observation with NA whenever $M=1$
$\mathbb{P}(M=m \mid X=x)$	Missingness mechanism
$X^{(m)}$	Observed sub-vector of X under missingness pattern m
$\pi^{(m)}$	Density of $X^{(m)}$
$\pi_m^{(m)}$	Conditional density of $X^{(m)}$ given missingness pattern $M=m$
$\mathbb{P}^{(m)}_{X\mid M=m}$	Conditional distribution of $X^{(m)}$ given missingness pattern $M=m$
$\mathcal{F}[\rho]$	f-divergence between $\rho^{(m)}$ and $\pi_m^{(m)}$ across all missingness patterns
h	Associated to f-divergence by $h(r)=rf'(r)-f(r)$
$r_t^{(m)}$	Density ratio $\pi_m^{(m)} \,/\, \rho_t^{(m)}$
X_t	Solution to the exact WGF
$\tilde{X}_t$	Solution to the approximated and discretized WGF
v_t	Velocity field of the exact WGF
$\tilde{v}_t$	Approximated velocity field
$w_{t,m}$	Local linear approximation of $\nabla(h\circ r_t^{(m)})$

A.2 Algorithm and additional details

In this section, we first discuss the system of linear equations that is crucial to our algorithm and then present pseudo-code for our procedure in Algorithm 1. In addition, we provide extensive details on the implementation.

Linear System of Equations Liu et al. [2024] propose to use gradient descent for the optimization in (4). Of course, we could use the same technique to approximate the solution of (8). However, since we are specifically interested in the KL divergence and, in that case, $\psi^*(g)=g^2/2$ has a linear derivative $(\psi^*)'(d)=d$, the optimization problem reduces to a system of linear equations that can be solved directly.

Specifically, the objective function for a fixed x^* and m is given by

$$L(w,b)=\frac{1}{n_m}\sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i})\Big((X_{m,i}^{(m)})^T w+b\Big)-\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i})\psi^*\Big((\tilde{X}_{t,i}^{(m)})^T w+b\Big)\,,$$

where we use the abbreviation $k_\sigma^{(m)}(x) \coloneqq k_\sigma(x^{(m)}, x^{*(m)})$. To find the maximum, we solve

$$\nabla L(w,b)=\begin{pmatrix} \frac{1}{n_m}\sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i})X_{m,i}^{(m)}-\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i})\Big((\tilde{X}_{t,i}^{(m)})^T w+b\Big)\tilde{X}_{t,i}^{(m)} \\ \frac{1}{n_m}\sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i})-\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i})\Big((\tilde{X}_{t,i}^{(m)})^T w+b\Big) \end{pmatrix}=0\,,$$

which can be rewritten as a system of linear equations $A(w,b)^T=c$ with

$$A=\begin{pmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{pmatrix} \quad \text{and} \quad c=\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}\,,$$

where

$$A_{11} = \begin{pmatrix} \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,1}^{(m)} \tilde{X}_{t,i,1}^{(m)} & \cdots & \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,1}^{(m)} \tilde{X}_{t,i,d_m}^{(m)} \\ \vdots & \ddots & \vdots \\ \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,d_m}^{(m)} \tilde{X}_{t,i,1}^{(m)} & \cdots & \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,d_m}^{(m)} \tilde{X}_{t,i,d_m}^{(m)} \end{pmatrix},$$

$$A_{21} = \begin{pmatrix} \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,1}^{(m)} & \cdots & \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}) \tilde{X}_{t,i,d_m}^{(m)} \end{pmatrix},$$

$$A_{22} = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} k_\sigma^{(m)}(\tilde{X}_{t,i}), \tag{11}$$

and

$$c_1 = \begin{pmatrix} \frac{1}{n_m} \sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i}) X_{m,i,1}^{(m)} & \cdots & \frac{1}{n_m} \sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i}) X_{m,i,d_m}^{(m)} \end{pmatrix}^T,$$

$$c_2 = \frac{1}{n_m} \sum_{i=1}^{n_m} k_\sigma^{(m)}(X_{m,i}). \tag{12}$$

In practice, we apply Tikhonov regularization to A, replacing it with $A + \varepsilon I$ where $\varepsilon = 10^{-5}$, to ensure numerical stability during inversion.

Algorithm 1: FLOWGEM (FLOW-based GEneration for Missing data)

Data: Observations with missing values $(\mathbf{X}, \mathbf{M}) \in \mathbb{R}^{n \times d} \times \{0, 1\}^{n \times d}$,
Method to sample from ρ_0 (`initial_sample`), Final time T, Step size η,
Kernel function k_σ, Bandwidth σ, New sample size $\tilde{n}$
Result: New complete sample $\{\tilde{X}_{T,i}\}_{i=1}^{\tilde{n}}$ approximately distributed as $\mathbb{P}_X$

$\tilde{\mathbf{X}}_0 \leftarrow$ `initial_sample(`$\mathbf{X}, \mathbf{M}, \tilde{n}$`)` `// ` $\in \mathbb{R}^{\tilde{n} \times d}$
for $t \leftarrow 0$ **to** $T - 1$ **do**
  for $i \leftarrow 0$ **to** $\tilde{n} - 1$ **do** `// can be parallelized or vectorized`
    $x^* \leftarrow \tilde{\mathbf{X}}_{t,i}$
    for $m \in$ `unique_rows(`$\mathbf{M}$`)` **do** `// can be parallelized`
      $\mathbf{X}_m \leftarrow$ rows $\mathbf{X}_j$ with $\mathbf{M}_j = m$
      $A \leftarrow$ `compute_A` $(k_\sigma, m, x^*, \tilde{\mathbf{X}}_t)$ `// according to (11)`
      $c \leftarrow$ `compute_c` $(k_\sigma, m, x^*, \mathbf{X}_m)$ `// according to (12)`
      $(w_m, b_m) \leftarrow$ `solve` (A, c)
    end
    $\tilde{\mathbf{X}}_{t+1,i} \leftarrow \tilde{\mathbf{X}}_{t,i} + \eta \sum_m \frac{n_m}{n} w_m$
  end
end
return $\tilde{\mathbf{X}}_T$

Standardization Some datasets such as "scm20d" have widely different scales for different variables. To minimize numerical issues, we standardize the columns by using a diagonal matrix Λ and an estimated mean μ. To motivate this approach, we are able to make use of the well-known fact that the KL divergence is invariant to affine transformations. For all m and all t, let $\Lambda_m = (\Lambda_{i,j})_{i:m_i=0, j:m_j=0}$ be the d_m-dimensional submatrix of Λ and similarly, $\mu_m = (\mu_j)_{j:m_j=0}$. Moreover, let

$$Y^{(m)} = \Lambda_m^{-1/2}(X^{(m)} - \mu_m) \quad \text{and} \quad \tilde{Y}_t = \Lambda^{-1/2}(\tilde{X}_t - \mu)$$

be the standardized data, and define by

$$\tilde{\pi}_m^{(m)}(y^{(m)}) = \det(\Lambda_m)^{1/2} \pi_m^{(m)}(\Lambda_m^{1/2} y^{(m)} + \mu),$$
$$\tilde{\rho}_t(y) = \det(\Lambda)^{1/2} \rho_t(\Lambda^{1/2} y + \mu)$$

the respective scaled densities. Then by a simple change-of-variable argument in the integration, it holds that

$$\begin{aligned}\mathbb{E}_{M\sim\mathbb{P}_M}\left[\mathrm{KL}\left(\mathbb{P}^{(M)}_{X|M}\,\|\,\mathbb{P}^{(M)}_{\tilde{X}_t}\right)\right] &= \sum_{m\in\mathcal{M}}\mathbb{P}(M=m)\int\log\left(\frac{\pi^{(m)}_m(x^{(m)})}{\rho^{(m)}_t(x^{(m)})}\right)\pi^{(m)}_m(x^{(m)})dx^{(m)}\\ &= \sum_{m\in\mathcal{M}}\mathbb{P}(M=m)\int\log\left(\frac{\tilde{\pi}^{(m)}_m(y^{(m)})}{\tilde{\rho}^{(m)}_t(y^{(m)})}\right)\tilde{\pi}^{(m)}_m(y^{(m)})dy^{(m)}\,.\end{aligned}$$

The last expression is simply the KL divergence between the scaled distributions. Combined with Proposition 1, this shows that minimizing the KL divergence between the scaled distribution will recover P_X after rescaling. Thus, we may standardize X and X_t using any μ, Λ. In our case, they are obtained by calculating standard deviations and means separately for each column only on the observed data.

