Pattern Recognition 129 (2022) 108720

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Variational cycle-consistent imputation adversarial networks for general missing patterns

Woojin Lee[a], Sungyoon Lee[b], Junyoung Byun[c], Hoki Kim[c], Jaewook Lee[c,*]

[a] School of AI Convergence, Dongguk University-Seoul, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea
[b] Center for Artificial Intelligence and Natural Sciences, Korea Institute for Advanced Study (KIAS), 85 Hoegiro, Dongdaemun-gu, Seoul 02455, Republic of Korea
[c] Industrial Engineering, Seoul National University, Gwanakro 1, Seoul 08826, Republic of Korea

ARTICLE INFO

Article history:
Received 20 October 2020
Revised 2 September 2021
Accepted 18 April 2022
Available online 20 April 2022

Keywords:
Imputation
Missing data
Cycle-consistent

ABSTRACT

Imputation of missing data is an important but challenging issue because we do not know the underlying distribution of the missing data. Previous imputation models have addressed this problem by assuming specific kinds of missing distributions. However, in practice, the mechanism of the missing data is unknown, so the most general case of missing pattern needs to be considered for successful imputation. In this paper, we present cycle-consistent imputation adversarial networks to discover the underlying distribution of missing patterns closely under some relaxations. Using adversarial training, our model successfully learns the most general case of missing patterns. Therefore our method can be applied to a wide variety of imputation problems. We empirically evaluated the proposed method with numerical and image data. The result shows that our method yields the state-of-the-art performance quantitatively and qualitatively on standard datasets.

1. Introduction

Deep learning society has grown tremendously in recent years, leading to increasing demand for large amounts of data. However, since procuring complete data is rarely feasible in the real world, it is very important to recover at least enough information to analyze from incomplete data [1,2]. A common method to deal with this problem is to build an appropriate algorithm to impute the missing data.

There are three types of missing pattern assumptions used according to the dependency: (1) The data is missing completely at random (MCAR) if its missingness occurs completely randomly regardless of the value of the data. Assuming the missing pattern to be MCAR is a simple approach, but an unrealistically strong assumption in practice. (2) The data is missing at random (MAR) if the missing data depends on observed data but not on unobserved data. (3) The last type of missing pattern is missing not at random (MNAR), where the missing data depends on unobserved data as well as observed data [3]. It is the most general case of missing, since it can encompasses all types of missing patterns. In the real world, since the true missing pattern is unknown, the most general case of missing pattern needs to be considered for successful imputation.

Previous approaches to missing data imputation are based on a sequential equation [4] or the EM algorithm [5]. Recently, deep learning has been widely adopted in missing data imputation [6–11]. Deep learning-based methods have extended their imputation applications to high-dimensional image data [12].

Constructing an missing value imputation model is a different and more challenging problem than standard learning procedures, due to two main reasons. First, there are no complete and incomplete data pairs, i.e., it is impossible to know the true value of missing part of the incomplete data. Therefore it is infeasible to train a model that predicts the missing value because the true value is not recorded in data. Second, it is hard to know the missing pattern of the data. If we can identify the missing pattern of the data, it will be able to generate a synthetic pair of complete and incomplete data.

Learning-based imputation methods can be categorized as either reconstruction or adversarial training. Reconstruction based approach focuses on reconstructing a predicted sample similar to the original data, by generating synthetic missing data from complete data. Adversarial training based approach tries to generate a realistic imputed sample, by using GAN [13] based loss functions.

* Corresponding author. Industrial Engineering, Seoul National University, Gwanakro 1, Seoul 08826, Republic of Korea
E-mail address: jaewook@snu.ac.kr (J. Lee).

https://doi.org/10.1016/j.patcog.2022.108720
0031-3203/

The former can reconstruct the missing values of incomplete data. However this approach has a limitation that it is hard to build a model that can consider the missing pattern, unless we know the true distribution of missing. The latter approach can be applied to a wide variety of unknown missing distributions. However, since this approach does not predict the missing value by training, it may not show robust performance under high missing rate.

To overcome such difficulties, in this paper, we suggest a method that combines two different imputation approaches. We assume that we have access for the complete data distribution and the suggested algorithm can be optimized through deep learning framework. Under this assumptions, we suggest a novel imputation method that identifies the missing pattern of the data by using adversarial training approach and builds an imputation model by using reconstruction approach.

Our proposed method consists of two simultaneously trained generators. One generator creates synthetic missing samples indistinguishable from real incomplete data. At the same time, the other reconstructs full data from synthetic missing samples similar to the corresponding original data. In this way, our suggested model can handle various imputation problems without any assumptions on the missing patterns.

We demonstrated the theoretical basis of our proposed method and empirically evaluated it on several numerical and image datasets. From the imputation results of the numerical datasets, we showed that our method outperforms previous methods in various missing patterns and datasets. Through experiments on image datasets, we visualize that our method successfully captures the characteristics of underlying missing patterns from incomplete dataset, and successfully reconstructs the missing parts of the data. Although a theoretical basis for convergence of our proposed method is not given, we have provided empirical results on synthetic Gaussian examples that quantitatively shows our method can estimate the ground-truth missing pattern.

Our main contributions are as follows:

1) The proposed model can learn all types of data missing patterns without having any assumptions on the missing patterns. It makes our model applicable to a wide variety of imputation problems.
2) We provide theoretical backgrounds of our suggested method based on variational approach. From the perspective of variational inference, we demonstrate that our method approximates the joint distribution between incomplete data and complete data, and it can learn the underlying missing and imputation processes, when we have access to the complete data and the model converges.
3) Our model outperforms state-of-the-art models by introducing the framework of cross-domain translation. We quantitatively demonstrate the superior performance of our method on numerical datasets. In addition, we qualitatively visualized that our method successfully learns the missing patterns and imputes the missing values.

2. Literature review

2.1. Missing cases and imputation

We use the following notations to describe the missing patterns. X denotes a random variable for the data in the d-dimensional space $\mathcal{X}$, and M is a random variable for a binary mask in $\{0,1\}^d$ representing observed entries as 1 and missing entries as 0. We then denote a random variable X^{obs} as $X^{obs} = f_M(X) \equiv (X \odot M, 1-M) \in \mathcal{X}^* \equiv \mathcal{X} \times \{0,1\}^d$, where $f_M(X)$ is a masking function that drops the values of X in the locations of missing parts denoted by $1-M$, and $\odot$ is an elementwise multiplication. We denote x and x^{obs} as realizations of the variable X and X^{obs} respectively.

We represent the data distribution as $X \sim q^*(X)$ and the imputation process as $X \sim p(X|X^{obs})$. Especially, we denote the missing process as $M \sim p(M|X)$ (or equivalently, $X^{obs} \sim p(X^{obs}|X)$) since M may depend on the data X. We also denote the data distribution of incomplete data as $X^{obs} \sim p^*(X^{obs})$. Now we introduce some useful assumptions on the missing patterns.

The missing pattern is said to be MCAR if the probability of being missing does not depend on the data. This implies that the reasons of the missing are not related to the data. It can be expressed as

$$p(M|X) = p(M). \tag{1}$$

Assuming MCAR in imputation problems is convenient, but this setting is often unrealistic for the data at hand.

The missing pattern is said to be MAR if the missing pattern may depend on observed information. It can be expressed as

$$p(M|X) = p(M|X^{obs}). \tag{2}$$

It is more general and realistic than MCAR, but still restricted to specific cases.

Finally, the missing pattern is MNAR if the probability to be missing depends on information of the data, including the missing values itself. It is the most general case that can consider all of the missing types, but the most challenging one.

In real world application of imputation methods, it is impossible to know the underlying distribution of the missing pattern what kinds of missing patterns they belongs to. Therefore it is reasonable to consider the most general missing pattern $p(M|X)$ without any assumption for solving the missing problem.

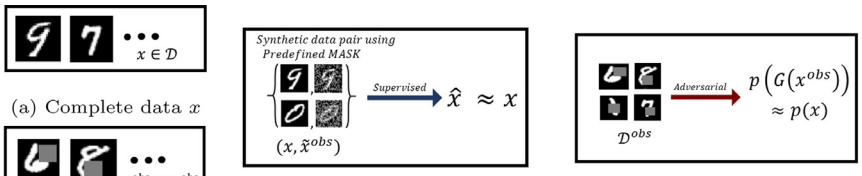
2.2. Imputation methods

In many scenarios, two different kinds of data are prepared to address the problem of missing data imputation: complete data $\mathcal{D}$ (Fig. 1a) and incomplete data $\mathcal{D}^{obs}$ (Fig. 1b). Until recently, most work on imputation methods exploited only one of the two to construct a model.

First, a reconstruction based approach uses complete data $\mathcal{D}$ for training. As illustrated in Fig. 1c, these models generate the synthetic training data pair $(x, \tilde{x}^{obs})$ by using all $x \in \mathcal{D}$, where $\tilde{x}^{obs}$ denote generated synthetic incomplete data that corresponds to x. Then they build a model $G(x^{obs})$ which can reconstruct data $\hat{x} = G(\tilde{x}^{obs})$, this is similar to the original data x, by minimizing $\mathcal{L}_{recon}(x, \hat{x})$. The advantage of this approach is that it works consistently well when the missing rate in the test stage is even high.

The performance of the reconstruction based approach depends on how similar the synthetic training pair is to the real ones. If the synthetic incomplete data deviates from the real incomplete data, the imputation model may not be well generalized to the test data. However, it is hard to generate synthetic incomplete data $\tilde{x}^{obs}$ unless we know the true distribution of missing pattern $p(M|X)$. Previous methods address this problem by using a predefined mask generator, but a generated mask may not reflect the true underlying missing distribution. MIDA [9], a Denoising Autoencodoer-based model, uses a drop-out approach to build synthetic data pairs $(x, \tilde{x}^{obs})$. VAEAC [14] also belongs to reconstruction based approach, which applies the idea of a Variational Autoencoder [15] to missing data imputation. It uses a drop-out mask and a rectangular mask in training.

An adversarial training-based approach uses incomplete data $\mathcal{D}^{obs}$. Given $x^{obs} \in \mathcal{D}^{obs}$, rather than recovering $\hat{x}$ similar to the corresponding x (which is not available), this approach tries to generate the imputed data $G(x^{obs})$, that seems to be sampled from the data distribution, encouraging $G(x^{obs})$ to simulate $p(X|X^{obs} = x^{obs})$.



(a) Complete data x

(b) Incomplete data x^{obs}

(c) Reconstruction based approach

(d) Adversarial training based approach

Fig. 1. 1a MNIST examples of complete data $x \in \mathcal{D}$, 1b: Incomplete data $x^{obs} \in \mathcal{D}^{obs}$, 1c: Reconstruction approach of imputation, 1d: Adversarial approach of imputation.

Table 1
Required assumptions on the missing pattern for previous imputation methods.

	Required assumption on the missing pattern	Reconstruction
MIDA	Missing pattern is known	○
MICE	Only under MCAR and MAR	×
Amelia	Only under MCAR and MAR	×
Missforest	Only under MCAR and MAR	×
GAIN	None	×
Ours	None	○

The process of an adversarial training-based method is illustrated in Fig. 1d. This approach is easier to apply because it does not require complete data, and can also be applied to various kinds of unknown missing data patterns. However, since the training of this approach heavily relies on GAN based loss functions, it does not show consistent imputation performance.

GAIN [10] has taken the approach of using an element-wise discriminator to generate imputed data indistinguishable from the observed part of the sample. Unfortunately, since GAIN does not use the information of true distribution $p(X^{obs})$, it does not show robust performance with high missing rate. Recent research in MisGAN [12] and HI-VAE [16] propose to use GAN and VAE-based framework respectively, when there only exists incomplete data. These studies also suggested an architecture to complete missing data by using a data generator which represents the true data distribution. Still, it may fail to reflect the true conditional distribution $p(X|X^{obs})$.

