

Correcting inherent bias in iterative imputation of missing data

Katarzyna Reluga, HU Berlin

Interdisciplinary Centre for Digitality and Digital Methods
12.06.2026

Joint work with



Luke Shannon

PhD Student

School of Mathematics,
University of Bristol



Song Liu

Associate Professor
School of Mathematics,
University of Bristol

Missing data is everywhere

Real-world datasets are almost never complete.

In digital humanities this is especially familiar:

- ▶ **Historical records** with blank dates, names, or locations
- ▶ **Survey respondents** who skip sensitive questions
- ▶ Linked **databases** where an entry appears in only one of them

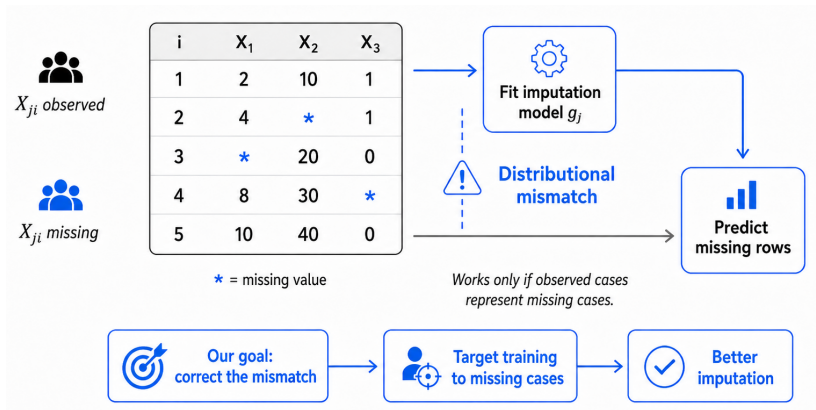
The standard fix is **imputation** – use the data you **do** have to predict the values you **don't**, by training a **predictive model** on the observed entries and using it to fill in the gaps.

Imputation is now a **routine** preprocessing step in statistical and machine learning pipelines – but is it done correctly?

Motivation: why missing data is difficult

Basic recipe of many modern imputation methods:

cycle through each variable $\rightarrow$ train a model on observed rows $\rightarrow$
fill in the missing rows $\rightarrow$ repeat



What we contribute

We address the mismatch between the cases we observe and the cases we need to impute.

Theory

We show that imputation can be reframed as a **weighted learning problem**. These weights make the observed data look more like the missing data.

Algorithm

Weighted Iterative Imputation – a lightweight wrapper around any existing iterative method, incorporating the weights at no substantial extra cost.

Empirical Evidence:

Across 8 real-world datasets and 3 model classes, the weighted version gives: **lower prediction error** and **better distributional recovery**.

- ▶ Motivation
- ▶ **Background**
- ▶ Our method
- ▶ Experiments
- ▶ Conclusions

What is missing data?

- ▶ $X = (X_1, \dots, X_D)$ – a complete data vector
- ▶ $R = (R_1, \dots, R_D)$ – the missingness pattern
- ▶ $\tilde{X}$ – a partially observed vector

$$R_j = \begin{cases} 1 & \text{if } X_j \text{ is observed} \\ 0 & \text{if } X_j \text{ is missing} \end{cases} \quad \tilde{X}_j = \begin{cases} X_j, & \text{if } R_j = 1, \\ *, & \text{if } R_j = 0. \end{cases}$$

$$\mathbf{X} = \begin{pmatrix} 2 & 10 & 1 \\ 4 & 15 & 1 \\ 6 & 20 & 0 \\ 8 & 30 & 1 \end{pmatrix}$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} 2 & 10 & 1 \\ 4 & * & 1 \\ * & 20 & 0 \\ 8 & 30 & * \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Standard missingness regimes:

- ▶ Missing completely at random (**MCAR**) — $p(R | X) = p(R)$: missingness unrelated to data.
 - ▶ Example: random scanning or upload failure.
- ▶ Missing at random (**MAR**) — $p(R | X) = p(R | X_{\text{obs}})$: missingness depends only on observed values
 - ▶ Example: author information missing more often for older documents.
- ▶ Missing not at random (**MNAR**) — missingness depends on the missing value itself.
 - ▶ Example: people with high alcohol consumption are less likely to report their alcohol intake.

Standard missingness regimes:

- ▶ Missing completely at random (**MCAR**) — $p(R | X) = p(R)$: missingness unrelated to data.
 - ▶ Example: random scanning or upload failure.
- ▶ Missing at random (**MAR**) — $p(R | X) = p(R | X_{\text{obs}})$: missingness depends only on observed values ← our setup
 - ▶ Example: author information missing more often for older documents.
- ▶ Missing not at random (**MNAR**) — missingness depends on the missing value itself.
 - ▶ Example: people with high alcohol consumption are less likely to report their alcohol intake.

