Data Mining and Knowledge Discovery (2022) 36:1498–1520
https://doi.org/10.1007/s10618-022-00837-0

Extended missing data imputation via GANs for ranking applications

Grace Deng[1] · Cuize Han[2] · David S. Matteson[1]

Received: 10 November 2021 / Accepted: 18 April 2022 / Published online: 15 May 2022

Abstract

We propose Conditional Imputation GAN, an extended missing data imputation method based on Generative Adversarial Networks (GANs). The motivating use case is learning-to-rank, the cornerstone of modern search, recommendation system, and information retrieval applications. Empirical ranking datasets do not always follow standard Gaussian distributions or Missing Completely At Random (MCAR) mechanism, which are standard assumptions of classic missing data imputation methods. Our methodology provides a simple solution that offers compatible imputation guarantees while relaxing assumptions for missing mechanisms and sidesteps approximating intractable distributions to improve imputation quality. We prove that the optimal GAN imputation is achieved for Extended Missing At Random and Extended Always Missing At Random mechanisms, beyond the naive MCAR. Our method demonstrates the highest imputation quality on the open-source Microsoft Research Ranking Dataset and a synthetic ranking dataset compared to state-of-the-art benchmarks and across various feature distributions. Using a proprietary Amazon Search ranking dataset, we also demonstrate comparable ranking quality metrics for ranking models trained on GAN-imputed data compared to ground-truth data.

Keywords Learning-to-rank · Missing data imputation · Generative adversarial networks

Responsible editor: Johannes Fürnkranz

✉ Grace Deng
gd345@cornell.edu

Cuize Han
cuize@amazon.com

David S. Matteson
dm484@cornell.edu

[1] Department of Statistics and Data Science, Cornell University, Ithaca, NY, USA

[2] Amazon Search, Palo Alto, CA, USA

1 Introduction

Missing data is a prevalent data quality issue found in all aspects of data science and machine learning. Modern data collection technology can often exhibit non-random gaps in data due to a variety of reasons, e.g., non-response bias. At the same time, many machine learning models require complete datasets for training, highlighting the need for missing data imputation methods that are broadly applicable to different types of datasets characterized by complex missing mechanisms.

1.1 Motivation

Our motivating application is the classic "learning-to-rank" problem for search, recommendation systems, and information retrieval (Li 2011; Burges 2010). The ranking dataset has a unique structure compared to panel, time series, or image datasets. It is characterized by query-groups, where individual results are associated with a query and ordered by a ranking model, and composite features with non-standard distributions that will vary both on the query-group and query-result level. Training on missing data leads to biased ranking models (Marlin and Zemel 2009) and dropping individual query-results with missing values is difficult given the ordered nature of ranking data.

A further challenge for imputation in ranking applications is violation of the Missing Completely At Random (MCAR) (Heitjan and Basu 1996; Doretti et al. 2018) and Gaussian distribution assumptions favored by classic imputation methods; see Fig. 1. Ranking datasets can include columns that are always observed, which influences the probability of missingness of other features, e.g., a product labeled "New" has more missing feature values due to lack of data. Meanwhile, the MCAR mechanism requires the probability of missingness to be independent of all data values. A more appropriate mechanism for ranking datasets would be Missing At Random (MAR), which is less restrictive and specifies that missingness only depends on observed components (Little and Rubin 2019). Finally, the mechanism is called Missing Not At Random (MNAR) if missingness depends on unobserved components.

Currently, standard imputation methods such as MICE and MissForest have strict assumptions on underlying missing mechanisms and feature distributions (Van Buuren and Groothuis-Oudshoorn 2011; Stekhoven and Bühlmann 2012). Alternatively, using prediction methods for missing values requires a custom model per feature as well as a set of predictors that are never missing; this is near impossible to achieve with real data. More recent methods (Yoon et al. 2018; Li et al. 2018; Luo et al. 2018), involving generative models do not necessarily address more complex dataset structures or account for auxiliary information that influence the underlying data generating process.

We propose a novel extended missing data imputation method by adapting Conditional Generative Adversarial Networks (CGANs) (Goodfellow et al. 2014; Mirza and Osindero 2014; Yoon et al. 2018). Our method aims to encompass more complex data structures and missing mechanisms; empirical ranking datasets are a prime example given its "ordered-grouping" characteristics and heterogeneous distributions across different query-groups. Furthermore, we define more realistic missing scenarios

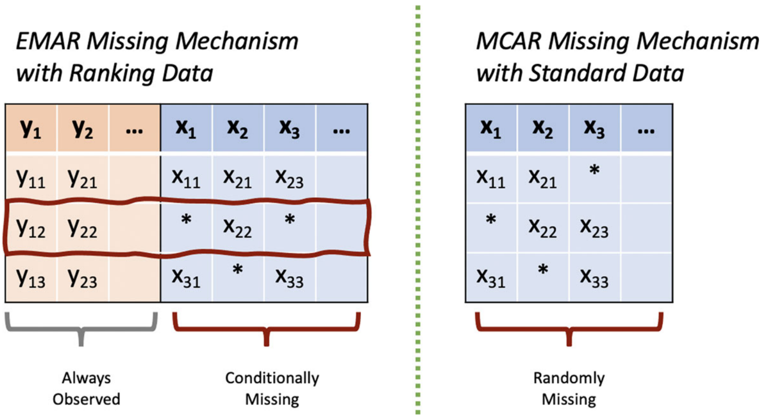
EMAR Missing Mechanism with Ranking Data
y_1 y_2 … x_1 x_2 x_3 …
y_{11} y_{21} x_{11} x_{21} x_{23}
y_{12} y_{22} * x_{22} *
y_{13} y_{23} x_{31} * x_{33}
Always Observed
Conditionally Missing
MCAR Missing Mechanism with Standard Data
x_1 x_2 x_3 …
x_{11} x_{21} *
* x_{22} x_{23}
x_{31} * x_{33}
Randomly Missing

Fig. 1 Missing data scenarios under EMAR versus MCAR mechanisms for an observed ranking dataset. Empirical ranking dataset with always observed columns is an example of EMAR; EAMAR mechanism generalizes EMAR to the population distribution. See Sect. 2.3.3 for formal definitions

Extended Missing At Random (EMAR) and Extended Always Missing At Random (EAMAR) based on MAR and show that GAN-generated imputations satisfy conditions for compatible imputations under these new mechanisms.

1.2 Related methods

The GAN architecture is comprised of two competing deep neural nets; the *generator* and the *discriminator*. The generator produces synthetic data by mimicking the underlying data distribution, and the discriminator tries to distinguish fake and real data. Training GANs is a balancing act in that neither system should dominate the other too quickly. A strong generator may lead to mode collapse (Thanh-Tung and Tran 2020) by memorizing select samples, while a strong discriminator can lead to near-zero gradients and non-convergence by perfectly classifying samples.

