Differentiable and Scalable Generative Adversarial Models for Data Imputation

Yangyang Wu, Jun Wang, Xiaoye Miao, Wenjia Wang, and Jianwei Yin

Abstract—Data imputation has been extensively explored to solve the missing data problem. The dramatically increasing volume of incomplete data makes the imputation models computationally infeasible in many real-life applications. In this paper, we propose an effective scalable imputation system named SCIS to significantly speed up the training of the differentiable generative adversarial imputation models under accuracy-guarantees for large-scale incomplete data. SCIS consists of two modules, differentiable imputation modeling (DIM) and sample size estimation (SSE). DIM leverages a new masking Sinkhorn divergence function to make an arbitrary generative adversarial imputation model differentiable, while for such a differentiable imputation model, SSE can estimate an appropriate sample size to ensure the user-specified imputation accuracy of the final model. Moreover, SCIS can also accelerate the autoencoder based imputation models. Extensive experiments upon several real-life large-scale datasets demonstrate that, our proposed system can accelerate the generative adversarial model training by 6.23x. Using around 1.27% samples, SCIS yields competitive accuracy with the state-of-the-art imputation methods in much shorter computation time.

Index Terms—Data imputation, generative adversarial network, large-scale incomplete data.

I. INTRODUCTION

Many real-life scenarios, such as e-commerce, transportation science, and health care, encounter the problem of missing data [1], [2], [3], [4] as long as the data collection is involved. Data might be missing for various reasons, including collection device failure [5], instable environment [2], privacy concerns [6], etc. The missing data problem poses a daunting challenge to the downstream data analysis.

Manuscript received 18 April 2022; revised 18 May 2023; accepted 2 July 2023. Date of publication 7 July 2023; date of current version 11 January 2024. This work was supported in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21P020005, in part by the National Natural Science Foundation of China under Grants 61825205 and 61902343, in part by the Fundamental Research Funds for the Central Universities under Grant 2021FZZX001-25, in part by the Key Research and Development Jiabing Program of Zhejiang Province under Grant 2023C01002, and in part by Hangzhou Major Project and Development Program under Grant 2022AZD0140. Recommended for acceptance by J. Lee. (Jun Wang is co-first author) (Corresponding author: Xiaoye Miao.)

Yangyang Wu and Xiaoye Miao are with the Center for Data Science, Zhejiang University, Hangzhou 310058, China (e-mail: zjuwy@zju.edu.cn; miaoxy@zju.edu.cn).

Jun Wang and Wenjia Wang are with the Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: jwangfx@connect.ust.hk; wenjiawang@ust.hk).

Jianwei Yin is with the Center for Data Science, College of Computer Science, Zhejiang University, Hangzhou 310058, China (e-mail: zjuyjw@cs.zju.edu.cn).

Digital Object Identifier 10.1109/TKDE.2023.3293129

To alleviate the missing data problem, a substantial amount of data imputation studies [7], [8] has been carried out, including the statistical ones [9], machine learning ones [10], multi-layer perceptron (MLP) based ones [11], autoencoder (AE) based ones [12], and generative adversarial network (GAN) based ones [13], [14]. Ideally, a preferable data imputation algorithm is to model the complex patterns and learn the true underlying data distribution well from the observed data, and count on the learned distribution to impute the missing data. GAN-based methods [13], [14] are attempts in this direction. They maximize the distributional proximity of generated and true underlying data by introducing an adversarial game. Many empirical investigations [13], [14] have demonstrated the promising performance brought by GAN-based imputation methods.

The ubiquity of data collection technologies with unprecedented processing power and substantial storage capacity has given rise to a dramatic increase in the volume of incomplete data. For example, a real-world COVID-19 case surveillance public use dataset [15] contains 22,507,139 cases with 7 clinical and symptom features, taking a 47.62% missing rate. The large volume of incomplete data, however, means that it is expensive and unwieldy to exploit the above algorithms. Although GAN-based methods achieve better imputation performance than other methods, they deplete exceedingly high training cost over large-scale incomplete data. Our experimental results show that, almost all imputation methods take more than a day on training over the million-size incomplete data. In particular, these imputation methods are trained offline (if necessary) and prediction online for various data types, e.g., time-series, image, text, tabular data, etc. In spite of this, the extremely long training time is unacceptable or infeasible, especially for large-scale incomplete data. Meanwhile, the highly updated data situation also requires the fast training. For instance, in many real-life scenarios, e.g., e-commerce, transportation science, and health care, the (in)complete data update frequently, resulting in a sharp drop in imputation accuracy. Thus, imputation model (re)training acceleration for online deployment is imminent in real-life applications. Consequently, the effective and scalable data imputation over large-scale incomplete data is indispensable in many real-life fields.

It is challenging to apply the GAN-based imputation methods to specific real-life scenarios, particularly for large-scale incomplete data. First, there is a strong theoretical evidence [16], [17], [18] showing that, the Jensen-Shannon (JS) divergence of the GAN-based imputation model fails in the case that the true underlying and generated data distributions have non-overlapping
It makes the JS divergence based imputation loss function non-differentiable and exhibit “vanishing” gradient problem in various incomplete datasets. Second, existing GAN-based imputation methods consume high training cost to calculate gradients with the mini-batch strategy for both generator and discriminator over the entire dataset. In general, the model complexity and training sample size are two primary factors that affect the efficiency of the GAN-based imputation.

Therefore, in this paper, we propose an effective SCalable Imputation System SCIS to optimize GAN-based imputation models. SCIS makes the GAN-based model differentiable via using the optimal transport theory. Then, for the differentiable model, it pays attention to the training sample size, and estimates an appropriate sample size for efficient and accuracy-guaranteed training. The system is composed of two modules, a differentiable imputation modeling (DIM) module and a sample size estimation (SSE) module. In terms of the first challenge, the DIM module leverages a masking Sinkhorn (MS) divergence to improve a GAN-based imputation model as a differentiable one, which can always provide reliable gradients to avoid the “vanishing” gradient problem. Regarding the second challenge, the SSE module estimates the minimum sample size to enable the trained differentiable GAN-based model to meet a user-specified error tolerance, instead of training the model with the whole dataset. Hence, SCIS employs the minimum sample size estimation to make the differentiable GAN-based model scalable on large-scale incomplete data. In addition, SCIS can also help to accelerate the AE-based imputation models. In summary, the main contributions are described as follows.

- We propose an effective and efficient scalable imputation system SCIS to accelerate GAN-based imputation models under accuracy guarantees.
- In the DIM module of SCIS, we put forward an MS divergence to quantify the closeness between the true underlying and generated data distributions. It employs the optimal transport theory to make the GAN-based imputation model differentiable, for avoiding the “vanishing” gradient issue.
- The SSE module of SCIS leverages the empirical estimation technique to estimate the minimum sample size for training an approximate GAN-based model, according to a user-specified imputation accuracy. It thus significantly saves the model training cost.

Extensive experiments using several real-life datasets demonstrate the computational benefits of SCIS, compared with the state-of-the-art methods.

The rest of the paper is organized as follows. We introduce the background in Section II. Section III gives an overview of the proposed system SCIS. We elaborate the DIM and SSE modules in Sections IV and V, respectively. Section VI reports the experimental results and findings. Finally, we conclude this work in Section VII.

### II. BACKGROUND

#### A. Existing Imputation Methods

Existing imputation algorithms can be categorized into statistical ones, machine learning ones, and deep learning ones [19]. The statistical methods substitute the missing values with the statistics [9], or the most similar ones among the data [20], [21]. These methods have a limited imputation ability, since they ignore the data distribution analysis.

In contrast, the machine learning imputation approaches are to train machine learning models [22] to estimate the missing values, including decision tree models like XGBI (XGBoost imputation) [23], MissFI (MissForest imputation) [10], and Baran [24], and regression models like MICE (multivariate imputation by chained equations) [7], imputation via individual model [25], and ICLR (imputation via clusterwise linear regression) [26]. These methods employ the batch gradient descent techniques [27] to calculate the gradient over the entire dataset. However, the incomplete dataset may be too large to fit in memory.

Moreover, the deep learning imputation methods leverage deep learning models [28] to impute missing values. This category consists of i) MLP-based ones like DataWig [29] and RRSI (round-robin Sinkhorn imputation) [11], ii) AE-based ones, e.g., MIADE (multiple imputation denoising autoencoder) [30], VAEI (variational autoencoder imputation) [31], EDDI (efficient dynamic discovery of highvalue information framework) [32], HVAEE (heterogeneous incomplete variational autoencoder) [33], MIWAE (missing data importance-weighted autoencoder) [8], and not-MIWAE (not-missing-at-random importance-weighted autoencoder) [12], and iii) GAN-based ones, such as GINN (graph imputation neural network) [14] and GAIN (generative adversarial imputation network) [13]. The above methods calculate the model gradients with a series of random partitions of the dataset. Nevertheless, both the iteration times and training cost of these methods are dramatically increasing with the rising volume of incomplete data.

In addition, the existing imputation algorithms can also be categorized as iterative ones and non-iterative ones [34] and [19]. The iterative methods build at least one prediction model for each incomplete variable, such as statistical ones and machine learning ones. In contrast, the non-iterative methods, including AE-based ones and GAN-based ones, build one or multiple prediction models for the whole dataset. Therefore, the complexity of non-iterative models is lower than iterative ones, except statistical methods.

