Missing Data Imputation by Mutual Information Minimization

Song Liu (song.liu@bristol.ac.uk)

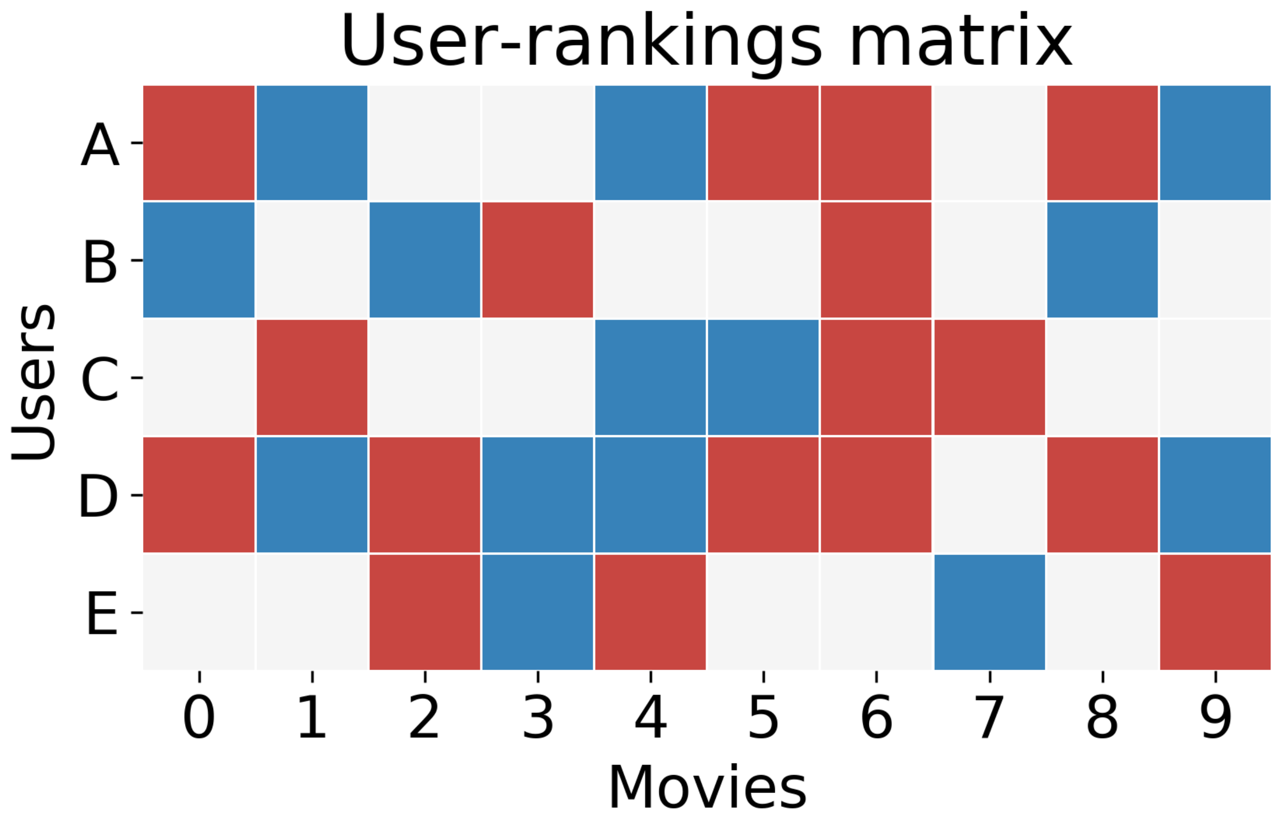
joint work with Jiahao Yu, Qizhen Ying, Leyang Wang, Ziyue Jiang

based on Section 5.3 of arXiv:2305.15577, ICML2024

and arXiv:2505.11749, NeurIPS2025

Problem: Missing Data (Intuition)

The Netflix Prize (https://en.wikipedia.org/wiki/Netflix_Prize)



Problem: Missing Data (Mathy)

We observe pairs of data and mask: $\mathcal{D} = \{(x, m)\}$,

$$x_j = \begin{cases} x_j^*, & \text{if } m_j = 1 \\ \text{NaN}, & \text{if } m_j = 0 \end{cases}.$$

- $x^* \in \mathbb{R}^d$, true data. $m \in \{0, 1\}^d$, missing pattern.
- **Goal**: Given $\mathcal{D}$, guess $\left\{ x_j^* \middle| m_j = 0 \right\}$.

Types of Missingness

Is m dependent of x^*?

- If $m \perp\!\!\!\perp x^*$, Missing Completely at Random (MCAR)
- If $m \perp\!\!\!\perp x^*_{1-m} | x^*_m$, Missing at Random (MAR)
- Otherwise, Missing not at Random (MNAR)

Example, Score = (Math, Physics)

- Exam scores are randomly deleted.
- Math scores are randomly deleted if Physics scores < 50.
- Math scores are randomly deleted if Math scores < 50.