The limitations of reconstruction and adversarial training based approaches arise from their use of only one domain of data. Because complete observed data set $\mathcal{D}$ and incomplete observed set $\mathcal{D}^{obs}$ are unpaired, MIDA and VAEAC chose to use the complete dataset, while GAIN and MisGAN chose the incomplete dataset.

Both approaches have critical problems when the assumption doesn't meet the true missing patterns. Nevertheless, many earlier studies, including above papers, didn't consider the case where there are relationships between the missing pattern and the data, such as MNAR or MAR. Even in the fields of image inpainting, they usually assumed missing pattern occur randomly or independently from the original images as well [17–19].

From the perspective of missing patterns, we summarized the types of missing patterns assumed by previous methods in the Table 1. In the case of MIDA [9], it can only consider the missing pattern that is predefined by the user. Therefore this method is effective if the user is well aware of the true missing pattern, which is unrealistic. GAIN [10] considers the missing pattern which was occurred in the dataset. Therefore, this method can handle general missing patterns same as ours. However, since GAIN does not generate the synthetic missing patterns, it can not be trained to estimate the missing value. Therefore, only the discriminator is responsible for whether the missing value is properly estimated in the training process, so even though it can theoretically handle general missing patterns, it does not show a competitive imputation performance. Traditional multiple imputation approaches [4,5,20] consider only the cases of MCAR and MAR, that ignored the case when the missing pattern can be conditional to the missing part of the data.

The reason why previous methods could not consider the most general cases in missing is because they could not access to the value of missing data x^{miss}. Therefore it was hard to identify the conditional relationship between the data x and the m, and to build an imputation model that can consider this relationship. In this paper, we solve this problem by modeling the missing pattern $p(M|X)$ by using GAN-based data-to-data translation framework. In addition, by using the parametrized missing generator, we solve the unpaired problem in missing data. In this way we can build an imputation model that does not assume any specific missing patterns.

Recently there have been several approaches using GAN for image-to-image translation [21–23]. CycleGAN [21] efficiently learns a generator so that the distribution of $G(X)$ is indistinguishable from the distribution of Y. The model couples it with an inverse mapping $F(Y)$ by introducing the concept of cycle-consistency to enforce $F(G(X)) \approx X$.

3. Method

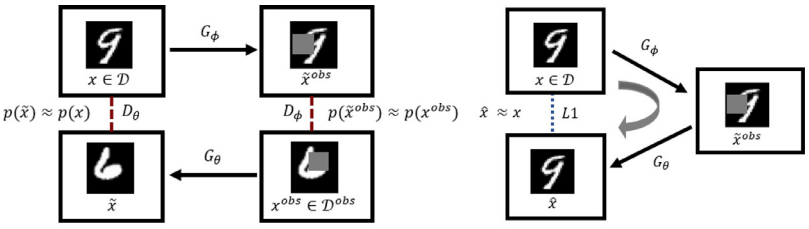
3.1. Formulations

Given observed data x^{obs}, our goal is to model the conditional distribution of corresponding imputed data $p(X|X^{obs})$ and to generate desired imputed samples $\tilde{x}$ from $p(X|X^{obs} = x^{obs})$. In general, to learn the conditional distribution $p(X|Y)$ by a conditional generative model framework, a set of pairs (x, y) is required. However, in the scenario of imputation, we are only given observed data x^{obs} for training, without access to the complete data x corresponding to x^{obs}.

In this paper, to solve a wider range of problems and bring some potential benefits, we put relaxation on this imputation problem by assuming that we have access to incomplete data x^{obs} and the fully complete data x. To explain in more detail, we assume that a pair of datasets, complete dataset $\mathcal{D} \subset \mathcal{X}$ and incomplete dataset $\mathcal{D}^{obs} \subset \mathcal{X}^*$, is available, while each pair of data (x, x^{obs}) is not available.

Through adopting this relaxation, we can potentially bring some advantages to the imputation process by accessing the full data distribution. Considering that the typical missingness problem formulation is purely based on incomplete data, it is an open problem for imputing the missing values and estimating the missingness mechanism especially when it is MNAR. However, if we use a complete dataset $\mathcal{D}$ during an imputation process, we can expect that the model has better explanatory power by observing the general distribution of the original data x than that of the setting not using a complete dataset.

Note that, for some cases, it might be unable to gain a complete dataset. For example, the case when a measuring instrument does not work for above a certain value. However, considering that missing patterns are usually stochastic, we believe that relaxation is a weak assumption in the real-world problem.



(a) Adversarial loss (b) Cycle-consistency loss

Fig. 2. The proposed method exploits two loss function. 2a: Adversarial loss ensure that each generated image $G_\phi(x)$ and $G_\theta(x^{obs})$ is indistinguishable from respective domain $\mathcal{X}$ and $\mathcal{X}^*$, 2b: Cycle-consistency loss ensure the reconstructed image $\hat{x}$ similar to the original image x.

Under this setting of an unpaired data-to-data translation problem, we propose to exploit adversarial loss and cycle-consistency loss to learn a mapping between the two domains $\mathcal{X}$ and $\mathcal{X}^*$.

3.2. Proposed method

Our goal is to learn the underlying missing and imputation process with neural networks. We model the missing procedure as G_ϕ with the missing process parameter ϕ and the imputation process as G_θ with the imputation process parameter θ. Since it requires G_ϕ and G_θ to be stochastic sampling networks for multiple imputation, we concatenate the input x or x^{obs} with a random noise $\epsilon \sim p(\epsilon)$. For example, to get the imputed sample from a given x^{obs}, we can randomly draw a noise vector ϵ and compute $G_\phi(x, \epsilon)$. To generate samples indistinguishable from true data, we use adversarial training framework for each process.

As shown in Fig. 2a, we train D_ϕ and G_ϕ to model the missing mechanism $q_\phi(X^{obs}|X)$ by the generator $G_\phi(X, \epsilon)$. Simultaneously, we model the imputation process $p_\theta(X|X^{obs})$ by learning D_θ and G_θ. The objectives for the variational GAN minimax problem are defined as follows:

$$V_{miss}(D_\phi, G_\phi) = \mathbb{E}_{q^*(X)p^*(X^{obs})}[\ln(D_\phi(X^{obs}; X))] + \mathbb{E}_{q^*(X)p(\epsilon)}[\ln(1 - D_\phi(G_\phi(X, \epsilon); X))] \quad (3)$$

$$V_{impute}(D_\theta, G_\theta) = \mathbb{E}_{p^*(X^{obs})q^*(X)}[\ln D_\theta(X; X^{obs})] + \mathbb{E}_{p^*(X^{obs})p(\varepsilon))}[\ln(1 - D_\theta(G_\theta(X^{obs}, \varepsilon); X^{obs}))] \quad (4)$$

In addition, we combine this objective with the cycle-consistency loss in Eq. (5) to encourage $\hat{x} = G_\theta(G_\phi(X, \epsilon), \varepsilon) \approx x$ as represented in Fig. 2b.

$$V_{cyc}(G_\phi, G_\theta) = \mathbb{E}_{q^*(X)p(\epsilon)}[\|G_\theta(G_\phi(X, \epsilon), \varepsilon) - X\|_2^2] \quad (5)$$

The reason for not using full cycle loss is that the other side of cycle loss forces the missing generator to be deterministic but we wanted it to be more stochastic. For more details of full cycle loss, please refer to Section 5.3.

Finally, our total loss combining (3)–(5) can be obtained as follows:

$$\min_{G=(G_\phi, G_\theta)} \max_{D=(D_\phi, D_\theta)} V(D, G) = V_{miss}(D_\phi, G_\phi) + \lambda_1 V_{impute}(D_\theta, G_\theta) + \lambda_2 V_{cyc}(G_\phi, G_\theta) \quad (6)$$

where λ_1 and λ_2 control the importance of each objective. In the experiments, we use $\lambda_1 = 1$ and $\lambda_2 = 10$ for numerical dataset and $\lambda_1 = 1$ and $\lambda_2 = 100$ for image dataset. They were determined experimentally on a logarithmic scale, respectively.

The algorithm for our proposed method is in Algorithm 1. We have denoted the random variable X^{obs} that contains the information of the binary mask M that represents the observed entries as 1 and missing entries as 0. For the empirical experiment, we used value of the incomplete data X^{obs} as well as the mask M as an input of the imputation generator G_θ.

Algorithm 1: Proposed method.

Input : incomplete data $(X_k^{obs})_{k=1}^N$ and complete data $(X_k)_{k=1}^N$.
Output: imputation generator and discriminator G_θ, D_θ, missing generator and discriminator G_ϕ, D_ϕ.

for $1, \ldots,$*#epochs* **do**
- *Sample mini-batches of m examples from incomplete data* $B^{obs} = \{X_1^{obs}, \ldots, X_m^{obs}\}$ *and complete data* $B = \{X_1, \ldots, X_m\}$
- *Sample m noise samples* $\{\epsilon_1, \ldots, \epsilon_m\}$ *from* $\mathcal{N}(0, I)$
- ***GAN loss with*** X^{obs}
  - $V_{miss} = \frac{1}{m}\sum_{i=1}^m [\ln(D_\phi(X_i^{obs})) + \ln(1 - D_\phi(\tilde{X}_i^{obs}))]$
  - *where* $\tilde{X}_i^{obs} = G_\phi(X_i, \epsilon_i)$
- ***GAN loss with*** X
  - $V_{impute} = \frac{1}{m}\sum_{i=1}^m [\ln(D_\theta(X_i)) + \ln(1 - D_\theta(\tilde{X}_i))]$
  - *where* $\tilde{X}_i = G_\theta(X_i^{obs}, \epsilon_i)$
- ***Cycle-consistency loss***
  - $V_{cyc} = \frac{1}{m}\sum_{i=1}^m [\|\hat{X}_i - X_i\|_2^2]$
  - *where* $\hat{X}_i = G_\theta(\tilde{X}_i^{obs}, \epsilon_i)$
- $V = V_{miss} + \alpha V_{impute} + \beta V_{cyc}$
- $\theta, \phi = Adam(\theta, \phi, \nabla_\theta V, \nabla_\phi V, V)$

end

4. Theoretical results

In this section, we establish theoretical results for our method based on the variational approach. First, from the perspective of variational inference, we demonstrate that our method approximates the joint distribution between the incomplete data and the complete data. We assume that we have access to the complete data and the algorithm can be optimized through adversarial learning and cycle-consistency loss. Under this assumption, we argue that through optimizing the proposed objective in Eq. (6), we can attain G_ϕ and G_θ modeling the real missing and imputation processes, which are represented as $q(X^{obs}|X)$ and $p(X|X^{obs})$.

Now, we show that our method approximates the joint distribution between the incomplete data and the complete data by using variational inference. Suppose we have complete data $X \sim q^*(X)$ and incomplete data $X^{obs} \sim p^*(X^{obs})$.

We assume that X^{obs} is governed by its underlying hidden complete variable X, which is drawn from a prior $q(x)$ and related to x through the likelihood $q_\phi(x^{obs}|x)$. Accordingly, the joint density of x and x^{obs} is given by

$$q_\phi(x, x^{obs}) = q_\phi(x^{obs}|x)q^*(x) \quad (7)$$

We specify the likelihood through a mapping G_ϕ that takes as input random noise ϵ and complete data x,

$$x^{obs} \sim q_\phi(x^{obs}|x)$$
$$x^{obs} = G_\phi(x, \epsilon), \quad \epsilon \sim p(\epsilon) \quad (8)$$

Here, for simplicity of notation, we assume that x^{obs} contains the information of the mask $\tilde{m}$, where x^{obs} can be expressed as $x \odot \tilde{m}$, similar to random variable X^{obs}. Then we define the variational approximation to the joint distribution as

$$p_\theta(x, x^{obs}) = p_\theta(x|x^{obs})p^*(x^{obs}). \tag{9}$$

We specify the likelihood through an inverse mapping G_θ that takes as input random noise ε and observable incomplete data x^{obs},

$$x \sim p_\theta(x|x^{obs}) \tag{10}$$

$$x = G_\theta(x^{obs}; \varepsilon), \quad \varepsilon \sim p(\varepsilon). \tag{11}$$

Recall that $q^*(x)$ and $p^*(x^{obs})$ denote the empirical distribution with complete data and the incomplete data.

