Jointly Imputing Multi-View Data with Optimal Transport

Yangyang Wu[1], **Xiaoye Miao**[1*], **Xinyu Huang**[3], **Jianwei Yin**[1,2]

[1] Center for Data Science, Zhejiang University, Hangzhou, China
[2] College of Computer Science, Zhejiang University, Hangzhou, China
[3] Data Science Institute, Columbia University, New York, USA
{zjuwuyy, miaoxy}@zju.edu.cn, xh2511@columbia.edu, zjuyjw@cs.zju.edu.cn

Abstract

The multi-view data with incomplete information hinder the effective data analysis. Existing multi-view imputation methods that learn the mapping between complete view and *completely missing* view are not able to deal with the common multi-view data with *missing feature* information. In this paper, we propose a generative imputation model named Git with optimal transport theory to *jointly* impute the missing features/values, conditional on *all* observed values from the multi-view data. Git consists of two modules, i.e., a *multi-view joint generator* (MJG) and a *masking energy discriminator* (MED). The generator MJG incorporates a *joint autoencoder* with the *multiple imputation rule* to learn the data distribution from all observed multi-view data. The discriminator MED leverages a new *masking energy* divergence function to make Git differentiable for imputation enhancement. Extensive experiments on several real-world multi-view data sets demonstrate that, Git yields over 35% accuracy gain, compared to the state-of-the-art approaches.

Introduction

Multi-view data (Guo 2013) are captured from different modalities, sources, spaces, and other forms. It has become one of the main data types in many real-life scenarios, such as video surveillance, entertainment media, medical detection, etc. Due to various reasons like the collection device failure, instable system environment, or privacy concerns (Miao et al. 2022a, 2021; Wu et al. 2022), it is common that a fraction of features/values in some data views are not collected, resulting in the missingness of multi-view data. For example, the public medical dataset *PhysioNet* (Silva et al. 2012) collects patients' physiological signals and measurements from multiple views, including respiration, blood pressure, and electrocardiograms. It takes above 80% average missing rate, making it difficult to analyze. Hence, such feature missing problem poses a daunting challenge to the multi-view data analysis (Yan et al. 2021; Miao et al. 2022b).

Existing multi-view imputation studies (Farhangfar, Kurgan, and Pedrycz 2007; Jaques et al. 2017; Tran et al. 2017; Yoon, Jordon, and Schaar 2018; Spinelli, Scardapane, and Uncini 2019) focus on the *view-level* missing problem, i.e., a special case of the *feature-level* missing problem when *all* features in the target-view are completely missing. They learn the mapping between the source-view and target-view in the multi-view data to predict the missing target-view conditional on the given source-view. These multi-view imputation algorithms are *neither efficient nor effective*. For imputing the target-view, one has to train a group of mapping models. The number of possible source-views determines how many models should be learnt. Particularly, four mapping models are built for multi-view imputation over the karolinska directed emotional faces dataset *KDEF* (Goeleven et al. 2008) with five views. Moreover, the real-life multi-view data generally lose a proportion of features in each view in practice. Existing methods do not consider the feature missing state/information in each view. The corresponding *root mean squared error* for missing feature imputation is even higher than 30 on some multi-view datasets. Therefore, it is not proper, even infeasible, to apply existing multi-view imputation approaches for dealing with the multi-view data in real-life scenarios with *arbitrarily missing features*.

In this paper, we propose a novel generative multi-view imputation model, termed as Git, with the support of optimal transport theory. Git is composed of two modules, a *multi-view joint generator* (MJG) and a *masking energy discriminator* (MED). The generator MJG leverages a *multiple imputation* based autoencoder to learn a mapping between all observed multi-view data and target-view data, so as to jointly and efficiently impute missing features in the multi-view data. However, the typical Jensen-Shannon divergence used in the generative imputation model (Arjovsky and Bottou 2017) is not continuous and non-differentiable, resulting in the "vanishing" gradient problem. In view of this, we develop a *masking energy* (ME) divergence in MED to enable both modules in Git to be differentiable. It thus can always provide reliable gradients to *avoid* the "vanishing" gradient problem for better imputation ability. In summary, the main contributions of this paper are described as follows.

- We propose a generative multi-view imputation model Git that is able to efficiently and effectively predict missing features in the multi-view data. To the best of our knowledge, this is the first proposal to address the missing feature problem within multi-view data.
- The MJG module in Git estimates the missing features of the target-view data conditional on all observed multi-

*Corresponding author: miaoxy@zju.edu.cn

view data via leveraging a joint autoencoder with the multiple imputation rule.

- In the MED module, we put forward an ME divergence to measure the closeness between the true underlying and generated data distributions of the target-view data. It employs the optimal transport theory to make Git differentiable, avoiding the "vanishing" gradient issue.
- Extensive experiments using several real-life multi-view data sets demonstrate that, Git substantially outperforms the state-of-the-art methods.

Related Work

There are a series of data imputation methods proposed to impute missing values for *single-view* data, such as MissForest (Stekhoven and Bühlmann 2011), MICE (Royston and White 2011), RRSI (Boris et al. 2020), not-MIWAE (Ipsen, Mattei, and Frellsen 2020), GAIN (Yoon, Jordon, and Schaar 2018), etc. However, these methods have limited ability to impute multi-view data, since they focus on the target-view data, ignoring the information from other views.

For multi-view data, existing multi-view imputation methods adopt traditional machine learning models (Farhangfar, Kurgan, and Pedrycz 2007), or autoencoder model (Jaques et al. 2017; Tran et al. 2017), e.g., the multi-view denoising autoencoder (Jaques et al. 2017), cascaded residual autoencoder (CRA) (Tran et al. 2017), and multi-view information bottleneck (MIB) (Federici et al. 2020), or generative adversarial network (GAN) (Yoon, Jordon, and Schaar 2018; Spinelli, Scardapane, and Uncini 2019) to imputing missing views. For example, based on the typical linear regression model and k nearest neighbor model, the missing view can be imputed by the synthetic-CT-based multi-view registration method (Zheng et al. 2019) and the isomorphic linear correlation analysis method (Zhang et al. 2018).

Particularly, among multi-view imputation algorithms with GAN model, the view imputation generative adversarial network (VIGAN) (Shang et al. 2017) uses a GAN model to identify view-to-view mappings and employs a denoising autoencoder to reconstruct the missing view. The encoder-decoder multi-view generative adversarial network (EMGAN) (Cai et al. 2018) utilizes a 3D encoder-decoder network to capture the relationship between the source-view and target-view. It uses an additional adversarial training strategy to generate high-quality data. The conditional autoencoder generative adversarial network (CAE-GAN) (Yang, Qian, and Fan 2020) incorporates the variational autoencoder and GAN under a conditional process to tackle the multi-view imputation problem.