GANs have shown impressive performance in generative tasks, including high-resolution image generation, text-to-image synthesis, image-to-image translation, video synthesis, and audio generation/synthesis (Oza et al. 2020; Sheng et al. 2019). A ‘vanilla’ GAN has a tendency to suffer from non-convergence and mode collapse. The introduction of Deep-Convolutional GANs (DC-GAN) with convolution layers (Radford et al. 2016) greatly improved the stability of GANs during training and showed that the generator and discriminator learned a hierarchy of representations. Other techniques for GAN training include feature mapping, batch normalization (Salimans et al. 2016), leaky-relu activation functions (Xu et al. 2015), and modifications of the objective loss functions, e.g., Wasserstein loss (Arjovsky et al. 2017). Conditional GANs (Mirza and Osindero 2014) allow for greater control over the modes of generated data by conditioning on auxiliary information such as class labels.

The Generative Adversarial Imputation Networks (GAIN) algorithm was first proposed (Yoon et al. 2018) to address this problem under the naive MCAR assumption. This algorithm generated better imputations benchmarked against distributive missing data methods such as MICE (Van Buuren and Groothuis-Oudshoorn 2011) and MissForest (Stekhoven and Bühlmann 2012), and generative methods such as Expectation-Maximization (EM). These methods are limited by assuming an underlying parametric distribution for missing data (Van Buuren 2018), and MICE in particular assumes MCAR. Imputation with GANs addresses the gaps in current imputation methods where the generator strives to accurately impute missing data, and the discriminator strives to distinguish between observed and imputed data while minimizing the traditional minimax loss function. Applied to ranking, there are two drawbacks: the restrictive assumption of MCAR missing mechanism and the inability to account for heterogeneous subgroups with different data distributions.

Other methods are limited to images. Mis-GAN (Li et al. 2018) utilized two separate GANs for data and mask (a matrix for indicating missing values) imputation for images, under various data corruption scenarios. Colla-GAN (Lee et al. 2019) proposed converting the image imputation problem to a multi-domain images-to-image translation problem, resulting in imputations with higher visual quality. GAMIN (Yoon and Sull 2020) specifically targets high missingness levels ($> 85\%$). Non-image data use cases include imputation for sequential data such as multivariate time series (Luo et al. 2018; Kim et al. 2020; Guo et al. 2019; Zhang et al. 2021).

1.3 Our approach and contributions

We propose an extended Conditional Imputation GAN with three key contributions:

1. We introduce two new missing mechanisms, EMAR and EAMAR, that encompass broader empirical dataset types and provide theoretical guarantees for compatible imputations via GANs.
2. We propose a Conditional Imputation GAN that allows for flexible imputation across (i) different data distributions, (ii) heterogeneous subgroups based on auxiliary information, and (iii) our new extended missing mechanisms.
3. We illustrate the superior imputation quality of our method against state-of-the-art benchmarks using open-source Microsoft Research ranking dataset and a proprietary 1.8 million query-group Amazon Search dataset.

To our knowledge, there has been no prior work exploring GAN imputation for machine-learned ranking (MLR) applications. Our method greatly expands the theoretical basis for GAN-based imputation methods for complex datasets and missing mechanisms, and is the first to adapt Conditional GANs for imputation on industry-scale ranking datasets.

Through empirical evaluations, we showcase the superior imputation quality of our method against benchmarks using three ranking datasets: a public Microsoft Research[1] ranking dataset with heterogeneous subgroups, a simulated ranking dataset with extensive feature distributions, and a proprietary 1.8 million query-group Amazon Search dataset. The Conditional Imputation GAN is particularly effective for imputing data under non-MCAR scenarios with non-standard distributions, as well as being computationally efficient for large-scale datasets.

[1] Data available at https://www.microsoft.com/en-us/research/project/mslr/.

As an investigation to downstream application impact, we also train standard ranking models on the imputed data versus the ground-truth data based on a shared target (e.g., clicks or purchases). We then evaluate standard ranking quality measures such as Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR). Our results demonstrate standard ranking models trained on imputed data has comparable performance to models trained on ground-truth complete data, indicating potential broader applicability to other business applications that are also impacted by pervasive data quality (missingness) problems.

2 Methodology

We briefly summarize Conditional GANs and how they can be adapted for imputation that better reflects non-MCAR missingness and heterogeneous feature distributions in ranking datasets. New missing mechanisms EMAR and EAMAR are introduced, and a theoretical analysis is provided for compatible imputations via the Conditional Imputation GAN.

2.1 Conditional GAN

Standard GANs consist of two adversarial models: a generator G that mimics the true data distribution p_{data} and a discriminative model D that predicts the probability that a sample comes from the true distribution or the generated distribution p_G from G (Goodfellow et al. 2014). The models G and D can theoretically be any non-linear mapping function, such as deep neural nets, with a variety of tuning parameters and configurations. This is set-up as a two-player min-max game with value function $V(D, G)$:

$$\begin{aligned}\min_G \max_D V(D, G) = \mathbb{E}_{X \sim p_{data}(X)}[\log D(X)] \\ + \mathbb{E}_{Z \sim p_Z(Z)}[\log(1 - D(G(Z)))]\end{aligned} \tag{1}$$

Suppose that there is auxiliary information Y about the data X. We modify the standard GAN structure by conditioning on Y in both the discriminator and generator, and combine the input noise $p_Z(Z)$ and Y as a joint hidden representation. The resulting Conditional GAN (CGAN) (Mirza and Osindero 2014) value function then becomes:

$$\begin{aligned}\min_G \max_D V(D, G) = \mathbb{E}_{X \sim p_{data}(X)} \log(D(X|Y)) \\ + \mathbb{E}_{Z \sim p_Z(Z)}(\log(1 - D(G(Z|Y))))\end{aligned} \tag{2}$$

2.2 Conditional Imputation GAN

We introduce the Conditional Imputation GAN structure after briefly summarizing the GAIN structure[2] (Yoon et al. 2018). First we define $\boldsymbol{X} = (X_1, \ldots, X_d)$ as a random

[2] Code available at https://github.com/jsyoon0823/GAIN.

vector that could take on either continuous or discrete values (ranking features), and our training data are realizations of $\boldsymbol{X}$. We define the random vector $\boldsymbol{M}$ with the same dimensions as $\boldsymbol{X}$ which takes on values in $\{0, 1\}^d$; this is the missingness indicator matrix (Little and Rubin 2019), or mask matrix for short. Define a new random vector for observed data $\tilde{\boldsymbol{X}}$ as follows:

$$\tilde{X}_i = \begin{cases} X_i, & \text{if } M_i = 1 \\ *, & \text{otherwise} \end{cases} \tag{3}$$

where $*$ represents an unobserved or missing value replaced by noise value Z_i. Hence, M explicitly indicates which values of $\tilde{X}_i$ are observed ($M_i = 1$) and which are missing ($M_i = 0$).