Difficulty: MCAR < MAR < MNAR

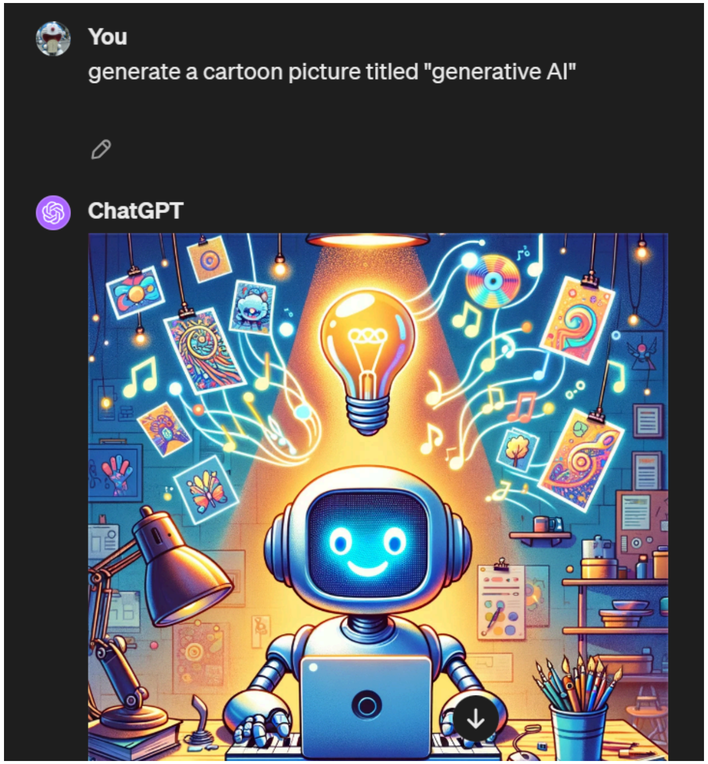
How to Impute?

- **One-shot** (usually as a placeholder):
  - Fill $x_{1-m}$with the mean/median the observed values.
  - Does not fully utilize information in the dataset.
- **Round-robin** (e.g., MICE, missForest, Hyperimpute):
  i. Initialize the dataset with One-shot impute.
  ii. For all j,
     a. Fit a node-wise probability model $p(x_j|x_{-j})$.
     b. Impute x_j with a sample from $p(x_j|x_{-j})$, if $m_j = 0$.
- **Others** (OTimpute, MIRACLE, MIWAE)

Impute without Model?

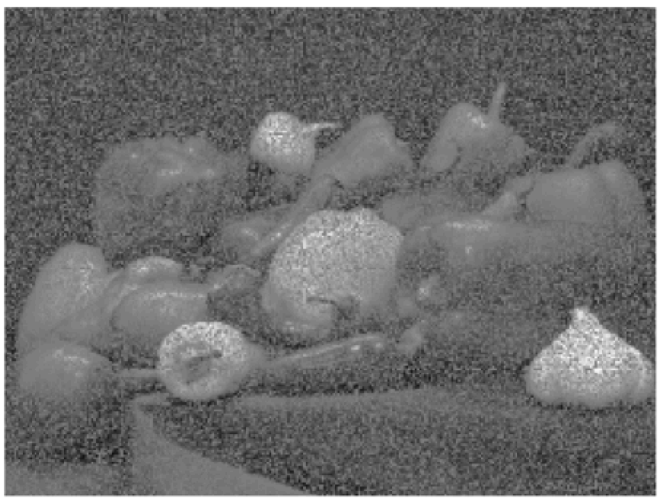
- Much earlier works rely on parametric or non-parametric model fitting.
- Can I impute missing data without fitting a probabilistic model?

Generative AI



- Imputation is naturally a generative modelling problem.
- impute = generate conditionally!

Generative Adversarial Impute Net (intuition) (Yoon et al., 2018)



- Good imputation $\approx$ I cannot tell whether a pixel is imputed or not.

Generative Adversarial Impute Net (intuition)