First, we consider directly approximating the joint distribution $q_\phi(x, x^{obs})$ through a variational joint $p_\theta(x, x^{obs})$ closest in KL divergence (equivalent to maximum likelihood estimator when q_ϕ is a true joint distribution) as

$$\begin{aligned}
&\mathrm{KL}[q_\phi(x, x^{obs})||p_\theta(x, x^{obs})] \\
&= \mathbb{E}_{q_\phi(x,x^{obs})}[\ln q_\phi(x, x^{obs}) - \ln p_\theta(x, x^{obs})] \\
&= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[-\ln p_\theta(x|x^{obs})] \\
&\quad + \mathbb{E}_{q^*}\mathrm{KL}[q_\phi(x^{obs}|x)||p^*(x^{obs})] - \mathbb{E}_{q^*}[-\ln q^*(x)],
\end{aligned} \tag{12}$$

where the last term $\mathbb{E}_{q^*}[-\ln q^*(x)]$ is the entropy of $q^*(x)$ and a constant w.r.t. parameters θ and ϕ.

The first term of the Eq. (12) is the negative expected log posterior (NELP), defined as

$$\begin{aligned}
\mathcal{L}_{NELP}(\theta, \phi) &= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[-\ln p_\theta(x|x^{obs})] \\
&= \mathbb{E}_{q^*(x)p(\epsilon)}[-\ln p_\theta(x|G_\phi(x, \epsilon))]
\end{aligned} \tag{13}$$

where the last equality comes from reparameterization trick as in Kingma and Welling [15].

We demonstrate that minimizing the cycle-consistency loss in Eq. (5) corresponds to minimizing the NELP 13.

Theorem 1. *If* $p_\theta(x|x^{obs}) = \mathcal{N}(x|G_\theta(x^{obs}), \frac{1}{2\gamma}I)$, *then* $\min_{G_\phi, G_\theta} V_{cyc}(G_\phi, G_\theta)$ *reduces to* $\min \mathcal{L}_{NELP}(\theta, \phi)$.

The second term in Eq. (12) is the KL divergence between $q_\phi(x^{obs}|x)$ and the empirical density p^*.

Theorem 2. *Minimizing* $\mathbb{E}_{q^*}\mathrm{KL}[q_\phi(x^{obs}|x)||p^*(x^{obs})]$ *over* ϕ *can be achieved by solving*

$$\min_{G_\phi} \max_{D_\phi} V_{missing}(D_\phi, G_\phi)$$

Theorems 1 and 2 implies that

$$\min_{G=(G_\phi, G_\theta)} \max_{D=(D_\phi, D_\theta)} V_{miss}(D_\phi, G_\phi) + \gamma V_{cyc}(G_\phi, G_\theta) \tag{14}$$

reduces to $\min_{\phi,\theta} \mathrm{KL}[q_\phi(x, x^{obs})||p_\theta(x, x^{obs})]$ and $\mathbb{E}_{q^*}\mathrm{KL}[q_\phi(x^{obs}|x)||p^*(x^{obs})]$, thereby approximating the joint distribution $q_\phi(x, x^{obs})$ through a variational joint $p_\theta(x, x^{obs})$ as well as matching the conditional distributions for missing mechanism. However, it only models a possible translation between two domains through joint distribution, not necessarily the true conditional distributions for imputing mechanism. Therefore, it can mislead us to learn generators matching any pair in the two domains, failing to reflect the target conditional relations for imputation (to match $p_\theta(x|x^{obs})$ to $q^*(x)$ for each x^{obs}).

To this end, we next plan to approximate $q^*(x)$ by seeking a variational conditional $p_\theta(x|x^{obs})$ closest in the mean KL divergence with respect to x^{obs} as follows

$$\mathbb{E}_{p^*(x^{obs})}\mathrm{KL}[\ln p_\theta(x|x^{obs})||q^*(x)] \tag{15}$$

Table 2
Description of numerical datasets.

Dataset	# of attributes	# of samples
Boston Housing	12	506
Glass	9	215
Satellite	36	500
Shuttle	9	500
Soybean	60	208
Vehicle	18	846
Vowel	9	991

Table 3
Network architectures for Numerical datasets.

Model	Structure
$G_\phi = G_{miss}$	FC($2 \times Dim$)-FC($2 \times Dim$)-FC($1 \times Dim$)
$G_\theta = G_{impute}$	FC($4 \times Dim$)-FC($2 \times Dim$)-FC($1 \times Dim$)
$D_\theta = D_{miss}$	FC($2 \times Dim$)-FC($1 \times Dim$)-FC(1)
$D_\theta = D_{impute}$	FC($2 \times Dim$)-FC($1 \times Dim$)-FC(1)

Theorem 3. *Minimizing* $\mathbb{E}_{p^*(x^{obs})}\mathrm{KL}[\ln p_\theta(x|x^{obs})||q^*(x)]$ *over* θ *can be achieved by solving*

$$\min_{G_\theta} \max_{D_\theta} V_{impute}(D_\theta, G_\theta)$$

Hence, the proposed method provides an imputation generator G_θ^* that can achieve the best imputation results if it can learn the missing mechanism $p(X^{obs}|X)$ using the missing generator G_ϕ^*.

5. Experiments

First, we tested our suggested method on numerical datasets. We evaluated missing imputation performance on various UCI datasets [24] and compared it with recent state-of-the-art methods.

However, evaluation on the numerical datasets has some limitations. Through experiments on numerical datasets, it is not easy to check our method successfully learned the missing patterns at a glance.

Considering the above limitations, we continue by extending our experiment to image datasets. We observed how well our method imitates the mask for various kinds of missing patterns. For image datasets, we used the MNIST [25] and Fashion-MNIST datasets [26] to visualize imputation results and measure the prediction performance.

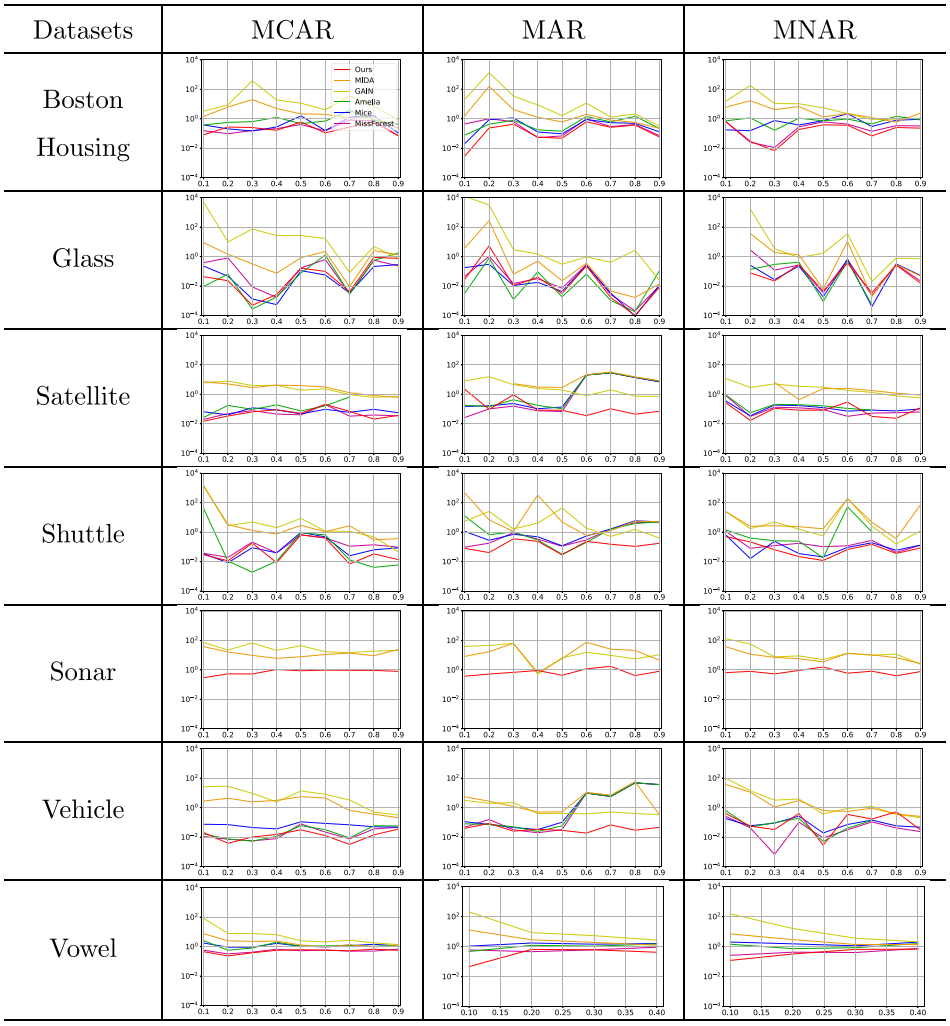
5.1. Numerical datasets

5.1.1. Settings

Data. We evaluated our method on datasets from the UCI repository [24] (Boston Housing, Glass, Satellite, Shuttle, Soybean, Vehicle, and Vowel). We removed categorical and ordinal variables and used only numerical variables for missing data imputation. We randomly split the data into training set (70%), validation set (10%), and test set (20%). After splitting, we applied various missing patterns (MCAR,MAR, and MNAR) to each of the sets. Then we used training set to build a model, and conducted hyper-parameter tuning by using X^{obs} in validation set, finally evaluated on X^{obs} in test set. By mitigating the probability of missing, we experimented on various missingness proportion from 10% to 90% (Table 2).

Network architecture. In numerical dataset experiments, we used Fully Connected(FC) layer structures. Generator G_ϕ, G_θ and discriminator D_ϕ, D_θ are composed of three FC layers. For all layers, ReLU is used as activation function, except following cases. Detailed structures are in Table 3.

Table 4
Imputation results in numerical data in *Random* setting, differing missingness proportion from 10% to 90%, in terms of MSE. The y-axis in the figure represents log scale of regularized MSE value of each method divided by MSE of mean imputation (MSE_{method}/MSE_{Mean}). Methods are illustrated in different color; MIDA : Orange, GAIN : Yellow, Amelia : Green, MICE : Blue, Missforest : Purple, and Ours : Red.



Missing pattern. We used three kinds of missing data generation processes for numerical data experiment: MCAR, MAR, and MNAR. For MCAR case, we randomly erased a% of the data as 0, where a is proportion of missingness from 10% to 90%. Meanwhile, for MAR and MNAR case, we followed the method in Gondara and Wang [9].

First, we randomly sampled two attributes x_1 and x_2 from the dataset and calculated their medians. We used two types of missing attribute selection.

- *Random setting*: We randomly select the half of the attributes that can be missing.
- *Uniform setting*: We select all of the attributes that can be missing.

Then we append a value a to all observations, and for each observation set the values of selected the attributes as 0 where $(x_1 \leq 1.5 \times median(x_1)$ or $x_2 \geq 0.5 \times median(x_2))$. The difference between MAR and MNAR is that for MAR the values x_1 and x_2 does not be missed while in MNAR cases, the values can be missed.

Additionally, we experimented MAR and MNAR settings by using three attributes (x_1, x_2, and x_2) to demonstrate the imputation performance of our method in more complicates cases. We refer this settings as MAR^3 and MNAR^3. We tested this settings on datasets that have more than 10 attributes (Boston Housing, Satellite, Sonar, and Vehicle).