---

**TABLE I**

| Symbol | Description |
|--------|-------------|
| X (M)  | an input incomplete dataset with its mask matrix |
| \(M_1/M_x\) | an imputation/optimized model |
| \(X/X\) | the reconstructed/imputed matrix of \(X\) |
| \(X_0 (M_0)\) | the initial dataset matrix with its mask matrix |
| \(X_v (M_v)\) | the validation dataset matrix with its mask matrix |
| \(n_0/N_0\) | the initial/validation sample size |
| \(c/\alpha\) | the user-tolerated error bound/confidence level |

---

1. The support of a distribution is the complement of the largest open set of the random variables having zero-probability.

---

WU et al.: DIFFERENTIABLE AND SCALABLE GENERATIVE ADVERSARIAL MODELS FOR DATA IMPUTATION 491

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
One recent work EDIT [35] derives a set of high-effect samples to speed up the parametric model training. For each sample in the entire large-scale dataset, EDIT has to evaluate its influence power, resulting in a high computational cost. By contrast, SCIS is much more efficient, since it sidesteps the costly influence power computation, and directly and quickly gets a minimum set of training samples via an empirical estimation technique to achieve an accuracy-guaranteed GAN-based imputation model.

B. Problem Definition

The input incomplete dataset is stored in a matrix $\mathbf{X} = (x_{1}, \ldots, x_{N})$, in which each data sample $x_{i} = (x_{i1}, \ldots, x_{id})$ with $x_{ij} \in X_{j}$ takes values from a $d$-dimensional space $\mathcal{X} = X_{1} \times \cdots \times X_{d}$. For encoding its missing information, we define a mask matrix $\mathbf{M} = (m_{1}, \ldots, m_{N})$, where each mask vector $m_{i} = (m_{i1}, \ldots, m_{id})$ corresponds to a data sample $x_{i}$. In particular, $m_{ij}$ takes value from $\{0, 1\}$, $i = 1, \ldots, s$, and $j = 1, \ldots, d; m_{ij} = 1$ (resp. 0) iff the $j$-th dimension is observed (resp. missing). Note that, we use $\mathbf{X}$ and $\mathbf{X}_{N}$ interchangeably throughout this paper. Table I lists the frequently used symbols throughout this paper.

Definition 1 (Data imputation): Given an incomplete set $\mathbf{X}$ with its mask matrix $\mathbf{M}$, the data imputation problem aims to build an imputation model $\mathbf{M}$ to find suitable values for missing ones in $\mathbf{X}$, i.e., the imputed matrix

$$\hat{\mathbf{X}} = \mathbf{M} \odot \mathbf{X} + (1 - \mathbf{M}) \odot \bar{\mathbf{X}}$$

(1)

where $\odot$ is the element-wise multiplication; $\hat{\mathbf{X}} = \mathbf{M}(\mathbf{X})$ is the reconstructed matrix predicted by $\mathbf{M}$ over $\mathbf{X}$. In this way, the model $\mathbf{M}$, (i) makes $\hat{\mathbf{X}}$ as close to the real complete dataset (if it exists) as possible, or (ii) helps downstream prediction tasks to achieve better performance if adopting $\hat{\mathbf{X}}$ that than only with original $\mathbf{X}$.

In this paper, our study mission is to empower the GAN-based imputation model with efficiency and effectiveness for large-scale incomplete data, such that for the optimized model, (i) the training cost is minimized, and (ii) the imputation accuracy satisfies a user-tolerated error bound.

In particular, the GAN-based imputation model builds an adversarial training platform with the Jensen-Shannon (JS) divergence for the generator and discriminator. The generator produces the data values as close to the true underlying (observed) data distribution as possible. The discriminator distinguishes the difference between the generated data and true underlying data as correctly as possible. The objective function of GAN-based imputation is defined as a minimax problem over the generator and discriminator. As a result, the GAN-based model employs the optimized generator to impute missing values via using (1).

III. SYSTEM OVERVIEW

In this section, we present an overview of the proposed system SCIS. It consists of a differentiable imputation modeling (DIM) module and a sample size estimation (SSE) module.

To facilitate the effective and scalable imputation on large-scale incomplete data, SCIS first uses the DIM module to improve a GAN-based imputation model as a differentiable one, for avoiding the “vanishing” gradient problem. Then, for such a differentiable GAN-based imputation model, SCIS employs the SSE module to estimate the training sample size under accuracy-guarantees to accelerate the imputation. The general procedure is described in Algorithm 1. The inputs of SCIS include an incomplete set $\mathbf{X}$ with its mask matrix $\mathbf{M}$, a validation size $N_{v}$, an initial size $n_{0}$, a GAN-based model $\mathbf{M}$, a user-tolerated error bound $\varepsilon$, and a confidence level $\alpha$. It outputs the trained model $\mathbf{M}_{*}$ and the data $\hat{\mathbf{X}}$ imputed by $\mathbf{M}_{*}$.

At first, SCIS samples a size-$N_{v}$ validation dataset $\mathbf{X}_{v} \in \mathbb{R}^{N_{v} \times d}$ (with the validation mask matrix $\mathbf{M}_{v}$) from the incomplete input dataset $\mathbf{X} \in \mathbb{R}^{N \times d}$ (with the mask matrix $\mathbf{M}$), while it samples a size-$n_{0}$ dataset $\mathbf{X}_{0} \in \mathbb{R}^{n_{0} \times d}$ (with the initial mask matrix $\mathbf{M}_{0}$) from the rest (line 1). Then, SCIS invokes the DIM module to train an initial model $\mathbf{M}_{0}$ with the masking Sinkhorn (MS) divergence based imputation loss function over $\mathbf{X}_{0}$ and $\mathbf{M}_{0}$ (line 2).

Next, with the support of the differentiability of MS divergence, SCIS consults the SSE module to estimate the minimum sample size $n_{*}$ (with $n_{0} \leq n_{*} \leq N$) based on $\mathbf{M}_{0}$, such that the imputation difference of the models over the size-$n_{*}$ dataset and the (whole) size-$N$ dataset would not exceed the user-tolerated error bound $\varepsilon$ with probability at least $(1 - \alpha)$. In particular, if $n_{*}$ is equal to $n_{0}$, SCIS simply outputs $\mathbf{M}_{0}$ and the matrix $\mathbf{X}$ imputed by $\mathbf{M}_{0}$. Otherwise, SCIS constructs a size-$n_{*}$ sample set $\mathbf{X}_{s}$ and its mask matrix $\mathbf{M}_{s}$ from $\mathbf{X}$ and $\mathbf{M}$. It invokes the DIM module again to retrain $\mathbf{M}_{0}$ using $\mathbf{X}_{s}$ and $\mathbf{M}_{s}$ (lines 3-5). Finally, SCIS returns $\mathbf{M}_{*}$ and the matrix $\mathbf{X}$ imputed by $\mathbf{M}_{*}$.

In particular, SCIS can also make the AE-based imputation models scalable with the support of the DIM and SSE modules. Similarly, SCIS employs the DIM module to replace the maximum likelihood estimation based imputation loss function of AE-based model with the MS divergence based imputation loss function. It then invokes the SSE module to estimate the minimum sample size to enable the trained AE-based model to meet a user-specified error tolerance, and thus makes the model scalable.

IV. DIFFERENTIABLE IMPUTATION MODELING

For the GAN-based imputation model, the intersection in the supports of the true underlying and generated data distributions is usually negligible [16]. In such case, the JS divergence makes the GAN-based models non-differentiable, and even suffers from the “vanishing” gradient problem in diverse datasets. This problem prevents the generator parameter of GAN-based models from updating its value or even stop the generator from further training.

In the SCIS, the differentiable imputation modeling (DIM) module designs a masking Sinkhorn (MS) divergence by deploying the mask matrix from the incomplete dataset and optimal transport theory. It makes the GAN-based imputation models differentiable, and thus obtains reliable gradients.
Algorithm 1: The Procedure of SCIS.

Input: an incomplete set $X$ with its mask matrix $M$, a validation size $N_v$, an initial size $n_0$, a GAN-based model $M$, a user-tolerated error bound $\varepsilon$, and a confidence level $\alpha$.

Output: the trained model $M_*$ and imputed data $\hat{X}$.

1: sample a size-$N_v$ validation dataset $X_0$ (with $M_0$) and a size-$n_0$ initial dataset $X_0$ (with $M_0$);
2: invoke DIM to train an initial model $M_0$ with the MS divergence loss function over $X_0$ and $M_0$;
3: consult SSE to derive the minimum size $n_*$ to satisfy the error bound $\varepsilon$ with probability at least $(1 - \alpha)$;
4: if $n_* = n_0$ then $M_* \leftarrow M_0$;
5: else invoke DIM to train $M_0$ on the minimum sample set $X_*$ and $M_*$ to obtain the optimized $M_*$;
6: reconstruct $X$ via using $M_*$ to obtain $\hat{X}$;
7: $X = M \odot X + (1 - M) \odot \hat{X}$;
8: return $M_*$ and $\hat{X}$.

through the model training and circumvents the “vanishing” gradient problem. In this section, we first introduce the MS divergence based imputation loss function. Then, we elaborate how to optimize the GAN-based imputation model via the MS divergence.