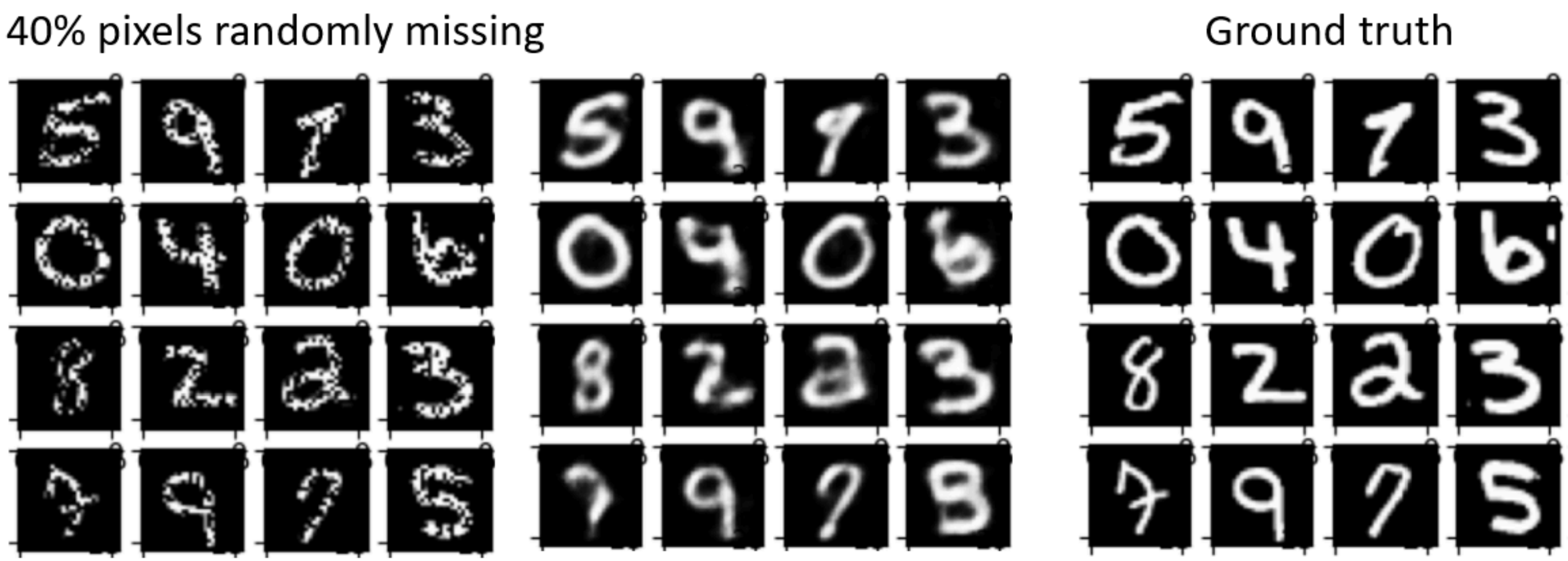
1. Impute the dataset with **One-shot** impute.
2. **Train a classifier** to predict whether a feature is imputed or not given the current imputation.
3. **Imputer data** so that the classifier trained in step 2 cannot tell whether a feature is imputed or not.
4. Repeat 2 and 3.

Generative Adversarial Impute Net (mathy)

1. Given the initial imputation and mask $(\hat{x}, m)$.
2. For each feature j, train a binary classifier to **predict missingness** m_j with $\hat{m}_j := f_j(\hat{x}, m_{-j}) \in [0,1]$ by
$$\min_{f_j} \mathrm{CrossEnt}(m_j, \hat{m}_j)$$
3. Update $\hat{x}$, so the above loss is maximized across all j:
$$\max_{\hat{x}} \sum_j \mathrm{CrossEnt}(m_j, \hat{m}_j)$$
4. Repeat 2 and 3.

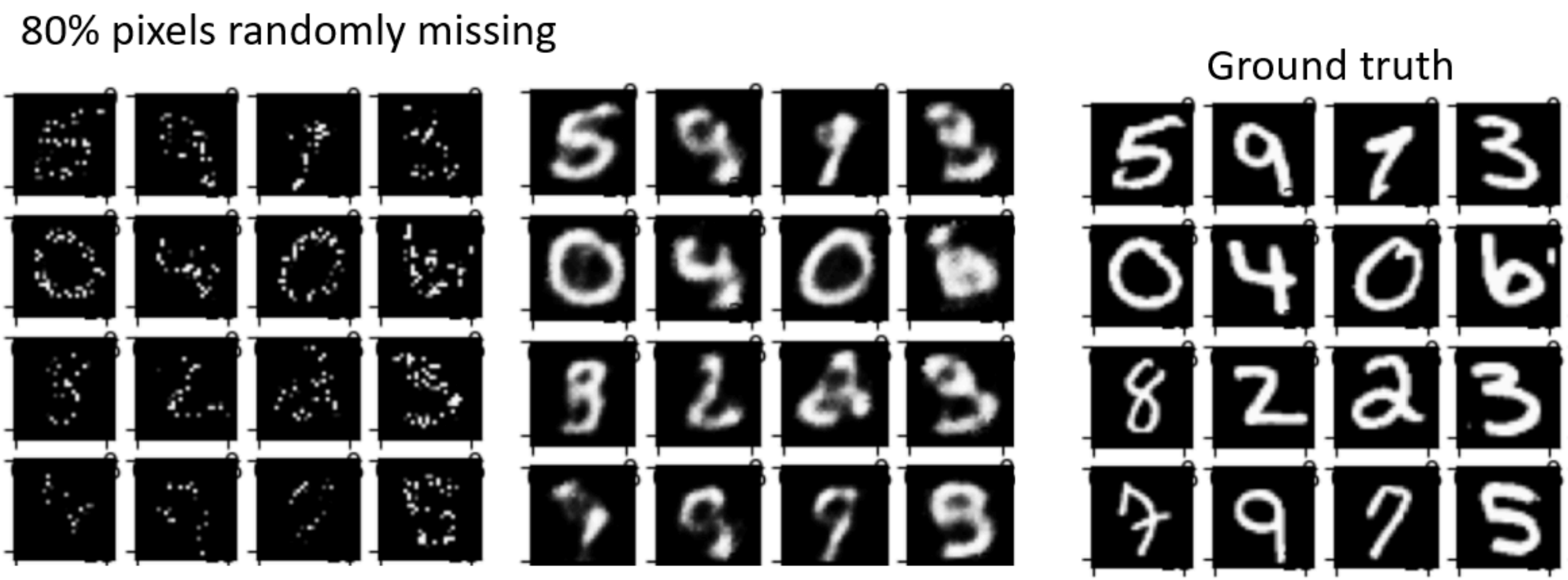
This approach is called GAIN (Yoon et al., 2018).