Early Stopping and step-halving In the real data experiments, we employ an early stopping mechanism. In particular, we measure the relative change of the particles in (9) at each t and stop if

$$\frac{\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\|\sum_{m\in\mathcal{M}_n}\frac{n_m}{n}w_{t,m}(\tilde{X}_{t,i})\|_2}{\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\|\tilde{X}_{t,i}\|_2}<\varepsilon\,,$$

using $\varepsilon = 0.01$. This not only makes the algorithm faster, but also impedes the algorithm from overshooting in case η was chosen too large. Finally, if the norm of the gradient increases in an iteration, we halve η for all subsequent iterations.

A.3 Details on experiments and additional results

Real datasets Table 3 contains more details on the datasets and their sources. We note that the values of n and d are different than the total number of rows and columns in each dataset, because (1) columns with less than 10% of unique values were removed for the final dataset and (2) we split the dataset in two for training and test set.

Runtimes All computations were performed using Google Colaboratory Pro with a TPU v6e-1 accelerator. The host runtime used Python 3.11. Experiments were run between March 2026 and April 2026. Average computation times per method for the simulated datasets are reported in Table 4.

Table 3: Real data examples considered in Section 4.2. For each dataset, we report the number of rows, the number of columns, the actual number of observations n and dimensions d used in the analysis, as well as their source. UCI refers to Lichman [2013] and was downloaded from `https://github.com/treforevans/uci_datasets/tree/master`. Allergen Chip Challenge refers to Société Française d'Allergologie and AllergoBioNet [2024].

Name	Rows	Cols	n	d	Source
scm1d	9803	296	4901	272	Bischl et al. [2025]
scm20d	8966	77	4482	58	Bischl et al. [2025]
pumadyn32nm	8192	33	4095	33	UCI
parkinsons	5875	21	2937	18	UCI
allergens	2351	112	1175	49	Allergen Chip Challenge
concrete	1030	9	514	8	Yeh [1998]
stock	536	12	267	9	UCI
forest	517	13	257	7	UCI
housing	506	14	252	10	UCI
windspeed	433	6	215	6	van Buuren et al. [2023]

Missing Value Generation We use the ampute function of Schouten et al. [2018] as follows: We randomly generate a set of patterns, including a fully observed one, and generate MAR missingness with the ampute function according to those patterns, such that every pattern is observed with equal probability. This results in about 40–50% missing values, calculated as total number of missing entries divided by $n \times d$, as in Grzesiak et al. [2025]. We note that it is often left ambiguous in other papers how "x% of missing values" is defined. In our way of measuring, 40% of missing values is substantial, in fact the recent benchmark by Grzesiak et al. [2025] only considers up to 30%.

Table 4: Average computation time (in seconds) per method for the simulation study of Section 4.1 with $n = 2000$, $d = 3$, and uniform distribution. Mean and standard deviation over $B = 20$ replications are reported.

Method	Mean	Std. Dev.
FLOWGEM	22.85	1.36
MICE	0.56	0.00
GAIN	2.80	0.64
Hyperimpute	8.68	5.63
MIRI	1611.91	2588.70
Bayes	2038.28	520.23
MissDiff	19.22	1.03
NewImp	18.16	0.56

Standardized Energy Distance We assess the performance of imputation methods by measuring the distributional distance between the generated dataset and a set of independent complete data. To this end, we use the energy distance from Székely [2003], defined as

$$e^2(X, Y) = 2\mathbb{E}\|X - Y\|_2 - \mathbb{E}\|X - X'\|_2 - \mathbb{E}\|Y - Y'\|_2 \,,$$

where X and Y are random variables (here observed and generated). Before computing the energy distance, we standardized all variables in both imputed and fully observed datasets using the column-wise means and standard deviations estimated from the fully observed dataset *prior to imposing missingness*. This ensures that the distance measure is not dominated by variables with large natural variance and is comparable across datasets.

Alternative metrics Additionally, we assess the performance of FLOWGEM and its competitors via maximum mean discrepancy (MMD) distance and estimated 2-Wasserstein distance as implemented in Muzellec et al. [2020]. The results for all metrics and methods are illustrated in Table 5.

Continuous Features As our method intrinsically assumes the distribution $\mathbb{P}_X$ to be continuous (with density π), it was designed to handle continuous features. To respect that in our experiments, we filtered out columns in the datasets that had less than 10% of unique values for our application.

Initialization For real data, we generate $\tilde{X}_0$ by imputing the dataset using the MICE implementation in the Python `hyperimpute` package with 10 iterations (the full MICE implementation uses 100 iterations). This gives a reasonable and quick-to-calculate data-adaptive starting value for our algorithm.

Competitors We compare our method against MICE [van Buuren and Groothuis-Oudshoorn, 2011], GAIN [Yoon et al., 2018], Hyperimpute [Jarrett et al., 2022], MIRI [Yu et al., 2025], the Bayesian density estimation of Näf and Chérief-Abdellatif [2026], MissDiff [Ouyang et al., 2023] and NewImp [Chen et al., 2024]. For GAIN, Hyperimpute and MICE, we use the `hyperimpute` Python package [vanderschaarlab, 2022] with default hyperparameter settings. For MIRI, MissDiff and NewImp, we use the implementation available at `https://github.com/yujhml/MIRI-Imputation` (last accessed in March 2026). As the original implementation only supports float32, we adapted the code to float64 to ensure a fair comparison with the remaining methods. For MIRI, we use the hyperparameters specified in Yu et al. [2025], namely `max_rounds=15`, `batchsize=500`, `maxepochs=900`, `odesteps=100`, while for MissDiff and NewImp, we use the default settings. We note that, for essentially all competing methods, performance may be improved through more extensive hyperparameter tuning. In particular, Yu et al. [2025] mention a different specification with higher number of rounds. However, our current setting was already straining our computational resources. While computation times of the current MIRI code are remarkably resistant to higher d, completing in about 70 minutes on the "scm1d" dataset, the code was excruciatingly slow on the smaller datasets. This can be seen for instance in the simulation with $d = 3$ in Table 4. Moreover, we saw no obvious convergent behavior and instead the code usually started producing `nans` for too many iterations.

Table 5: Distance between generated and held-out samples across ten real-world datasets (lower is better): (a) Standardized energy distance, (b) Estimated MMD distance, (c) Estimated 2-Wasserstein distance. Mean and standard deviation over ten runs are reported, except for dataset scm1d, where only a single run was performed due to long runtimes. "–" indicates the method exceeded the runtime limit. The results are normalized per dataset and metric. For each dataset, the two best results are marked in bold. Across all metrics and datasets, our method's mean error never exceeds 1.44 times that of the best method.

(a)	Dataset	FLOWGEM	MICE	GAIN	Hyperimp.	MIRI	Bayes	MissDiff	NewImp
	scm1d	**1.03** (—)	**1.00** (—)	> 100	1.22 (—)	> 100	–	17.91 (—)	> 100
	scm20d	**1.00** (0.02)	**1.23** (0.02)	> 100	1.73 (0.00)	> 100	–	14.28 (1.50)	> 100
	pumadyn32nm	**1.09** (0.03)	**1.00** (0.08)	72.87 (33.39)	9.78 (0.69)	> 100	–	20.51 (1.90)	> 100
	parkinsons	**1.00** (0.31)	**1.54** (0.06)	68.40 (15.58)	1.57 (0.00)	> 100	–	3.00 (0.70)	> 100
	allergens	**1.41** (0.77)	6.91 (0.15)	7.15 (8.08)	**1.00** (0.22)	> 100	–	3.80 (0.97)	> 100
	concrete	**1.00** (0.07)	3.76 (0.20)	13.49 (4.14)	**2.11** (0.01)	> 100	> 100	3.43 (0.46)	> 100
	stock	**1.00** (0.07)	**1.09** (0.13)	49.94 (30.05)	1.74 (0.04)	13.99 (22.42)	> 100	1.84 (0.60)	> 100
	forest	**1.00** (0.03)	1.13 (0.13)	1.16 (0.16)	**1.03** (0.02)	1.90 (0.81)	2.62 (0.54)	1.10 (0.12)	> 100
	housing	**1.03** (0.10)	**1.00** (0.08)	20.48 (5.71)	2.93 (0.15)	> 100	6.30 (3.45)	1.50 (0.20)	> 100
	windspeed	**1.03** (0.02)	1.05 (0.10)	1.19 (0.10)	**1.00** (0.00)	> 100	27.46 (15.89)	1.39 (0.18)	> 100