However, the aforementioned multi-view imputation methods only focus on the view-level missing problem. It is inappropriate, even infeasible, for them to handle the feature-level missing problem within multi-view data, since they ignore the feature-level missing state in each view. In contrast, our proposed imputation model Git fully utilizes all observed data information of multi-view data. It further leverages the optimal transport theory to make it differentiable to solve the "vanishing" gradient issue.

Solution Overview

Let the multi-view dataset with the sample size n and view size v be stored in a matrix $\mathbf{X} = \{\mathbf{X}^1; \cdots; \mathbf{X}^v\} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$. The i-th multi-view data sample $\mathbf{x}_i = (\mathbf{x}_i^1; \cdots; \mathbf{x}_i^v)$ is in the form $\mathbf{x}_i = (x_{i1}^1, \cdots, x_{id_1}^1; \cdots; x_{i1}^v, \cdots, x_{id_v}^v)$. To encode the missing information, we define a mask matrix $\mathbf{M} = \{\mathbf{M}^1; \cdots; \mathbf{M}^v\} = \{\mathbf{m}_1, \cdots, \mathbf{m}_n\}$, where each mask vector $\mathbf{m}_i = (\mathbf{m}_i^1; \cdots; \mathbf{m}_i^v) = (m_{i1}^1, \cdots, m_{id_1}^1; \cdots; m_{i1}^v, \cdots, m_{id_v}^v)$ corresponds to the data sample $\mathbf{x}_i$. In particular, m_{ij}^k takes value in $\{0, 1\}$, $k = 1, \cdots, v$, $i = 1, \cdots, n$, and $j = 1, \cdots, d_v$; $m_{ij}^k = 1$ if the j-th dimensional value of the k-th view in $\mathbf{x}_i$ is observed, otherwise $m_{ij}^k = 0$.

Definition 1. Multi-view data imputation. *Given an incomplete multi-view dataset $\mathbf{X}$ with the mask matrix $\mathbf{M}$, the multi-view data imputation problem aims to build an imputation model $\mathcal{M}$ to find appropriate values for missing components in the target-view data $\mathbf{X}^t$ with the mask matrix $\mathbf{M}^t$, so as to (i) make the imputed data $\hat{\mathbf{X}}^t$ as* close *to the real* complete *target-view data (if it exists) as possible, or (ii) help downstream multi-view analysis to achieve better performance when adopting $\hat{\mathbf{X}}^t$ than that only with $\mathbf{X}^t$.*

Our proposed multi-view imputation model Git is mainly composed of a *multi-view joint generator* (MJG) module and a *masking energy discriminator* (MED) module. It builds an adversarial training platform with the generator MJG and discriminator MED for the multi-view imputation task. MJG jointly integrates all the observed values in the multi-view data to impute the missing features via using a joint autoencoder with the multiple imputation rule. MED is inspired by the outstanding performance of energy divergence (Zhao, Mathieu, and LeCun 2017) that assumes that the distributions of any two mini-batches extracted randomly from true underlying data and that of the two corresponding mini-batches from generated data should be close. Hence, it designs a masking energy (ME) divergence to make Git differentiable, for avoiding the "vanishing" gradient problem. Similar as the typical GAN, the generator MJG produces the data values as close to the true underlying (observed) target-view data distribution as possible, while the discriminator MED distinguishes the difference between the generated and true underlying target-view data as correctly as possible.

The objective function of Git is defined as a minimax problem with the ME divergence over MJG and MED. As a result, it employs the optimized generator MJG to impute missing components in the target-view data $\mathbf{X}^t$. Namely, the *imputed* matrix is calculated by $\hat{\mathbf{X}}^t = \mathbf{M}^t \odot \mathbf{X}^t + (1 - \mathbf{M}^t) \odot \bar{\mathbf{X}}^t$, where $\odot$ is the element-wise multiplication. $\bar{\mathbf{X}}^t$ is the reconstructed matrix predicted by the imputation model $\mathcal{M}$.

Algorithm 1 gives the pseudo-code of Git. It takes an incomplete multi-view dataset $\mathbf{X}$ with its mask matrix $\mathbf{M}$ and the target-view $\mathbf{X}^t$ as inputs. It outputs the optimized MJG G^* and the imputed target-view data $\hat{\mathbf{X}}^t$. Git starts to solve the minimax optimization problem in an iterative manner. During each training iteration, Git first samples two independent mini-batches $\mathbf{P}$ and $\mathbf{Q}$ from $\mathbf{X}$. It then updates the MJG module G with the fixed MED module D (lines 3-7).

Algorithm 1: The procedure of Git

Input: the incomplete multi-view dataset $\mathbf{X}$ with mask matrix $\mathbf{M}$, the target-view $\mathbf{X}^t$, training times C, and MED's iteration times K

Output: the optimized G^* and the imputed $\hat{\mathbf{X}}^t$

1: **for** *epoch* $= 1$ to C **do**
2: sample two mini-batches $\mathbf{P}$ and $\mathbf{Q}$ (w.r.t. mask matrices $\mathbf{M}^p$ and $\mathbf{M}^q$) from $\mathbf{X}$ (w.r.t. $\mathbf{M}$)
/* **MJG module updating** */
3: $\bar{\mathbf{P}}^t \leftarrow G(\mathbf{P}, \mathbf{M}^p, \mathbf{X}^t)$, $\bar{\mathbf{Q}}^t \leftarrow G(\mathbf{Q}, \mathbf{M}^q, \mathbf{X}^t)$
4: calculate the weighted reconstruction losses ($\mathcal{L}_r(\mathbf{P}^t)$ and $\mathcal{L}_r(\mathbf{Q}^t)$) and ME divergence loss $\mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t)$
5: calculate the total loss of the generator G, i.e., $\mathcal{L}_G \leftarrow \mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t) + \alpha \cdot \left[\mathcal{L}_r(\mathbf{P}^t) + \mathcal{L}_r(\mathbf{Q}^t)\right]$
6: update θ_G in G with $\mathcal{L}_G$
/* **MED module updating** */
7: **for** $k = 1$ to K **do**
8: $\bar{\mathbf{P}}^t \leftarrow G(\mathbf{P}, \mathbf{M}^p, \mathbf{X}^t)$, $\bar{\mathbf{Q}}^t \leftarrow G(\mathbf{Q}, \mathbf{M}^q, \mathbf{X}^t)$
9: calculate the MED module's loss $\mathcal{L}_D$ over $\mathbf{P}^t$ and $\mathbf{Q}^t$ (with $\bar{\mathbf{P}}^t$ and $\bar{\mathbf{Q}}^t$)
10: update θ_D in D with $\mathcal{L}_D$
11: estimate missing values in $\mathbf{X}^t$ by G^* with optimized θ_G to obtain the imputed target-view data $\hat{\mathbf{X}}^t$
12: **return** G^* and $\hat{\mathbf{X}}^t$

To be more specific, G produces the reconstructed target-view data $\bar{\mathbf{P}}^t$ and $\bar{\mathbf{Q}}^t$ for $\mathbf{P}$ and $\mathbf{Q}$ (line 3), respectively. On top of $\bar{\mathbf{P}}^t$ and $\bar{\mathbf{Q}}^t$, G calculates the total generative loss $\mathcal{L}_G$ (lines 4-6), which consists of the weighted reconstruction losses, i.e., $\mathcal{L}_r(\mathbf{P}^t)$ and $\mathcal{L}_r(\mathbf{Q}^t)$, and ME divergence adversarial loss, i.e., $\mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t)$. G updates its model parameters θ_G over $\mathcal{L}_G$ via using the gradient descent (line 7).