$\tilde{\boldsymbol{X}}$, $\boldsymbol{M}$, and $\boldsymbol{Z}$ are now inputs into the generator G, which will generate an output vector of imputations $\bar{\boldsymbol{X}} = G(\tilde{\boldsymbol{X}}, \boldsymbol{M}, (\boldsymbol{1} - \boldsymbol{M}) \odot \boldsymbol{Z})$ of the same dimension as $\tilde{\boldsymbol{X}}$. The function $\odot$ denotes element-wise multiplication. Note that $\boldsymbol{Z}$ is independent of all other variables; it can be Gaussian noise but can also be designated otherwise depending on the dataset. $\bar{\boldsymbol{X}}$ is a GAN-generated estimate of the true data vector $\boldsymbol{X}$, but we are only interested in the values of $\bar{\boldsymbol{X}}_i$ for which $\boldsymbol{M}_i = 0$, that is, when the value is unobserved. Hence, the GAN-completed data vector $\hat{\boldsymbol{X}}$ with imputations from $\bar{\boldsymbol{X}}$ is

$$\hat{\boldsymbol{X}} = \boldsymbol{M} \odot \tilde{\boldsymbol{X}} + (1 - \boldsymbol{M}) \odot \bar{\boldsymbol{X}} \tag{4}$$

The discriminator D then tries to recover the true $\boldsymbol{M}$ from the completed data vector $\hat{\boldsymbol{X}}$, by predicting the probability of whether each $\hat{\boldsymbol{X}}_i$ is real (observed) or fake (imputed). The resulting vector of probabilities is denoted as $\hat{\boldsymbol{M}} = D(\hat{\boldsymbol{X}})$. Hence, the goal of the generator-discriminator pair is to minimize the distance between $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$.

Given an arbitrary loss function $\mathcal{L}$, the value function is a two-player min-max game. Using the cross-entropy loss function gives:

$$\begin{aligned} \min_G \max_D V(D, G) &= \mathbb{E}[\mathcal{L}(\boldsymbol{M}, \hat{\boldsymbol{M}})] \\ &= \mathbb{E}[\sum_{i=1}^{n} (\boldsymbol{M}_i \log(\hat{\boldsymbol{M}}_i) + (1 - \boldsymbol{M}_i) \log(1 - \hat{\boldsymbol{M}}_i))] \end{aligned} \tag{5}$$

However, this set-up is too naive and fails to account for heterogeneity across different subsets or class labels within many real-world datasets; hence, we propose adapting Conditional GANs to address these concerns.

Suppose we have auxiliary information $\boldsymbol{Y}$ that is always observed along with data $\tilde{\boldsymbol{X}}$ that is conditionally missing under EMAR; $\boldsymbol{Y}$ may influence the probability of missingness or underlying data distribution in other features. We then condition on $\boldsymbol{Y}$ by feeding $\tilde{\boldsymbol{X}}$, $\boldsymbol{M}$, $\boldsymbol{Z}$, and $\boldsymbol{Y}$ into the generator G. This will generate an output vector of imputations $\bar{\boldsymbol{X}} = G(\tilde{\boldsymbol{X}}, \boldsymbol{M}, (1 - \boldsymbol{M}) \odot \boldsymbol{Z}|\boldsymbol{Y})$, which we then use to form the completed data vector $\hat{\boldsymbol{X}}$ in (4) and feed into the discriminator in order to recover $\boldsymbol{M}$. The output of probabilities is now $\hat{\boldsymbol{M}} = D(\hat{\boldsymbol{X}}|\boldsymbol{Y})$. Finally, we write the objective function of the

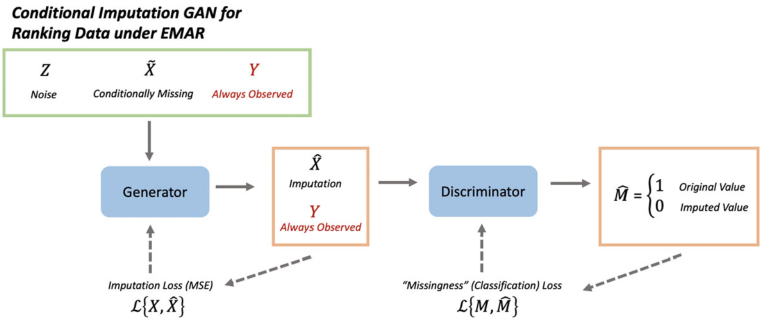
Conditional Imputation GAN for Ranking Data under EMAR
Z
Noise
$\tilde{X}$
Conditionally Missing
Y
Always Observed
Generator
$\hat{X}$
Imputation
Y
Always Observed
Discriminator
$\hat{M} = \begin{cases} 1 & \text{Original Value} \\ 0 & \text{Imputed Value} \end{cases}$
Imputation Loss (MSE)
$\mathcal{L}\{X, \hat{X}\}$
"Missingness" (Classification) Loss
$\mathcal{L}\{M, \hat{M}\}$

Fig. 2 Conditional Imputation GAN overview

Conditional Imputation GAN as:

$$\min_G \max_D V(D, G) = \mathbb{E}[\mathcal{L}(\boldsymbol{M}, \hat{\boldsymbol{M}})] = \mathbb{E}[\mathcal{L}(\boldsymbol{M}, D(\hat{\boldsymbol{X}}|\boldsymbol{Y}))] \tag{6}$$

Using (6), we expand GAN imputation for empirical ranking datasets by conditioning on columns that are always observed during training and separating them from the imputation loss function. See Fig. 2 and Algorithm 1 for details.

Algorithm 1 Psuedo-code for Conditional Imputation GAN

```
Given mini-batch size $n_{mb}$
for number of training iterations do
    Draw $n_{mb}$ samples $(\tilde{\boldsymbol{x}}, \boldsymbol{m}, \boldsymbol{z}, \boldsymbol{y})$ from $\tilde{\boldsymbol{X}}, \boldsymbol{M}, \boldsymbol{Z}$, and $\boldsymbol{Y}$
    (1) Generator $G$
    Generate imputation $\bar{\boldsymbol{x}} \leftarrow G(\tilde{\boldsymbol{x}}, \boldsymbol{m}, \boldsymbol{z}, \boldsymbol{y})$
    Complete data $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{m} \odot \tilde{\boldsymbol{x}} + (1 - \boldsymbol{m}) \odot \bar{\boldsymbol{x}}$
    Compute $\mathcal{L}_G = -\sum(1 - \boldsymbol{m}) \odot \log(\hat{\boldsymbol{m}})$ where $\hat{\boldsymbol{m}} = D(\hat{\boldsymbol{x}}, \boldsymbol{y})$
    Update $G$ using Adam optimizer
    (2) Discriminator $D$
    Predict $\hat{\boldsymbol{m}} \leftarrow D(\hat{\boldsymbol{x}}, \boldsymbol{y})$
    Compute $\mathcal{L}_D = -\sum[\boldsymbol{m} \odot \log(\hat{\boldsymbol{m}}) + (1 - \boldsymbol{m}) \odot \log(1 - \hat{\boldsymbol{m}})]$
    Update $G$ using Adam optimizer
end
```