(b)	Dataset	FLOWGEM	MICE	GAIN	Hyperimp.	MIRI	Bayes	MissDiff	NewImp
	scm1d	**1.12** (—)	1.17 (—)	20.61 (—)	**1.00** (—)	> 100	–	2.60 (—)	> 100
	scm20d	**1.13** (0.00)	1.27 (0.01)	18.83 (2.08)	**1.00** (0.00)	> 100	–	2.00 (0.36)	> 100
	pumadyn32nm	1.18 (0.00)	1.17 (0.00)	1.39 (0.09)	**1.11** (0.01)	> 100	–	**1.00** (0.00)	> 100
	parkinsons	**1.00** (0.11)	**1.11** (0.10)	15.23 (13.08)	2.38 (0.00)	> 100	–	4.87 (0.28)	> 100
	allergens	1.14 (0.04)	1.41 (0.01)	1.63 (0.67)	**1.00** (0.05)	> 100	–	**1.12** (0.07)	> 100
	concrete	**1.03** (0.03)	2.12 (0.12)	2.79 (0.60)	**1.00** (0.00)	> 100	> 100	1.19 (0.06)	> 100
	stock	**1.00** (0.02)	**1.02** (0.03)	5.65 (3.69)	1.03 (0.01)	> 100	> 100	1.02 (0.04)	> 100
	forest	1.44 (0.10)	1.77 (0.22)	**1.09** (0.37)	1.40 (0.06)	3.31 (1.39)	3.29 (1.32)	**1.00** (0.16)	> 100
	housing	**1.00** (0.09)	**1.03** (0.12)	10.96 (4.33)	2.38 (0.07)	> 100	2.04 (0.17)	1.61 (0.22)	> 100
	windspeed	1.09 (0.02)	1.08 (0.03)	**1.03** (0.03)	**1.00** (0.00)	> 100	5.17 (1.52)	1.14 (0.04)	> 100

(c)	Dataset	FLOWGEM	MICE	GAIN	Hyperimp.	MIRI	Bayes	MissDiff	NewImp
	scm1d	**1.00** (—)	**1.02** (—)	> 100	1.69 (—)	> 100	–	61.71 (—)	> 100
	scm20d	**1.00** (0.03)	**1.04** (0.11)	> 100	2.45 (0.00)	> 100	–	38.47 (7.91)	> 100
	pumadyn32nm	**1.15** (0.04)	**1.00** (0.10)	96.02 (43.16)	12.73 (0.91)	> 100	–	30.42 (2.85)	> 100
	parkinsons	**1.00** (0.34)	**1.38** (0.41)	62.22 (50.46)	15.45 (0.00)	> 100	–	34.15 (1.62)	> 100
	allergens	**1.00** (0.53)	5.36 (0.13)	19.33 (23.49)	**1.70** (0.86)	> 100	–	10.48 (1.66)	> 100
	concrete	**1.00** (0.06)	3.97 (0.34)	16.39 (6.13)	**2.96** (0.04)	4.07 (0.68)	> 100	4.61 (0.72)	> 100
	stock	**1.00** (0.09)	**1.22** (0.18)	78.32 (34.84)	2.17 (0.13)	8.82 (2.90)	> 100	2.74 (1.00)	> 100
	forest	**1.14** (0.05)	1.72 (0.21)	1.32 (0.57)	**1.00** (0.05)	1.56 (0.42)	5.12 (2.47)	1.71 (0.35)	27.86 (0.38)
	housing	**1.07** (0.19)	**1.00** (0.22)	27.44 (6.82)	9.26 (0.36)	> 100	6.03 (1.33)	7.77 (0.85)	45.98 (1.65)
	windspeed	**1.10** (0.03)	1.15 (0.17)	1.32 (0.21)	**1.00** (0.01)	6.14 (2.13)	34.75 (18.81)	1.94 (0.37)	> 100

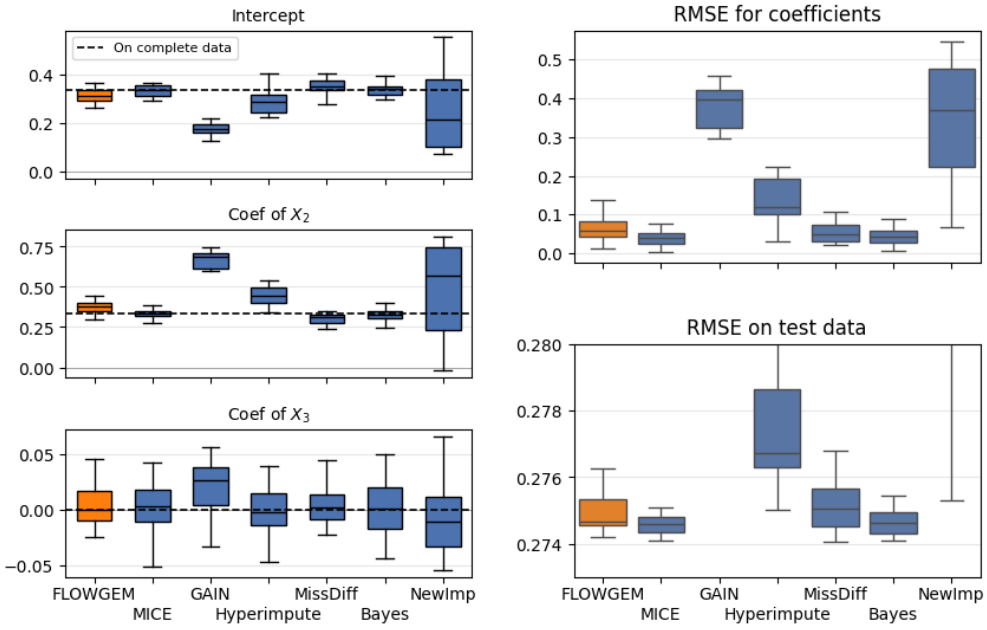


Figure 3: Regression results for the simulation study in Section 4.1. Left: Estimated regression coefficients (intercept, X_2, X_3) for each imputation method. The dashed line marks the true coefficient value. Right, top: Euclidean (ℓ_2) distance between the estimated and true coefficient vectors. Right, bottom: RMSE on (complete) test data following the same data distribution. The results for MIRI are omitted because the RMSE for coefficients exceeded 1000 in 16 out of 20 runs for this method.

Regression simulation In the simulation study in Section 4.1, we considered the challenge formulated in Näf et al. [2026], Näf [2026] of estimating the 0.1 quantile of X_1. To investigate FLOWGEM's performance on other downstream tasks, we additionally run a regression analysis on the same simulated data. In particular, we fit a linear regression for X_1 given X_2 and X_3. It is easy to compute that the true regression function is given by

$$\mathbb{E}\left[X_1 \mid X_2 = x_2, X_3 = x_3\right] = \frac{1}{3} + \frac{1}{3}x_2.$$

Using the same generated/imputed datasets from FLOWGEM and all other methods that Figure 2 is based on, we get the results illustrated in Figure 3. Also in this task, FLOWGEM is among the best-performing methods, together with MICE. Rather surprisingly, MissDiff and Bayes perform equally well on the regression task, even though Figures 2 and 5 show that they do not capture the data distribution properly.

Gaussian simulation Finally, we present the Gaussian version of the simulated example in Section 4.1. For $d = 3$, we take $X \sim \mathcal{N}(0, \Sigma)$, where $\mathcal{N}(\mu, \Sigma)$ denotes the Gaussian distribution with mean μ and covariance matrix Σ. Here, Σ is chosen to have diagonal elements of one and a correlation of 0.7 between X_1, X_2, with no correlation between X_1, X_2 and X_3. We then define the following missingness mechanism: For $(m_1, m_2, m_3) = ((0,0,0), (0,1,0), (1,0,0))$, let

$$\begin{aligned}
\mathbb{P}(M = m_1 \mid X = x) &= (\Phi(x_1) + \Phi(x_2))/3\,; \\
\mathbb{P}(M = m_2 \mid X = x) &= (2 - \Phi(x_1))/3\,; \\
\mathbb{P}(M = m_3 \mid X = x) &= (1 - \Phi(x_2))/3\,,
\end{aligned}$$

where Φ is the standard Gaussian distribution function. Figure 4 presents the result. Again, we are far better than GAIN, Hyperimpute, MissDiff, NewImp and MIRI, only narrowly beaten by MICE and Bayes. However, we note that both MICE and the Bayes method essentially use Gaussian regression and Gaussian approximations respectively, giving them a somewhat unfair edge in this example. Despite this edge, it turns out that increasing T to 1500 further boosts FLOWGEM and closes the gap between our method and MICE or Bayes, but we did not include this result here. Figures 5 and 6

show scatter plots of the first two dimensions of the generated samples for the uniform and Gaussian simulation, respectively, visually confirming the results in Figures 2 and 4.