Performs well!



The blurriness is partly due to the CNN architecture.

Performs well!



- Generative method works reasonably well on high-dimensional data.
- Round-robin method usually struggle on high-dimensional data due to its dimension-wise structure.

Why does GAIN work?

- The classifier f is a probabilistic model of the missingness $p(m_j|\hat{x}, m_{-j})$, minimizing a cross-entropy loss.
- The imputer maximizes the cross entropy loss:
  - $\max_{\hat{x}} \sum_j \mathbb{E}_{\hat{x},m} - \log p(m_j|\hat{x}, m_{-j})$
  - Pseudo-likelihood approximation $\sum_j \log p(m_j|\hat{x}, m_{-j}) \approx \log p(m|x)$.
- Thus, the imputer approximately minimizes the mutual information between $\hat{x}$ and m
  - $\min_{\hat{x}} \mathbb{E}_{\hat{x},m} \log p(m|\hat{x}) \approx \min_{\hat{x}} \mathrm{MI}[\hat{x}, m]$

Why does GAIN work?

- GAIN imputes by breaking dependency between m and $\hat{x}$.
- If we can impute the missing values **perfectly**, i.e., $\hat{x} = x^*$, then $\hat{x} \perp\!\!\!\perp m$, by MCAR assumption.
- GAIN enforces a **necessary condition** of a **perfect imputation**.
    - Necessary, not sufficient, as missing data imputation is ill-defined problem.
    - GAIN has additional loss terms to ensure the imputation stays close to the true values.

Imputation by Minimizing MI

- GAIN only approximately minimizes the MI
  - Joint probability $\approx$ Pseudo-likelihood
- Can we minimize MI exactly?
- Minimizing of MI = minimizing of KL divergence.
  - $\mathrm{MI}[\hat{x}, m] := \mathrm{KL}[p_{\hat{x},m} | p_m p_{\hat{x}}]$.
- **Problem**: KL is intractable.
  - not without complicated approximation scheme.
  - e.g., maximizing Donsker and Varadhan lower bound.

Simple Iterative KL Minimization

Set initial imputation $\hat{x}^{(0)}$, $t = 1$

Repeat

- $\hat{x}^{(t)} = \arg\min_{\hat{x}} \mathrm{KL}[p_{\hat{x},m} | p_m p_{\hat{x}^{(t-1)}}]$.
- $t \leftarrow t + 1$

Does it Work?

Proposition:

$\mathrm{KL}[p_{\hat{x}^{(t-1)},m} | p_m p_{\hat{x}^{(t-1)}}] \geq \mathrm{KL}[p_{\hat{x}^{(t)},m} | p_m p_{\hat{x}^{(t)}}]$, i.e., The mutual information between $\hat{x}$ and m is non-increasing over iterations.

Proof by Gibbs inequality.

The Optimal Imputer

- At iteration t, what is the optimal imputer $\hat{x}^{(t)}$?
- Recall $\hat{x}^{(t)} = \arg\min_{\hat{x}} \mathrm{KL}[p_{\hat{x},m} | p_m p_{\hat{x}^{(t-1)}}]$.

Proposition:

$\hat{x}^{(t)}$ is the optimal imputer if and only if
$p_{\hat{x}^{(t)}}(x_{1-m}|x_m, m) = p_{\hat{x}^{(t-1)}}(x_{1-m}|x_m)$.

- The optimal imputer gradually removes the influence of m over iterations.
- Finding the optimal imputer is the same as **matching two conditional distributions**.

Constructing The Optimal Imputer

Intuition:

Construct an ODE that "transports" $\hat{x}^{(t-1)}$ to $\hat{x}^{(t)}$.

Mathy:

Formally speaking, the imputer $\hat{x}^{(t)}$ is the solution of an ODE

$$\mathrm{d}z(\tau) = v[z(\tau), \tau]\mathrm{d}\tau$$

at $\tau = 1$ with the initial condition $z(0) = \hat{x}^{(t-1)}$.