Imputation functions

An imputation function is a map

$$g : \tilde{\mathcal{X}} \rightarrow \mathcal{X},$$

which takes a partially observed vector $\tilde{x}$ and returns a complete vector

$$g(\tilde{x}) = (g_1(\tilde{x}), \dots, g_d(\tilde{x})).$$

The map must preserve observed entries:

$$g_i(\tilde{x}) = \tilde{x}_i \quad \text{whenever } r_{\tilde{x},i} = 1.$$

In words: imputation only changes the missing values; observed values are copied unchanged.

Imputation as risk minimisation

Under MSE loss, the imputation risk decomposes over coordinates that can be missing:

$$\mathcal{J}(g) = \sum_{i \in \mathcal{P}} \mathbb{P}(R_i = 0) \mathcal{J}_i(g_i),$$

where

$$\mathcal{J}_i(g_i) = \mathbb{E} \left[\{g_i(\tilde{X}) - X_i\}^2 \mid R_i = 0 \right].$$

In words: for each feature, we want the imputed value to be close to the true value among the entries that are actually missing.

The catch: this risk depends on $X_i \mid R_i = 0$ (**not observed** X_i).

Existing methods often train on observed rows, $R_i = 1$, thereby **ignoring the possible distributional shift** between observed and missing populations.

Iterative imputation

Missing entries in multiple columns are handled by a round-robin algorithm:

1. Initialise all missing values with column means
2. For each partially observed variable X_j :
 - (a) Use the current imputed dataset as input
 - (b) Fit a model g_j predicting X_j from all other variables, using observed rows only
 - (c) Replace missing entries of X_j with predictions from $\hat{g}_j$
3. Repeat until convergence

Well-known examples: **MICE**, **MissForest**, **HyperImpute**.

Step 2(b) **is where the distributional mismatch enters** — and where we apply our correction.

Example toy data:

Original data

i	X_1	X_2	X_3
1	2	10	1
2	4	*	1
3	*	20	0
4	8	30	*
5	10	40	0

$\Rightarrow$

Initialise

Initialised data

i	X_1	X_2	X_3
1	2	10	1
2	4	25	1
3	6	20	0
4	8	30	0.5
5	10	40	0

- ▶ X_2 has missing values – fit $X_2 = g_2(X_1, X_3)$ using observed cases $R_2 = 1$.
- ▶ For a missing entry, e.g. $X_{2,2}$, impute $\hat{X}_{2,2} = \hat{g}_2(X_{1,2}, X_{3,2})$.
- ▶ The model is trained on observed cases, $R_2 = 1$, but applied to missing cases, $R_2 = 0$.

- ▶ Motivation
- ▶ Background
- ▶ **Our method**
- ▶ Experiments
- ▶ Conclusions

The key result: rewriting the risk

Proposition. Under MAR and an overlap condition, the true imputation risk can be expressed entirely over observed data:

$$\mathcal{J}_i(g_i) = \mathbb{E}_{X, R_{\neg i}} \left[w_i(X_{\text{obs}}, R_{\neg i}) \left\{ g_i(\tilde{X}) - X_i \right\}^2 \mid R_i = 1 \right]$$

where the importance weight is the density ratio:

$$w_i(x_{\text{obs}}, r_{\neg i}) := \frac{p(x_{\text{obs}}, r_{\neg i} \mid r_i = 0)}{p(x_{\text{obs}}, r_{\neg i} \mid r_i = 1)}.$$

In words: we can use observed rows, but must give higher weight to those whose covariate values resemble the missing population.

- ▶ $w_i > 1$: this observed point looks like the missing population – upweight it
- ▶ $w_i < 1$: this observed point is over-represented among observed rows – downweight it

Surrogate risk in the round-robin algorithm

In practice, we replace $\mathcal{J}_i(g_i)$ by a **surrogate risk** evaluated on the current imputed dataset $\hat{X}^{(t)}$. At iteration t , when updating variable X_i :

- ▶ treat the current imputed predictors $\hat{X}_{-i}^{(t)}$ as fixed;
- ▶ fit g_i only on rows where X_i is observed, $R_i = 1$;
- ▶ reweight observed rows so that they mimic the missing population, $R_i = 0$.

The surrogate objective is

$$\mathcal{J}_i^{(t)}(g_i) = \mathbb{E} \left[w_i(\hat{X}_{-i}^{(t)}) \{g_i(\hat{X}_{-i}^{(t)}) - X_i\}^2 \mid R_i = 1 \right].$$

In words: we train on observed outcomes, but weight them to look like the cases where X_i is missing.