A. Masking Sinkhorn Divergence

We devise the masking Sinkhorn (MS) divergence by introducing the mask matrix, entropic regularizer, and corrective terms into the optimal transport metric, so as to equip it with the GAN-based imputation model to get the differentiability. Specifically, we first design a masking optimal transport (MOT) metric to empower the optimal transport metric with the mask matrix for data imputation. Then, built upon the MOT metric, we put forward the entropic regularizer to make the masking regularized optimal transport (MROT) metric differentiable and programmable. We further equip the MROT metric with corrective terms to correct the bias from the entropic regularizer, which underlies the MS divergence.

To be more specific, the masking optimal transport metric is introduced by leveraging the mask matrix and optimal transport metric, as stated in Definition 2. It is based on a simple intuition that the observed and generated data should share the same distribution. Let $\hat{\mu}_x \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$, $\hat{\mu}_m \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} \delta_{m_i}$, and $\hat{\nu}_x \overset{def}{=} \frac{1}{\sum_{i=1}^{n} \delta_{x_i}} \delta_{x_i}$ denote the empirical measures over a size-$n$ data matrix $X \subset \mathbb{R}$, its mask matrix $M$, and the reconstructed matrix $X_0$ respectively, where $\delta_{x_i}$, $\delta_{m_i}$, and $\delta_{x_i}$ are the Dirac distributions.

Definition 2 (The masking optimal transport, MOT): The MOT metric $OT^m(\hat{\nu}_x, \hat{\mu}_m)$ over $\hat{\nu}_x$ and $\hat{\mu}_m$ is defined as the optimal transport metric $OT(\hat{\nu}_x \otimes \hat{\mu}_m, \hat{\mu}_x \otimes \hat{\mu}_m)$, i.e., $OT^m(\hat{\nu}_x, \hat{\mu}_m) = OT(\hat{\nu}_x \otimes \hat{\mu}_m, \hat{\mu}_x \otimes \hat{\mu}_m) = \min_{\Gamma \in \Gamma_{n,n}} \langle P, C_m \rangle$, where $\nu_x \otimes \mu_m$ (resp. $\hat{\nu}_x \otimes \hat{\mu}_m$) stands for the product measure of $\nu_x$ (resp. $\hat{\nu}_x$) and $\mu_m$. The transportation plan matrix $P$ is from $\Gamma_{n,n} \overset{def}{=} \{ P \in \mathbb{R}^{n \times n} : P 1_n = \frac{1}{n} 1_n, P^T 1_n = \frac{1}{n} 1_n \}$. The masking cost matrix $C_m$ is defined as $C_m = (f(c(x, y) = ||x - y||_2^2)$ is the cost function. $\langle P, C_m \rangle = tr(P^T C_m)$ is the Frobenius dot-product of $P$ and $C_m$.

However, the MOT metric is still not differentiable [36]. There exists a costly linear program for computing the transport plan $P$. As a result, we introduce a masking regularized optimal transport metric, as stated in Definition 3. This metric becomes differentiable and programmable through an entropic regularization term [37].

Definition 3 (The masking regularized optimal transport, MROT): The MROT metric $OT^m(\hat{\nu}_x, \hat{\mu}_m)$ over $\hat{\nu}_x$ and $\hat{\mu}_m$ is defined as $OT^m(\hat{\nu}_x, \hat{\mu}_m) = \min_{\Gamma \in \Gamma_{n,n}} \langle P, C_m \rangle + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} \log p_{ij}$, where $\lambda$ is a hyper-parameter.

Due to the entropic regularizer $h(P)$, the resolution of the optimal transport plan $P^*$ can be tractable by using Sinkhorn’s algorithm [38], and thus the MROT metric becomes algorithmically differentiable and easily programmable. However, this entropic regularizer makes the MROT metric no longer positive. This may make the imputation models focus too much on learning the mean of the true underlying data distribution (i.e., ignoring the whole distribution). To get rid of this issue, we add corrective terms into the MS divergence to assure itself of positivity.

Definition 4 (The masking Sinkhorn divergence, MS divergence): The MS divergence $S_m(\hat{\nu}_x, \hat{\mu}_m)$ between the empirical measures $\hat{\nu}_x$ and $\hat{\mu}_x$ is defined as

$$S_m(\hat{\nu}_x, \hat{\mu}_x) = 2 OT^m(\hat{\nu}_x, \hat{\mu}_m) - OT^m(\hat{\nu}_x, \hat{\nu}_x) + OT^m(\hat{\mu}_x, \hat{\mu}_x).$$

Moreover, we instantiate the “vanishing” gradient problem and how the MS divergence handles this problem below. We also show the non-differentiable property of the JS divergence and differentiable property of the MS divergence. In this paper, we make the assumption that the data are missing completely at random (i.e., MCAR).

Example 1: For the imputation task defined in Definition 1, we consider $d = 1$ and $X_1 = \mathbb{R}$. Under MCAR mechanism, the mask vector $m \in \{0, 1\}$ is independent of the sample $x \in X_1$, i.e., $p_m(1|x) = p_m(1)$. Thus, a joint distribution of $x$ and $m$ can be defined as $p(x, m) = p_x(x)p_m(m)$. In particular, the true underlying and generated data distributions are defined as $p_x(x) = \delta_0$ and $p_x(x) = \delta_0$ with the parameter $\theta \in \mathbb{R}$, respectively. Besides, the missingness distribution $p_m(m)$ is supposed to follow a Bernoulli distribution $Ber(\theta)$ with a constant $\theta (0, 1)$. For simplicity, we denote $p_0$ and $p_0$ as the distributions of $p_x(x)p_m(m)$ and $p_x(x)p_m(m)$, respectively. Thus, the JS divergence between $p_0$ and $p_0$ is calculated by

$$JS(p_0, p_0) = \sum_{x \in \mathcal{X}} \left[ \int p_x(x) \log \frac{2p_x(x)q}{p_x(x)q + p_x(x)(1-q)} dx + \int p_x(x)(1-q) dx \right]$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[
\begin{align*}
\times \log \frac{2p_\lambda(x)(1-q)}{p^0_\lambda(x)(1-q) + p^0_\lambda(x)(1-q)}
= \begin{cases} 
0 & \text{if } \theta = 0 \\
2 \log 2 & \text{else}
\end{cases}
\end{align*}
\]

We can find that, \(JS(p_0||p_\theta)\) is not continuous at \(\theta = 0\), and thus non-differentiable. Moreover, the gradients of \(JS(p_0||p_\theta)\) are 0 almost everywhere. Thus, \(JS(p_0||p_\theta)\) provides useless gradient information to update the model parameter \(\theta\), which underlies the “vanishing” gradient problem. In contrast, the differentiable MS divergence between \(p_0\) and \(p_\theta\) is calculated by
\[
\begin{align*}
S_m(p_0||p_\theta) = 2OT^m_\lambda(p_0, p_\theta) - [OT^m_\lambda(p_0, p_\theta)] + OT^m_\lambda(p_\theta, p_0),
\end{align*}
\]
where \(OT^m_\lambda(p_0, p_\theta) = 0\). \(OT^m_\lambda(p_\theta, p_0)\) is calculated by \(OT^m_\lambda(p_\theta, p_0) = \lambda[(1-q) \log(1-q) + q \log q]\). Moreover, \(OT^m_\lambda(p_\theta, p_\theta)\) is defined as
\[
\begin{align*}
&OT^m_\lambda(p_\theta, p_\theta) = \min_{\gamma \in \Pi(p_\theta, p_\theta)} \mathbb{E}(y, y') - \gamma(y - y')^2 \\
&\quad + \lambda \log \gamma(y, y')
\end{align*}
\]
where \(\Pi(p_\theta, p_\theta)\) denotes the set of all joint distributions \(\gamma(y, y')\), whose marginals are \(p_\theta \) and \(p_\theta \), respectively. The second equality exploits the fact that the optimal joint distribution \(\gamma^*(y, y')\) is calculated by
\[
\begin{align*}
\gamma^*(y, y') = \begin{cases} 
1 - q & \text{if } y = 0 \text{ and } y' = 0 \\
q & \text{if } y = 0 \text{ and } y' = \theta \\
0 & \text{else}
\end{cases}
\end{align*}
\]

Thus, \(S_m(p_0||p_\theta) = 2q\theta^2 + \lambda[(1-q) \log(1-q) + q \log q]\). It is obvious that, \(S_m(p_0||p_\theta)\) is differentiable everywhere w.r.t \(\theta\). The gradients of \(S_m(p_0||p_\theta)\) vary linearly, which can always provide reliable gradients to update \(\theta\), and thus dispose of the “vanishing” gradient problem.

In general, the MS divergence based imputation loss function can be defined as \(L_\gamma(X, M) = \frac{1}{2\theta}S_m(\nu_\lambda||\tilde{\nu}_\lambda)\). By virtue of the differentiable MS divergence, the MS divergence based imputation loss function can provide a usable and reliable gradient during the GAN-based model training, and thus it helps to get rid of the “vanishing” gradient issue.