2.3 Theoretical analysis

Prior works trying to extend or improve GAN imputation (Lee et al. 2019; Li et al. 2018; Camino et al. 2019; Kim et al. 2020) have all restricted theoretical guarantees for the generated distributions to the Missing Completely At Random (MCAR) assumption. In practice, this is too restrictive and rarely satisfied by real-world missing data. Empirically, we often find GAN imputations working quite well for missing data under MAR or even MNAR, and there is a clear gap between theoretical guarantees and empirical results. Here, we aim to close this gap by investigating more general conditions on missing mechanisms and extend theories beyond the MCAR assumption.

In the following analysis, we use small case letters to represent the n independent realizations of $\boldsymbol{X}$, $\boldsymbol{M}$ and $\tilde{\boldsymbol{X}}$ as $\boldsymbol{x}_i = (x_{i1,}, \ldots, x_{id})$, $\boldsymbol{m}_i = (m_{i1,}, \ldots, m_{id})$ and $\tilde{\boldsymbol{x}}_i = (\tilde{x}_{i1,}, \ldots, \tilde{x}_{id})$, $i = 1, \ldots, n$. For a vector $\boldsymbol{x}$ of dimension d, we use the notation $\boldsymbol{x}|_{\boldsymbol{m}}$ to represent the subvector of $\boldsymbol{x}$ that corresponds to the positions where the elements of $\boldsymbol{m}$ is 1, i.e., the observed data components. Curly brackets within conditional probability statements are used for readability.

First we state the main theoretical result in Yoon et al. (2018) which we will utilize to extend the theoretical analysis. A necessary and sufficient condition for $\hat{\boldsymbol{X}}$ being generated by an ideal generator is

$$\mathbb{P}\Big(\hat{\boldsymbol{X}} = \boldsymbol{x} \,\Big|\, \{\boldsymbol{H} = \boldsymbol{h}, M_i = t\}\Big) = \mathbb{P}\Big(\hat{\boldsymbol{X}} = \boldsymbol{x} \,\Big|\, \boldsymbol{H} = \boldsymbol{h}\Big) \tag{7}$$

for every $i \in \{1, \ldots, d\}$, $t \in \{0, 1\}$, $\boldsymbol{x} \in \mathcal{X}^d$ and $h \in \mathcal{H}$ such that $\mathbb{P}(\boldsymbol{H} = \boldsymbol{h} \mid M_i = t) > 0$. Here $\boldsymbol{H}$ is a hint mechanism that takes value in the space $\mathcal{H}$ and it is a random vector defined by us given $\boldsymbol{M}$ and $\tilde{\boldsymbol{X}}$. $\mathbb{P}$ is the underlying probability measure and to keep notations simple, we assume, without loss of generality, all the random variables involved are discrete. Note that this result holds without any assumptions on the joint distribution of $(\boldsymbol{X}, \boldsymbol{M})$. Hence, we can utilize this to extend beyond the naive MCAR assumption.

2.3.1 MAR and AMAR: Missing mechanism

A missing model is the specification of the conditional distribution $\boldsymbol{M}|\boldsymbol{X}$ which governs the missing data generation process. A missing mechanism is certain assumptions made to $\boldsymbol{M}|\boldsymbol{X}$ that can be satisfied by a set of missing models. Three classic types of missing mechanisms are MCAR, MAR (Missing At Random) and MNAR (Missing Not At Random) (Little and Rubin 2019). Roughly speaking, MCAR means whether the data is missing or not is independent of the data, MAR requires the probability of missingness only depends on the observed data and MNAR allows missingness to depend on unobserved data. There are some subtleties in the definition of MAR. As it involves the observed data, do we mean that the assumption is only being made on our realized sample at hand $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i)$, $i = 1, ., n$ or on any future sample that we may observe? Clearly the latter is a stronger assumption. This has been made clear and discussed thoroughly in Seaman et al. (2013) and Mealli and Rubin (2015). We follow Mealli and Rubin (2015) to define MAR as assuming probability of missingness depends only on realized samples, and AMAR (Always Missing At Random) as depending on any future sample. Formally, we say $(\boldsymbol{X}, \boldsymbol{M})$ is **MAR** given the realized sample $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i)$, $i = 1, \ldots, n$ if

$$\mathbb{P}\Big(\boldsymbol{M} = \boldsymbol{m}_i \,\Big|\, \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}\}\Big) = \mathbb{P}\Big(\boldsymbol{M} = \boldsymbol{m}_i \,\Big|\, \boldsymbol{X} = \boldsymbol{x}\Big) \tag{8}$$

for any $i = 1, \ldots, n$ and $\boldsymbol{x}$ such that $\boldsymbol{x}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}$.

$(\boldsymbol{X}, \boldsymbol{M})$ is **AMAR** if

$$\mathbb{P}(\boldsymbol{M} = \boldsymbol{m} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}\}) = \mathbb{P}(\boldsymbol{M} = \boldsymbol{m} \mid \boldsymbol{X} = \boldsymbol{x}) \tag{9}$$

for any $\boldsymbol{m} \in \{0, 1\}^d$ and $\boldsymbol{x} \in \mathcal{X}^d$. By the property of conditional probability and the fact that $\{\boldsymbol{X} = \boldsymbol{x}\}$ implies $\{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}\}$, (8) and (9) are equivalent to

$$\begin{aligned}&\mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}, \boldsymbol{M} = \boldsymbol{m}_i\})\\ &\quad = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}\})\end{aligned} \tag{10}$$

and

$$\begin{aligned}&\mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}, \boldsymbol{M} = \boldsymbol{m}\})\\ &\quad = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}\})\end{aligned} \tag{11}$$

Given a fixed $\boldsymbol{m}_0 \in \{0, 1\}^d$, we emphasize here that both conditions do not imply the conditional independence of $\boldsymbol{X}$ and $\boldsymbol{M}$ given $\boldsymbol{X}|_{\boldsymbol{m}_0}$ which is a much stronger assumption that requires

$$\begin{aligned}&\mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_0} = \boldsymbol{x}|_{\boldsymbol{m}_0}, \boldsymbol{M} = \boldsymbol{m}\})\\ &\quad = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_0} = \boldsymbol{x}|_{\boldsymbol{m}_0}\})\end{aligned} \tag{12}$$

for any $\boldsymbol{m} \in \{0, 1\}^d$ and $\boldsymbol{x} \in \mathcal{X}^d$. Many different missing mechanisms can be defined through those conditional probability equations where different mechanisms correspond to different restrictions on the set of variable values that satisfy the equations. See Doretti et al. (2018) for more examples.