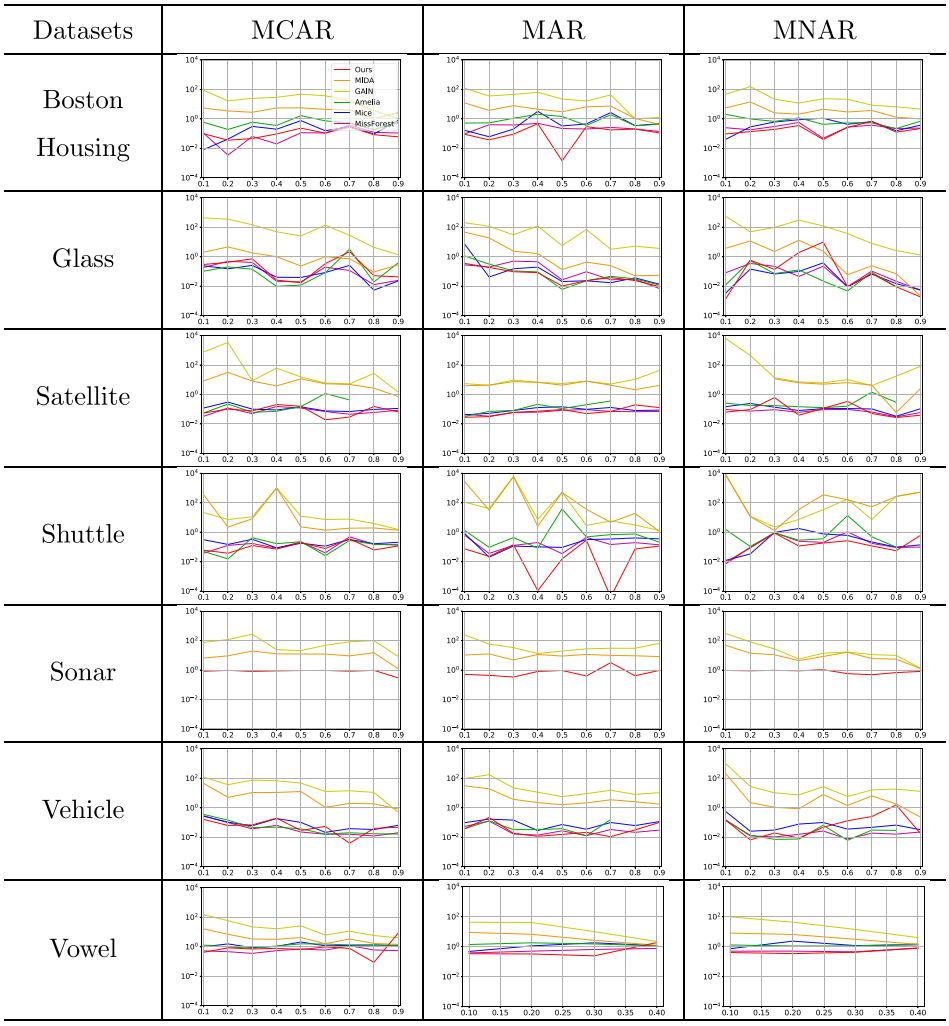
Evaluation. We used Mean Squared Error (MSE) and sum of Root Mean Squared Error (RMSE) as the evaluation metrics of this experiment. Since MSE measures the squared error of each column, the performance of MSE can depend on variables of high value and variance. Therefore, we also used RMSE to compensate for weaknesses of MSE. They are defined as follows :

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{x}_i - x_i)^2\right), \tag{16}$$

$$RMSE = \sum_{i=1}^{m}\sqrt{\mathbb{E}\left(\sum_{i=1}^{n}(\hat{x}_i - x_i)^2\right)}. \tag{17}$$

Comparison We evaluated the imputation performance of our method, comparing it with six different imputation models: Amelia [5], MICE [4], MIDA [9], GAIN [10], Missforest [20] and mean imputation. For Amelia, MICE, and Missforest, we used the R-package provided by the authors. We also used source code provided by

Table 5
Imputation results in numerical data in *Uniform* setting, differing missingness proportion from 10% to 90%, in terms of MSE. The *y*-axis in the figure represents log scale of regularized MSE value of each method divided by MSE of mean imputation (MSE_{method}/MSE_{Mean}).



the authors in MIDA and GAIN. We also compared our method with mean imputation, the most popular missing data imputation method. In all settings we focused on imputing the missing part of the $\mathcal{D}^{obs}$ by using the knowledge of the complete dataset $\mathcal{D}$ and the observed part of $\mathcal{D}^{obs}$.

5.1.2. Imputation performance

The results of numerical experiments are summarized in Tables 4–7. Tables 4 and 5 show imputation results in RMSE, while Tables 6 and 7 show results in terms of RMSE. In each figures, *x*-axis represents missingness proportion while *y*-axis represents log scale of regularized MSE or RMSE which is divided by MSE or RMSE results of Mean imputation. In every figures, we can find that our method (Red) shows robust performance compared to other methods (the lower the better).

Fig. 3 illustrates the Histogram of the number of experiments with the best performance for each method. For MCAR imputation tasks (Fig. 3a), our methods showed best results in 132 tasks out of 250 tasks. In MAR and MNAR tasks (Fig. 3b, our method was best in 129 tasks and 154 tasks, respectively. This results show that our method shows superior results in all three missing imputation problems. In addition, since our method can handle the imputation problems when missing pattern is conditional to the data (MAR and MNAR), it performed much better in those two tasks. In all three missing patterns tasks, Missforest, Amelia, and MICE also showed competitive results. This can be interpreted that models that assumed a complex missing pattern showed better performance.

Among 712 experimental results, our method shows best performance in 415 cases, while Missforest and Amelia show 164 and 72, respectively. We can find that our suggested model outperformed previous models in various missing experiments.

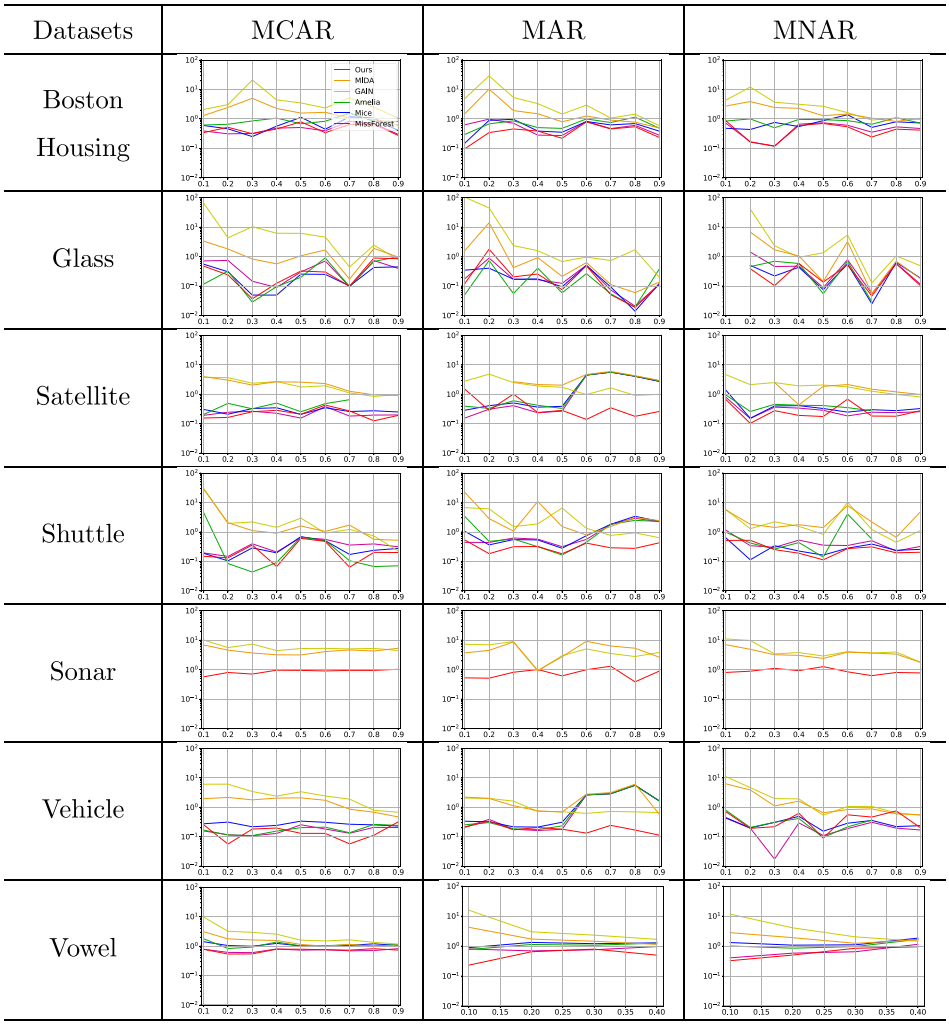
Imputation results on MAR^3 and $MNAR^3$ are summarized in Tables 8 and 9. For MAR^3 imputation tasks, our method showed best results in 78 tasks our of 144 tasks, while Missforest and Amelia showed 44 and 13, respectively. In $MNAR^3$ tasks, our method was best in 89 tasks. This results show that our method demonstrates superior performance in complicated missing tasks.

5.2. Image datasets

5.2.1. Settings

Data. MNIST is a dataset consist of 60,000 training examples. In the experiment, we divided the entire training set into halves, and used them as $\mathcal{D}$ and $\mathcal{D}^{obs}$, respectively. In the case of DATA-DEPENDENT missing evaluation, we only used samples with a la-

Table 6
Imputation results in numerical data in *Random* setting. differing missingness proportion from 10% to 90%, in terms of RMSE. The *y*-axis in the figure represents log scale of regularized RMSE value of each method divided by RMSE of mean imputation ($RMSE_{method}/RMSE_{Mean}$).



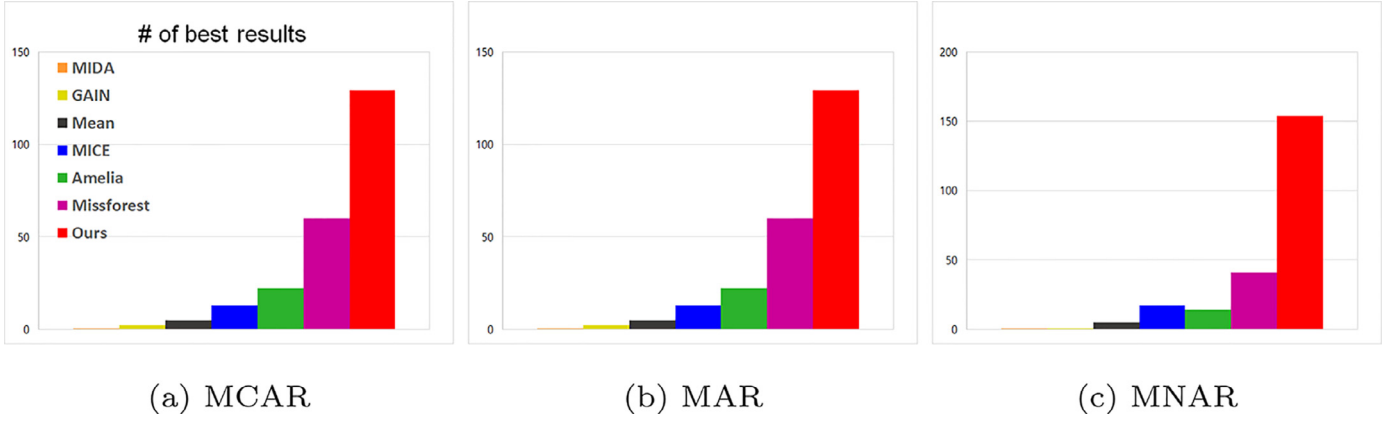


Fig. 3. Histogram of the number of experiments with the best performance for each methode.

bel of 0, 1, and 2. For Fashion-MNIST, we use the same setting as MNIST.