In the following, Git updates the MED module D with the updated MJG module G (lines 8-11). To be more specific, D first predicts the reconstructed target-view data $\bar{\mathbf{P}}^t$ and $\bar{\mathbf{Q}}^t$ via using G (line 9). It then calculates D's loss $\mathcal{L}_D$ (line 10), i.e., the ME divergence loss, and updates its model parameters θ_D over $\mathcal{L}_D$ via using the gradient descent (line 11). In particular, such optimization process of D is repeated K times. Eventually, Git returns the optimized G^* and the target-view data $\hat{\mathbf{X}}^t$ imputed by G^* (line 15).

The Multi-View Joint Generator

The *multi-view joint generator* (MJG) module of Git attempts to impute the missing features of the target-view data as accurately as possible. Its essential idea is to adopt a joint autoencoder with the multiple imputation rule to aggregate reconstruction (imputation) results learned from all observed components of multi-view data.

The basic architecture of MJG module G is depicted in Figure 1. In particular, the MJG module G takes the incomplete multi-view dataset $\mathbf{X}$ as input, and it produces a series of reconstructed target-view data matrices $\{\bar{\mathbf{X}}^{1t}; \cdots; \bar{\mathbf{X}}^{vt}\}$

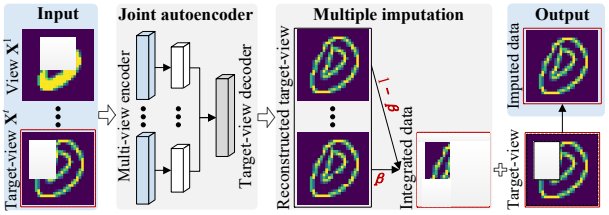


Figure 1: The architecture of MJG module

with respect to v views in $\mathbf{X}$. Following Rubin's multiple imputation rule (Little and Rubin 2019), the imputed target-view data $\hat{\mathbf{X}}^t$ can be finally calculated by integrating these reconstructed data $\{\bar{\mathbf{X}}^{1t}; \cdots; \bar{\mathbf{X}}^{vt}\}$. Formally, $\hat{\mathbf{X}}^t = \mathbf{M}^t \odot \mathbf{X}^t + (1 - \mathbf{M}^t) \odot [(1 - \beta) \cdot \sum_{s=1:v}^{s \neq t} \bar{\mathbf{X}}^{st} + \beta \cdot \bar{\mathbf{X}}^{tt}]$, where β is a weighted hyper-parameter. It is used to control the importance degree of the reconstructed matrix of the target-view (w.r.t. $\bar{\mathbf{X}}^{tt}$) in contrast with other views (w.r.t. $\bar{\mathbf{X}}^{st}, s \neq t$) for target-view imputation.

In the MJG module, we develop a joint autoencoder with a multi-view encoder and a target-view decoder to learn the mapping between the observed components of the multi-view data and the target-view data. To be more specific, the multi-view encoder embeds different types of views with different neural networks, e.g., convolutional neural network for image processing. On top of the embedded attributes, the target-view decoder then leverages a neural network (e.g., transpose convolutional network for image processing) to map the embedded attributes into the feature space of target-view, so as to obtain the reconstructed target-view data matrix $\bar{\mathbf{X}}^{st}$ w.r.t. the observed data in the s-th view data, $s = 1, \cdots, v$. To minimize model parameters and speed up training, we share the parameters of neural networks w.r.t. the same type of source-views in the multi-view encoder.

During model training, the objective of MJG module contains two types of losses, including the masking energy (ME) divergence adversarial loss and the weighted reconstruction loss. The ME divergence adversarial loss is produced by the MED module, as to be elaborated in the next section. The weighted reconstruction loss is to enforce the consistency between the true underlying and generated/reconstructed target-view data. Formally, with the support of Rubin's multiple imputation rule, the weighted reconstruction loss $\mathcal{L}_r(\mathbf{X}^t)$ for the target-view $\mathbf{X}^t$ is $\mathbf{M}^t \odot [(1 - \beta) \sum_{s=1:v}^{s \neq t} \ell_e(\mathbf{X}^t, \bar{\mathbf{X}}^{st}) + \beta \cdot \ell_e(\mathbf{X}^t, \bar{\mathbf{X}}^{tt})]$, where ℓ_e is the function of the *root mean absolute error*.

Hence, the overall objective of MJG module over the mini-batches $\mathbf{P}$ and $\mathbf{Q}$ is $\mathcal{L}_G = \mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t) + \alpha \cdot [\mathcal{L}_r(\mathbf{P}^t) + \mathcal{L}_r(\mathbf{Q}^t)]$, where $\mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t)$ is the ME divergence adversarial loss produced by the MED module. α is a hyper-parameter. The MJG module is trained to minimize $\mathcal{L}_G$ using the newly updated MED module.

Note that, the MJG module of Git can also simultaneously impute all incomplete views with a minor modification. It needs to develop one decoder for each target-view to map the embedded attributes into the target space.

The Masking Energy Discriminator

Following the standard GAN model (Goodfellow et al. 2014), we employ a discriminator to compete with the MJG module in Git, which can help the MJG module to impute data as truly as possible. However, the true underlying and generated target-view data distributions of the GAN-based multi-view imputation model usually have a negligible intersection (Arjovsky and Bottou 2017). The Jensen-Shannon (JS) divergence of the discriminator is not continuous and non-differentiable, giving rise to the "vanishing" gradient problem. This problem may prevent the model parameter of such discriminator from changing its value or even stop model from further training. It thus degrades the performance of multi-view imputation model.

To this end, in Git, we introduce a *masking energy* discriminator (MED) module. It introduces a *masking energy* (ME) divergence by the optimal transport metric, to make Git *differentiable*, so as to prevent the "vanishing" gradient problem, for the stable model training.