2.3.2 Compatible imputations

Prior work (Yoon et al. 2018) showed that, under the MCAR assumption, the ideal imputation $\hat{\boldsymbol{X}}$ has the same distribution as the original data. This is perfect but may be too stringent if we only care about the imputation quality for the missing data given the observed data. Thus, we define two compatible conditions for imputation. We say $\hat{\boldsymbol{X}}$ is a compatible imputation for the missing data $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i)$, $i = 1, \ldots, n$ if

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}, \boldsymbol{M} = \boldsymbol{m}_i\}) =\\ &\quad \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}, \boldsymbol{M} = \boldsymbol{m}_i\})\end{aligned} \tag{13}$$

for any $i = 1, \ldots, n$ and $\boldsymbol{x}$ such that $\boldsymbol{x}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}$. We say $\hat{\boldsymbol{X}}$ is always compatible for imputing $(\boldsymbol{X}, \boldsymbol{M})$ if

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}, \boldsymbol{M} = \boldsymbol{m}\}) =\\ &\quad \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}, \boldsymbol{M} = \boldsymbol{m}\})\end{aligned} \tag{14}$$

for any $\boldsymbol{m} \in \{0,1\}^d$ and $\boldsymbol{x} \in \mathcal{X}^d$. The compatible conditions are really what we desire for imputations. We will show that the GAN imputation $\hat{\boldsymbol{X}}$ still enjoys compatibility for many missing mechanisms beyond MCAR.

2.3.3 EMAR and EAMAR: extended missing mechanisms

We formally define a new missing mechanism that we call EMAR (Extended Missing At Random). Formally, we say $(\boldsymbol{X}, \boldsymbol{M})$ is EMAR given the realized sample $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i), i = 1, \ldots, n$ if

$$\begin{aligned} &\mathbb{P}\left(\boldsymbol{X} = \boldsymbol{x} \,\middle|\, \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}, \boldsymbol{M} = \boldsymbol{m}\}\right) \\ &\quad = \mathbb{P}\left(\boldsymbol{X} = \boldsymbol{x} \,\middle|\, \{\boldsymbol{X}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}\}\right) \end{aligned} \tag{15}$$

for any $i = 1, \ldots, n$, $\boldsymbol{x}$ such that $\boldsymbol{x}|_{\boldsymbol{m}_i} = \tilde{\boldsymbol{x}}_i|_{\boldsymbol{m}_i}$ and $\boldsymbol{m} = \boldsymbol{m}_i$ or $\mathbf{1}$. $(\boldsymbol{X}, \boldsymbol{M})$ is EAMAR (Extended Always Missing At Random) if

$$\begin{aligned} &\mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}, \boldsymbol{M} = \boldsymbol{m}'\}) \\ &\quad = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x} \mid \{\boldsymbol{X}|_{\boldsymbol{m}} = \boldsymbol{x}|_{\boldsymbol{m}}\}) \end{aligned} \tag{16}$$

for any $\boldsymbol{x} \in \mathcal{X}^d$ and $\boldsymbol{m}' = \boldsymbol{m}$ or $\mathbf{1}$. Examples of EMAR or EAMAR are much more prevalent in real datasets instead of MCAR, e.g., a ranking dataset with query-group columns that are always observed. For both missing mechanisms, we can then state:

Theorem 1 *EMAR on $(\boldsymbol{X}, \boldsymbol{M})$ giventhe realized sample $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i), i = 1, \ldots, n$ is a sufficient condition for $\hat{\boldsymbol{X}}$ being a compatible imputation for the missing data $(\tilde{\boldsymbol{x}}_i, \boldsymbol{m}_i), i = 1, \ldots, n$. EAMAR on $(\boldsymbol{X}, \boldsymbol{M})$ is a sufficient condition for $\hat{\boldsymbol{X}}$ being always compatible for imputing $(\boldsymbol{X}, \boldsymbol{M})$.*

Proof Let $\boldsymbol{I}$ be a uniform random subset of $\{1, 2, \ldots, d\}$ that is independent of $(\boldsymbol{X}, \boldsymbol{M}, \hat{\boldsymbol{X}})$. Given $\boldsymbol{I}$, define the random variable J that uniformly takes value in $\{1, 2, \ldots, d\} \backslash \boldsymbol{I}$. Given $\tilde{\boldsymbol{X}}, \boldsymbol{M}, \boldsymbol{I}, J$, we define a pair of hint vectors $(\boldsymbol{H}, \tilde{\boldsymbol{H}})$:

$$H_j = \begin{cases} \tilde{X}_j & j \in \boldsymbol{I} \\ \# & j \notin \boldsymbol{I} \end{cases}, \quad \tilde{H}_j = \begin{cases} M_j & j \neq J \\ 0.5 & j = J \end{cases} \tag{17}$$

where $\# \notin \mathcal{X}$ and it is different from the symbol $*$ that indicates missing.

We will see that the proof for EAMAR implies always compatible and EMAR for realization implies compatible for realization is the same. We will focus on the former. Take any $\boldsymbol{x} \in \mathcal{X}^d$, $\boldsymbol{m}_0 \in \{0,1\}^d$ and let $I_0 = \{j : j \in \{1, 2 \ldots, d\}, m_{0j} = 1\}$. Also take $j_0 \in \{1, 2, \ldots, d\} \backslash I_0$. Let $\boldsymbol{h} = (h_1, \ldots, h_d)$ and $\tilde{\boldsymbol{h}} = (\tilde{h}_1, \ldots, \tilde{h}_d)$ to be

$$h_j = \begin{cases} \tilde{x}_j & j \in I_0 \\ \# & j \notin I_0 \end{cases}, \quad \tilde{h}_j = \begin{cases} m_{0j} & j \neq j_0 \\ 0.5 & j = j_0 \end{cases}$$

From (7), we have

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}|\,\{\boldsymbol{H}=\boldsymbol{h},\tilde{\boldsymbol{H}}=\tilde{\boldsymbol{h}},M_{j_0}=0\})\\=&\,\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{H}=\boldsymbol{h},\tilde{\boldsymbol{H}}=\tilde{\boldsymbol{h}},M_{j_0}=1\}).\end{aligned}\tag{18}$$