It is worth while to note that, the proposed MS divergence from SCIS is to minimize the MS divergence between the true underlying data distribution in the observed data and generated data distribution for GAN-based imputation models. Hence, with the support of the MS divergence, the true underlying data distribution can be well preserved in the imputed data from observed values. It is different from the Sinkhorn divergence used in the round-robin Sinkhorn imputation algorithm (RRSI) [11], where the Sinkhorn divergence between any two imputed batches is minimized for the MLP-based imputation model. In essence, the data distribution predicted by RRSI easily converges to a mixed distribution of the observed data and initial imputed data, rather than the true underlying one, especially when there exist a large amount of missing data. More intuitively, at the initial step,

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{The architecture of the DIM module.}
\end{figure}

RRSI fills the missing data by the mean value of the observed data for each incomplete feature separately. Thus, RRSI is likely to regard two samples with similar missing patterns as similar ones, while these two samples are actually different in the ground true space.

\section{B. Imputation Optimization With MS Divergence}

In DIM module, we improve a GAN-based imputation model as a differentiable one, by taking the MS divergence to measure the closeness between the true underlying data distribution in \(X_n\) and the generated data distribution in \(\hat{X}_n\). In pursuit of deriving the GAN-based imputation model to predict \(\hat{X}_n\) as similar to the observed data in \(X_n\) as possible, our goal becomes to find the optimized parameter \(\theta\) minimizing \(S_m(\nu_\lambda||\tilde{\nu}_\lambda)\) from a parameter space \(\Theta\). Formally, we rewrite \(S_m(\nu_\lambda||\tilde{\nu}_\lambda)\) as \(S_m(X_n \bigoplus M_n, X_n \bigoplus M_n)\). Therefore, the MS divergence based imputation loss minimizer is given by \(\theta = \arg \min_{\theta \in \Theta} \frac{1}{2n}S_m(X_n \bigoplus M_n, X_n \bigoplus M_n)\). By using the barycentric transport map [39] that obtains the gradient of the optimal transport metric w.r.t. each sample, the MS divergence based gradient function can be derived based on the chain rule.

\textbf{Proposition 1:} The MS divergence based gradient function \(g(\theta)\) can be calculated by
\[
g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[ \sum_{j=1}^{n} P_{ij}^* (\hat{x}_i \bigoplus m_i - x_j \bigoplus m_j) T(m_i) \right] \nabla \theta x_i,
\]
where \(\nabla \theta S_m(\cdot)\) and \(\nabla \theta X_n S_m(\cdot)\) are the derivatives of \(S_m(\cdot)\) with the parameter \(\theta\) and the reconstructed matrix \(X_n\), respectively; \(\nabla \theta X_n\) is the derivative of \(X_n\) with \(\theta\); \(P^* = (P_{ij}^*)_{ij}\) is the optimal transport plan; \(T(m_i)\) is to transform a mask vector \(m_i\) to a diagonal matrix.

\textbf{Proof:} First, by using the barycentric transport map [39], we obtain that \(\forall i \in 1, \ldots, n, \nabla \theta X_n OT^m_\lambda (X_n \bigoplus M_n, X_n \bigoplus M_n) = [\sum_{j=1}^{n} P_{ij}^* (\hat{x}_i \bigoplus m_i - x_j \bigoplus m_j) T(m_i)]\). Then, by using the chain rule, the gradient of \(L_\gamma(X_n, M_n)\) is calculated by
\[
g(\theta) = \left[ \frac{1}{2n} \nabla \theta X_n S_m(X_n \bigoplus M_n, X_n \bigoplus M_n) \right] \nabla \theta X
\]
\[
= \frac{1}{n} \sum_{j=1}^{n} \left[ \sum_{j=1}^{n} P_{ij}^* (\hat{x}_i \bigoplus m_i - x_j \bigoplus m_j) T(m_i) \right] \nabla \theta x_i.
\]

Consequently, as depicted in Fig. 1, the DIM module takes the data matrix \(X_n\) and its mask matrix \(M_n\) as inputs, and outputs the optimized GAN-based imputation model \(\hat{M}_n\) and the data matrix \(\hat{X}_n\) imputed by \(\hat{M}_n\). Using the MS divergence based gradient function, DIM employs a mini-batch gradient descent

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
technique [40] to solve the optimization problem for the differentiable GAN-based imputation model. To be more specific, similar as the studies [17], [41], the discriminator is trained to maximize the MS divergence between the true underlying and generated data distributions, while the generator is trained by minimizing the MS divergence metric evaluated by the newly updated discriminator. As a result, the DIM module employs the optimized GAN-based imputation model \( M_n \) to impute missing values in \( X_n \) via using (1).

V. SAMPLE SIZE ESTIMATION

With the support of the differentiable GAN-based imputation model in the DIM module, the goal of the sample size estimation (SSE) module is to analytically estimate the minimum sample size \( n_\star \) (with \( n_0 \leq n_\star \leq N \)) for training an approximate GAN-based model under accuracy guarantees, in contrast to using all samples in the whole dataset to train the model. With the SSE module, the imputation difference \( \text{D}(M_\star, M_N) \) between the model \( M_\star \) w.r.t. the size-\( n_\star \) (in)complete samples from \( X \) and the model \( M_N \) w.r.t. the given incomplete dataset \( X \) can be estimated and well-controlled without additional model training cost, i.e.,

\[
P(\text{D}(M_\star, M_N) \leq \varepsilon) \geq 1 - \alpha,
\]

where \( \varepsilon \) is a user-tolerated error bound of distance between imputed values produced by \( M_\star \) and \( M_N \); \( \alpha \) is a confidence level; \( m \) is the mask vector of \( X \); \( \hat{X} \) and \( \tilde{X} \) are the vectors reconstructed by \( M_\star \) and \( M_N \) over \( X \), respectively.

The core idea of SSE is to infer the distribution of the parameters \( \theta_\star \) in an unknown model \( M_\star \) w.r.t. size-\( n_\star \) (\( n_0 \leq n_\star \leq N \)) samples and the unknown model \( M_N \) for estimating the probability \( P(\text{D}(M_\star, M_N) \leq \varepsilon) \), also denoted \( P(\text{D}(\theta_\star, \theta_N) \leq \varepsilon) \). By doing so, we can further determine the minimum sample size \( n_\star \) that satisfies accuracy guarantees described in (2). In the following, i) we leverage a minimum distance estimator to infer the distribution of the parameters \( \theta_\star \) and \( \theta_N \) with the given parameter \( \theta_0 \) of the initial model \( M_0 \), as shown in Theorem 1; ii) we employ an empirical estimation to calculate the probability \( P(\text{D}(\theta_\star, \theta_N) \leq \varepsilon) \) via the distributions of \( \theta_\star \) and \( \theta_N \) estimated from Theorem 1, as shown in Proposition 2; and iii) with the estimation of \( P(\text{D}(\theta_\star, \theta_N) \leq \varepsilon) \), we use binary search to find the minimum size-\( n_\star \) samples to satisfy the users’ expectation on imputation accuracy in (2).

**Parameter Distribution Inference:** To begin with, inspired by the asymptotic normality of minimum distance estimator [42], we study the inference on the distribution of the parameter \( \theta_\star \) in an unknown differentiable GAN-based imputation model \( M_\star \) with a size-\( n_\star \) (\( n_0 \leq n_\star \leq N \)) (in)complete sample set from \( X \), by using the parameter \( \theta_0 \) of \( M_0 \), as stated in Theorem 1. For brevity, we make a notation that for two positive-value functions \( g \) and \( h \) over the same domain and upper bounded by a positive constant, \( g \sim h \) iff \( g/h \) and \( h/g \) are bounded by a constant. In particular, the MS divergence based gradient function makes Theorem 1 work for GAN-based imputation models, while the corresponding theorem in [43] is only tailored for traditional machine learning models.

**Theorem 1:** Let \( \hat{\theta}_n \) be the random variable, to characterize the parameter distribution of \( M_n \), of which \( \hat{\theta}_n \) is an instance. Given the parameter \( \theta_0 \) of a differentiable GAN-based imputation model \( M_\star \), \( \theta_\star \) follows

\[
\hat{\theta}_n|\theta_0 \rightarrow \mathcal{N}(\theta_0, \eta \mathbf{H}^{-1})
\]

with

\[
\eta \simeq e^{\frac{\eta_0}{2} + \frac{1}{\lambda} \left(1 + \frac{1}{\lambda^2} \right)^2 \left( \frac{1}{n_0} - \frac{1}{n} \right)},
\]

as \( n_0 \to \infty \) and \( n \to \infty \), where \( \lambda \) is a hyper-parameter in the MS divergence; \( \mathbf{H} \) is the invertible Hessian matrix of the MS divergence based imputation loss function with \( \theta_0; \mathcal{N}(\theta_0, \eta \mathbf{H}^{-1}) \) denotes a multivariate normal distribution with mean \( \theta_0 \) and covariance matrix \( \eta \mathbf{H}^{-1} \).

**Proof:** Let \( \theta_0 \) and \( \theta_N \) be the random variables to characterize the parameter distribution of \( M_0 \) and \( M_N \), respectively. To estimate the distribution of \( \theta_\star \), we first derive the distribution of \( \theta_\star \) by the multivariate central limit theorem, where \( \theta_\star \) is the conceptual optimal parameter when the sample size approaches infinity. In particular, the multivariate central limit theorem [43] depicts that the statistical behavior of model parameters will approximate a Gaussian distribution as the sample size becomes larger. Then, we infer the distribution of \( (\theta_0 - \theta_\star) \) based on that of \( (\theta_0 - \theta_{\infty}) \). Finally, we exploit Bayes’ theorem to estimate the distribution of \( \theta_\star \) from that of \( (\theta_0 - \theta_\star) \).