Network architecture. For image dataset experiments, we used convolutional layer structures as well as fully connected layers to capture the spatial information. Table 10 shows the structure of the models used in image datasets. Note that Fully Connected(FC), Convolutional(C), 2×2 Max-Pooling(M) and Transposed Convolutional(TC) layers are used. In the case of FC, C and TC, we add specific information of each layer as (Out Channel, Kernel Size, Stride, Padding). After all Convolutional layers, including Transposed Convolutional layers, we add Batch Normalization. *Dim* denotes the number of features in each dataset. For ex-

Table 7
Imputation results in numerical data in *Uniform* setting, differing missingness proportion from 10% to 90%, in terms of RMSE. The *y*-axis in the figure represents log scale of regularized RMSE value of each method divided by RMSE of mean imputation ($RMSE_{method}/RMSE_{Mean}$). Methods are illustrated in different color; MIDA : Orange, GAIN : Yellow, Amelia : Green, MICE : Blue, Missforest : Purple, and Ours : Red.

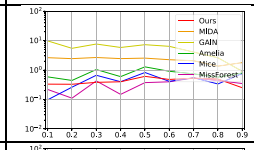
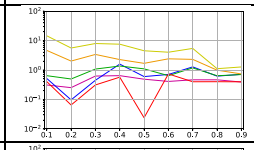
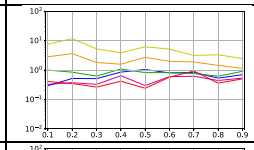
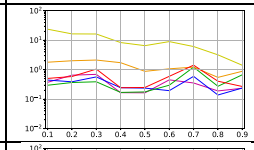
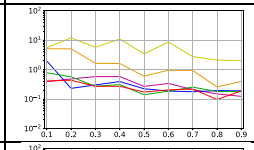
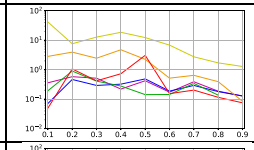
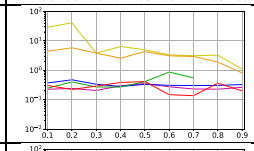
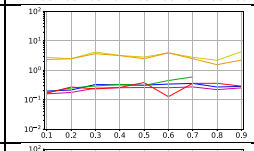
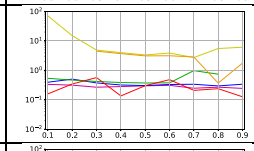
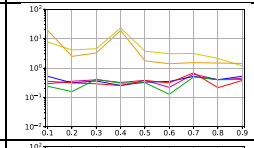
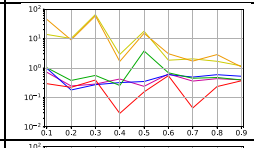
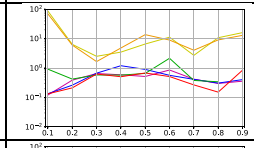
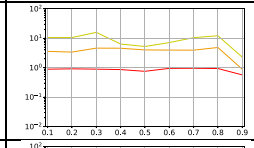
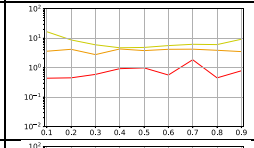
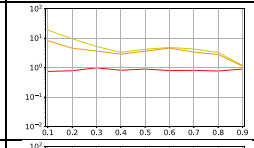
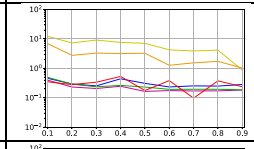
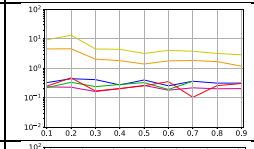
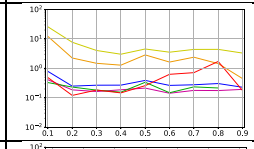
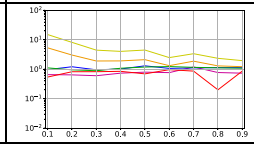
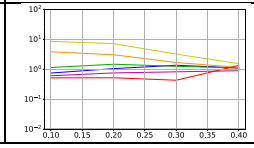
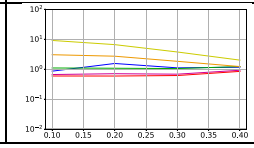
Datasets	MCAR	MAR	MNAR
Boston Housing			
Glass			
Satellite			
Shuttle			
Sonar			
Vehicle			
Vowel			

Table 8
Imputation results of numerical data in MAR[3] and MNAR[3] settings, in terms of MSE.

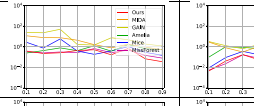
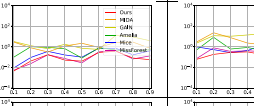
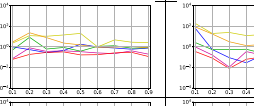
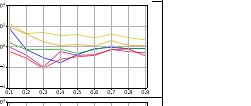
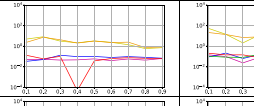
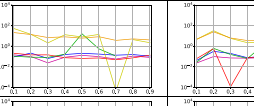
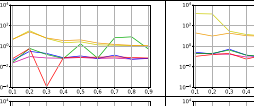
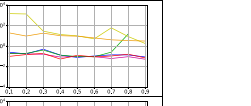
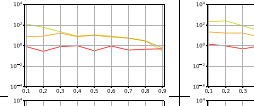
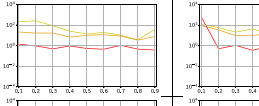
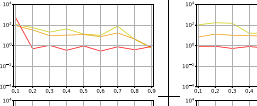
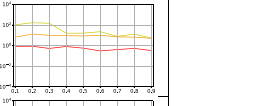
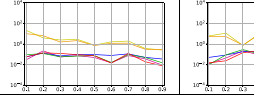
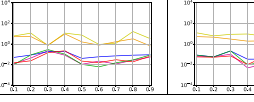
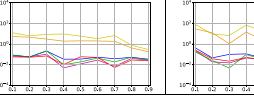
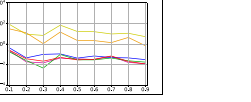
	MAR		MNAR	
	random	uniform	random	uniform
Boston Housing				
Satellite				
Sonar				
Vehicle				

Table 9
Imputation results of numerical data in MAR[3] and MNAR[3] settings, in terms of RMSE.

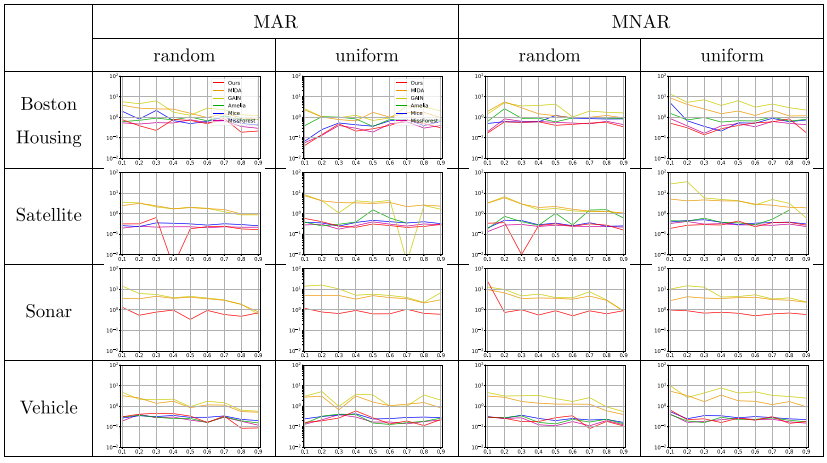


Table 10
Network architectures for Image datasets.

Model	Structure
$G_\phi = G_{miss}$	C(16,5,0,0)-C(32,5,0,0)-M-C(64,5,0,0)-M -FC(100)-TC(200,4,1,0)-TC(400,4,1,0)-TC(200,4,2,1)-TC(1,4,2,1)
$G_\theta = G_{impute}$	C(16,5,0,0)-C(32,5,0,0)-M-C(64,5,0,0)-M -TC(32,5,2,1)-TC(16,4,2,1)-TC(1,4,2,1)
$D_\theta = D_{miss}$	C(100,4,2,1)-C(200,4,2,1)-C(100,4,1,0)-C(1,4,1,0)
$D_\theta = D_{impute}$	FC(14×14)-FC(7×7)-FC(1)

ample, MNIST and Fashion-MNIST have the same dimension as $Dim = 28 \times 28$.

For all layers, ReLU is used as activation function, except following cases. (1) The last activation function of all Discriminators is Sigmoid$(\cdot)$. (2) $G_\phi = G_{miss}$ uses ReLU(Tanh$((\cdot))$ as a last activation to generate mask in range of [0,1]. (3) $D_\theta = D_{miss}$ uses LeakyReLU$(\cdot)$ instead of ReLU$(\cdot)$.

Missing pattern. As far as we are aware, it is the first attempt to extend a numeric imputation method to image datasets. Thus, we newly defined missing patterns. We considered three types of missing data distribution:

1. *UNIFORM-MCAR*: Each pixel is independently missing with the same probability. We experimented with missing rate from 20% to 80% at the same interval of 20% (Fig. 4a).
2. *SQUARE-MCAR*: Each picture is missing a square-shaped mask, independent of its class. We experimented with increasing the size of the mask from 6×6 to 15×15 pixels at the same interval of 3 pixels (Fig. 4b).
3. *DATA-DEPENDENT*: We added triangle-shaped missing masks never suggested in previous papers for considering complex missing shapes. Thus, depending on its class number, pictures are missing different masks: random, square or triangle (Fig. 4c and d).

We qualitatively assess how well our model learns true mask distribution in different missing cases. We considered all three missing patterns and missing rates stated above. Fig. 4a and b are samples from the mask generation results for UNIFORM-MCAR and SQUARE-MCAR, respectively. It is shown that the proposed model imitates not only the shape of various masks but also the missing rate (or size) very well. Moreover, it did not face mode collapse without taking any particular structure or distance.

Fig. 4c shows the real and generated masks in the DATA-DEPENDENT case. In the learning phase, each real image is masked by its class number(0 : random, 1 : square, and 2 : triangle). The result shows that in most cases the model generated the appropriate type of missingness conditioned on the number. The same experiment was also conducted on the Fashion-MNIST data ('T-shirt/top' : random, 'Trouser' : square, and 'Pullover' : triangle), with the results summarized in Fig. 4d.

Comparison. For the experiments on the image dataset, we compared with MIDA and GAIN. Since these methods are based on deep learning based framework, they are suitable to be applied to image datasets.

5.2.2. Imputation performance

Fig. 5 shows the imputation results of models in various conditions of missingness. We can see the our method produces better samples than others, especially when the missing rate is low.

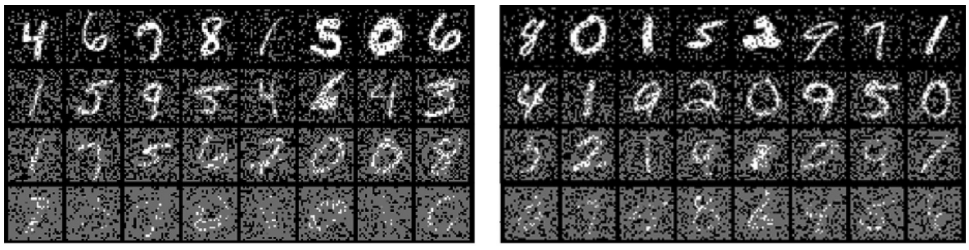
We infer the reasons for the lack of imputation capabilities of other models as follows. MIDA tries to generate pairs $(x_i, \tilde{x}_i^{obs})$ from given fully-observed data x_i by sampling a random mask (usually uniform mask) $m_i \sim U(M)$ and $\tilde{x}_i^{obs} = f_{m_i}(x_i)$.

However, this generated data may fail to represent the joint distribution $p(X, X^{obs})$ and the conditional distribution $p(X|X^{obs})$ since $U(M)$ is not the true mask distribution. If we sample m from distributions $U(M)$ that have a larger support than that of the true distribution such as uniform distribution, most of the data is useless for modeling the imputation relationship. The data sampled from a distribution with a smaller support is not sufficient to model the imputation process. Therefore, particularly in the SQUARE-MCAR situation, MIDA generally produces blurry images.