Specifically, inspired by the simple intuition of the multi-view imputation task that, the true underlying and generated target-view data should share the same distribution, we first integrate the optimal transport metric with the mask matrix, to develop a *masking optimal transport* (MOT) metric for multi-view imputation, as stated in Definition 2. Let $\hat{\mu}_{\mathbf{x}}^{t|p} = \frac{1}{b}\sum_{i=1}^{b}\delta_{\mathbf{x}_i^t}^{p}$ (resp. $\hat{\mu}_{\mathbf{x}}^{t|q} = \frac{1}{b}\sum_{i=1}^{b}\delta_{\mathbf{x}_i^t}^{q}$), $\hat{\mu}_{\mathbf{m}}^{t|p} = \frac{1}{b}\sum_{i=1}^{b}\delta_{\mathbf{m}_i^t}^{p}$ (resp. $\hat{\mu}_{\mathbf{m}}^{t|q} = \frac{1}{b}\sum_{i=1}^{b}\delta_{\mathbf{m}_i^t}^{q}$), and $\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}=\frac{1}{b}\sum_{i=1}^{b}\delta_{\bar{\mathbf{x}}_i^t}^{p}$ (resp. $\hat{\nu}_{\bar{\mathbf{x}}}^{t|q}=\frac{1}{b}\sum_{i=1}^{b}\delta_{\bar{\mathbf{x}}_i^t}^{q}$) denote the empirical measures over the size-b target-view data matrix $\mathbf{P}^t$ (resp. $\mathbf{Q}^t$) of the mini-batch $\mathbf{P}$ (resp. $\mathbf{Q}$), its mask matrix $\mathbf{M}^{t|p}$ (resp. $\mathbf{M}^{t|q}$), and the reconstructed matrix $\bar{\mathbf{P}}^t$ (resp. $\bar{\mathbf{Q}}^t$) predicted by the MJG module for the target-view data $\mathbf{X}^t$, respectively. $\delta_{\mathbf{x}_i^t}^{p}$, $\delta_{\mathbf{x}_i^t}^{q}$, $\delta_{\mathbf{m}_i^t}^{p}$, $\delta_{\mathbf{m}_i^t}^{q}$, $\delta_{\bar{\mathbf{x}}_i^t}^{p}$, and $\delta_{\bar{\mathbf{x}}_i^t}^{q}$ are the Dirac distributions.

Definition 2. (The masking optimal transport metric, MOT). *For the target-view $\mathbf{X}^t$ in $\mathbf{P}$, the MOT metric, denoted by $OT_{\mathbf{m}}$, over the empirical measures of $\mathbf{P}^t$ and its reconstructed $\bar{\mathbf{P}}^t$, i.e., $\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}$ and $\hat{\mu}_{\mathbf{x}}^{t|p}$, is $OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}, \hat{\mu}_{\mathbf{x}}^{t|p}) = \min_{\mathcal{Y}_t^p \in \Gamma_{b,b}} \langle \mathcal{Y}_t^p, \mathcal{C}_{\mathbf{m}}^{t|p} \rangle + \lambda h(\mathcal{Y}_t^p)$, where $h(\mathcal{Y}_t^p) = \sum_{i=1}^{b}\sum_{j=1}^{b} y_{ij}\log y_{ij}$; λ is a hyper-parameter; $\hat{\nu}_{\bar{\mathbf{x}}}^{t|p} \otimes \hat{\mu}_{\mathbf{m}}^{t|p}$ (resp. $\hat{\mu}_{\mathbf{x}}^{t|p} \otimes \hat{\mu}_{\mathbf{m}}^{t|p}$) stands for the product measure of $\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}$ (resp. $\hat{\mu}_{\mathbf{x}}^{t|p}$) and $\hat{\mu}_{\mathbf{m}}^{t|p}$. The transportation plan matrix $\mathcal{Y}_t^p$ is from the set $\Gamma_{b,b} \overset{def}{=} \{\mathcal{Y}_t^p \in \mathbb{R}^{b\times b} : \mathcal{Y}_t^p \mathbb{1}_b = \frac{1}{b}\mathbb{1}_b, \mathcal{Y}_{t|p}^{\top}\mathbb{1}_b = \frac{1}{b}\mathbb{1}_b\}$. $\langle \mathcal{Y}_t^p, \mathcal{C}_{\mathbf{m}}^{t|p} \rangle = \mathrm{tr}((\mathcal{Y}_t^p)^{\top}\mathcal{C}_{\mathbf{m}}^{t|p})$ is the Frobenius dot-product of $\mathcal{Y}_t^p$ and $\mathcal{C}_{\mathbf{m}}^{t|p}$. The masking cost matrix $\mathcal{C}_{\mathbf{m}}^{t|p}$ is defined as $(1 - \frac{(\mathbf{m}_i^t \odot \bar{\mathbf{x}}_i^t)\cdot(\mathbf{m}_j^t \odot \mathbf{x}_j^t)}{||\mathbf{m}_i^t \odot \bar{\mathbf{x}}_i^t||_2 ||\mathbf{m}_j^t \odot \mathbf{x}_j^t||_2})_{ij} \in \mathbb{R}^{b\times b}$, where $\odot$ is the element-wise multiplication.*

The MOT metric measures the closeness between empirical distributions of the true underlying and generated data, conditional on all observed values from the target-view data. The remarkable advantage of MOT divergence is that, it makes GAN-based multi-view imputation model differentiable and tractable, eliminating the model instabilities incurred by the JS divergence.

However, with the fixed mini-batch size, the gradients of the MOT divergence are not unbiased estimators of the gradients of the original optimal transport problem for multi-view imputation (Bellemare et al. 2017). As a result, in order to get *stable* and *unbiased* gradients, we further introduce a *masking energy* (ME) divergence over the two mini-batches $\mathbf{P}$ and $\mathbf{Q}$ in $\mathbf{X}$, as the basis of MED modeling.

Definition 3. (The masking energy divergence, ME divergence). *For the target-view $\mathbf{X}^t$ in $\mathbf{P}$ and $\mathbf{Q}$, the ME divergence, denoted by $\mathcal{E}_{\mathbf{m}}$, among the empirical measures $\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}$, $\hat{\mu}_{\mathbf{x}}^{t|p}$, $\hat{\nu}_{\bar{\mathbf{x}}}^{t|q}$, and $\hat{\mu}_{\mathbf{x}}^{t|q}$ is defined as $OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}, \hat{\mu}_{\mathbf{x}}^{t|p}) + OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|q}, \hat{\mu}_{\mathbf{x}}^{t|p}) + OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}, \hat{\mu}_{\mathbf{x}}^{t|q}) + OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|q}, \hat{\mu}_{\mathbf{x}}^{t|q}) - 2OT_{\mathbf{m}}(\hat{\mu}_{\mathbf{x}}^{t|p}, \hat{\mu}_{\mathbf{x}}^{t|q}) - 2OT_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}, \hat{\nu}_{\bar{\mathbf{x}}}^{t|q})$.*

This new ME divergence combines both the *differentiable* MOT metric and the energy divergence with an unbiased estimator (Zhao, Mathieu, and LeCun 2017), resulting in a highly discriminative divergence function with stable and unbiased gradients. In general, for the target-view $\mathbf{X}^t$, the differentiable ME divergence loss function $\mathcal{L}_a(\mathbf{P}^t, \mathbf{Q}^t)$ over the two mini-batches $\mathbf{P}$ and $\mathbf{Q}$ can be defined as $\frac{1}{2b}\cdot\mathcal{E}_{\mathbf{m}}(\hat{\nu}_{\bar{\mathbf{x}}}^{t|p}, \hat{\mu}_{\mathbf{x}}^{t|p}, \hat{\nu}_{\bar{\mathbf{x}}}^{t|q}, \hat{\mu}_{\mathbf{x}}^{t|q})$, where b is the mini-batch size.