Specifically, since the MS divergence is differentiable, \( \theta_0 \) is obtained by finding the parameter at which \( g(\theta) \) becomes 0, i.e., \( \theta_0 \) satisfies \( g(\theta_0) = 0 \). According to the mean-value theorem, there exists \( \hat{\theta}_0 \) between \( \theta_0 \) and \( \theta_{\infty} \) that satisfies \( g'(\hat{\theta}_0) \cdot (\theta_0 - \theta_{\infty}) = -g(\theta_0) \). Therefore, we can find

\[
\hat{\theta}_0 - \theta_{\infty} = -g'(\theta_0)^{-1} \left( \sum_{j=1}^{n_0} \left[ \sum_{i=1}^{n_0} \mathbf{P}_{ij} (\hat{x}_i \odot m_i - x_j \odot m_j) \mathbf{T}(m_j) \nabla_{\theta_j} \hat{x}_i \right] \right)
\]

\[
\frac{n_0 \to \infty}{\mathcal{N}(0, \frac{1}{n_0} \zeta(\lambda) \mathbf{H}^{-1})},
\]

where \( \zeta(\lambda) \simeq e^{\frac{\lambda}{2} + \frac{1}{\lambda} \left(1 + \frac{1}{\lambda^2} \right)^2 \left( \frac{1}{n_0} - \frac{1}{n} \right)} \); \( \mathbf{J} \) and \( \mathbf{H} \) are the information matrix and Hessian matrix of the MS divergence based imputation loss function with \( \theta_0 \), respectively. The transition from (4) to (5) is based on the multidimensional central limit theorem and the fact that the sample complexity for Sinkhorn divergence is \( O(\frac{\log^2 \| \mathbf{X} \| + \| f_c \| \infty}{\kappa_\lambda}) \) [44], here, \( \| \cdot \| \) is the diameter \( \left\| \mathbf{x} - \mathbf{x}' \right\|_1 \) of \( \mathbf{x}' \). L is the Lipschitz constant for the cost function \( f_c \). In our case, \( \| \cdot \| \) and \( L \) will be \( \sqrt{d} \) and \( \sqrt{d} \) respectively. It is because that the input data will be normalized to \( \left[0, 1\right]^d \) and the cost function \( f_c \) is a squared two-norm function. Thus, \( \zeta(\lambda) \simeq e^{\frac{\lambda}{2} + \frac{1}{\lambda} \left(1 + \frac{1}{\lambda^2} \right)^2} \). Moreover, \( \mathbf{J} = -\mathbf{H} \) implies the last equation.

Then, with the estimated distribution of \( \theta_\star \), we further infer the distribution of \( (\hat{\theta}_0 - \theta_\star) \). To be more specific, since \( X_n \) can be regarded as a union of \( X_0 \) and \( (X_n - X_0) \),
we introduce two random variables $V_1$ and $V_2$ independently following $N(0, \zeta(\lambda)H^{-1})$ to capture the randomness of $X_0$ and $(X_n - X_0)$, respectively. Note that $\hat{\theta}_0 - \theta_\infty \to \frac{1}{\sqrt{n}}V_1$. In this way, we can find

$$\hat{\theta}_0 - \hat{\theta}_n = \frac{1}{\sqrt{n}}H^{-1}\left( \frac{1}{\sqrt{n}} \sum_{i=1}^{n_0} \left[ \sum_{i=1}^{n_0} P^*_i (\tilde{x}_i \odot m_i) - x_j \odot m_j \right] \mathcal{T}(m_i) \mathcal{V}_{\theta_n} \tilde{x}_j \right)$$

$$+ \frac{1}{\sqrt{n}}H^{-1}\left( \frac{1}{\sqrt{n}} \sum_{i=1}^{n_0} \left[ \sum_{i=1}^{n_0} P^*_i (\tilde{x}_i \odot m_i) - x_j \odot m_j \right] \mathcal{T}(m_i) \mathcal{V}_{\theta_n} \tilde{x}_j \right)$$

Moreover, we exploit the fact that $(\hat{\theta}_0 - \hat{\theta}_n)$, which is the sum of two independent normal random variables, asymptotically following a normal distribution. Thus, following [43], when both $n_0$ and $n$ approach infinity, $(\hat{\theta}_0 - \hat{\theta}_n)$ follows

$$\hat{\theta}_0 - \hat{\theta}_n \xrightarrow{n_0 \text{ and } n \to \infty} \left( \frac{1}{\sqrt{n_0}} - \frac{\sqrt{n_0}}{n} \right) V_1 - \frac{\sqrt{n} - n_0}{n} V_2$$

$$\sim \mathcal{N} \left( 0, \left( \frac{1}{n_0} - \frac{1}{n} \right) \cdot \zeta(\lambda) \cdot H^{-1} \right).$$

Finally, we exploit Bayes’ theorem to estimate the distribution of $\hat{\theta}_n|\theta_0$ based on the distribution of $(\hat{\theta}_0 - \hat{\theta}_n)$. Observe that $(\hat{\theta}_0 - \hat{\theta}_n)$ and $(\hat{\theta}_n - \theta_\infty)$ are independent because the covariance between them is zero, i.e.,

$$\text{Cov}(\hat{\theta}_0 - \theta_0, \hat{\theta}_n - \theta_\infty) = \frac{1}{2} \left( \text{Var}(\hat{\theta}_0 - \theta_\infty) - \text{Var}(\hat{\theta}_0 - \hat{\theta}_n) - \text{Var}(\hat{\theta}_n - \theta_\infty) \right)$$

$$= \frac{1}{2} \left( \frac{1}{n_0} - \left( \frac{1}{n_0} - \frac{1}{n} \right) \right) \cdot \zeta(\lambda) \cdot H^{-1}.$$
Therefore, according to Proposition 2, the probability $\Pr(\mathcal{D}(\theta_n, \theta_N) \leq \epsilon)$ can be approximated by

$$
\Pr(\mathcal{D}(\theta_n, \theta_N) \leq \epsilon) \approx \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}[\mathcal{D}(\theta_{n,i}, \theta_{N,i}) \leq \epsilon],
$$

where $(\theta_{n,i}, \theta_{N,i})$ is a pair sampled from the conditional probability distributions $\theta_{n,i}|\theta_n$ and $\theta_{N,i}|\theta_N$, respectively. Moreover, $\Pr(\mathcal{D}(\theta_n, \theta_N) \leq \epsilon)$ is an increasing function with $n$, which is derived in the similar way as [43].

**Minimum Size Search**: Based on Proposition 2, the SSE module uses binary search to find the minimum size $n_*$ (with $n_0 \leq n_* \leq N$) of samples in $X$ to satisfy the users’ expectation on imputation accuracy. The search procedure terminates, if $n_*$ satisfies (2) while $(n_* - 1)$ does not.

VI. EXPERIMENT

In this section, we evaluate the performance of our proposed system SCIS via comparisons with twelve state-of-the-art imputation methods. All algorithms were implemented in Python. The experiments were conducted on an Intel Core 2.80 GHz server with TITAN Xp 12GiB (GPU) and 192 GB RAM, running Ubuntu 18.04 system.

**Datasets**: In the experiments, we use six real-world incomplete datasets over several scenarios. In particular, COVID-19 trials tracker [46] (Triad) dataset shows the clinical trial registries on studies of COVID-19 and tracks the availability of the studies’ results. It contains 6,433 trials with 9 features and a 9.63% missing rate. Emergency declarations time-series [47] (Emerg) dataset is emergency declarations and mitigation policies for each US state starting on January 20, 2020. It contains 8,364 samples with 22 features and a 62.69% missing rate. Government response time-series [48] (Resp) dataset is a summary of a government’s response starting on January 01, 2020 [49]. It includes 200,737 samples with 19 features and a 5.66% missing rate. Symptom search trends-time-series [50] (Sear) dataset is a daily or weekly time series that shows how Google search patterns for 424 symptoms change based on the relative frequency of searches for each symptom in 2,792 specific regions. It contains 948,762 samples with an 81.35% missing rate. Daily weather time-series [51] (Weat) dataset shows the 9 weather attributes from the nearest station in the specific regions starting on March 30, 2020. It has 4,911,011 samples from 19,284 regions with a 21.56% missing rate. COVID-19 case surveillance public use [52] (Surv) dataset has 7 symptom features for 22,507,139 cases with a 47.62% missing rate.

**Metrics**: In the evaluation, we use the training time and root mean squared error (RMSE) [11], [12], [13] to measure the efficiency and effectiveness of imputation models. We also report the training sample rate $R_v$, i.e., how many samples are used for training models (100% for basic original ones and $\frac{n}{N} \times 100\%$ for SCIS). The smaller the metric value, the better the imputation performance. The original natural missing in the real-world incomplete datasets has no ground truth (i.e., don’t know the corresponding observed value). Hence, it cannot be used for the imputation task evaluation, while it is useful for the post-imputation task. To obtain the RMSE values, we randomly remove 20% observed values (only for imputation task evaluation), and thus we use these observed values as the ground-truth to the missing values. In evaluation, each value is reported by averaging five times of experimental results under different data random divisions. In addition, for each dataset, the numerical features are normalized via using the min-max normalization to prevent a few dimensions dominating the model training, and the categorical features are represented with one-hot encoding.