On the other hand, GAIN tries to generate samples according to $p(X|X^{obs})$ by adding MSE loss between $M \odot X$ and $M \odot \bar{X}$, where $\bar{X}$ denotes the output of the generator. This framework may work well for numerical data, but analyzing an image on a pixel-by-pixel basis can result in a poor understanding of the overall structure because it focuses on the local view. Consequently, the images generated from GAIN often do not look like numbers.

5.2.3. Prediction performance

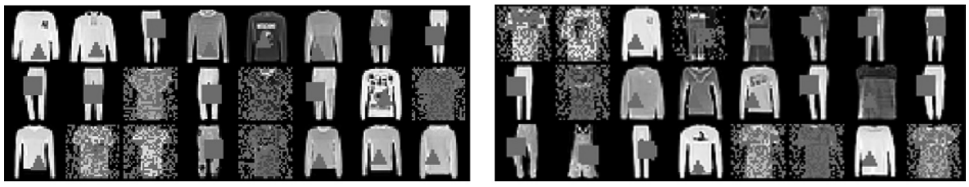
Now we evaluate the post-imputation prediction performance of our method by comparing the classification accuracy of the



(a) UNIFORM-MCAR with MNIST

(b) SQUARE-MCAR with MNIST

(c) DATA-DEPENDENT with MNIST



(d) DATA-DEPENDENT with Fashion-MNIST

Fig. 4. Sampled real(left) and generated(right) masks of the proposed method with different missingnesses.

Table 11
Classification results in DATA-DEPENDENT. Results are displayed in the form of Average ± Std of accuracy (%) of 10 different experiments. Best performance method is highlighted in bold.

Method	MNIST	Fashion-MNIST
GAIN	89.54 ± 0.467	88.19 ± 0.532
MIDA	84.14 ± 0.606	89.71 ± 0.404
Ours(MCAR)	98.82 ± 0.202	94.98 ± 0.286
Ours	**99.23 ±0.113**	**95.95 ±0.221**
Optimal	99.25 ± 0.092	97.36 ± 0.131

imputed image. For DATA-DEPENDENT missingness, as we experiment with only 3 labels, the basic classifier records accuracy of 99.8% for data without missing. Table 11 summarizes the prediction results for the DATA-DEPENDENT case, which shows that our model generally outperforms previous methods. 'Optimal' means optimal performance baseline with a model that used prior knowledge of missing distribution in training. In both experiments, our model showed the closest performance to the optimal model.

5.2.4. Data-dependent analysis

To address the data dependent missing situation, We consider the importance of the model to receive the complete image x along with a random vector z as input for in the mask generation and discrimination process. However, in different missing cases, x is not necessary as an input since the missing is independent of the observed data. Therefore, we construct a variant of our model with a mask generator that uses only z as its input. We refer to the new model as the MCAR version.

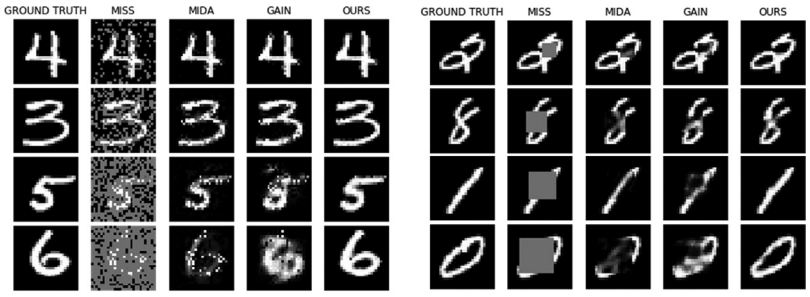
Fig. 6 illustrates the mask generation results of the MCAR version model for DATA-DEPENDENT missingness. It shows that the model generated any shape of mask, regardless of the digit label. Table 11 shows the original model scores higher in accuracy and lower in standard deviation than the MCAR version. This result supports our hypothesis that by using x as a mask generator input, the generator can better learn the data-dependent missing pattern, leading in turn to the better imputation performance.

5.3. Additional study : two cycle loss

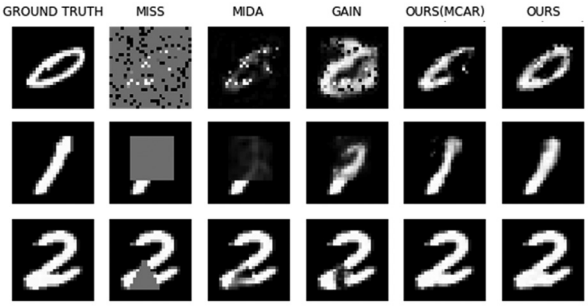
In all experiments we use only one-sided cycle loss of $X \rightarrow X^{obs} \rightarrow X$, which is denoted as Eq. (5). However, one might think of why the proposed method does not use the other cycle-consistent loss used in Zhu et al. [21].

$$\mathbb{E}_{p^*(X^{obs})p(\varepsilon)}[\|G_\phi(G_\theta(X^{obs},\varepsilon),\epsilon) - X^{obs}\|_2^2] \tag{18}$$

The reason for not using $X^{obs} \rightarrow X \rightarrow X^{obs}$ is because we designed the missing generator G_θ to produce stochastic outputs.

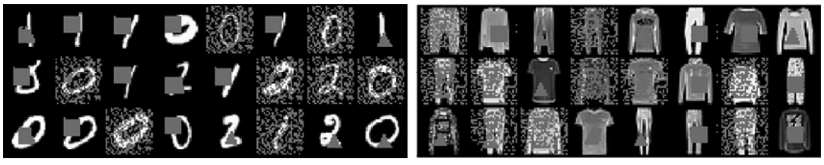


(a) UNIFORM-MCAR with MNIST (b) SQUARE-MCAR with MNIST



(c) DATA-DEPENDENT with MNIST

Fig. 5. Sampled imputation results for different missingnesses. From the left, ground-truth, missingness, MIDA imputation, GAIN imputation, and OURS imputation are shown in order.



(a) MNIST (b) Fashion-MNIST

Fig. 6. Sampled generated masks with MCAR version for DATA-DEPENDENT missingness.

Since missing pattern in the real world is rarely deterministic, reconstructing the exact incomplete data in a deterministic manner doesn't fit the purpose of the proposed model. Using the cycle loss in Eq. (5), the missing generator G_θ fails to learn the missing pattern if there is randomness in the true missing pattern. Since, it is impossible to learn the randomness, G_θ converges to a wrong solution that does not generate masks.

Fig. 7 illustrates the results of our method (left) and two cycle version (right). The first row shows the task of $X \rightarrow X^{obs} \rightarrow X$ and the second row shows the task of $X^{obs} \rightarrow X \rightarrow X^{obs}$. Our method successfully generates a clear mask on the original image x as shown $\tilde{x}^{obs}$ in the first row. Furthermore, our method filled and generated missing patterns more naturally than the given image.

However, the missing generator of two cycle version is converged to the wrong point which rarely produces masks. As shown in the right figure of Fig. 7, x is almost same as $\tilde{x}^{obs}$ in the first row. In the second row, $\tilde{x}$ and $\hat{x}^{obs}$ are definitely the same which indicates the missing generator did not produce masks. Considering the chance of two independent missing images is nearly 0, the loss that reduces the difference between x^{obs} and $\tilde{x}^{obs}$ induces the model to produce no masks under the stochastic characteristic of missing patterns. Considering that the two cycle version of the Fig. 7 is similar to Zhu et al. [21], this result illustrates the difference between our method and [21].

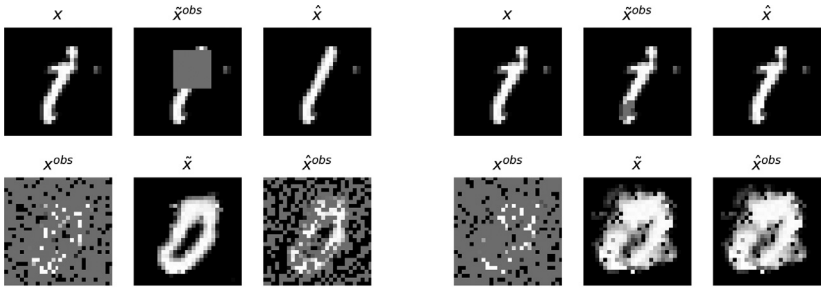
5.4. Additional study : imputation for unseen MNAR incomplete data

In the real-world missing problems, there might be cases when the missing patterns does not exist in the MNAR incomplete data. In this case, since the missing patterns are 'unseen' in the training phase, it is difficult to impute the missing value.

However, we believe that our method can better impute the missing in this case, since it can consider the distribution of the complete data distribution $\mathcal{D}$. This leads the model to understand the underlying distribution of complete data and it can serve as a benefit when the model reconstructs incomplete data.

To show our proposed method handles unseen incomplete data better than comparison methods, we conducted additional studies. Based on the models that were trained on DATA-DEPENDENT settings, we have tested the models on the missing data that were not seen in the training data. For example, since the square missing pattern only appears on the class 1 in the training set, we have checked if the models can handle the square missing pattern in class 0 or 2.

The results are shown in Fig. 8. Compared to MIDA and GAIN we can find that our method successfully imputed the missing part even though the missing pattern incomplete data has not been observed in the training data.



(a) Ours (b) Two Cycle

Fig. 7. Imputation and missing results of our method 7a and two cycle version 7b. The first row shows the task of $X \rightarrow X^{obs} \rightarrow X$ while the second row shows $X^{obs} \rightarrow X \rightarrow X^{obs}$.

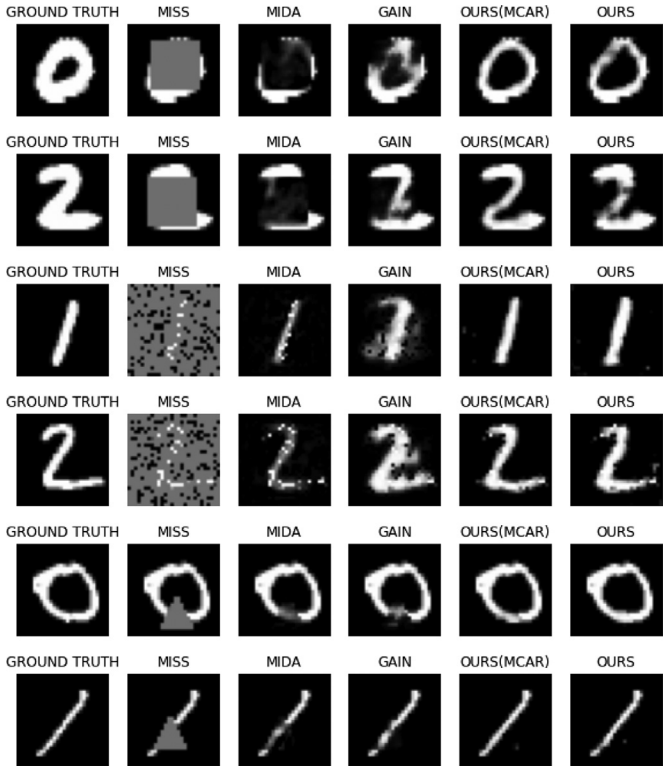


Fig. 8. Imputation results on unseen missing patterns where the models are trained on the DATA-DEPENDENT settings in Fig. 5c.

Table 12
Classification results in unseen missing patterns. Results are displayed in the form of accuracy (%).

Missing pattern	MIDA	GAIN	Ours (MCAR)	Ours
Random	81.97	67.26	96.55	**96.97**
Square	55.81	75.52	79.03	**93.03**
Triangle	99.48	99.45	99.58	**99.61**

Moreover, the prediction results for unseen incomplete data are summarized in Table 12. Our model showed the best results in all three unseen incomplete data settings.