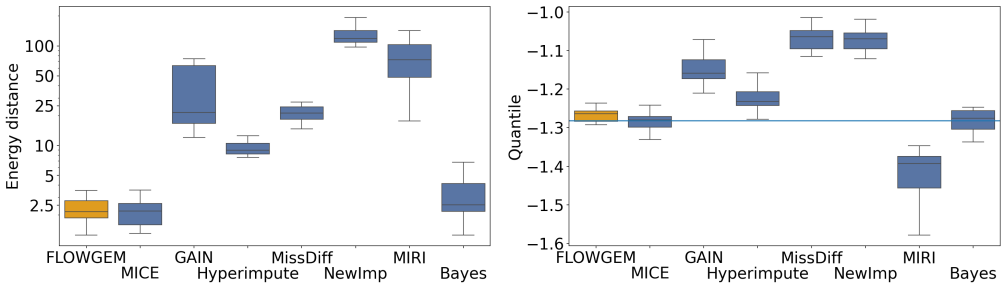


Figure 4: Standardized energy distance in log-scale (left) and quantile estimate (right) for the simulated example with Gaussian distribution, $n = 2000$, and repeated over $B = 20$ times. The blue line on the right indicates the true quantile value of the uncontaminated distribution. Our method, FLOWGEM, belongs to one of the best methods, minimizing the energy distance and estimating the quantile most accurately. We note that both MICE and the Bayes method essentially use Gaussian regression and Gaussian approximations respectively, giving them a somewhat unfair edge in this example. Despite this edge, it turns out that increasing T to 1500 further boosts FLOWGEM and closes the gap between our method and MICE or Bayes, but we did not include this result here.

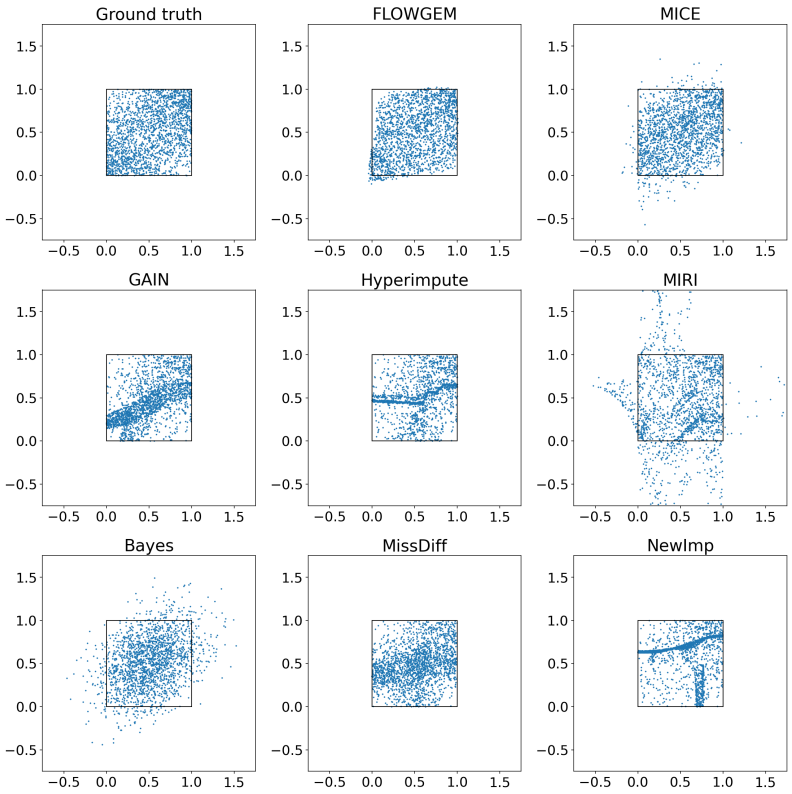


Figure 5: Scatter plots of the first two dimensions of the generated samples for each method, for a single replication of the simulation study in Section 4.1 with $n = 2000$, $d = 3$, and uniform distribution. The black square indicates the support $[0, 1]^2$ of the true distribution.

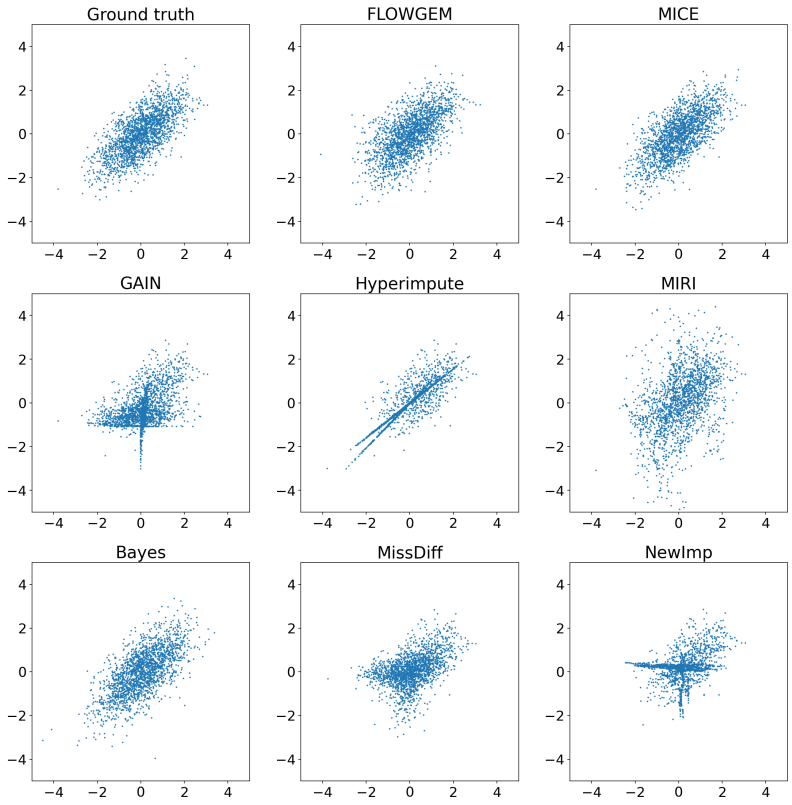


Figure 6: Scatter plots of the first two dimensions of the generated samples for each method, for a single replication of the simulation study in Section 4.1 with $n = 2000$, $d = 3$, and Gaussian distribution. We note that both MICE and the Bayes method essentially use Gaussian regression and Gaussian approximations respectively, giving them a somewhat unfair edge in this example.

A.4 Sensitivity Analysis

Further experiments were performed on three of the real world datasets (concrete, forest, and housing) to investigate FLOWGEM's sensitivity to the choice of its hyperparameters and to different amputation settings.

Choice of the bandwidth First, we vary σ on a multiplicative grid around the currently employed median heuristic choice $\hat{\sigma}$, and keep the other parameters fixed ($\eta = 0.01$, $T = 1000$). The results are displayed in Table 6. The performance either barely varies for different choices of the bandwidth or is best at the median heuristic itself.

Table 6: Sensitivity analysis for the bandwidth parameter σ, using a multiplicative grid around the currently employed median heuristic choice $\hat{\sigma}$. The results show the mean standardized energy distance over 5 runs with different seeds, together with the standard deviation in parentheses. Each value is divided by the reference value (default parameter choice) in the same row.

Dataset	$0.25\hat{\sigma}$	$0.5\hat{\sigma}$	$1\hat{\sigma}$	$2\hat{\sigma}$	$4\hat{\sigma}$
concrete	2.57 (0.52)	1.13 (0.45)	1.00 (0.10)	1.76 (0.16)	3.60 (0.41)
forest	1.03 (0.09)	1.03 (0.06)	1.00 (0.04)	1.08 (0.03)	1.16 (0.05)
housing	0.99 (0.07)	0.91 (0.10)	1.00 (0.10)	1.13 (0.07)	0.99 (0.07)

Choice of the step size Second, we study the sensitivity of our method to the choice of the initial step size η, keeping the bandwidth σ fixed at its median-heuristic value. We do not analyze the maximal number of steps T separately, since it only bounds the runtime and, with step-halving and

early stopping (described in Appendix A.2), a larger T should not worsen the result. When varying η, we set $T = 5000$, which is large enough for early stopping to terminate the run in 13 of 15 settings. In the remaining two settings, the step limit was reached, so these results may be slightly affected by the limit on T. The results are shown in Table 7. It can be seen that the choice of η has a very mild effect on the performance, which is exactly what the adaptive step size rule was designed for. Only for the housing dataset, the outcome differs more significantly. To increase the interpretability of the outcomes, we also indicate the mean effective number of iterations and final step size in the table. In the concrete dataset, the adaptive step size rule is not triggered. In fact, the number of iterations directly scales with the step size, keeping the final time point $T\eta$ fixed, which explains why the results only vary mildly. For the forest dataset, the adaptive step size rule almost always pushes η to a value of 0.0025, which explains that the results remain constant. A similar effect can be observed for the housing dataset. Yet, the largest considered initial step size, namely $\eta = 0.04$, triggers repeated step-halving, resulting in an extremely small effective step size and (surprisingly) improved performance.