This avoids pattern-specific models by updating each variable on the current completed data matrix.

Estimating the weights

We do not know $p(x \mid R_j = 0)$ and $p(x \mid R_j = 1)$ directly.

Solution: train a binary classifier $\hat{\eta}_j$ to distinguish observed rows ($R_j = 1$) from missing rows ($R_j = 0$): $\hat{\eta}_j(x) \approx p(R_j = 1 \mid x)$

By Bayes' rule, the density ratio can be rewritten in terms of conditional missingness probabilities:

$$\frac{p(x \mid R_j = 0)}{p(x \mid R_j = 1)} = \frac{p(R_j = 0 \mid x)}{p(R_j = 1 \mid x)} \cdot \frac{p(R_j = 1)}{p(R_j = 0)}.$$

Dropping the constant marginal factor gives: $\hat{w}_j(x) = \frac{1 - \hat{\eta}_j(x)}{\hat{\eta}_j(x)}$

If missingness is predictable from the other variables, the classifier captures the observed–missing distribution shift; the odds convert this shift into a correction weight.

Our algorithm: weighted iterative imputation

1. Initialise all missing values (column means)
2. For each partially observed variable X_j :
 - (a) Fit a binary classifier to predict R_j from all other (imputed) variables $\Rightarrow$ get $\hat{w}_j$
 - (b) Fit a weighted regression model minimising:

$$\sum_{\{k: R_j^{(k)}=1\}} \hat{w}_j^{(k)} (g_j(\hat{x}_{-j}^{(k)}) - x_j^{(k)})^2$$

- (c) Re-impute missing entries of X_j with $\hat{g}_j$
3. Repeat until convergence

Works with any model class: linear regression, random forest, MLP.

The weighting step is a lightweight wrapper around existing iterative imputation.

- ▶ Motivation
- ▶ Background
- ▶ Our method
- ▶ **Experiments**
- ▶ Conclusions

Experimental setup

We simulate missingness under a Missing At Random mechanism.

For each dataset:

- ▶ Select up to four continuous variables to make partially observed: $\mathcal{M} \subseteq \{1, \dots, d\}$
- ▶ For each incomplete variable X_i , choose fully observed covariates: $\mathcal{C}_i \subseteq \{1, \dots, d\} \setminus \mathcal{M}$.
- ▶ Generate missingness using a logistic model:
$$\mathbb{P}(R_i = 1 \mid X) = \sigma\left(\alpha \sum_{j \in \mathcal{C}_i} X_j + \beta_i\right), \sigma(z) = \frac{1}{1+e^{-z}}.$$

Here, α controls the MAR strength, while β_i is chosen to obtain a 30% missingness rate.

We use 35 random seeds and vary $\alpha \in \{-3, -2, -1, 0, 1, 2, 3\}$, giving $35 \times 7 = 245$ simulation settings per dataset.

Data sets and performance metrics

8 real-world datasets:

- ▶ Tabular: Blood, Airfoil, California, Letter
- ▶ Time series: Gas, Metro, Motor Acceleration
- ▶ Image: MNIST

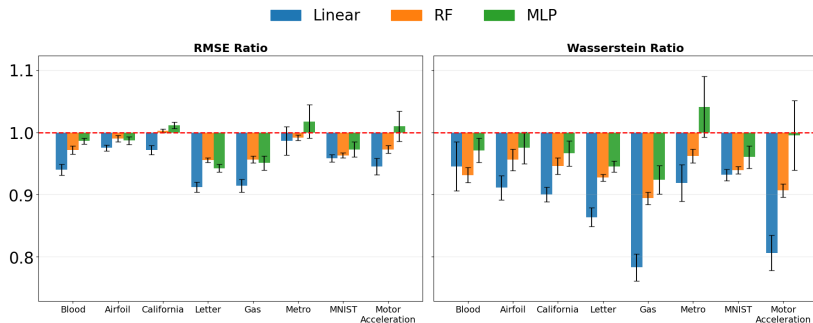
Three model classes: linear regression, Random Forest (RF), MLP.

Metrics:

- ▶ **RMSE** over imputed entries (lower = better)
- ▶ **Wasserstein distance:** how well the imputed distribution matches the true distribution (lower = better)

Reported as ratio weighted / unweighted. Values below the red dashed line at 1.0 mean our method wins.