By virtue of the *differentiable* ME divergence, Git can provide a usable and unbiased gradient during training, and thus helps to get rid of the "vanishing" gradient issue and improve the imputation. There is an explicit illustration on how Git handles the "vanishing" gradient problem under the ME divergence, as well as it cannot be done under the JS divergence in Appendix A.

Let $\hat{\nu}_{\bar{\mathbf{x}}}^{t}$ be the generated distribution of target-view, and $\hat{\mu}_{\mathbf{x}}^{t}$ be the true underlying distribution of target-view. The data distribution produced by Git converges to the true underlying target-view data distribution under a weak assumption (Bernton et al. 2019), as stated in Lemma 1.

Assumption 1 The multi-view imputation of Git makes the ME divergence converge to 0 as the sample size $n \to \infty$.

Lemma 1 Under Assumption 1, the target-view data distribution $\hat{\nu}_{\bar{\mathbf{x}}}^{t}$ produced by the MJG module in Git converges to the true underlying data distribution $\hat{\mu}_{\mathbf{x}}^{t}$, (i.e., $\hat{\nu}_{\bar{\mathbf{x}}}^{t} \to \hat{\mu}_{\mathbf{x}}^{t}$) with the optimization of Git, as $n \to \infty$.

The proof of Lemma 1 completes based on the definition of ME divergence and Assumption 1, and the proof details are shown in Appendix B.

Experiment

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

Experimental Settings

Datasets. In the experiments, we use five public real-world multi-view datasets. In particular, the mixed national institute of standards and technology dataset [1] (*MNIST*) is a

[1] http://yann.lecun.com/exdb/mnist/

Method	RMSE				PSNR			
	MNIST	*CUB*	*CityStreet*	*KDEF*	*MNIST*	*CUB*	*CityStreet*	*KDEF*
CRA	29.90± 1.02	32.89± 0.98	25.12± 0.89	17.89± 0.92	35.52± 0.93	29.02± 0.82	25.21± 0.93	47.21± 1.21
MIB	30.83± 1.12	33.25± 0.97	–	18.02± 0.95	34.49± 0.94	28.09± 0.98	–	45.92± 1.13
VIGAN	30.06± 0.82	32.41± 1.21	24.73± 0.87	18.66± 0.82	34.86± 0.82	28.49± 0.89	25.75± 0.92	45.91± 0.98
EMGAN	30.22± 0.92	32.45± 0.96	26.71± 0.91	17.61± 0.99	34.27± 0.92	28.44± 0.93	22.39± 0.89	39.84± 0.97
CAEGAN	30.41± 0.98	32.32± 0.89	27.16± 0.92	16.78± 0.98	34.42± 0.82	28.75± 0.82	21.82± 0.94	46.57± 0.99
Git	**19.66± 0.81**	**27.08± 0.78**	**15.03± 0.83**	**11.33± 0.82**	**53.47± 0.84**	**32.38± 0.79**	**42.78± 0.89**	**56.26± 0.79**

Table 1: Performance comparison of missing feature imputation

Method	RMSE				PSNR			
	MNIST	*CUB*	*CityStreet*	*KDEF*	*MNIST*	*CUB*	*CityStreet*	*KDEF*
CRA	9.28± 0.58	12.31± 0.44	33.28± 0.65	25.89± 0.55	67.02± 0.89	58.25± 0.98	40.03± 0.82	39.98± 0.79
VIGAN	9.18± 0.61	12.16± 0.36	32.70± 0.68	25.99± 0.61	67.86± 0.96	58.21± 0.92	40.98± 0.79	40.42± 0.81
EMGAN	9.28± 0.65	11.24± 0.55	33.07± 0.59	24.10± 0.56	67.55± 0.84	59.06± 0.89	37.49± 0.85	41.71± 0.77
CAEGAN	9.69± 0.56	13.21± 0.52	33.92± 0.65	24.21± 0.59	66.60± 0.86	58.89± 0.91	39.36± 0.78	42.58± 0.75
Git	**8.78± 0.34**	**10.89± 0.32**	**31.84± 0.49**	**23.98± 0.36**	**68.02± 0.78**	**61.11± 0.79**	**41.36± 0.65**	**43.07± 0.61**

Table 2: Performance comparison of missing view imputation

widely known benchmark hand-written digit dataset with 70,000 examples. A separate view is created following the method in (Liu and Tuzel 2016) where the authors produce a new digit image from an original image by only maintaining the edge of the number. The Caltech-UCSD Birds-200-2011 dataset [2] (*CUB*) contains different categories of birds, where the first 10 categories are used. In *CUB*, visual features from GoogLeNet and text features using doc2vec are used as two views. It contains 11,788 images of 200 bird species, and a vocabulary of 28 attribute groupings and 312 binary attributes. The busy city street multi-view video dataset [3] (*CityStreet*) is collected from the videos with one hour long and 2.7K resolution at 30 fps. 500 multi-view images are uniformly sampled from the videos with 3 different views. The karolinska directed emotional faces dataset [4] (*KDEF*) is a set of totally 4,900 pictures with 7 emotional expressions, where each subject is imaged from 5 different views. The Database of Faces dataset (*ORL*) (Yan et al. 2021) contains 400 face images from 40 distinct subjects. Due to the similar data type of *KDEF* and *ORL* and the space constraint, the experimental results over *ORL* are presented in Appendix C. For each dataset, we randomly choose 10% samples for the test, 10% samples for validation, and the rest for training.

Baselines. In the experiments, we compare five state-of-the-art multi-view imputation methods, i.e., CRA (Tran et al. 2017), MIB(Federici et al. 2020) VIGAN (Shang et al. 2017), EMGAN (Cai et al. 2018), and CAEGAN (Yang, Qian, and Fan 2020). Since all of these methods cannot be trained with incomplete multi-view data, we initialize the missing values in these datasets as zero.