**Imputation Methods**: In the experiments, the baselines include fifteen state-of-the-art imputation methods, namely five machine learning ones: MissF [10], XGBI [23], Baran [24], MICE [7], and ICLR [26], two MLP-based ones: DataWig [29] and RRSI [11], five AE-based ones: MIDAE [30], VAEI [31], MIWAE [8], EDDI [32], and HIVAE [33], two GAN-based ones: GINN [14] and GAIN [13], and EDIT-GAIN [35] (a variant of GAIN improved by EDIT). We evaluate SCIS on top of seven generative imputation methods (i.e., AE-based ones and GAN-based ones).

**Implementation Details**: For each baseline, we adopt the default parameter settings from the papers or their source codes. For all machine learning imputation methods, the learning rate is 0.3, and the number of iterations is 100. The number of decision trees in MissFI and XGBI is 100. Baran employs AdaBoost as the prediction model. The imputation times in MICE are 20. For all deep learning imputation methods, the learning rate is 0.001, the dropout rate is 0.5, the training epoch is 100, and the batch size is 128. The ADAM algorithm is utilized to train networks. MIDAE is a 2-layer with 128 units per layer network. For VAEI, the encoder and decoder are fully connected networks with two hidden layers, each with 20 neurons per layer, and the latent space is 10-dimensional. HIVAE uses only one dense layer for all the parameters of the encoder and decoder, each with 10 neurons per layer. In GINN, the discriminator is a simple 3-layer feed-forward network trained 5 times for each optimization step of the generator. In GAIN, both generator and discriminator are modeled as 2-layer fully connected network. In EDIT, the user-specified hyper-parameter $\alpha$ is set to 100%. Moreover, in SCIS, the hyper-parameter $\lambda$ is 130, the confidence level $\alpha$ is 0.05, the hyper-parameter $\beta$ is 0.01, the number of parameter sampling $k$ in SSE is 20, the user-tolerated error bound $\epsilon$ is 0.001. The initial size $n_0$ is 500 for Tria, 500 for Emerg, 2,000 for Resp, 6,000 for Sear, 20,000 for Weat, and 20,000 for Surv. In addition, for all datasets, we randomly choose 10% data as the test data, $N_v$-size data as the validation data, and the rest as the training data. The validation size $N_v$ is equal to $n_0$.

A. Scalability Evaluation

Tables II and III report the performance of imputation methods over six real-world incomplete datasets. Some results are unavailable (represented by ‘-‘), since the corresponding methods are not able to finish within $10^5$ seconds.

One can observe that, SCIS takes less training time and smaller training sample rate $R_v$ than baselines, while it achieves competitive imputation accuracy (i.e., similar RMSE value). It adopts only 12.22% training samples and saves 31.64% training time.
TABLE II

| Method | Tria | Emer | Resp |
|--------|------|------|------|
|        | Time (s) | R_e (%) | RMSE (Bias) | Time (s) | R_e (%) | RMSE (Bias) | Time (s) | R_e (%) | RMSE (Bias) |
| MissF | 152 | 100 | 0.417 (± 0.023) | 782 | 100 | 0.377 (± 0.033) | - | - | - |
| XGBI | 167 | 100 | 0.408 (± 0.021) | 582 | 100 | 0.372 (± 0.032) | - | - | - |
| Baran | 1,204 | 100 | 0.412 (± 0.025) | - | - | - | - | - | - |
| MICE | 592 | 100 | 0.402 (± 0.021) | - | - | - | - | - | - |
| ICLR | 625 | 100 | 0.411 (± 0.024) | - | - | - | - | - | - |
| DataWig | 329 | 100 | 0.485 (± 0.042) | 3,252 | 100 | 0.389 (± 0.031) | - | - | - |
| RRSI | 413 | 100 | 0.398 (± 0.011) | 4,673 | 100 | 0.358 (± 0.025) | - | - | - |
| MIDA-E | 532 | 100 | 0.445 (± 0.025) | 2,321 | 100 | 0.378 (± 0.032) | - | - | - |
| SCIS-MIDA-E | 431 | 26.22 | 0.443 (± 0.029) | 2,093 | 14.29 | 0.376 (± 0.027) | 35,311 | 3.51 | 0.430 (± 0.020) |
| VAEI | 321 | 100 | 0.663 (± 0.023) | 892 | 100 | 0.598 (± 0.036) | - | - | - |
| SCIS-VAI | 298 | 18.97 | 0.451 (± 0.027) | 796 | 21.02 | 0.392 (± 0.021) | 32,160 | 3.21 | 0.434 (± 0.026) |
| SCIS-MIWA-E | 692 | 100 | 0.396 (± 0.015) | - | - | - | - | - | - |
| SCIS-MIWA | 739 | 17.98 | 0.390 (± 0.022) | 34,311 | 15.22 | 0.371 (± 0.023) | - | - | - |
| EDDI | 132 | 100 | 0.404 (± 0.018) | 662 | 100 | 0.363 (± 0.019) | - | - | - |
| SCIS-EDDI | 111 | 18.71 | 0.399 (± 0.024) | 560 | 14.82 | 0.359 (± 0.026) | 44,112 | 3.13 | 0.420 (± 0.020) |
| HVAE | 124 | 100 | 0.396 (± 0.020) | 450 | 100 | 0.357 (± 0.020) | 969 | 100 | 0.398 (± 0.018) |
| SCIS-HVAE | 113 | 17.62 | 0.393 (± 0.021) | 416 | 16.93 | 0.349 (± 0.022) | 726 | 3.35 | 0.394 (± 0.021) |
| GINN | 529 | 100 | 0.432 (± 0.030) | 1,572 | 100 | 0.373 (± 0.028) | - | - | - |
| SCIS-GINN | 309 | 16.76 | 0.432 (± 0.028) | 969 | 13.94 | 0.373 (± 0.024) | 7,224 | 2.48 | 0.410 (± 0.020) |
| GAIN | 90 | 100 | 0.396 (± 0.024) | 340 | 100 | 0.352 (± 0.025) | 649 | 100 | 0.396 (± 0.031) |
| EDIT-GAIN | 87 | 21.35 | 0.384 (± 0.020) | 302 | 12.42 | 0.341 (± 0.020) | 610 | 1.61 | 0.386 (± 0.025) |
| SCIS-GAIN | 85 | 23.56 | 0.386 (± 0.017) | 286 | 12.32 | 0.342 (± 0.019) | 595 | 1.50 | 0.389 (± 0.021) |

TABLE III

| Method | Sear | Weat | Surv |
|--------|------|------|------|
|        | Time (s) | R_e (%) | RMSE (Bias) | Time (s) | R_e (%) | RMSE (Bias) | Time (s) | R_e (%) | RMSE (Bias) |
| HVAE | - | - | - | 14,612 | 100 | 0.174 (± 0.014) | 35,041 | 100 | 0.440 (± 0.008) |
| SCIS-HVAE | 23,453 | 0.89 | 0.260 (± 0.018) | 3,893 | 2.10 | 0.169 (± 0.017) | 4,012 | 0.91 | 0.438 (± 0.008) |
| GINN | - | - | - | - | - | - | - | - | - |
| SCIS-GINN | 78,121 | 100 | 0.252 (± 0.014) | 9,252 | 100 | 0.165 (± 0.014) | 29,102 | 100 | 0.440 (± 0.012) |
| GAIN | 36,372 | 1.05 | 0.280 (± 0.012) | 4,172 | 1.89 | 0.151 (± 0.012) | 10,261 | 0.69 | 0.435 (± 0.018) |
| EDIT-GAIN | 3,580 | 0.76 | 0.252 (± 0.013) | 2,275 | 1.90 | 0.163 (± 0.016) | 3,019 | 0.67 | 0.438 (± 0.013) |

Moreover, the speedup of SCIS is not very obvious on the first three datasets in Table II, compared with the others in Table III. It is because that, the initial sample size \( n_0 \) for the first three datasets are significantly smaller than that of the other ones. The smaller \( n_0 \), the larger the variance derived by (3), resulting in higher training sample rate and training time. The training time (resp. training sample rate) even decreases to 4.52% (resp. 0.67%) on Sear (resp. Surv) for GAIN. It is attributed to the SSE module in SCIS that minimizes the required training sample size of GAN-based methods. In addition, the competitive (even better) accuracy with (than) original methods results from the MS divergence based imputation loss function employed in SCIS that measures the closeness between the true underlying data and generated data distributions. Also, the accuracy guarantee in SSE benefits the accuracy of SCIS. It even increases 3.02% accuracy for GAIN on Tria. In particular, the experimental results of SCIS-GINN are unavailable over Sear and Surv datasets, since GINN has a high complexity on construction of the similarity graph. For further extensive experimental study on SCIS, we employ GAIN as the baseline, since GAIN can work on these datasets, providing a clear comparison benchmark.