Problem: How to set v?

- There exists many $v[z(\tau), \tau]$ that transports $\hat{x}^{(t-1)}$ to $\hat{x}^{(t)}$.

Training a (conditional) Rectified Flow

- One v can be found by solving a least squares:

$$v^* = \arg\min_v \int_\tau \mathbb{E} \left\| \tilde{x}^{(t-1)} - \hat{x}^{(t-1)} - v_\tau \left[x_{1-m}(\tau), \hat{x}_m^{(t-1)}, \hat{x}_m^{(t)} \right] \right\|^2$$

- where $x(\tau) = \tau\tilde{x}^{(t)} + (1-\tau)\hat{x}^{(t-1)}$ and $\tilde{x}$ is an independent copy of $\hat{x}$.

- The above objective is a special type of Rectified Flow (RF).
  - RF learns an ODE that transport samples between two distributions p_{x_0} and p_{x_1} by following the interpolation path $x(\tau) = \tau x_1 + (1-\tau)x_0$.

Training a (conditional) Rectified Flow

v^* is indeed optimal:

Theorem

The above ODE characterized by the velocity field v^*, with initial condition $z(0) = \hat{x}^{(t-1)}$, has a solution $z(1) = \hat{x}^{(t)}$ at $\tau = 1$.

i.e., v^* indeed transport $\hat{x}^{(t-1)}$ to $\hat{x}^{(t)}$.

Algorithm: Mutual Information Reducing Iterations (MIRI)

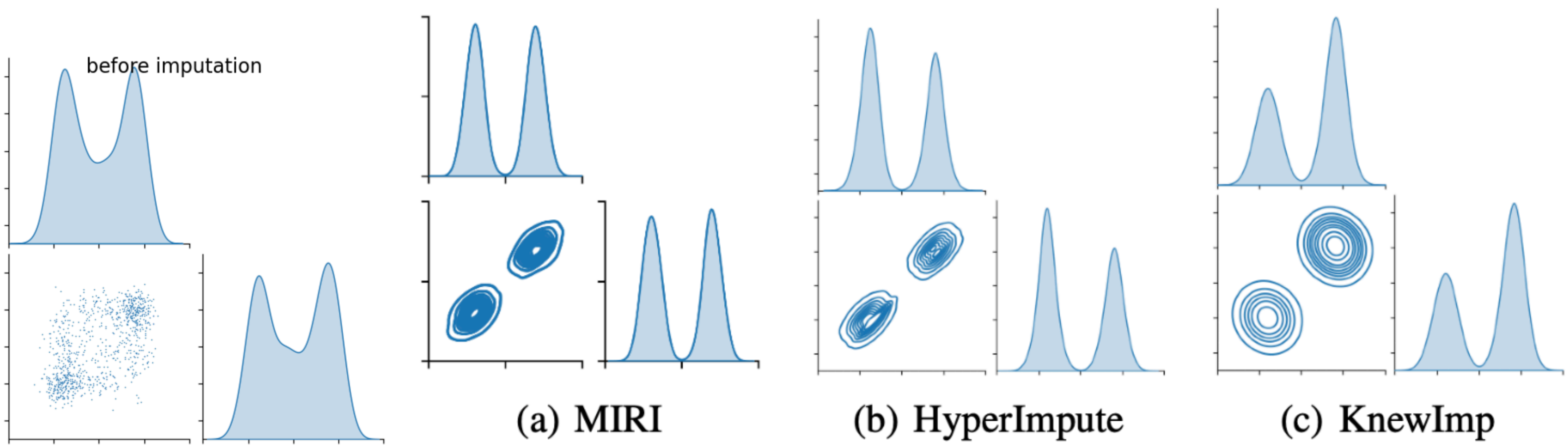
Set initial imputation $\hat{x}^{(0)}$, $t = 1$

Repeat

- Obtain $\tilde{x}^{(t-1)}$ by shuffling $\hat{x}^{(t)}$.
- Train v^* using LS with $(\hat{x}^{(t-1)}, m)$ and $\tilde{x}^{(t-1)}$.
- $\hat{x}^{(t)} = \mathrm{ODESolve}(\mathrm{init} = \hat{x}^{(t-1)}, \mathrm{velocity} = v, \tau = 1)$.
- $t \leftarrow t + 1$

Toy Data

Recovering a bimodal Gaussian mixture with 30% missing data, using 6000 samples to train v.