Let $\boldsymbol{m}_0^t$ to be the vector that equals $\boldsymbol{m}_0$ component-wise except for $m_{0j_0}^t=t$. Note that the event $\{\boldsymbol{H}=\boldsymbol{h},\tilde{\boldsymbol{H}}=\tilde{\boldsymbol{h}},M_{j_0}=t\}$ is equivalent to $\{\boldsymbol{I}=I_0,J=j_0,X_j=x_j,\forall j\in I_0,M_{j'}=m_{0j'},\forall j'\neq j_0,M_{j_0}=t\}$. So we have

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{H}=\boldsymbol{h},\tilde{\boldsymbol{H}}=\tilde{\boldsymbol{h}},M_{j_0}=t\})\\&\quad=\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{X_j=x_{0j},\forall j\in I_0,M_{j'}=m_{0j'},\forall j'\neq j_0,M_{j_0}=t\})\\&\quad=\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0^t\})\end{aligned}\tag{19}$$

where the first equality is because of $\boldsymbol{I}$ and J's independence with other random variables. From (18)(19), we have

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0^1\})\\&\quad=\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0^0\})\end{aligned}\tag{20}$$

By taking all the other $j_0'\in\{1,2,\ldots,d\}\backslash I_0$ and follow the same procedure, we see that for any $\boldsymbol{m}\in\{0,1\}^d$ such that $\boldsymbol{m}\geq\boldsymbol{m}_0$ (componentwise), we have

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}\})\\&\quad=\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0^0\})\\&\quad=\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0\})\end{aligned}\tag{21}$$

On the other hand, for the special case of $\boldsymbol{m}=\mathbf{1}$, we have

$$\begin{aligned}&\mathbb{P}(\hat{\boldsymbol{X}}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\mathbf{1}\})\\&\quad=\mathbb{P}(\boldsymbol{X}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\mathbf{1}\})\\&\quad=\mathbb{P}(\boldsymbol{X}=\boldsymbol{x}\mid\{\boldsymbol{X}|_{\boldsymbol{m}_0}=\boldsymbol{x}|_{\boldsymbol{m}_0},\boldsymbol{M}=\boldsymbol{m}_0\})\end{aligned}\tag{22}$$

where the first equality holds because given $\boldsymbol{M}=\mathbf{1}$, $\hat{\boldsymbol{X}}=\boldsymbol{X}$ and the second one is due to the EAMAR assumption (16). Thus combining (21), (22), we prove the theorem.

To summarize, we demonstrated the advantages of Conditional Imputation GAN by showing the compatibility of optimal imputations under extended missing mechanisms EMAR and EAMAR. For the observed missing patterns, EMAR requires that the data distribution conditional on the observed values is the same as if they were not missing. EMAR is a stronger assumption compare to MAR, but it is much less restrictive than MCAR. In Theorem 1, we proved that EMAR, which includes a collection of missing models, is a sufficient condition for compatibility of optimal GAN imputation, and

Table 1 Synthetic ranking data: feature distribution by category

		Books	Furniture	Beauty	Clothes	Electronics
Gaussian	$N(\mu, \sigma^2)$	$N(0, 1)$	$N(1, 1)$	$N(2, 1)$	$N(3, 1)$	$N(4, 1)$
LogNormal	$LN(\mu, \sigma^2)$	$LN(2, 1)$	$LN(1.5, 1)$	$LN(1, 1)$	$LN(0.5, 1)$	$LN(0, 1)$
Exponential	$Exp(\lambda)$	$Exp(0.5)$	$Exp(1)$	$Exp(1.5)$	$Exp(2)$	$Exp(2.5)$
Poisson	$Poisson(\lambda)$	$Pois(2)$	$Pois(4)$	$Pois(6)$	$Pois(8)$	$Pois(10)$
Uniform	$Unif(a, b)$	$Unif(0, 2)$	$Unif(0, 4)$	$Unif(0, 6)$	$Unif(0, 8)$	$Unif(0, 10)$

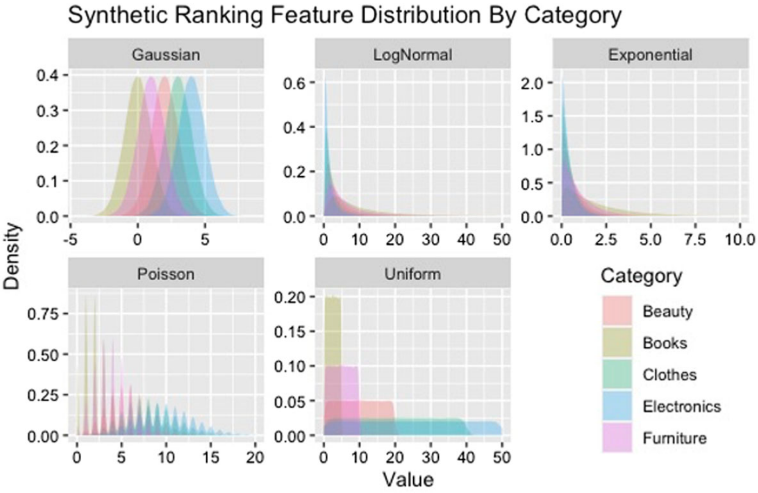

Fig. 3 Synthetic—ranking feature dist. by category. Ranking features are simulated to follow both discrete and continuous distributions commonly found in empirical ranking datasets

EAMAR is a sufficient condition for always compatibility. Whether the optimal GAN imputation is compatible under MNAR remains open for future work.

3 Simulation: imputation quality by data distribution

3.1 Data and methodology

To illustrate how our method performs across a variety of data distributions, we first simulate a 10K query-group ranking dataset, with feature columns sampled from 5 distributions: Gaussian, LogNormal, Exponential, Poisson, Uniform. Each query-group has 64 query-results and is associated with a hypothetical product type (“Category”) that is always observed, in accordance with the EMAR and EAMAR assumption. The 5 types are Books, Furniture, Beauty, Clothes, Electronics. The product category determines the true distribution parameters from which the ranking features are sampled; see Table 1 for details and Fig. 3 for how ranking feature distributions vary by product type.

We want to compare imputation quality as measured by RMSE across four methods: Conditional Imputation GAN, GAIN, MICE, and MissForest. Given that the Category column will always be observed, we select four levels of missingness (5%, 10%, 20%, 30%) and randomly mask feature values in each query-group as missing; this is aligned with the EMAR missing mechanism.

For each method and missingness level, 10 imputations of the simulated ranking dataset are generated. In TensorFlow[3], each GAN replicate was trained for 50 epochs with mini-batch size 256 after standardizing the data to zero mean and unit variance; the Adam optimizer (Kingma and Ba 2015) is used with default learning rate 0.001. No dropout was used given the moderate data size. The generator and discriminator both utilized a standard architecture of fully-connected layers with leaky-relu activation. In R, default settings of MissForest and MICE are used, with only a 10% random sample used for MissForest given the computational cost. We then compute average RMSE and standard errors over imputations across all ranking features and also separately (column-wise) for features from each distribution. See Table 2.