Table 7: Sensitivity analysis for the initial step size parameter η, using a multiplicative grid around the default value of 0.01. (a) Mean standardized energy distance over 5 runs with different seeds, together with the standard deviation in parentheses. Each value is divided by the reference value (default parameter choice) in the same row. (b) Effect of the early stopping and step-halving mechanisms: mean effective number of iterations with mean final step size in parentheses.

(a)	Dataset	$\eta = 0.0025$	$\eta = 0.005$	$\eta = 0.01$	$\eta = 0.02$	$\eta = 0.04$
	concrete	1.07 (0.09)	1.00 (0.04)	1.00 (0.04)	1.00 (0.04)	1.00 (0.04)
	forest	1.00 (0.02)	1.00 (0.02)	1.00 (0.02)	1.00 (0.02)	1.11 (0.12)
	housing	1.00 (0.07)	1.00 (0.07)	1.00 (0.07)	1.00 (0.07)	0.85 (0.06)

(b)	Dataset	$\eta = 0.0025$	$\eta = 0.005$	$\eta = 0.01$	$\eta = 0.02$	$\eta = 0.04$
	concrete	5000 (0.0025)	3547 (0.005)	1774 (0.01)	888 (0.02)	444 (0.04)
	forest	3778 (0.0025)	1889 (0.005)	3764 (0.0025)	3750 (0.0025)	4131 (0.0015)
	housing	4915 (0.0025)	2458 (0.005)	4901 (0.0025)	4890 (0.0025)	5000 ($\approx 10^{-19}$)

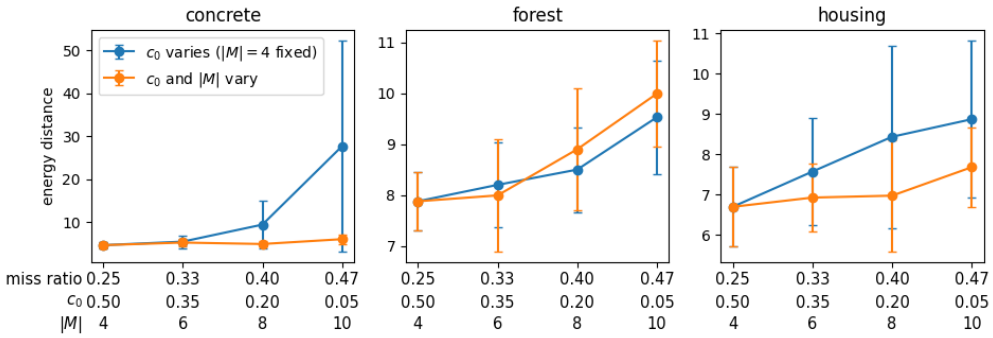
Missing value generation The original experiments are performed on a single amputed dataset based on a randomly generated set of missingness patterns. To study the role of the missing value generation on FLOWGEM's performance, we perform two more experiments. In the first one, we fix the number of missingness patterns $|\mathcal{M}| = 4$ and decrease the share $c_0 = \mathbb{P}(M = 0)$ of fully observed rows from 50% to 25%, which increases the missing ratio. In the second experiment, we simultaneously increase the number of patterns $|\mathcal{M}| \in \{4, 6, 8, 10\}$ and decrease c_0. In each of 5 draws, we sample 10 distinct random patterns, where the first $|\mathcal{M}|$ are used for each $|\mathcal{M}| < 10$ and the average fraction of missing variables per incomplete row is within three percentage points of 50% for every subset. Accordingly, the overall missing ratio does not change due to the number of patterns but only because of the share of fully observed rows. Comparing the two experiments at equal missing ratio thus separates the effect of the number of patterns from that of the missing ratio. We run FLOWGEM with three seeds per draw. The results are illustrated in Figure 7. In both experiments, the increased missing ratio makes the problem more difficult and thus leads to an increase of the energy distance. In light of Proposition 5, we would expect the error to increase more severely when the number of patterns increase. In fact, this is the case for the forest dataset. For the other two datasets, the effect seems to be the opposite. We note that the result in Proposition 5 gives an upper bound on the local approximation error and that further effects may influence the overall error.

A.5 Relation to Stein Variational Gradient Descent

Another method that is conceptually related to ours is Stein Variational Gradient Descent (SVGD). In fact, the connection to SVGD is already discussed in the prior paper [Liu et al., 2024, Appendix N], which our approach is based on. Our method differs from SVGD in several key aspects:

- *Score information.* A key difference is that SVGD assumes the score of the target density to be known, whereas we only assume to be given samples from that distribution.

Figure 7: Sensitivity analysis for the missing ratio and number of patterns. Standardized energy distance on the test data, averaged over seeds, as mean and standard deviation across draws, so that the reported deviations reflect the variability of the missing value generation.



- *Objective functional.* With the score known, SVGD targets the minimization of the backward KL, $\mathrm{KL}(\rho\|\pi)$, as objective functional, while our method is built for the forward KL, $\mathrm{KL}(\pi\|\rho)$, where π denotes the true density and ρ the approximated one.
- *Gradient-flow geometry.* We are specifically interested in the Wasserstein gradient flow. SVGD, on the other hand, can be interpreted as a gradient flow of the KL divergence under a different Riemannian metric structure on the space of density functions induced by Stein operator. [Liu, 2017]
- *Missing values.* Finally, our setting introduces missing values and therefore requires adapting both the objective and the induced velocity field to operate on partially observed data.

Nevertheless, one could consider reframing the estimation of $h \circ r$ as a global functional estimation problem in a reproducing kernel Hilbert space as done in SVGD, instead of using our local approximation at a point x^*. Depending on the kernel choice, this may improve smoothness and reduce the "local stitching" artifacts that arise from purely local regressions. While RKHS methods could likely be adapted, doing so would require nontrivial changes to the estimation procedure and potentially to the assumptions used in our analysis. We consider this an interesting direction for future work.

A.6 Assumptions of Proposition 5

In this section, we derive the assumptions for the approximation error bound in Proposition 5. We start by translating the assumptions of Liu et al. [2024], not considering missing values, into our notation.

Assumption 3. *There exists a constant $C_k > 0$ independent of σ, such that for all t,*

$$\frac{1}{2}\int \rho_t(\sigma y + x^*)\exp\left(-\|y\|^2/2\right)\cdot\|y\|^2\cdot\left(\|\sigma y + x^*\|^2+1\right)^{1/2}\,\mathrm{d}y \le C_k.$$

Assumption 4. *For all t, as $\sigma \to 0$,*

$$\mathrm{Var}_{X\sim\pi}\left[\frac{1}{\sigma^d}k_\sigma(X,x^*)X_j\right] = O\left(\frac{1}{\sigma^d}\right), \quad \textit{for all } j=1,\dots,d,$$

$$\mathrm{Var}_{X\sim\pi}\left[\frac{1}{\sigma^d}k_\sigma(X,x^*)\right] = O\left(\frac{1}{\sigma^d}\right),$$

$$\mathrm{Var}_{X\sim\rho_t}\left[\frac{1}{\sigma^d}k_\sigma(X,x^*)\cdot(\langle w_t^*, X\rangle + b_t^*)X_j\right] = O\left(\frac{1}{\sigma^d}\right), \quad \textit{for all } j=1,\dots,d,$$

$$\mathrm{Var}_{X\sim\rho_t}\left[\frac{1}{\sigma^d}k_\sigma(X,x^*)\cdot(\langle w_t^*, X\rangle + b_t^*)\right] = O\left(\frac{1}{\sigma^d}\right),$$

where $w_t^* := \nabla(h\circ r_t)(x^*)$ and $b_t^* := h(r_t(x^*)) - \langle\nabla(h\circ r_t)(x^*), x^*\rangle$.

Assumption 5. *For all t,*

$$\sup_{x\in\mathcal{X}}\left\|\nabla^2\left(\frac{\pi}{\rho_t}\right)(x)\right\| \leq \kappa_{\rightarrow}\,.$$

Assumption 6. *For all t, taking $\boldsymbol{X} := [X^\top, 1]^\top$*

$$\lambda_{\min}\left\{\mathbb{E}_{X\sim\rho_t}\left[k_\sigma(X,x^*)\boldsymbol{X}\boldsymbol{X}^\top\right]\right\} \geq \sigma^d\cdot\Lambda_{\rightarrow} > 0\,. \tag{15}$$

For our Proposition 5 to hold, Assumptions 3–6 must hold for the marginals $\rho_t^{(m)}$ and $\pi_m^{(m)}$ for all m. We now study this in more detail and show that this results in natural smoothness assumptions on the missing value mechanism.