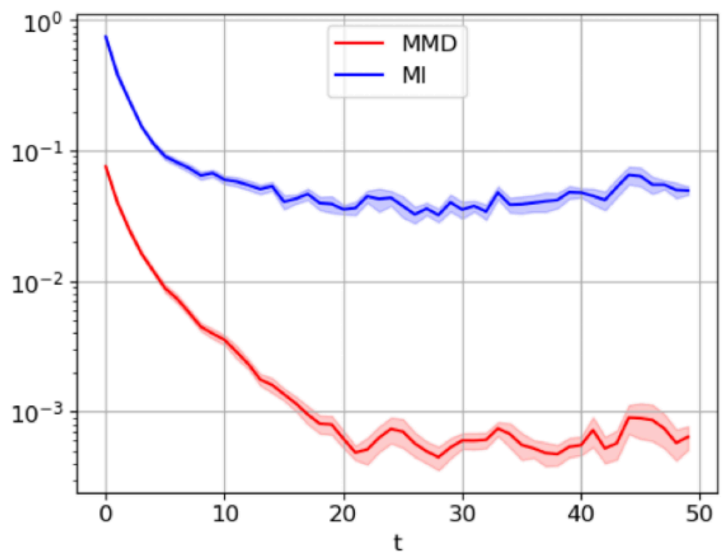
(a) MIRI (b) HyperImpute (c) KnewImp

- MIRI reconstruct each mode accurately.

Toy Data

How well can MIRI reconstruct the ground turth?

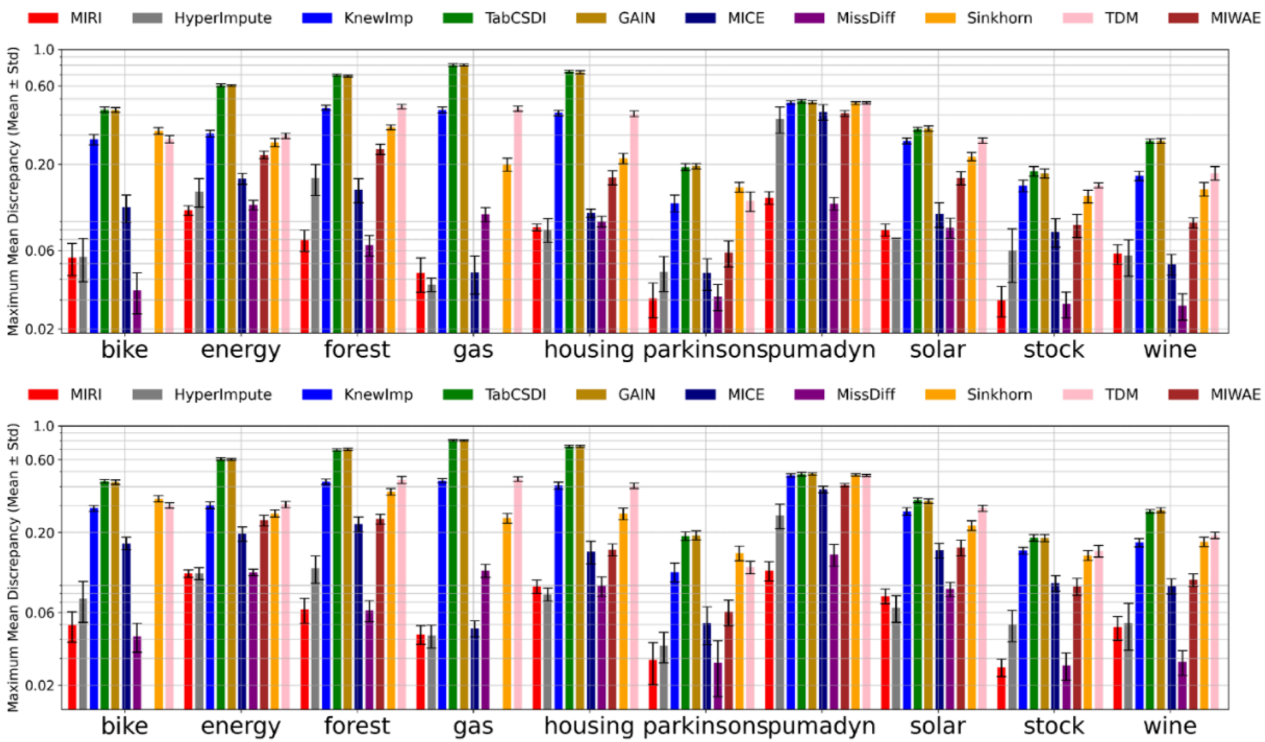
Performance Metric: $\mathrm{MMD}(x^*, \hat{x}^{(t)})$. The lower the better.



$\mathrm{MI}(\hat{x}^{(t)}, m)$ aligns with MMD well, indicating it is a good "loss function" for the missing data imputation.