In addition, Fig. 2 depicts the missing value prediction time of GAIN and SCIS over six real-world datasets. The prediction time of them is comparable, since SCIS on top of GAIN does not modify the generator structure of GAIN, not producing additional prediction consumption.
B. Parameter Evaluation

Effect of \( R_m \): When varying the missing rate \( R_m \) (i.e., how many values in original observed data are dropped) from 10\% to 90\%, the corresponding results are depicted in Fig. 3. It also reports the time cost of the SSE module, which is the core module of SCIS. The reported SCIS training time has included the execution time of SSE. We can find that, compared with GAIN, SCIS takes much less training time and training samples to obtain a similar or even higher imputation accuracy in all cases. It is more robust with the increasing missing rate \( R_m \) than GAIN. The SSE module takes 28.31\% training time of SCIS on average. In addition, the imputation accuracy of both GAIN and SCIS is comparable, and it descends consistently with the growth of missing rate. The reason is that, as the missing rate increases, the observed information for algorithms becomes less, making imputation algorithms less effective.

Effect of \( \varepsilon \): With the user-tolerated error bound \( \varepsilon \) varying from 0.001 to 0.009, Fig. 4 plots the corresponding results. The sample rate of the initial training set \( X_0 \), i.e., \( R_1 = n_0 / N \) (and that of the minimum sample set \( X_\ast \), i.e., \( R_2 = n_\ast / N \)). The user-tolerated error is derived by \( R_{mse} + \varepsilon \), where \( R_{mse} \) is the RMSE value of GAIN with the MS divergence based imputation loss function trained on \( X \). In order to verify the effect of the DIM module on SCIS, we also report the imputation error \( R_{mse} + \varepsilon \) in the figure, where \( R_{mse} \) is the RMSE value of original GAIN trained on \( X \). Some results are unavailable in the figure, because the corresponding methods cannot finish within \( 10^3 \) seconds.

As inferred from the figure, SCIS has the higher imputation accuracy than the user-tolerated error \( R_{mse} + \varepsilon \) and the imputation error \( R_{mse} + \varepsilon \) in most cases. It means that, the SSE module can estimate an appropriate sample size to get as the good imputation accuracy as the users’ expectation, i.e., it indeed satisfies the accuracy requirement of users. Besides, the error \( R_{mse} \) derived by SCIS is smaller than \( R_{mse} \) derived by GAIN in many cases. It confirms that, the DIM module using the MS divergence does boost the imputation accuracy. Moreover, in most cases, the RMSE of SCIS increases with the growth of \( \varepsilon \), while \( R_2 \) is opposite. It is because, a smaller value of \( \varepsilon \) signifies a lower user-tolerated error. Hence, more samples (i.e., a larger \( R_2 \)) are needed to satisfy a lower RMSE requirement. In addition, when \( \varepsilon \) exceeds 0.005, the RMSE of SCIS changes slightly since \( n_\ast \) equals the lower bound \( n_0 \).

Effect of \( n_0 \): Fig. 5 depicts the experimental results of varying the initial sample size \( n_0 \). For SCIS with GAIN, different datasets require different optimal \( n_0 \). SCIS achieves the best imputation accuracy (in terms of RMSE) when the optimal \( n_0 \) is chosen, i.e., 500 for \( \text{Tria} \), 500 for \( \text{Emer} \), 2,000 for \( \text{Resp} \), 6,000 for \( \text{Sear} \), 20,000 for \( \text{Weat} \), and 20,000 for \( \text{Surv} \). Meanwhile, the time consumption and training sample rate remain reasonable and acceptable. In addition, the \( R_1 \) of SCIS increases with the decrease of \( n_0 \) in most cases. It is partially because that, the less the initial sample size \( n_0 \), the larger the variance derived by (3), leading to more training samples. It is worthwhile to note that, a smaller proportion of \( n_0 \) to the data scale (i.e., \( N \)) is required for a larger dataset. For example, for thousand-size datasets, the optimal \( n_0 \) exceeds 5\% of \( N \). For million-size datasets, the optimal \( n_0 \) can be set as 0.5\% of \( N \).

Effect of \( L_c \): When varying the segment length \( L_c \) of consecutive missing data over the three largest time-series datasets (i.e.,
Fig. 5. The performance of SCIS versus $n_0$.

Fig. 6. The performance in RMSE of SCIS versus $L_c$.

Fig. 7. Comparison under different missing mechanisms.

Resp, Sear, and Weat), the corresponding results (i.e., RMSE) are presented in Fig. 6. We randomly remove $L_c$-size consecutive values in each feature of the time-series datasets. We can find that the RMSE of SCIS increases with the growth of segment length $L_c$. It is because that, as $L_c$ turns large, the observed information for algorithms becomes less, making SCIS less effective. The maximum segment length that SCIS can effectively impute is 4,000 for Resp, 5,000 for Sear, and 30,000 for Weat. The RMSE of SCIS changes slightly when $L_c$ is smaller than the maximum segment length, while it changes obviously when $L_c$ exceeds the maximum segment length.

**Effect of Missing Mechanisms:** We vary the data missing mechanisms missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR), and the corresponding experimental results are plotted in Fig. 7. In particular, we follow the methodologies in [19], [53], [54] to generate missing values for simulating MCAR, MAR, and MNAR. We can observe that, the training time of imputation algorithms GAIN and SCIS is insensitive to different missing mechanisms for each dataset. The imputation algorithms in MNAR mechanism achieve the lower accuracy than that in the other two missing mechanisms. It partially attributes to the extremely biased simulation of MNAR. The performance of imputation algorithms in MAR is similar as that in MCAR. As expected, SCIS consistently takes much less training time to obtain a comparable imputation accuracy in each case. It costs 51.62%, 52.35%, and 54.28% of the training time required for GAIN on average under MCAR, MAR, and MNAR mechanisms, respectively.

**C. Ablation Study**

We investigate the influence of different modules of SCIS on the imputation performance. The corresponding experimental results of the RMSE, training time, and training sample rate $R_t$ are shown in Tables IV and V. DIM-GAIN is the variant of SCIS-GAIN without the SSE module over GAIN. Fixed-DIM-GAIN, a variant of DIM-GAIN, randomly selects ten percentage of samples as the training data to accelerate the model training process. In Table V, the results of DIM-GAIN are unavailable (represented by “−”), since they are not able to finish within $10^5$ seconds.

We can observe that, DIM-GAIN gains better imputation accuracy (i.e., smaller RMSE value) than GAIN, while requires higher training time. It is because that, the DIM module can provide reliable gradients for GAIN to alleviate the “vanishing” gradient problem and thus achieve a better imputation performance. It confirms the effectiveness of the MS divergence based imputation loss function in the DIM module. Besides, compared with DIM-GAIN and Fixed-DIM-GAIN, SCIS-GAIN takes significantly less training time and training samples, while shows a negligible decrease in imputation accuracy, especially on the last three incomplete datasets. Specifically, DIM-GAIN increases 3.24% accuracy for GAIN on average, but its training time is 4.68x averagely of GAIN. It confirms the effectiveness of the MS divergence based imputation loss function in the DIM module. SCIS-GAIN takes 6.79% (resp. 67.88%) training samples and saves 72.29% (resp. 20.27%) training time of DIM-GAIN (resp. Fixed-DIM-GAIN) that use the same imputation model (i.e., DIM-GAIN), the more training samples (i.e., higher training sample rate $R_t$) the imputation method requires, the higher imputation accuracy it achieves. It is because that, the more the training samples, the
higher the available information for imputation models, making models more powerful.

Moreover, Fig. 8 shows the $l_2$-norm on the gradient of generator (in GAIN and DIM-GAIN) and the RMSE values during 200 training epochs over two generated 2-dimensional datasets [11], i.e., a spiral curve and concentric circles. We randomly remove 20% observed values in each dataset. One can realize that, when the epoch exceeds 100, the $l_2$-norm on the gradient of generator in GAIN converges to 0, resulting in the stable but high RMSE of GAIN. It is because that, GAIN suffers from the “vanishing” gradient problem, which prevents its generator parameter from updating values. In contrast, when the epoch exceeds 100, the generator of DIM-GAIN still has stable and non-zero gradients and a steady performance improvement. Besides, DIM-GAIN consistently gets better imputation accuracy than GAIN. It further confirms that, the DIM module can get rid of the “vanishing” gradient problem in GAIN, and it does boost the imputation accuracy.

### D. Evaluation on Post-Imputation Prediction

In the last set of experiments, we verify the superiority of SCIS over GAIN on the post-imputation prediction task.

For the classification task over Tria and Surv and the regression task over Emer, Resp, Sear, and Weat, the corresponding prediction results are depicted in Table VI. The larger AUC value corresponds to the better prediction effect, while MAE is opposite. The imputation methods are first employed to impute missing values in the incomplete datasets. Then, a regression/classification model is trained with three fully-connected layers over the imputed data. The training epoch is 30, the learning rate is 0.005, the dropout rate is 0.5, and the batch size is 128.

We can observe that, the prediction performance under different imputation algorithms is consistent with the imputation performance of these algorithms, i.e., SCIS-GAIN gains competitive (even better) accuracy with (than) GAIN. Specifically, SCIS-GAIN decreases 0.51% MAE for GAIN on the regression task, while increases 0.27% AUC for GAIN on the classification task. In addition, on the regression (resp. classification) task, SCIS-GAIN achieves the larger improvement over the Weat (resp. Resp) dataset. Thus, it further confirms the effectiveness of SCIS.