Metrics. In the evaluation, we use the *root mean squared error* (RMSE) (Jeffery, Garofalakis, and Franklin 2006) and *term peak signal-to-noise ratio* (PSNR) (Najafipour, Babaee, and Shahrtash 2013) to measure the effectiveness of imputation models. The smaller RMSE value corresponds to the better imputation performance, while PSNR is opposite. To obtain the RMSE and PSNR values, we first remove a square (for image) or continuous words (for text) of fixed size (i.e., 20% of total features) centered at a random location in each view for imputation (Yu et al. 2020; Tran et al. 2017), and thus we use these values as the ground-truth to the missing values. Then, we treat a view from the data as the target-view at each time, so as to get the average metric values with the support of the removed and ground-truth values. Each result value is reported by averaging five times of experimental results under different data random divisions.

Implementation details. For all multi-view imputation methods, the learning rate is 0.001, the dropout rate is 0.1, and the batch size is 16. The ADAM algorithm is utilized to train networks. The training epoch is 50, 30, 30, and 500, over *MNIST*, *CUB*, *CityStreet*, and *KDEF*, respectively. CRA contains fully connected networks with 4 hidden layers. In VIGAN, the generator contains two stride-2 convolutions, 9 residual blocks, and 2 fractionally strided convolutions, while the discriminator contains 8 convolutional layers with an increasing number of 3×3 filter kernels. For EMGAN, the depth of the generator is 5. Each layer contains a downsampling block and a corresponding upsampling block. In the downsampling block, EMGAN employs two 3D convolutional layers to extract features. Then, it uses four 3D convolutional layers and a fully connected layer for the discriminator. For CAEGAN, the generator consists of a fully connection layer and 5 deconvolutional layers, while the discriminator is composed of 5 convolutional layers and a fully connection layer. In Git, the hyperparameter α is 0.7, β is 100, and the MED module's iteration times k is 2.

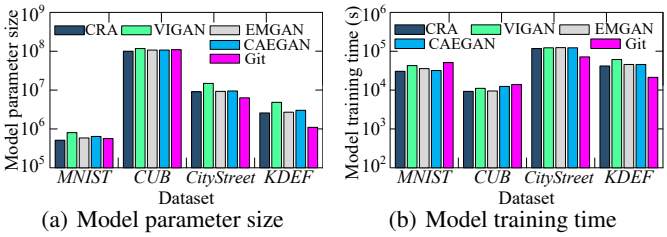
Comparison Study

Table 1 reports the experimental results of multi-view imputation methods over four datasets. In particular, the results of MIB are unavailable over *CityStreet*, since they are not able to finish within 10^5 seconds. One can observe that, Git substantially outperforms all baselines. In terms of imputation accuracy (i.e., RMSE and PSNR), Git exceeds the best per-

[2]http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

[3]https://visal.cs.cityu.edu.hk/research/citystreet/

[4]https://www.kdef.se/



(a) Model parameter size (b) Model training time

Figure 2: Imputation efficiency evaluation

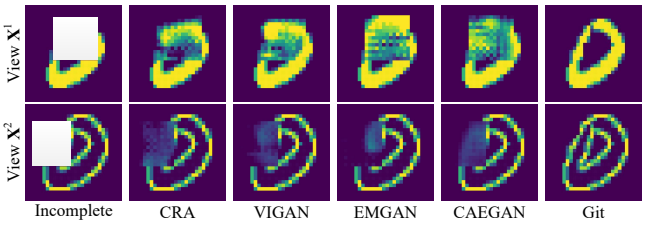
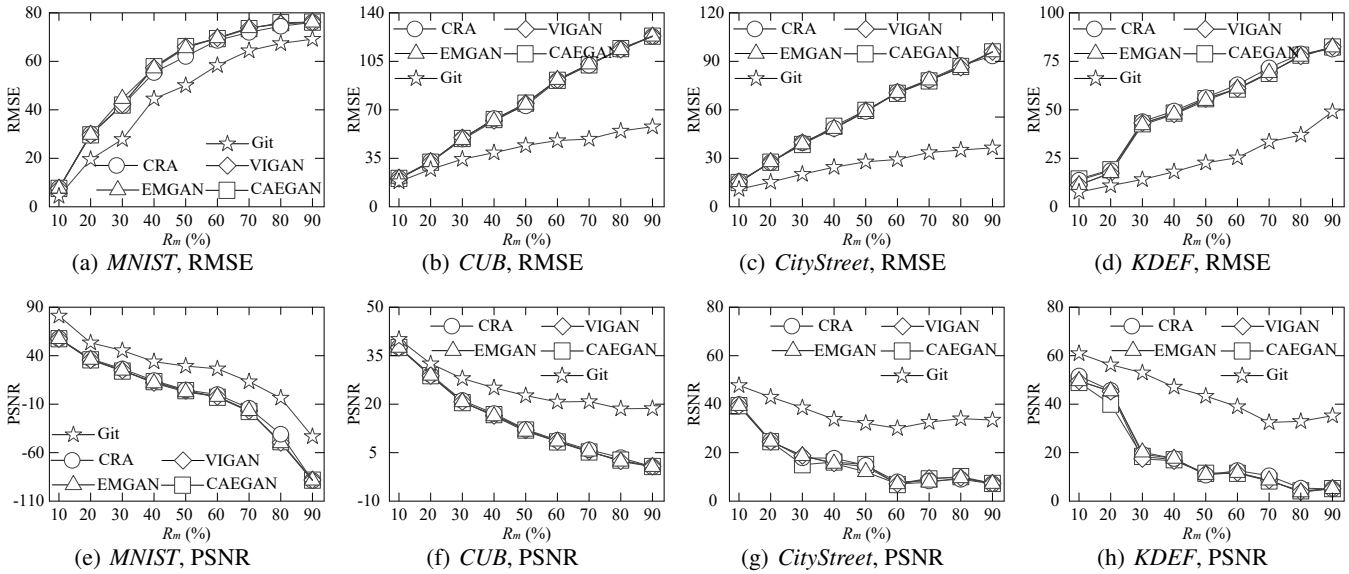


Figure 3: Visualization of multi-view imputation on *MNIST*



(a) *MNIST*, RMSE (b) *CUB*, RMSE (c) *CityStreet*, RMSE (d) *KDEF*, RMSE

(e) *MNIST*, PSNR (f) *CUB*, PSNR (g) *CityStreet*, PSNR (h) *KDEF*, PSNR

Figure 4: The performance of multi-view imputation algorithms vs. R_m

forming baseline (i.e., VIGAN) by 35.66% in average, and it even increases up to 70.36% over CAEGAN on the *CityStreet* dataset. Further, Git is more stable, since it has only 88.58% of the standard deviation produced by baselines on average. This is because, Git adopts an effective GAN model with the joint autoencoder and ME divergence, and thereby enhances the imputation accuracy. Due to the high complexity, we omit MIB in the rest of experiments.