5.5. Estimating ground-truth missing patterns

As illustrated in Fig. 9, our proposed model tries to optimize both imputation and missing generator (G_θ and G_ϕ) similar to the ground-truth missing pattern $p(x|x^{obs})$ and imputation process $q(x^{obs}|x)$, respectively.

Until now, we have measured the imputation and the prediction performance of our proposed method. As in Fig. 9, we have checked how our method can optimize the imputation generator similar to the true imputation process $G_\theta \approx p(x|x^{obs})$.

To push further, we show our proposed method can learn the ground-truth missing pattern $p(x|x^{obs})$. First, we will check how does the distribution of the generated incomplete data $\tilde{x}^{obs}$ $(= G_\phi(x))$ is similar to the original incomplete data x^{obs}. We visualized the results to qualitatively show out method can learn the missing pattern.

Second, we have done experiments on large synthetic datasets where missing patterns are generated on causal graphs. We checked the performance of our framework quantitatively on various number of variables and sample sizes.

Finally, by setting the ground-truth missing pattern $p(x|x^{obs})$ as neural-network based model, we check if our method can estimate the ground-truth parameters.

Estimating ground-truth missing distribution In this experiment we check how our proposed method can generate synthetic incomplete data $\tilde{x}^{obs}$ similar to the true incomplete data x^{obs}. To do this, we have generated two-dimensional original data where the two attributes x_1, x_2 are generated as follows:

$$x_1 \sim U(0,1),$$

$$x_2 = 2x_1 + \mathcal{N}(0,1).$$

x_1 is a uniform variable and x_2 is a function of x_1 with a Gaussian error. We let x_2 as the variable that can be missing, and use three missing algorithms to generate true incomplete data x^{obs}. The three ground-truth missing mechanisms are a follows:

1. MCAR : Missingness of x_2 is completely random
2. MAR : Missingness of x_2 depends on the value of x_1
3. MNAR : Missingness of x_2 depends on the value of x_2

For the missing imputation setting and the graphical visualization was inspired by the work in Van Buuren [27].

Table 13 shows the scatter plot of the original data x, ground-truth incomplete data x^{obs}, and the generated synthetic incomplete data $\tilde{x}^{obs}$. We denote the missing data as red and the data without missing as black.

In the MCAR case, since the missing is completely random, the red points in x^{obs} are randomly distributed, and our generated incomplete data $\tilde{x}^{obs}$ also shows similar results. For MAR and MNAR cases, because missing depends on the value of the original data, the mean value of the x_1 and x_2 ($\overline{x_1^{obs}}$ and $\overline{x_2^{obs}}$) is different from the original distribution. Although, there is big differences between mean values, the values of $\overline{\tilde{x}_1^{obs}}$ and $\overline{\tilde{x}_2^{obs}}$ were found to be similar with the ground-truth incomplete data in both MAR and MNAR

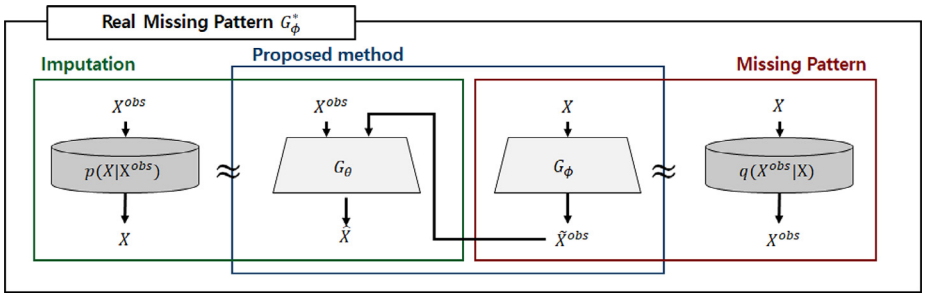


Fig. 9. Illustration of our proposed method and experiments.

Table 13
Scatter plot of the original data x, ground-truth incomplete data x^{obs}, and the generated synthetic incomplete data $\tilde{X}^{obs}$ on three different missing mechanisms. We denote the missing data as red and the data without missing as black.

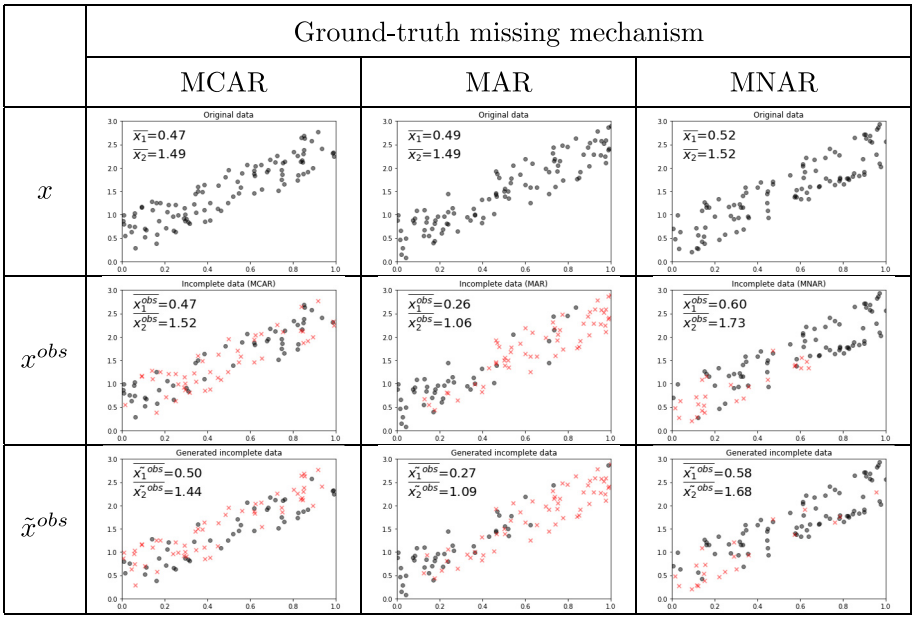


Table 14
Results of our method in predicting missing patterns with different number of variables.

# of variables	Accuracy	Precision	Recall	F1-Score
10	0.6613	0.6281	0.3659	0.4624
20	0.7820	0.5714	0.2486	0.3465
50	0.9140	0.5250	0.0878	0.1505

Table 15
Results of our method in predicting missing patterns with different number of samples.

# of samples	Accuracy	Precision	Recall	F1-Score
500	0.7372	0.6050	0.2703	0.3736
1000	0.6502	0.6992	0.2992	0.4190
10,000	0.6613	0.6281	0.3659	0.4624
50,000	0.6940	0.7928	0.2974	0.4326
100,000	0.6993	0.8031	0.2813	0.4167

experiments. This is also supported by the fact that the scatter plot in the third row $\tilde{x}^{obs}$ is similar to the second row x^{obs}.

Predicting ground-truth missing patterns on synthetic datasets We have evaluated our method on large synthetic datasets, where the missing patterns are generated on randomly causal graphs. We have followed the settings in Tu et al. [28], using MNAR settings for missing generation methods.

We evaluated the performance of estimating the missing patterns by calculating the accuracy, precision, recall, and F1 score on predicting whether the variable will be missing or not. We experimented with different numbers of variables and sample sizes to see how the method performed in different environments.

The results of our method in predicting missing patterns with different number of variables and samples are summarized in Tables 14 and 15. We can find that our proposed method shows consistent performance regardless of the number of samples. However, as the number of variables increases, the F1-score of our method also decreased. We assume that since the ratio of missing becomes low when the number of variables increases, our model's performance on predicting the missing location decreases. However, considering that our missingness is randomly generated, our proposed method has well predicted the unknown missing pattern on a synthetic large dataset.

Estimating ground-truth missing parameter To verify that our method can estimate the ground-truth missing parameters, we have conducted an additional experiment on numerical datasets. Given original data x, we have manually set the ground-truth missing mechanism as follows:

$$\mathrm{ReLU}(x\theta_{gt}), \tag{19}$$

Here, we consider the output with zero as the missing value for given ground-truth missing parameter θ_{gt}. Thus, in this experiment, the missing pattern is totally dependent with the

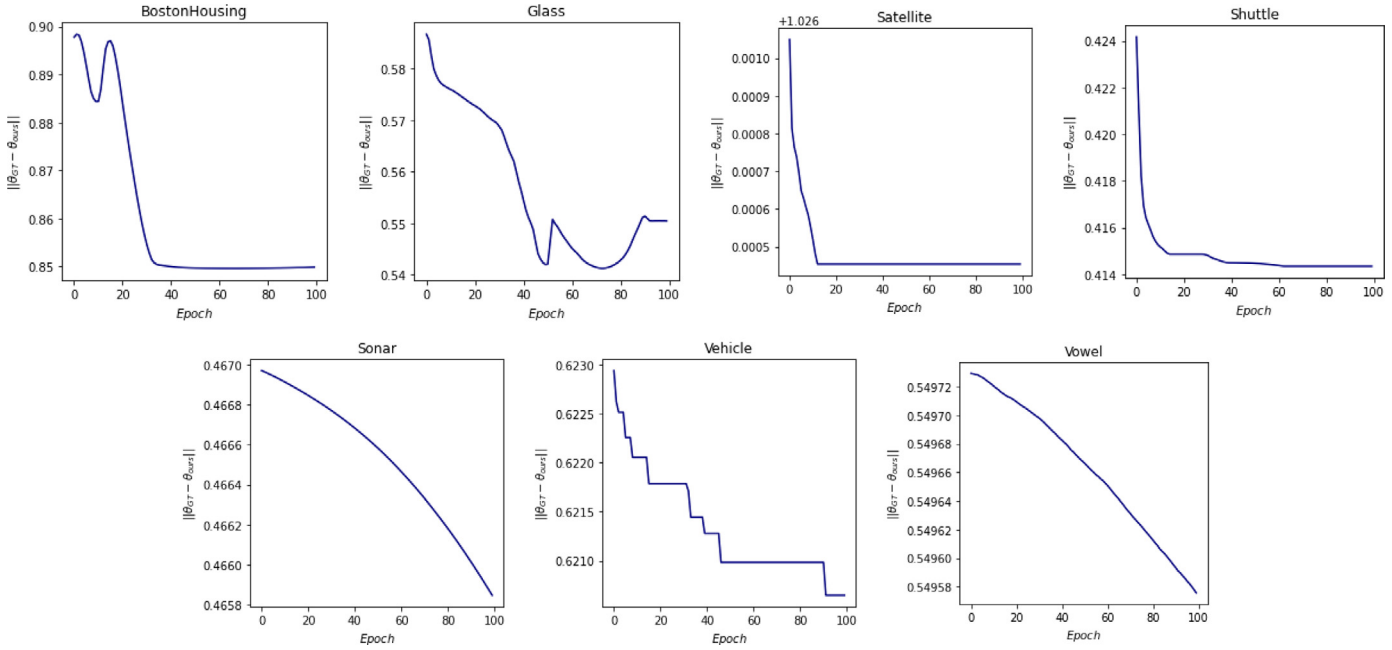


Fig. 10. L2 distance between the ground-truth missing parameter θ_{gt} and our parameter θ_{ours}.

parameter θ_{gt} so that θ_{gt} is the parameter we want to estimate.

Then, we apply our method on this setting, where we set our missing generator G_θ as the same structure as Eq. (19), as follows:

$$G_\theta = \mathrm{ReLU}(x\theta_{ours}), \tag{20}$$

where θ_{ours} is the parameter that will be optimized by Algorithm 1.

Fig. 10 shows the L2 distance between θ_{gt} and θ_{ours} during the training process. In all seven datasets, we can find that during training, the distance between two parameters decreases. It implies that our proposed method has successfully estimated the ground-truth missing mechanism G_θ by optimizing the weights θ_{ours}.

6. Discussion and future work

In this paper, we have proposed a effective imputation method that can be applied to various types of missing data patterns, including data. Our model learns to generate fake incomplete data indistinguishable from real missing data, and thus better impute the incomplete data by considering the missing pattern.