We first note that Assumption 3,6, and the second part of Assumption 4 involving ρ_t are only altered by considering the missing values in the sense that we need to consider the marginals $\rho_t^{(m)}$ for all m. Thus, we focus on Assumption 5 and the first part of Assumption 4, where it is necessary to consider

$$\pi_m^{(m)}(x^{(m)}) = \pi^{(m)}(x^{(m)})\frac{\mathbb{P}(M=m\mid X^{(m)}=x^{(m)})}{\mathbb{P}(M=m)},$$

for all m such that $\mathbb{P}(M=m)>0$. We abbreviate in the following $\omega_m(x^{(m)}) = \mathbb{P}(M=m\mid X^{(m)}=x^{(m)})/\mathbb{P}(M=m)$ and note that

$$\omega_m(x^{(m)}) \leq \frac{1}{\mathbb{P}(M=m)},\ \int\omega_m(x^{(m)})\pi^{(m)}(x^{(m)}) = 1.$$

Concerning Assumption 4, we notice that, for any real-valued measurable function f of $X^{(m)}$,

$$\begin{aligned}\mathrm{Var}_{X^{(m)}\sim\pi_m^{(m)}}\left[f(X^{(m)})\right] &= \arg\min_{a\in\mathbb{R}}\mathbb{E}_{X^{(m)}\sim\pi_m^{(m)}}\left[(f(X^{(m)})-a)^2\right]\\ &\leq \mathbb{E}_{X^{(m)}\sim\pi_m^{(m)}}\left[\left(f(X^{(m)})-\mathbb{E}_{\pi^{(m)}}\left[f(X^{(m)})\right]\right)^2\right].\end{aligned}$$

The last expectation can be bounded as

$$\begin{aligned}&\mathbb{E}_{X^{(m)}\sim\pi_m^{(m)}}\left[\left(f(X^{(m)})-\mathbb{E}_{\pi^{(m)}}\left[f(X^{(m)})\right]\right)^2\right]\\ &\quad= \int\left(f(x^{(m)})-\mathbb{E}_{X^{(m)}\sim\pi^{(m)}}\left[f(X^{(m)})\right]\right)^2\pi^{(m)}(x^{(m)})\omega_m^{(m)}(x^{(m)})dx^{(m)}\\ &\quad\leq \frac{1}{\mathbb{P}(M=m)}\int\left(f(x^{(m)})-\mathbb{E}_{X^{(m)}\sim\pi^{(m)}}\left[f(X^{(m)})\right]\right)^2\pi^{(m)}(x^{(m)})dx^{(m)}\\ &\quad= \frac{\mathrm{Var}_{X^{(m)}\sim\pi^{(m)}}[f(X^{(m)})]}{\mathbb{P}(M=m)},\end{aligned}$$

showing that Assumption 4 can be written in terms of $\pi^{(m)}$ instead of $\pi_m^{(m)}$. Concerning Assumption 5, we note that since

$$\begin{aligned}\nabla^2\left(\frac{\pi_m^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)}) = \omega_m(x^{(m)})\nabla^2\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)}) + \left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\nabla^2\omega_m(x^{(m)})\\ + \nabla\omega_m(x^{(m)})\left(\nabla\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right)^\top\\ + \nabla\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\left(\nabla\omega_m(x^{(m)})\right)^\top,\end{aligned}$$

it follows that

$$\left\|\nabla^2\left(\frac{\pi_m^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right\| \le |\omega_m(x^{(m)})|\left\|\nabla^2\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right\| + \left|\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right| \|\nabla^2\omega_m(x^{(m)})\| + 2\left\|\nabla\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right\| \left\|\nabla\omega_m(x^{(m)})\right\|.$$

Now, we will assume the second derivatives are uniformly bounded, and that for all m and all t,

$$\sup_{x^{(m)}}\left|\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right| < \infty. \tag{13}$$

By the Landau-Kolmogorov inequality [Kovalenko, 2026] applied over each coordinate, it follows that

$$\sup_{x^{(m)}}\left\|\nabla\omega_m(x^{(m)})\right\| \le C_m\left(\sup_{x^{(m)}}|\omega_m(x^{(m)})|\right)^{1/2}\sup_{x^{(m)}}\left(\left\|\nabla^2\omega_m(x^{(m)})\right\|\right)^{1/2},$$

and similarly with $\nabla(\pi^{(m)}/\rho_t^{(m)})$. Thus, uniform boundedness of second derivatives, as well as $\sup_{x^{(m)}}|\omega_m(x^{(m)})| \le 1$ and (13) imply uniform boundedness of the first derivative. This shows that, no additional assumption on the *first* derivative is needed and leads us to the final set of assumptions:

Assumption 7. *There exists a constant $C_k > 0$ independent of σ, such that for all t and m with $\mathbb{P}(M = m) > 0$,*

$$\frac{1}{2}\int \rho_t^{(m)}(\sigma y + x^*)\exp\bigl(-\|y\|^2/2\bigr)\cdot\|y\|^2\cdot\bigl(\|\sigma y + x^*\|^2 + 1\bigr)^{1/2}\,\mathrm{d}y \le C_k.$$

Assumption 8. *For all t, as $\sigma \to 0$,*

$$\mathrm{Var}_{X^{(m)}\sim\pi^{(m)}}\left[\frac{1}{\sigma^d}k_\sigma(X^{(m)}, x^{*(m)})X_j\right] = O\left(\frac{1}{\sigma^d}\right)\ \forall j \text{ with } m_j = 0,$$

$$\mathrm{Var}_{X^{(m)}\sim\pi^{(m)}}\left[\frac{1}{\sigma^d}k_\sigma(X^{(m)}, x^{*(m)})\right] = O\left(\frac{1}{\sigma^d}\right),$$

$$\mathrm{Var}_{X^{(m)}\sim\rho_t^{(m)}}\left[\frac{1}{\sigma^d}k_\sigma(X^{(m)}, x^{*(m)})\cdot(\langle w_{t,m}^*, X^{(m)}\rangle + b_{t,m}^*)X_j\right] = O\left(\frac{1}{\sigma^d}\right)\ \forall j \text{ with } m_j = 0,$$

$$\mathrm{Var}_{X^{(m)}\sim\rho_t^{(m)}}\left[\frac{1}{\sigma^d}k_\sigma(X^{(m)}, x^{*(m)})\cdot(\langle w_{t,m}^*, X^{(m)}\rangle + b_{t,m}^*)\right] = O\left(\frac{1}{\sigma^d}\right),$$

where $w_{t,m}^* \coloneqq \nabla(h\circ r_t^{(m)})(x^*)$ and $b_{t,m}^* \coloneqq h(r_t^{(m)}(x^*)) - \langle\nabla(h\circ r_t^{(m)})(x^*), x^*\rangle$.

Assumption 9. *For all t, and all m such that $\mathbb{P}(M = m) > 0$,*

$$\sup_{x^{(m)}}\left\|\nabla^2\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right\| \le \kappa_{\to},\ \sup_{x^{(m)}}\left|\left(\frac{\pi^{(m)}}{\rho_t^{(m)}}\right)(x^{(m)})\right| < \infty,\ \sup_{x^{(m)}}\left\|\nabla^2\omega_m(x^{(m)})\right\| < \infty.$$

Assumption 10. *For all t, and all m such that $\mathbb{P}(M = m) > 0$, taking $\boldsymbol{X}^{(m)} := [(X^{(m)})^\top, 1]^\top$*

$$\lambda_{\min}\left\{\mathbb{E}_{X^{(m)}\sim\rho_t^{(m)}}\left[k_\sigma(X^{(m)}, (x^*)^{(m)})\boldsymbol{X}^{(m)}(\boldsymbol{X}^{(m)})^\top\right]\right\} \ge \sigma_m^d\cdot\Lambda_{\to} > 0\,. \tag{15}$$

A.7 Proofs

Next, we provide the proofs of the results in the main text.