3.2 Simulation results

Conditional Imputation GAN yields the best RMSE overall and for each type of data distribution, performing fairly consistently for each feature. Our method performed particularly well for Gaussian, Poisson, and Uniform distributed features; the slightly higher RMSE for Log Normal and Exponential distribution is due to sampling from uniform initial starting values for the two right-skewed distributions, and is mitigated with longer training time. In practice, initial values can also be sampled from the distribution of observed values for each feature. In contrast, both of the non-GAN benchmarks MICE and MissForest have reasonable RMSE for Gaussian and Exponential distributions, but perform poorly with Log Normal and Uniform distributions. Furthermore, these two benchmarks had higher standard errors across distributions, an indication that GAN-based methods provide more robust imputations overall. These results validate the flexibility of Conditional GAN-based imputations given the ground truth of different underlying distribution by category. Conditioning on auxiliary information greatly improved imputation quality under non-MCAR mechanisms.

4 MSR ranking data: imputation quality by heterogeneous subgroups

4.1 Data and methodology

To demonstrate how the Conditional Imputation GAN accounts for heterogeneous subgroups within ranking data, we utilize the 10K query-group ranking dataset (MSLR-WEB10K) (Qin and Liu 2013) made available from Microsoft Research. The Fold 1 dataset is split into roughly 80% training (∼724 K rows) and 20% testing (∼242 K rows), with 136 features. Each query-group represents a search query in Microsoft Bing, and each row represents one webpage url returned as a poten-

[3] Code available at https://github.com/gdeng96/cond-imp-gan-ranking.

Table 2 Simulation results—imputation quality (RMSE) by distributions

Distribution	Method	5% Missing	10% Missing	20% Missing	30% Missing
Overall	Cond. Imp. GAN	**0.841** ± (0.0002)	**0.861** ± (0.0001)	**0.893** ± (0.0001)	**0.91** ± (0.0001)
	GAIN	1.014 ± (0.0002)	1.028 ± (0.0001)	1.063 ± (0.0001)	1.08 ± (0.0002)
	MICE	8.914 ± (0.1911)	9.011 ± (0.1063)	9.129 ± (0.2074)	6.536 ± (0.0683)
	MissForest	6.536 ± (0.0683)	6.514 ± (0.0831)	7.018 ± (0.0338)	7.018 ± (0.0338)
Exponential	Cond. Imp. GAN	**0.988** ± (0.0004)	**1.037** ± (0.0002)	**0.971** ± (0.0002)	**0.979** ± (0.0002)
	GAIN	1.161 ± (0.0003)	1.044 ± (0.0002)	1.009 ± (0.0001)	1.004 ± (0.0001)
	MICE	1.584 ± (0.0313)	1.574 ± (0.0144)	1.587 ± (0.0113)	1.068 ± (0.0330)
	MissForest	1.068 ± (0.0330)	1.18 ± (0.0174)	1.212 ± (0.0173)	1.212 ± (0.0173)
Gaussian	Cond. Imp. GAN	**0.707** ± (0.0003)	**0.715** ± (0.0002)	**0.706** ± (0.0002)	**0.874** ± (0.0003)
	GAIN	0.9 ± (0.0004)	1.098 ± (0.0003)	1.075 ± (0.0002)	1.108 ± (0.0001)
	MICE	1.731 ± (0.0073)	1.74 ± (0.0086)	1.781 ± (0.0046)	1.303 ± (0.0112)
	MissForest	1.303 ± (0.0112)	1.292 ± (0.0106)	1.42 ± (0.0124)	1.42 ± (0.0124)
LogNormal	Cond. Imp. GAN	**0.966** ± (0.0003)	**0.926** ± (0.0002)	**1.016** ± (0.0002)	**0.972** ± (0.0002)
	GAIN	0.997 ± (0.0003)	0.958 ± (0.0003)	1.139 ± (0.0001)	1.242 ± (0.0001)
	MICE	13.468 ± (0.6489)	13.856 ± (0.3277)	14.022 ± (0.6454)	9.187 ± (0.2773)
	MissForest	9.187 ± (0.2773)	9.526 ± (0.1508)	10.483 ± (0.0765)	10.483 ± (0.0765)
Poisson	Cond. Imp. GAN	**0.74** ± (0.0003)	**0.781** ± (0.0001)	**0.794** ± (0.0002)	**0.852** ± (0.0002)
	GAIN	0.995 ± (0.0003)	0.996 ± (0.0002)	1.064 ± (0.0001)	0.948 ± (0.0001)
	MICE	3.908 ± (0.0273)	3.953 ± (0.017)	4.039 ± (0.0127)	2.979 ± (0.0371)
	MissForest	2.979 ± (0.0371)	3.036 ± (0.0268)	3.218 ± (0.0175)	3.218 ± (0.0175)
Uniform	Cond. Imp. GAN	**0.756** ± (0.0003)	**0.808** ± (0.0002)	**0.937** ± (0.0003)	**0.865** ± (0.0002)
	GAIN	1 ± (0.0003)	1.038 ± (0.0002)	1.022 ± (0.0001)	1.076 ± (0.0001)
	MICE	13.862 ± (0.1199)	13.89 ± (0.0566)	14.044 ± (0.0502)	10.587 ± (0.0871)
	MissForest	10.587 ± (0.0871)	10.512 ± (0.1881)	11.13 ± (0.0836)	11.13 ± (0.0836)

Bold values indicate that the associated results (imputation as measured by RMSE, etc.) are the "best" out of all benchmarks/method comparisons

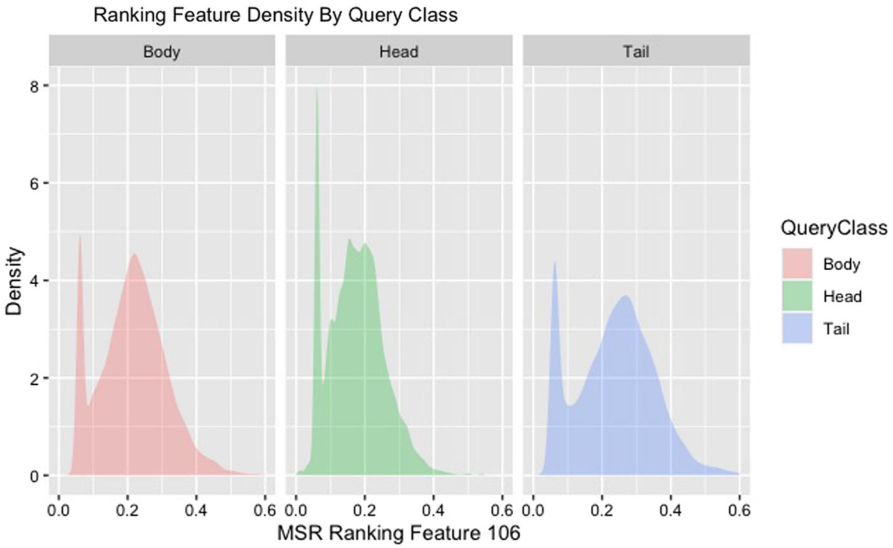

Fig. 4 MSR—Ranking Feature 106 distribution by Query Class. Note that tail queries display a lower concentrated peak between 0.05 and 0.1 and greater variance with more observations skewed to the right. Conditioning on Query Class allows for imputation with respect to these subgroups