Main results: weighted vs. unweighted

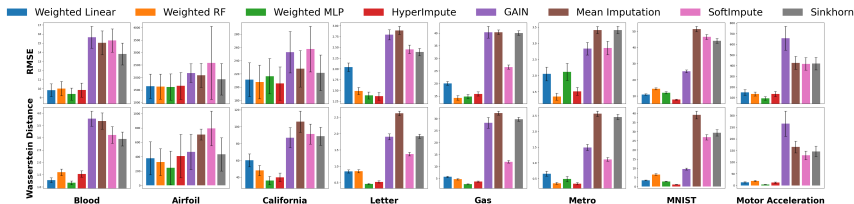


Each bar: mean ratio (weighted / unweighted).

Below 1.0 = improvement.

RMSE improves in 20/24 settings; Wasserstein distance improves in 23/24 settings.

Comparison with the state-of-the-art methods

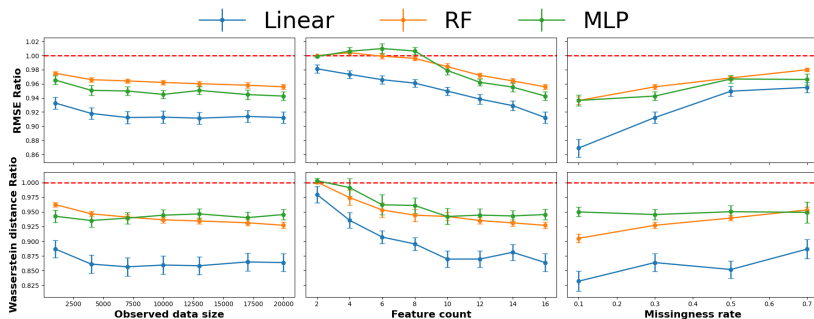


Comparison with **HyperImpute**, **GAIN**, **Sinkhorn**, **SoftImpute**.

Lower values indicate better performance. The figure compares a broad range of methods, showing that the weighted variants are competitive across all scenarios.

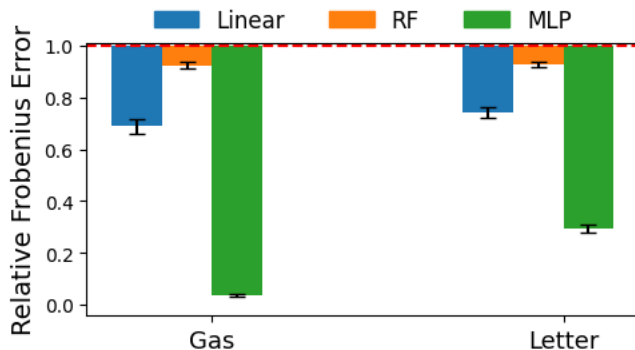
Bars denote mean performance, with error bars given by ± 1.96 SEM, corresponding to approximate 95% confidence intervals.

Sensitivity: varying dataset characteristics



Varying sample size (left), number of features (centre), and missingness rate (right) on the Letter dataset. In 55 of 57 settings the method does not degrade performance and often improves it.

Downstream tasks: Correlation matrix recovery



- ▶ Does the imputation preserve the **joint structure** of the data?
- ▶ For each imputed dataset, we compute its correlation matrix and compare it with the complete data correlation matrix.
- ▶ Performance is measured using the Frobenius error:
$$\|\widehat{\text{Cor}}(X_{\text{imp}}) - \text{Cor}(X_{\text{true}})\|_F.$$
- ▶ Lower values $\rightarrow$ better recovery of the dependence structure.

- ▶ Motivation
- ▶ Background
- ▶ Our method
- ▶ Experiments
- ▶ Conclusions

Summary

- ▶ Iterative imputation trains on observed data but must predict for missing data — this is a covariate shift problem
- ▶ Existing methods often ignore this shift, producing imputation models optimised for the wrong population
- ▶ Under MAR, the covariate shift yields an importance-weighted risk over observed data.
- ▶ Our method account for this shift with minimal extra cost.

Empirical results:

- ▶ **3% lower RMSE** and **7% lower Wasserstein distance** on average.
- ▶ Robust across 8 datasets, 3 model classes, and varying missingness rates.

Take-home message: *who you train on* matters; reweighting observed rows improves imputation.

Limitations and future work

Current limitations:

- ▶ Weighting can increase variance for flexible models (MLP) – bias–variance trade-off
- ▶ Limited gains under weak MAR / near-MCAR regimes
- ▶ Requires MAR; performance degrades gradually under MNAR (no catastrophic failure)

Future directions:

- ▶ Methods robust to weak distributional shift
- ▶ Uncertainty quantification
- ▶ More sophisticated density ratio estimation
- ▶ Application to text and categorical data
- ▶ Extension to the MNAR setting

Questions?

Manuscript: Shannon, Liu, Reluga (2026) Distribution Shift in Missing Data Imputation: A Risk-Based Perspective and Importance-Weighted Correction under MAR