Tabular Data (UCI Benchmark)

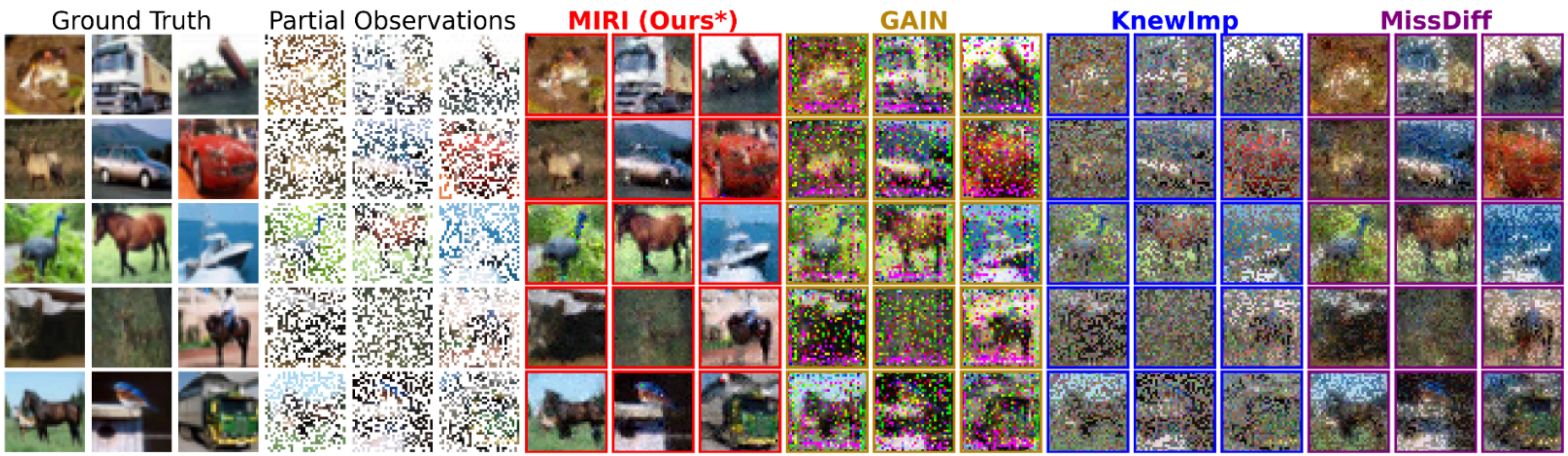
Reconstruct UCI datasets with 60% missing. Performance measured by MMD (the lower the better).



- MIRI is among the best. Round-robin methods are strong.

Image Data (CIFAR10)

32 by 32 images, with 60% randomly missing, using 5000 samples for training v.

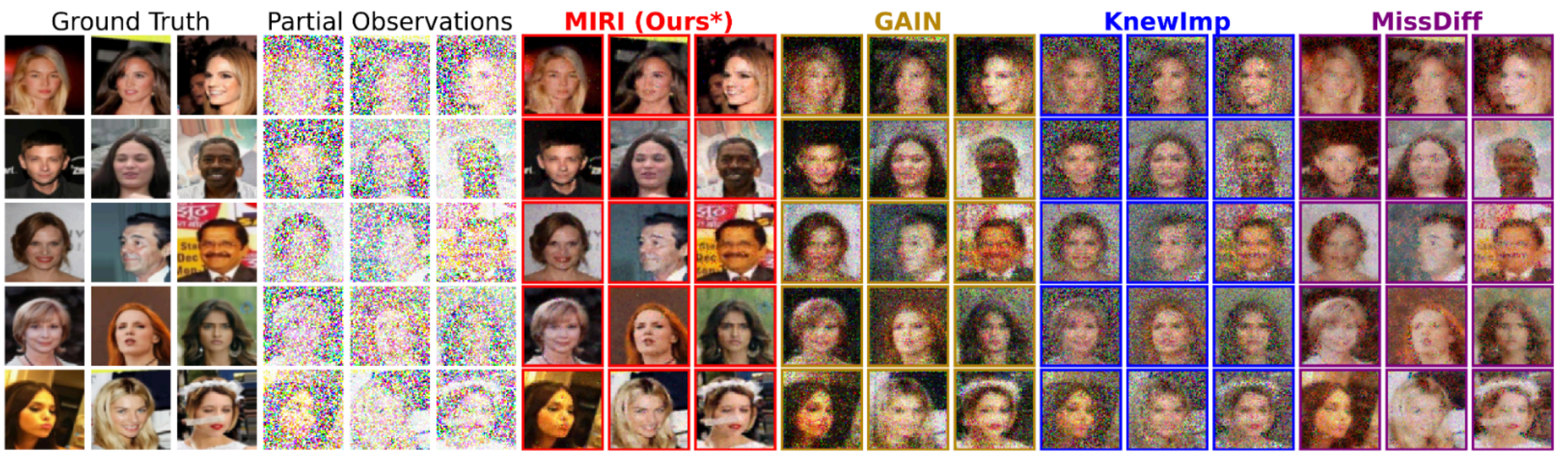


(a) 15 uncurated 32×32 CIFAR-10 images and their imputations. Pixels are removed from *all RGB channels*.

- MIRI has the best visual quality.
- Round-robin methods (e.g., MICE) is too slow.

Image Data (CelebA)

64 by 64 images, with 60% randomly missing, using 5000 samples for training v.