Proof of Proposition 1. Recall that $\mathbb{P}_X^{(M)}$ has density $\pi^{(M)}$, while $\mathbb{P}_{X|M}^{(M)}$ has density $\pi_M^{(M)}$. For an arbitrary distribution P_X with density p, we may write

$$\begin{aligned}
&\mathbb{E}_{M\sim\mathbb{P}_M}\left[\mathrm{KL}\left(\mathbb{P}_{X|M}^{(M)} \,\|\, P_X^{(M)}\right)\right] \\
&\quad= \sum_{m\in\mathcal{M}} \mathbb{P}(M=m) \int \log\left(\frac{\pi_m^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi_m^{(m)}(x^{(m)})dx^{(m)} \\
&\quad= \sum_{m\in\mathcal{M}} \mathbb{P}(M=m) \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\frac{\mathbb{P}(M=m \mid x^{(m)})}{\mathbb{P}(M=m)}\right) \pi_m^{(m)}(x^{(m)})dx^{(m)} \\
&\quad= \sum_{m\in\mathcal{M}} \mathbb{P}(M=m) \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi_m^{(m)}(x^{(m)})dx^{(m)} \\
&\qquad+ \sum_{m\in\mathcal{M}} \mathbb{P}(M=m) \int \log\left(\frac{\mathbb{P}(M=m \mid x^{(m)})}{\mathbb{P}(M=m)}\right) \pi_m^{(m)}(x^{(m)})dx^{(m)}\,.
\end{aligned}$$

The second term above is not influenced by the choice of P_X, so we might ignore it for the $\arg\min$ and focus on the first term:

$$\begin{aligned}
&\sum_{m\in\mathcal{M}} \mathbb{P}(M=m) \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi_m^{(m)}(x^{(m)})dx^{(m)} \\
&\quad= \sum_{m\in\mathcal{M}} \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi^{(m)}(x^{(m)})\mathbb{P}(M=m \mid x^{(m)})dx^{(m)}\,.
\end{aligned}$$

Following the same proof as in [Näf and Chérief-Abdellatif, 2026, Proposition 3.1], we obtain that under MAR,

$$\sum_{m\in\mathcal{M}} \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi^{(m)}(x^{(m)})\mathbb{P}(M=m \mid x^{(m)})dx^{(m)} \geq 0\,,$$

while zero is clearly attained for $p^{(m)} = \pi^{(m)}$. This shows that $\mathbb{P}_X$ is one solution to (5). Moreover, as shown in [Näf and Chérief-Abdellatif, 2026, Proposition 3.2], under MAR, $\mathbb{P}(M=0 \mid x) > 0$ for almost all x implies that,

$$\sum_{m\in\mathcal{M}} \int \log\left(\frac{\pi^{(m)}(x^{(m)})}{p^{(m)}(x^{(m)})}\right) \pi^{(m)}(x^{(m)})\mathbb{P}(M=m \mid x^{(m)})dx^{(m)} = 0 \iff \mathbb{P}_X = P_X\,,$$

showing that $\mathbb{P}_X$ is the *unique* solution to (5). On the other hand, if $\mathbb{P}(M=0 \mid x) = 0$ for $x \in A$ with $\mathbb{P}_X(A) > 0$, we can construct a density P_X that is equal to $\mathbb{P}_X$ on A^c and has the same $k \leq d-1$ dimensional marginals, but such that $P_X \neq \mathbb{P}_X$ on A. This difference in turn will be masked by $\mathbb{P}(M=0 \mid x) = 0$. □

Proof of Lemma 2. Using the chain rule and the law of total probability, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0}\mathcal{F}[\rho_\varepsilon] &= \sum_m \mathbb{P}(M=m)\int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0} f\left(\frac{\pi_m^{(m)}(x^{(m)})}{\rho_\varepsilon^{(m)}(x^{(m)})}\right)\rho_\varepsilon^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&= \sum_m \mathbb{P}(M=m)\int \left(f\left(\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}\right) - f'\left(\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}\right)\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}\right) \\
&\qquad\qquad \cdot \frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0}\rho_\varepsilon^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&= -\sum_m \mathbb{P}(M=m)\int \Big(h\circ r^{(m)}\Big)(x^{(m)})\frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0}\rho_\varepsilon^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&= -\sum_m \mathbb{P}(M=m)\int \Big(h\circ r^{(m)}\Big)(x^{(m)})\frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0}\int \rho_\varepsilon(x)\,\mathrm{d}x^{(1-m)}\,\mathrm{d}x^{(m)} \\
&= -\int \sum_m \mathbb{P}(M=m)\Big(h\circ r^{(m)}\Big)(x^{(m)})\frac{\mathrm{d}}{\mathrm{d}\varepsilon}|_{\varepsilon=0}\rho_\varepsilon(x)\,\mathrm{d}x\,.
\end{aligned}$$

The result follows from the definition of $\frac{\delta\mathcal{F}[\rho]}{\delta\rho}(x)$ being the first variation of $\mathcal{F}[\rho]$. □

Proof of Lemma 3. Assume ψ is convex and lower semi-continuous. Then, by duality, we have

$$\psi(u) = \sup_{v\in\mathrm{dom}(\psi^*)}\{vu-\psi^*(v)\}\,.$$

It follows that, for each $m\in\mathcal{M}$,

$$\begin{aligned}
&D_\psi\Big(\pi_m^{(m)},\rho^{(m)}\Big) \\
&\quad= \int \rho^{(m)}(x^{(m)})\psi\left(\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}\right)\mathrm{d}x^{(m)} \\
&\quad= \int \rho^{(m)}(x^{(m)})\sup_{v\in\mathrm{dom}(\psi^*)}\left\{v\cdot\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}-\psi^*(v)\right\}\mathrm{d}x^{(m)} \\
&\quad= \sup_{g_m:\,\mathbb{R}^{d_m}\to\mathbb{R}}\left\{\int \rho^{(m)}(x^{(m)})\left(g_m(x^{(m)})\frac{\pi_m^{(m)}(x^{(m)})}{\rho^{(m)}(x^{(m)})}-\psi^*\Big(g_m(x^{(m)})\Big)\right)\mathrm{d}x^{(m)}\right\} \\
&\quad= \sup_{g_m:\,\mathbb{R}^{d_m}\to\mathbb{R}}\left\{\int g_m(x^{(m)})\pi_m^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} - \int \psi^*\Big(g_m(x^{(m)})\Big)\rho^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)}\right\}.
\end{aligned}$$

Since, by duality, $\psi^*(\psi'(r)) = r\psi'(r)-\psi(r)$, the specific choice $g_m=\psi'\circ r^{(m)}$ yields

$$\begin{aligned}
&\int g_m(x^{(m)})\pi_m^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} - \int \psi^*(g_m(x^{(m)}))\rho^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&\quad= \int \psi'(r^{(m)}(x^{(m)}))\pi_m^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} - \int \psi^*(\psi'(r^{(m)}(x^{(m)})))\rho^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&\quad= \int \psi'(r^{(m)}(x^{(m)}))\pi_m^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&\qquad - \int \Big(r^{(m)}(x^{(m)})\psi'(r^{(m)}(x^{(m)})) - \psi(r^{(m)}(x^{(m)}))\Big)\rho^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&\quad= \int \psi(r^{(m)}(x^{(m)}))\rho^{(m)}(x^{(m)})\,\mathrm{d}x^{(m)} \\
&\quad= D_\psi\Big(\pi_m^{(m)},\rho^{(m)}\Big)\,.
\end{aligned}$$

The result follows from the observation that $h(r)=rf'(r)-f(r)=\psi'(r)$.

Note that Liu et al. [2024] state that ψ is chosen such that $\psi'(r)$ is equal to $rf'(r)-f(r)$ *up to some constant*. Accordingly, they use $\psi(r)=(r-1)^2/2$ and $\psi^*(g)=g^2/2+g$ for the KL divergence.

We want to emphasize that this is not a valid choice when using $f(r) = r\log(r)$, since Equation (3) only holds if $h(r) = \psi'(r)$. This is a minor issue, though, as it only influences the intercept b_t in equation (4) and not the sought-after slope w_t. □

Proof of Theorem 4. Let $(\tilde{X}_t, X) \sim \gamma_t$ according to Assumption 1. Then $(\tilde{X}_t + \eta v_t(\tilde{X}_t), X)$ is a coupling of $\bar{\rho}_{t+1}$ and π. Hence, we have

$$\begin{aligned}
\mathcal{W}_2^2(\bar{\rho}_{t+1}, \pi) &\leq \mathbb{E}_{(\tilde{X}_t, X)\sim\gamma_t}\left[\left\|\tilde{X}_t - X + \eta v_t(\tilde{X}_t)\right\|^2\right] \\
&\leq \mathbb{E}\left[\left\|\tilde{X}_t - X\right\|^2\right] + 2\eta\mathbb{E}\left[\langle \tilde{X}_t - X, v_t(\tilde{X}_t)\rangle\right] + \eta^2\mathbb{E}\left[\left\|v_t(\tilde{X}_t)\right\|^2\right] \\
&= \mathcal{W}_2^2(\tilde{\rho}_t, \pi) + 2\eta\int \langle x - y, v_t(x)\rangle \,\mathrm{d}\gamma_t(x,y) + \eta^2\int \|v_t(x)\|^2 \,\mathrm{d}\tilde{\rho}_t(x) \\
&\leq \mathcal{W}_2^2(\tilde{\rho}_t, \pi) - 2\eta\lambda\mathcal{W}_2^2(\tilde{\rho}_t, \pi) + \eta^2 L^2\mathcal{W}_2^2(\tilde{\rho}_t, \pi) \\
&= (1 - 2\eta\lambda + \eta^2 L)\mathcal{W}_2^2(\tilde{\rho}_t, \pi)\,.
\end{aligned}$$