### VII. Conclusion

In this paper, we propose an effective scalable imputation system SCIS to accelerate GAN-based imputation models. It consists of a DIM module and an SSE module. DIM makes the generative adversarial imputation models differentiable via leveraging a new MS divergence. SSE estimates the minimum training sample size for the differentiable imputation model, so that the final trained model satisfies a user-tolerated error bound. Extensive experiments over several real-world datasets demonstrate that, SCIS significantly accelerates the model training and meanwhile harvests the competitive imputation accuracy with the state-of-the-art GAN-based methods. In the future, we intend to enable SCIS to accelerate the time series generative imputation models on large-scale time series data.
REFERENCES

[1] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao, and J. Yin, “Generative semi-supervised learning for multivariate time series imputation,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 5983–5991.

[2] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, “HoloClean: Holistic data repairs with probabilistic inference,” in Proc. VLDB Endowment, vol. 10, no. 11, pp. 1190–1201, 2017.

[3] Y. Wu, X. Miao, X. Huang, and J. Yin, “Jointly imputing multi-view data with optimal transport,” in Proc. AAAI Conf. Artif. Intell., 2023, pp. 4747–4755.

[4] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Yin, and Y. Zheng, “Missing value imputation for multi-view urban statistical data via spatial correlation learning,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 1, pp. 686–698, Jan. 2023.

[5] F. Biessmann, D. Salinas, S. Schelter, P. Schmidt, and D. Lange, “Deep learning for missing value imputation in tables with non-numerical data,” in Proc. ACM Int. Conf. Knowl. Manage., 2018, pp. 2017–2025.

[6] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, “BRITS: Bidirectional recurrent imputation for time series,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 6775–6785.

[7] P. Royston and I. R. White, “Multiple imputation by chained equations (MICE): Implementation in stata,” J. Stat. Softw., vol. 45, no. 4, pp. 1–20, 2011.

[8] P.-A. Mattei and J. Frellsen, “MIWAE: Deep generative modelling and imputation of incomplete data sets,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 4413–4423.

[9] A. Farhangfar, L. A. Kurgan, and W. Pedrycz, “A novel framework for imputation of missing values in databases,” IEEE Trans. Syst., Man, Cybern. -Part A: Syst. Hum., vol. 37, no. 5, pp. 692–709, Sep. 2007.

[10] D. J. Stckehoven and P. Buhlmann, “MissForest non parametric missing value imputation for mixed-type data,” Bioinformatics, vol. 28, no. 1, pp. 112–118, 2011.

[11] M. Boris, J. Julie, B. Claire, and C. Marco, “Missing data imputation using optimal transport,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1–18.

[12] N. B. Ipsen, P.-A. Mattei, and J. Frellsen, “not-MIWAE: Deep generative modelling with missing not at random data,” 2020, arXiv:2006.12871.

[13] J. Yoon, J. Jordon, and M. van der Schaar, “GAIN: Missing value imputation with influence functions,” in Proc. VLDB Endowment, vol. 13, no. 3, pp. 624–632, 2020.

[14] G. Peyrè et al., “Computational optimal transport: With applications to data science,” Found. Trends Mach. Learn., vol. 11, no. 5/6, pp. 355–607, 2019.

[15] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, “Stochastic optimization for large-scale optimal transport,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2016, pp. 3440–3448.

[16] S. Ruder, “An overview of gradient descent optimization algorithms,” 2016, arXiv:1609.04747.

[17] G. Du, L. Zhou, Y. Yang, K. Lü, and L. Wang, “Deep multiple encoder-based multi-view clustering,” Data Sci. Eng., vol. 6, no. 3, pp. 323–338, 2021.

[18] F. Biessmann et al., “Datawig: Missing value imputation for tables,” J. Mach. Learn. Res., vol. 20, pp. 175:1–175:6, 2019.

[19] L. Gondara and K. Wang, “Multiple imputation using deep denoising autoencoders,” 2017, arXiv:1705.02737.

[20] J. T. McCoy, S. Kroon, and L. Auret, “Variational autoencoders for missing data imputation with application to a simulated milling circuit,” IFAC-PapersOnLine, vol. 51, no. 21, pp. 141–146, 2018.

[21] C. Ma, S. Tsiatsiashek, K. Palla, J. M. Hernández-Lobato, S. Nowozin, and C. Zhang, “EDDI: Efficient dynamic discovery of high-value information with partial VAE,” 2018, arXiv:1809.11142.

[22] A. Nazabal, P. M. Olmos, Z. Ghabramani, and I. Valera, “Handling incomplete heterogeneous data using VAEs,” 2018, arXiv:1807.03563.

[23] D. Jarrett, B. C. Cebere, T. Liu, A. Curth, and M. van der Schaar, “Hyperimpute: Generalized iterative imputation with automatic model selection,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 9916–9937.

[24] Y. Miao, Y. Wu, L. Chen, Y. Gao, J. Wang, and J. Yin, “Efficient and effective data imputation with influence functions,” in Proc. VLDB Endowment, vol. 15, no. 3, pp. 624–632, 2021.

[25] J. M. Hernández-Lobato, S. Nowozin, and C. Zhang, “EDDI: Efficient dynamic discovery of high-value information with partial VAE,” 2018, arXiv:1809.11142.

[26] A. Farhangfar, L. A. Kurgan, and W. Pedrycz, “A novel framework for imputation of missing values in databases,” IEEE Trans. Syst., Man, Cybern. -Part A: Syst. Hum., vol. 37, no. 5, pp. 692–709, Sep. 2007.

[27] D. J. Stckehoven and P. Buhlmann, “MissForest non parametric missing value imputation for mixed-type data,” Bioinformatics, vol. 28, no. 1, pp. 112–118, 2011.

[28] M. Boris, J. Julie, B. Claire, and C. Marco, “Missing data imputation using optimal transport,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1–18.

[29] N. B. Ipsen, P.-A. Mattei, and J. Frellsen, “not-MIWAE: Deep generative modelling with missing not at random data,” 2020, arXiv:2006.12871.

[30] J. Yoon, J. Jordon, and M. van der Schaar, “GAIN: Missing data imputation using generative adversarial nets,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 5675–5684.

[31] I. Spinelli, S. Scardapane, and A. Uncini, “Missing data imputation with adversarially-trained graph convolutional networks,” 2019, arXiv:1905.01907.

[32] O. Wältin, et al., “COVID-19 open-data: Curating a fine-grained, global-scale data repository for SARS-CoV-2,” 2020. [Online]. Available: https://github.com/covid-19-open-data.

[33] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” 2017, arXiv:1701.04862.

[34] M. G. Bellemare et al., “The cramner distance as a solution to biased Wasserstein gradients,” 2017, arXiv:1705.10743.

[35] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.

[36] X. Miao, Y. Wu, L. Chen, Y. Gao, and J. Yin, “An experimental survey of missing data imputation algorithms,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 7, pp. 6630–6650, Jul. 2023.

[37] N. S. Altman, “An introduction to kernel and nearest -neighbor nonparametric regression,” Amer. Statistician, vol. 46, no. 3, pp. 175–185, 1992.

[38] Y. Yang, J. Kim, and I.-H. Cho, “Parallel fractional hot deck imputation and variance estimation for big incomplete data curing,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 8, pp. 3912–3926, Aug. 2022.

[39] Y. Zhang, H. Zhang, Z. He, Y. Jing, K. Zhang, and X. S. Wang, “Parrot: A progressive analysis system on large text collections,” Data Sci. Eng., vol. 6, no. 1, pp. 1–19, 2021.

[40] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794.

[41] M. Mahdavi and Z. Abedjan, “Baran: Effective error correction via a unified context representation and transfer learning,” in Proc. VLDB Endowment, vol. 13, no. 12, pp. 1948–1961, 2020.

[42] A. Zhang, S. Song, Y. Sun, and J. Wang, “Learning individual models for imputation,” in Proc. IEEE Int. Conf. Data Eng., 2019, pp. 160–171.

[43] N. Karmita, S. Taheri, A. Bagirov, and P. Makinen, “Missing value imputation via clusteringwise linear regression,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 4, pp. 1889–1901, Apr. 2022.
Yangyang Wu received the master’s degree in information engineering from the Zhejiang University of Technology, China, in 2019. He is currently working toward the PhD degree with the Center for Data Science, Zhejiang University, China. His research interests include data imputation and data cleaning.

Jun Wang received the bachelor’s degree in information and computing science from Zhejiang University, China, in 2020. He is currently working toward the PhD degree with the Hong Kong University of Science and Technology, China. His research interests include machine learning algorithms for incomplete dataset and non-parametric statistics.

Xiaoye Miao received the PhD degree in computer science from Zhejiang University, China, in 2017. She is currently an assistant professor with the Center for Data Science, Zhejiang University, China. She was ever a research/postdoctoral fellow with the University of New South Wales and City University of Hong Kong, respectively. Her research interests include data services, data pricing, data cleaning, and graph management.

Wenjia Wang received the PhD degree in operations research from the Georgia Institute of Technology, Atlanta, USA, in 2018. Currently, he is an assistant professor with the Hong Kong University of Science and Technology. Prior to this, he was a postdoctoral fellow with the Statistical and Applied Mathematical Sciences Institute, and Duke University. His research interests focus on nonparametric methods with applications to computer experiments and machine learning.

Jianwei Yin received the PhD degree in computer science from Zhejiang University, in 2001. He is currently a professor with the College of Computer Science, Zhejiang University. He is a visiting scholar with the Georgia Institute of Technology, in 2008. His research interests include service computing and data management.