(b) 15 uncurated 64×64 CelebA images and their imputations. Pixels are removed from *each RGB channel independently*.

- MIRI has the best visual quality.

Wait, what about MAR data?

- Recall, MIRI is constructed to enforce the necessary condition of MCAR, i.e., $\hat{x} \perp\!\!\!\perp m$.
- For MAR data, the necessary condition of perfect imputation is $\hat{x}_{1-m} \perp\!\!\!\perp m | \hat{x}_m$, leading to minimizing a conditional MI: $\mathrm{MI}(\hat{x}_{1-m}, m; x_m)$.
- Long story short, the **optimal iterative imputer** that minimizes $\mathrm{MI}(\hat{x}_{1-m}, m; x_m)$ is also:

$$p_{\hat{x}^{(t)}}(x_{1-m} | x_m, m) = p_{\hat{x}^{(t-1)}}(x_{1-m} | x_m).$$

MIRI also works for MAR data.

MAR Tabular Data (UCI Benchmark)

Reconstruct UCI data with 40% and 80% missing MAR.

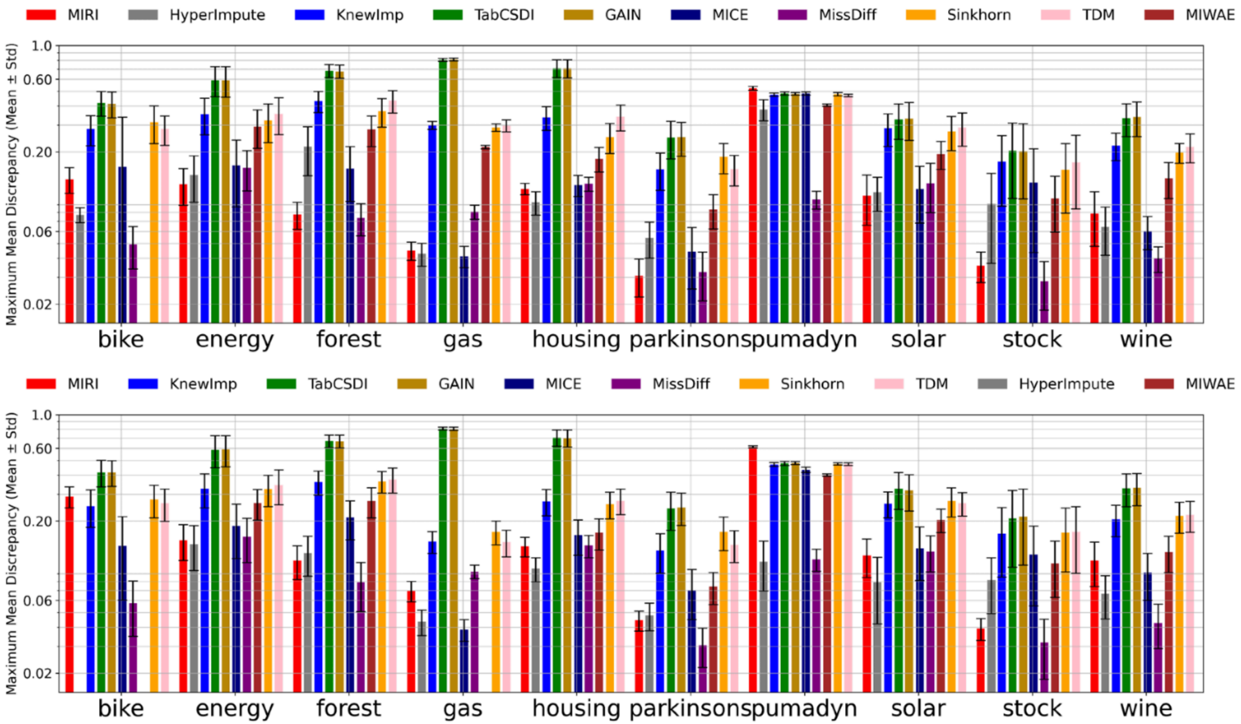


Figure 6: MAR MMD on 10 UCI datasets (Above: 40% missingness, Below: 80 % missingness). The lower the better.

- MIRI performance remains strong.

Conclusion

- MIRI is inspired by GAIN, and is a generative method for missing data imputation.
- It can be seen as enforcing the necessary condition of MCAR and MAR.
  - through MI minimization.
- The exact MI minimization can be carried out by repeated rectified flow, and its imputations are promising when compared with some recent imputers.

References

1. Jinsung Yoon, James Jordon, Mihaela van der Schaar, GAIN: Missing Data Imputation using Generative Adversarial Nets, ICML2018, 2018.

2. Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, Mihaela van der Schaar, HyperImpute: Generalized Iterative Imputation with Automatic Model Selection, ICML2023.

3. Xingchao Liu, Chengyue Gong, Qiang Liu, Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, ICLR 2023.