Now, if $\eta \leq \frac{\lambda}{L^2}$, it follows that

$$\mathcal{W}_2(\bar{\rho}_{t+1}, \pi) \leq \sqrt{1 - \eta\lambda}\mathcal{W}_2(\tilde{\rho}_t, \pi) \leq \left(1 - \frac{\eta\lambda}{2}\right)\mathcal{W}_2(\tilde{\rho}_t, \pi),$$

where we have used the fact that $\sqrt{1-x} \leq 1 - \frac{x}{2}$ for all $x \in [0,1]$. Next, we couple the exact and estimated updates using the same $\tilde{X}_t \sim \tilde{\rho}_t$. Then, we have

$$\begin{aligned}
\mathcal{W}_2(\tilde{\rho}_{t+1}, \bar{\rho}_{t+1}) &\leq \left(\mathbb{E}\left[\left\|\tilde{X}_t + \eta v_t(\tilde{X}_t) - (\tilde{X}_t + \eta\hat{v}_t(\tilde{X}_t))\right\|^2\right]\right)^{\frac{1}{2}} \\
&= \eta\left(\int \|v_t(x) - \hat{v}_t(x)\|^2 \,\mathrm{d}\tilde{\rho}_t(x)\right)^{\frac{1}{2}} \\
&\leq \eta\varepsilon\,,
\end{aligned}$$

by Assumption 2. With the triangle inequality, it follows that

$$\begin{aligned}
\mathcal{W}_2(\tilde{\rho}_{t+1}, \pi) &\leq \mathcal{W}_2(\tilde{\rho}_{t+1}, \bar{\rho}_{t+1}) + \mathcal{W}_2(\bar{\rho}_{t+1}, \pi) \\
&\leq \eta\varepsilon + \left(1 - \frac{\eta\lambda}{2}\right)\mathcal{W}_2(\tilde{\rho}_t, \pi)\,.
\end{aligned}$$

Iteration over t yields

$$\mathcal{W}_2(\tilde{\rho}_T, \pi) \leq \left(1 - \frac{\eta\lambda}{2}\right)^T \mathcal{W}_2(\tilde{\rho}_0, \pi) + \eta\varepsilon\sum_{k=1}^{T-1}\left(1 - \frac{\lambda\eta}{2}\right)^k.$$

The result follows since $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$, in particular for $x \in [0,1]$, and

$$\sum_{k=1}^{T-1}\left(1 - \frac{\lambda\eta}{2}\right)^k \leq \frac{2}{\lambda\eta}\left(1 - \left(1 - \frac{\lambda\eta}{2}\right)^T\right) \leq \frac{2}{\lambda\eta}\,.$$

□

Proof of Proposition 5. For fixed time step $t \in \{0, 1, \ldots, T-1\}$, missingness pattern $m \in \mathcal{M}$ and $x^* \in \mathcal{X}$, Liu et al. [2024, Theorem 4.8] states that

$$\left\|w_{t,m}(x^*) - \nabla(h \circ r_t^{(m)})(x^*)\right\|_2 \leq C_{t,m,x^*}^{(1)}\frac{1}{\sqrt{n_m\sigma^{d_m}}} + C_{t,m,x^*}^{(2)}\sigma^2\,. \tag{14}$$

By the triangle inequality, we have

$$\begin{aligned}
&\left\| \sum_{m\in\mathcal{M}} \frac{n_m}{n} w_{t,m}(x^*) - \mathbb{E}_M\left[\nabla(h\circ r_t^{(M)})(x^*)\right]\right\|_2 \\
&\quad\le \sum_{m\in\mathcal{M}} \frac{n_m}{n}\left\| w_{t,m}(x^*) - \nabla(h\circ r_t^{(m)})(x^*)\right\|_2 \\
&\qquad + \left\| \sum_{m\in\mathcal{M}} \frac{n_m}{n}\nabla(h\circ r_t^{(m)})(x^*) - \mathbb{E}_M\left[\nabla(h\circ r_t^{(M)})(x^*)\right]\right\|_2 .
\end{aligned}$$

The second term is the standard Monte Carlo estimation error vanishing at a rate of $\mathcal{O}\big(n^{-1/2}\big)$, while the first term can be bounded by

$$\begin{aligned}
&\sum_{m\in\mathcal{M}} \frac{n_m}{n}\left\| w_{t,m}(x^*) - \nabla(h\circ r_t^{(m)})(x^*)\right\|_2 \\
&\quad\le \sum_{m\in\mathcal{M}} \frac{n_m}{n}\left(C^{(1)}_{t,m,x^*}\frac{1}{\sqrt{n_m\sigma^{d_m}}} + C^{(2)}_{t,m,x^*}\sigma^2\right) \\
&\quad\le \left(\max_{m\in\mathcal{M}} C^{(1)}_{t,m,x^*}\right)\frac{1}{n\sqrt{\sigma^d}}\sum_{m\in\mathcal{M}}\sqrt{n_m} + \max_{m\in\mathcal{M}} C^{(2)}_{t,m,x^*}\sigma^2 \\
&\quad\le \left(\max_{m\in\mathcal{M}} C^{(1)}_{t,m,x^*}\right)\frac{1}{n\sqrt{\sigma^d}}\sqrt{|\mathcal{M}|\sum_{m\in\mathcal{M}} n_m} + \max_{m\in\mathcal{M}} C^{(2)}_{t,m,x^*}\sigma^2 \\
&\quad= \left(\max_{m\in\mathcal{M}} C^{(1)}_{t,m,x^*}\right)\sqrt{\frac{|\mathcal{M}|}{n\sigma^d}} + \max_{m\in\mathcal{M}} C^{(2)}_{t,m,x^*}\sigma^2 ,
\end{aligned}$$

where the first inequality applies (14), the second one follows from $\sigma \le 1$ and $d_m \le d$, the third one uses the Cauchy-Schwartz inequality, and the last equality holds because $\sum_m n_m = n$. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction list concrete contributions (optimality of KL minimization, FLOWGEM algorithm, approximation consistency, and empirical performance) which are established in the paper (Proposition 1, Equations (8) and (9), Proposition 5, and Table 1 and Figure 2, respectively). The scope is also clearly stated as missing values under the MAR assumption.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of the work are discussed in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All assumptions are stated in the propositions and lemmas in the main text, while their complete proofs are provided in Appendix A.7.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Details on the FLOWGEM algorithm are provided in Appendix A.2, while a detailed description of the experimental setup, real datasets, missing value generation, and the implementation of competing algorithms is provided in Appendix A.3. The code for replication of the experiments can be found on `https://github.com/gkremling/FLOWGEM-experiments`.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We only used open access datasets; their sources are provided in Table 3. The code for replication of the experiments can be found on `https://github.com/gkremling/FLOWGEM-experiments`. An open-source implementation of the proposed method is available as a Python package at `https://pypi.org/project/flowgem/`, with source code provided at `https://github.com/gkremling/FLOWGEM-package.git`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All details on the algorithm and experimental setup are given in Appendix A.2 and A.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: For the simulation study, we report boxplots (Figure 2), illustrating fluctuations over 20 replications and for the real datasets, we report mean and standard deviations over 10 replications of each method.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Details on the computer resources are provided in Appendix A.3, and computation times are reported in Table 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research complies fully with the NeurIPS Code of Ethics. It involves no human subjects, personal data, or private information. All datasets used are standard, publicly available benchmarks. The work advances fundamental methodology in generative modeling; no foreseeable dual-use risks or harmful applications are anticipated beyond those inherent to generative modeling in general.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This paper presents foundational methodological contributions to the treatment of missing values. Direct societal impact is not immediately foreseeable, though we acknowledge that advances in generative modeling may have broader downstream implications, both positive and negative, typical of the field.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not involve sensitive data or high-risk models.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We only use public datasets with appropriate references provided in Table 3. The code bases for the competing methods are also publicly available and properly credited in Appendix A.3.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: The code for replication of the experiments and the use of the method can be found on Github, and we will release the url with the camera-ready version. The repository also includes documentation.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard component.

    Guidelines:

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