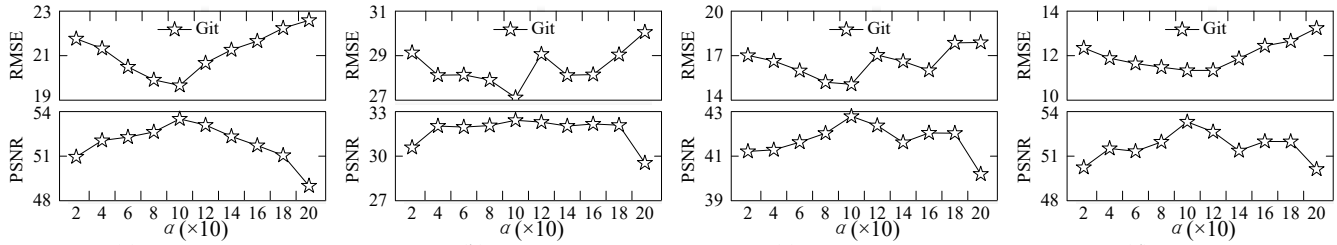
Moreover, in order to further confirm the effectiveness of Git, we set the target-view as completely missing, keeping same to the default setting of baselines. Table 2 presents the accuracy of multi-view imputation methods over such missing view imputation task. One can conclude that, Git consistently gets the better imputation accuracy than baselines in all cases. The accuracy of Git even increases up to 10.67% over CAEGAN on the *CUB* dataset. Thus, Git is still effective for the traditional missing view imputation task.

In addition, Figrue 2 lists the *model parameter size* and *model training time* of multi-view imputation methods. The smaller model parameter size or training time indicates the better efficiency. Git requires less model parameters and training time when the view size of dataset exceeds 2 (i.e., *CityStreet* and *KDEF*). It takes 82.98% model parameters of the smallest baseline CRA in average, while speeds up CRA by 1.21 times in average. In particular, the model parameters and model training time of Git are less sensitive to the view size than that of others. It is because that, every baseline requires to train $(v-1)$ models to learning the mappings from each of $(v-1)$ source-views to the target-view. In contrast, Git learns the distribution from all observed multi-view data with the joint autoencoder, so as to efficiently aggregate imputation results w.r.t. the observed data in v views.

Figure 3 visualizes the missing feature imputation of all multi-view imputation methods over a random sample with two views in *MNIST*. We can easily observe that, Git gets much better imputation accuracy than baselines. It further confirms the powerful imputation ability of Git for missing feature imputation problem.

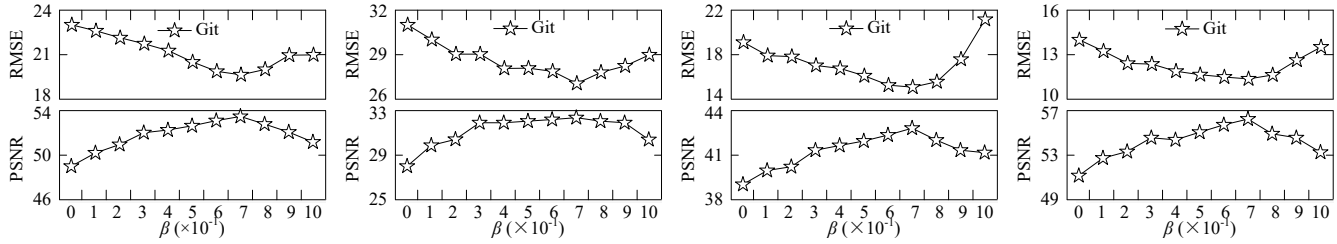
Parameter Evaluation

Effect of R_m. When varying the missing rate R_m (i.e., how many features/values in each view are dropped) from 10% to 90%, the corresponding experimental results are depicted in Figure 4. We can find that, with the growth of missing rate, the imputation accuracy (i.e., RMSE and PSNR) of each algorithm descends consistently. It is attributed to the less data information for imputation when the missing rate turns high. Among these algorithms, Git performs best in each case. Moreover, its accuracy becomes more stable with the increase of missing rate. In other words, Git is more robust



(a) *MNIST* (b) *CUB* (c) *CityStreet* (d) *KDEF*

Figure 5: The multi-view imputation performance of Git vs. α



(a) *MNIST* (b) *CUB* (c) *CityStreet* (d) *KDEF*

Figure 6: The multi-view imputation performance of Git vs. β

Method	RMSE				PSNR			
	MNIST	*CUB*	*CityStreet*	*KDEF*	*MNIST*	*CUB*	*CityStreet*	*KDEF*
Git-JS	19.98± 0.82	27.89± 0.86	15.23± 0.89	11.49± 0.86	52.67± 0.88	32.11± 0.81	42.01± 0.89	54.97± 0.90
Git-noD	21.32± 0.85	28.08± 0.78	16.67± 0.85	11.88± 0.83	52.03± 0.85	31.93± 0.82	41.29± 0.92	54.51± 0.85
Git	**19.66± 0.81**	**27.08± 0.78**	**15.03± 0.83**	**11.33± 0.82**	**53.47± 0.84**	**32.38± 0.79**	**42.78± 0.89**	**56.26± 0.79**

Table 3: The ablation study of Git

with the increasing missing rate R_m than others. The reason behind is that, Git learns the data distribution from all observed multi-view data conditional on the feature missing state of each view, so as to decrease or delay the impact of increased missing rate on the imputation performance.

Effect of α. When the hyper-parameter α that balances the weighted reconstruction loss and ME divergence adversarial loss for Git changes from 20 to 200, Figure 5 plots the corresponding experimental results. One can observe that, Git is the best, in terms of the smaller RMSE and larger PSNR, when α is 100. It confirms that, both weighted reconstruction and ME divergence adversarial loss functions in the MJG module benefit the imputation accuracy.

Effect of β. When the hyper-parameter β varies from 0 to 1, the corresponding experimental results are depicted in Figure 6. In particular, β weights the reconstructed target-view data matrix in contrast with other views in the weighted reconstruction loss function. We can find that, Git gets the best imputation accuracy when β is 0.7. It is because that, for Git, the data information learned from the target-view observed data is more useful than that of other views. It also confirms that, the information from the observed multi-view data benefits the imputation accuracy.

Ablation Study

Table 3 shows the experimental results of ablation study. Git-noD is the variant of Git without the MED module. Git-JS is the variant of Git that replaces ME divergence with JS divergence. We can observe that, both ME divergence and MED module in Git do have positive effect on the imputation performance. The imputation accuracy (i.e., RMSE and PSNR) decreases in average 1.67% and 7.65% without ME divergence and MED module, respectively. It signifies that, the MED module contributes more on Git. It also confirms the rationality and effectiveness of the ME divergence.