Our theoretical analysis approximates the ground-truth joint distribution of the complete and incomplete data $q_\phi(x, x^{obs})$ through a variational joint $p_\theta(x, x^{obs})$ as shown in Eq. (12). However, since the two processes q_ϕ and p_θ are estimated jointly with adversarial loss and cycle-consistency loss function as in Algorithm 1, it is not theoretically proved that our proposed method can converge to the ground-truth missing parameters. In response, we have done experiments in Section 5.5, showing that our method can estimate the ground-truth missing patterns in synthetic Gaussian experiments and real-world experiments. However, even though we observed that the proposed model can successfully model the underlying missing distribution, more theoretical analysis must be performed.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Sungyoon Lee is supported by a KIAS Individual Grant (AP083601) via the Center for AI and Natural Sciences at Korea Institute for Advanced Study. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF-2019R1A2C2002358). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2020-0-01789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Appendix A. Proofs of theoretical results

Proof of Theorem 1

Proof. From the definition of $\mathcal{L}_{NELP}(\theta, \phi)$ and by the assumption $p_\theta(x|x^{obs}) = \mathcal{N}(x|G_\theta(x^{obs}), \frac{1}{2\gamma}I)$, we have

$$\begin{aligned}
\mathcal{L}_{NELP}(\theta,\phi) &= \mathbb{E}_{q^*(x)p(\epsilon)}[-\ln p_\theta(x|G_\phi(x,\epsilon))] \\
&= \gamma\mathbb{E}_{q^*(x)p(\epsilon)}[||x - G_\theta(G_\phi(x,\epsilon))||_2^2] + \frac{d}{2}\ln(\pi/\gamma) \\
&= \gamma V_{cyc}(G_\phi, G_\theta) + \frac{d}{2}\ln(\pi/\gamma) \\
&\simeq \gamma\frac{1}{m}\sum_{i=1}^{m}[||x_i - G_\theta(G_\phi(x_i,\epsilon_i))||_2^2] \\
&\quad + \frac{d}{2}\ln(\pi/\gamma) \text{ as } m \to \infty
\end{aligned}$$

□

Proof of Theorem 2

Proof. The second term can be expressed as

$$\mathrm{KL}[q_\phi(x^{obs}|x)||p^*(x^{obs})] = \mathbb{E}_{q_\phi(x^{obs}|x)}[\ln \rho^*(x^{obs};x)], \tag{A.1}$$

where $\rho^*(x^{obs};x)$ is defined as follows:

$$\rho^*(x^{obs};x) = q_\phi(x^{obs}|x)/p^*(x^{obs}). \tag{A.2}$$

Since we cannot directly compute this density $\rho^*(x^{obs};x)$ due to intractable $p^*(x^{obs})$, we follow the derivation and the result in Nguyen et al. [29], Tiao et al. [30] of the lower bound for arbitrary f-divergences $D_f(p||q) = \mathbb{E}_p[f(q/p)]$:

$$D_f[p||q] \geq \max_{\hat{\rho}} \{\mathbb{E}_q[f'(\hat{\rho})] - \mathbb{E}_p[f^*(f'(\hat{\rho}))]\} \tag{A.3}$$

where f^* be the Fenchel conjugate of f and equality holds when $\hat{\rho} = q/p$.

Thus, applying the KL-divergence where $f(t) = t\ln t$ and $f^*(u) = e^{u-1}$ to (A.3), the density ratio estimator $\rho_\alpha(x^{obs};x)$ of ρ^* parametrized by α can be estimated by maximizing $\mathcal{L}_{\mathrm{KL}}^{missing}(\alpha,\phi)$ over α wherein ϕ is fixed as follows.

$$\mathbb{E}_{q^*}\mathrm{KL}[p^*(x^{obs})||q_\phi(x^{obs}|x)] \geq \max_\alpha \mathcal{L}_{\mathrm{KL}}^{missing}(\alpha,\phi)$$

where

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}}^{missing}(\alpha,\phi) &:= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[\ln \rho_\alpha(x^{obs};x)+1] - \mathbb{E}_{q^*(x)p^*(x^{obs})}[\rho_\alpha(x^{obs};x)] \\ &= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[\ln \rho_\alpha(x^{obs};x)] + \text{terms not involving } \phi\end{aligned} \tag{A.4}$$

and equality holds when $\rho_\alpha(x^{obs};x) = \rho^*(x^{obs};x)$. For the optimal α, we have

$$\min_\phi \mathcal{L}_{\mathrm{KL}}^{missing}(\alpha,\phi) = \min_\phi \mathbb{E}_{q^*}\mathrm{KL}[q_\phi(x^{obs}|x)||p^*(x^{obs})] \tag{A.5}$$

Using the Jensen Shannon divergence $f(t) = t\ln t - (t+1)\ln(t+1)$ and its Fenchel conjugate $f^*(u) = -\ln(1-e^u)$ in Eq. (A.3) and reparametrization of $q_\phi(x^{obs}|x)$, we have

$$\begin{aligned}\mathcal{L}_{JS}(\alpha,\phi) &= \mathbb{E}_{q^*(x)p^*(x^{obs})}[\ln \mathcal{D}_\alpha(x^{obs};x)] + \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[\ln(1-\mathcal{D}_\alpha(x^{obs};x))] \\ &= V_{miss}(D_\phi, G_\phi)\end{aligned}$$

where $\mathcal{D}_\alpha(x^{obs};x) = 1-\sigma(\ln \rho_\alpha(x^{obs};x))$, σ is the logistic sigmoid function, and $\rho_\alpha(x^{obs};x) = \rho^*(x^{obs};x)$ maximizes $\mathcal{L}_{JS}(\alpha,\phi)$.

$$\begin{aligned}\mathcal{L}_{KL}^{missing}(\alpha,\phi) &= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}\left[\ln \rho_\alpha(x^{obs};x)\right] \\ &= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}\left[\ln \frac{\sigma(\ln \rho_\alpha(x^{obs};x))}{1-\sigma(\ln \rho_\alpha(x^{obs};x))}\right] \\ &= \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[-\ln \mathcal{D}_\alpha(x^{obs};x)] \\ &\quad + \mathbb{E}_{q^*(x)q_\phi(x^{obs}|x)}[\ln(1-\mathcal{D}_\alpha(x^{obs};x))] \\ &= \mathcal{L}_{GAN}(\alpha,\phi) + \mathcal{L}_{JS}(\alpha,\phi) + \text{terms not involving } \phi.\end{aligned}$$

where the first term of the last equation the GAN loss function suggested by Goodfellow et al. [13] for the optimal α. Therefore, for the optimal α, $\min_\phi \mathcal{L}_{KL}^{missing}(\alpha,\phi)$ reduces to $\min_\phi \mathcal{L}_{JS}(\alpha,\phi)$, which is equivalent to $\min_\phi V_{missing}(D_\phi, G_\phi)$ for optimal D_ϕ. □

Proof of Theorem 3

Proof. Using (A.3) in the same way as above, we estimate the density ratio estimator $\rho_\beta(x;x^{obs})$ of $p_\theta(x|x^{obs})/q^*(x)$ parametrized by β by maximizing $\mathcal{L}_{\mathrm{KL}}^{impute}(\beta,\theta)$ over β wherein θ is fixed as follows.

$$\mathbb{E}_{p^*}\mathrm{KL}[q^*(x)||p_\theta(x|x^{obs})] \geq \max_\beta \mathcal{L}_{\mathrm{KL}}^{impute}(\beta,\theta)$$

where

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}}^{impute}(\beta,\theta) &:= \mathbb{E}_{p^*(x^{obs})p_\theta(x|x^{obs})}[\ln \rho_\beta(x;x^{obs})+1] - \mathbb{E}_{p^*(x^{obs})q^*(x)}[\rho_\beta(x;x^{obs})] \\ &= \mathbb{E}_{p^*(x^{obs})p_\theta(x|x^{obs})}[\ln \rho_\beta(x;x^{obs})] + \text{terms not involving } \theta\end{aligned} \tag{A.6}$$

and equality holds when $\rho_\beta(x;x^{obs}) = p_\theta(x|x^{obs})/q^*(x)$.

Therefore, minimizing the mean KL divergence with respect to x^{obs} in (15) can be summarized as follows:

$$\min_\theta \max_\beta \mathcal{L}_{\mathrm{KL}}^{impute}(\beta,\theta) \tag{A.7}$$

Rest of the proof is similar to that of Theorem 2. □

References

[1] Z.-g. Liu, Q. Pan, J. Dezert, A. Martin, Adaptive imputation of missing values for incomplete pattern classification, Pattern Recognit. 52 (2016) 85–95.
[2] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognit. 41 (12) (2008) 3692–3705.
[3] S. Sinharay, H.S. Stern, D. Russell, The use of multiple imputation for the analysis of missing data, Psychol. Methods 6 (4) (2001) 317.
[4] S. van Buuren, K. Groothuis-Oudshoorn, Mice: multivariate imputation by chained equations in R, J. Stat. Softw. 45 (2010) 1–68.
[5] J. Honaker, G. King, M. Blackwell, et al., Amelia II: a program for missing data, J. Stat. Softw. 45 (7) (2011) 1–47.
[6] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1096–1103.
[7] Y. Luo, X. Cai, Y. Zhang, J. Xu, et al., Multivariate time series imputation with generative adversarial networks, in: Advances in Neural Information Processing Systems, 2018, pp. 1603–1614.
[8] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, Brits: bidirectional recurrent imputation for time series, in: Advances in Neural Information Processing Systems, 2018, pp. 6775–6785.
[9] L. Gondara, K. Wang, MIDA: multiple imputation using denoising autoencoders, in: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 260–272.
[10] J. Yoon, J. Jordon, M. Schaar, Gain: Missing data imputation using generative adversarial nets, in: International Conference on Machine Learning, 2018, pp. 5689–5698.
[11] F.M. Bianchi, L. Livi, K.Ø. Mikalsen, M. Kampffmeyer, R. Jenssen, Learning representations of multivariate time series with missing data, Pattern Recognit. 96 (2019) 106973.
[12] S.C.X. Li, B. Jiang, B. Marlin, Misgan: learning from incomplete data with generative adversarial networks, arXiv preprint arXiv:1902.09599 (2019).
[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[14] O. Ivanov, M. Figurnov, D. Vetrov, Variational autoencoder with arbitrary conditioning, arXiv preprint arXiv:1806.02382 (2018).
[15] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, Stat 1050 (2014) 1.
[16] A. Nazabal, P.M. Olmos, Z. Ghahramani, I. Valera, Handling incomplete heterogeneous data using VAEs, Pattern Recognit. 107 (2020) 107501.
[17] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, A. Geiger, Sparsity invariant CNNs, in: Proceedings of the International Conference on 3D Vision (3DV), IEEE, 2017, pp. 11–20.
[18] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 85–100.
[19] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.
[20] D.J. Stekhoven, P. Bühlmann, Missforest–non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (1) (2012) 112–118.
[21] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.
[22] T. Kim, M. Cha, H. Kim, J.K. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 1857–1865.
[23] X. Huang, M.-Y. Liu, S. Belongie, J. Kautz, Multimodal unsupervised image–to-image translation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 172–189.
[24] M. Lichman, et al., UCI machine learning repository, 2013,
[25] Y. LeCun, The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/(1998).

[26] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).
[27] S. Van Buuren, Flexible Imputation of Missing Data, CRC Press, 2018.
[28] R. Tu, C. Zhang, P. Ackermann, K. Mohan, H. Kjellström, K. Zhang, Causal discovery in the presence of missing data, in: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1762–1770.
[29] X. Nguyen, M.J. Wainwright, M.I. Jordan, Estimating divergence functionals and the likelihood ratio by convex risk minimization, IEEE Trans. Inf. Theory 56 (11) (2010) 5847–5861.
[30] L.C. Tiao, E.V. Bonilla, F. Ramos, Cycle-consistent adversarial learning as approximate Bayesian inference, arXiv preprint arXiv:1806.01771 (2018).