In addition, we study the performance of multi-view imputation methods on post-imputation prediction (i.e., predicting the view labels). The detailed results are described in Appendix D. It reflects the imputation performance of each method. We can find that, Git still outperforms baselines in each case. It confirms the effectiveness of Git.

Conclusion

In this paper, we propose a multi-view generative imputation model Git with the support of optimal transport. It is able to effectively estimate missing features in the multi-view data. Git consists of the MJG module and the MED module. MJG learns the target-view data distribution with the joint autoencoder and the multiple imputation rule. MED leverages a new *masking energy* divergence function to make Git differentiable for imputation enhancement. Extensive experiments over several real-world data sets demonstrate that, Git significantly boosts the imputation performance, compared to the state-of-the-art multi-view imputation methods.

Acknowledgments

This work is partly supported by the National Key Research and Development Program of China under Grant No.2019YFE0126200, the Zhejiang Provincial NSF for Distinguished Young Scholars under Grant No.LR21F020005, the NSFC under Grants No.61825205 and No.61902343, and the Fundamental Research Funds for the Central Universities under Grant No.2021FZZX001-25. Xiaoye Miao is the corresponding author of the work.

References

Arjovsky, M.; and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. *ArXiv Preprint ArXiv:1701.04862*.

Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *ArXiv Preprint ArXiv:1705.10743*.

Bernton, E.; Jacob, P. E.; Gerber, M.; and Robert, C. P. 2019. On parameter estimation with the Wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4): 657–676.

Boris, M.; Julie, J.; Claire, B.; and Marco, C. 2020. Missing data imputation using optimal transport. In *ICML*, 1–18.

Cai, L.; Wang, Z.; Gao, H.; Shen, D.; and Ji, S. 2018. Deep adversarial learning for multi-modality missing data completion. In *SIGKDD*, 1158–1166.

Farhangfar, A.; Kurgan, L. A.; and Pedrycz, W. 2007. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics -Part A: Systems and Humans*, 37(5): 692–709.

Federici, M.; Dutta, A.; Forré, P.; Kushman, N.; and Akata, Z. 2020. Learning robust representations via multi-view information bottleneck. In *ICLR*, 1–12.

Goeleven, E.; De Raedt, R.; Leyman, L.; and Verschuere, B. 2008. The Karolinska directed emotional faces: A validation study. *Cognition and Emotion*, 22(6): 1094–1118.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.

Guo, Y. 2013. Convex subspace representation learning from multi-view data. In *AAAI*, volume 27, 387–393.

Ipsen, N. B.; Mattei, P.-A.; and Frellsen, J. 2020. not-MIWAE: Deep generative modelling with missing not at random data. *ArXiv Preprint ArXiv:2006.12871*.

Jaques, N.; Taylor, S.; Sano, A.; and Picard, R. 2017. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *ACII*, 202–208.

Jeffery, S. R.; Garofalakis, M.; and Franklin, M. J. 2006. Adaptive cleaning for RFID data streams. In *VLDB*, 163–174.

Little, R. J.; and Rubin, D. B. 2019. *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Liu, M.-Y.; and Tuzel, O. 2016. Coupled generative adversarial networks. In *NeurIPS*, 469–477.

Miao, X.; Wu, Y.; Chen, L.; Gao, Y.; Wang, J.; and Yin, J. 2022a. Efficient and effective data imputation with influence functions. *Proceedings of the VLDB Endowment*, 15(3): 624–632.

Miao, X.; Wu, Y.; Chen, L.; Gao, Y.; and Yin, J. 2022b. An experimental survey of missing data imputation algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 1(1): 1–20.

Miao, X.; Wu, Y.; Wang, J.; Gao, Y.; Mao, X.; and Yin, J. 2021. Generative semi-supervised learning for multivariate time series imputation. In *AAAI*, volume 35, 8983–8991.

Najafipour, A.; Babaee, A.; and Shahrtash, S. M. 2013. Comparing the trustworthiness of signal-to-noise ratio and peak signal-to-noise ratio in processing noisy partial discharge signals. *IET Science, Measurement & Technology*, 7(2): 112–118.

Royston, P.; and White, I. R. 2011. Multiple imputation by chained equations (MICE): Implementation in Stata. *Journal of Statistical Software*, 45(4): 1–20.

Shang, C.; Palmer, A.; Sun, J.; Chen, K.-S.; Lu, J.; and Bi, J. 2017. VIGAN: Missing view imputation with generative adversarial networks. In *Big Data*, 766–775.

Silva, I.; Moody, G.; Scott, D. J.; Celi, L. A.; and Mark, R. G. 2012. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *CinC*, 245–248.

Spinelli, I.; Scardapane, S.; and Uncini, A. 2019. Missing data imputation with adversarially-trained graph convolutional networks. *ArXiv Preprint ArXiv:1905.01907*.

Stekhoven, D. J.; and Bühlmann, P. 2011. MissForest non parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1): 112–118.

Tran, L.; Liu, X.; Zhou, J.; and Jin, R. 2017. Missing modalities imputation via cascaded residual autoencoder. In *CVPR*, 1405–1414.

Wu, Y.; Wang, J.; Miao, X.; Wang, W.; and Yin, J. 2022. Differentiable and scalable generative adversarial models for data imputation. *ArXiv Preprint ArXiv:2201.03202*.

Yan, X.; Hu, S.; Mao, Y.; Ye, Y.; and Yu, H. 2021. Deep multi-view learning methods: A review. *Neurocomputing*, 448: 106–129.

Yang, H.; Qian, P.; and Fan, C. 2020. An indirect multimodal image registration and completion method guided by image synthesis. *Computational and Mathematical Methods in Medicine*, 2020(1): 1–10.

Yoon, J.; Jordon, J.; and Schaar, M. 2018. GAIN: Missing data imputation using generative adversarial nets. In *ICML*, 5675–5684.

Yu, T.; Guo, Z.; Jin, X.; Wu, S.; Chen, Z.; Li, W.; Zhang, Z.; and Liu, S. 2020. Region normalization for image inpainting. In *AAAI*, volume 34, 12733–12740.

Zhang, L.; Zhao, Y.; Zhu, Z.; Shen, D.; and Ji, S. 2018. Multi-view missing data completion. *IEEE Transactions on Knowledge and Data Engineering*, 30(7): 1296–1309.

Zhao, J.; Mathieu, M.; and LeCun, Y. 2017. Energy-based generative adversarial network. In *ICLR*, 1–10.

Zheng, J.; Xia, K.; Zheng, Q.; and Qian, P. 2019. A smart brain MR image completion method guided by synthetic-CT-based multimodal registration. *Journal of Ambient Intelligence and Humanized Computing*, 1(1): 1–10.