tial match. Here note that each query-group can be categorized as Head, Body, or Tail queries depending on the number of urls found; this feature will be henceforth referred to as "Query Class" and is also an always observed column in practice. Head and tail queries usually indicate very different subgroups and ranking feature distributions; see Table 4 for an example. Query Class also influences the probability of missingness. For example, in an e-commerce setting, tail queries indicate rare or newly-launched items with higher probability of missing values due to lack of data. This would violate the naive MCAR setting and is an example of the extended EMAR and EAMAR mechanisms. Imputation quality will again be measured by RMSE averaged across multiple imputations (Fig. 4).

Similar to the simulation experiment, we condition on Query Class and select four levels of missingness (5%, 10%, 20%, 30%), with values from 5 ranking features randomly masked as missing. In TensorFlow, we train each GAN replicate using the same hyperparameters and generator-discriminator architecture as the previous experiment. In R, we use default MissForest and MICE settings to implement the two missing data imputation methods. 10 imputations of test ranking dataset are generated for each method and missingness level.

4.2 MSR results

Conditional Imputation GAN yields the lowest RMSE across the board, especially for higher missingness levels; it also learns ranking feature distributions with respect to auxiliary information such as Query Class. See Table 3, which compares average RMSE and standard errors against benchmarks.

Table 3 MSR results—imputation quality (RMSE)

Method	5% Missing	10% Missing	20% Missing	30% Missing
Cond. Imp. GAN	**0.232**	**0.237**	**0.239**	**0.239**
	(± 0.00001)	(± 0.00001)	(± 0.00001)	(± 0.00003)
GAIN	0.234	0.240	0.240	0.248
	(± 0.00002)	(± 0.0001)	(± 0.0001)	(± 0.0001)
MICE	0.309	0.311	0.312	0.314
	(± 0.0011)	(± 0.00057)	(± 0.00054)	(± 0.00044)
MissForest	0.240	0.243	0.249	0.250
	(± 0.00227)	(± 0.0027)	(± 0.00277)	(± 0.00202)

Bold values indicate that the associated results (imputation as measured by RMSE, etc.) are the "best" out of all benchmarks/method comparisons

In terms of non-GAN benchmarks, MissForest outperformed MICE. However, we do not recommend implementing MissForest for a large ranking dataset in general from a computational standpoint. Given that it is based on random forests, implementation can be computationally expensive (Tang and Ishwaran 2017). On a standard Google Colab notebook, both GAN imputations took only about 1 minute to train for the full MSR ranking data, and similarly only about 1 minute to train for default settings of MICE (5 iteration per multiple imputation). Meanwhile, it took more than 10 minutes to run for default settings of MissForest (10 iterations) on just a 10% MSR data sample, even though the data is low-dimensional.

5 Amazon Search ranking data

5.1 Data and methodology

We now utilize an Amazon Search ranking dataset consisting of 1.8 million query-groups. This large ranking dataset is used to conduct experiments in order to answer two key questions: (1) *What's the highest missingness level where Conditional Imputation GAN can still deliver good results?* (2) *How does imputation quality translate to downstream applications, e.g., ranking quality?*

A collection of 24 common features for ranking are chosen and fall under 3 groups: behavioral, semantic, and product characteristics. Ranking features selected have varying ranges, units, and spread, which increases the difficulty of imputation. Each feature is normalized by the mean and standard deviation, maintaining the same correlation structure as the original features. Higher correlation is observed amongst ranking features that belong to the same group. For confidentiality reasons, individual feature names are masked and referenced as Feature A, B, C, etc.

Similar to previous experiments, the Amazon ranking dataset is also characterized by descriptive columns that are always observed and ranking features with potentially missing values. Descriptive columns includes a 21-class column of product categories, which are quite imbalanced; conditional on these observed values, ranking features can

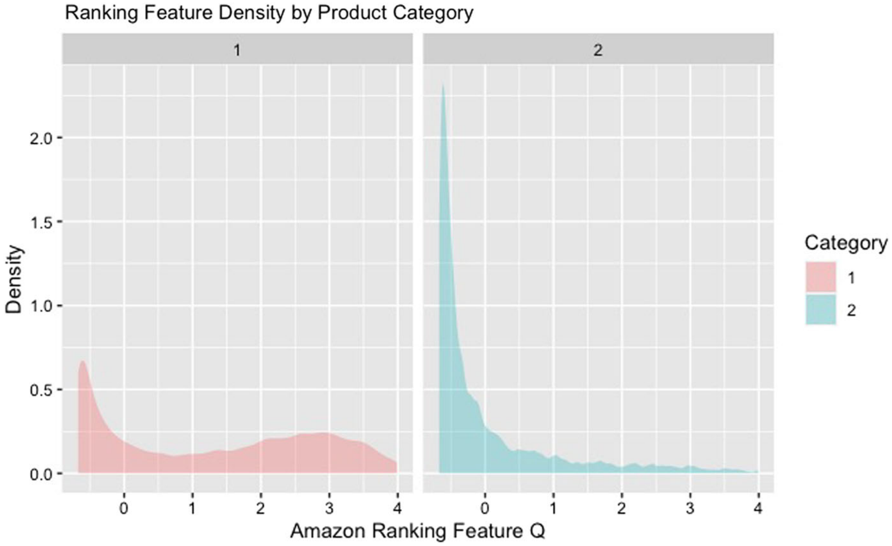

Fig. 5 Amazon—ranking feature dist. by category. Conditional distributions are significantly different, which motivates conditioning on product categories to improve imputation quality

display significantly disparate underlying distributions. For example, a histogram of Feature Q across two product categories 1 and 2 is bi-modal and heavily right-skewed, respectively; see Fig. 5 where conditional distributions are significantly different by both t-test and Kolmogorov-Smirnov, $p < 0.05$. Hence, it is logical to condition on these product categories as auxiliary information to improve imputation quality.

5.2 Imputation quality: RMSE by missingness level

To evaluate imputation quality by missingness level, we sample 500K query-groups for training and evaluate RMSE on a 300K query-group hold-out set. For each method, we specify product category columns that are always observed as auxiliary information to condition upon, and randomly mask ranking feature values as missing from 10% to 90%. This is aligned with the more expansive EMAR missing mechanism, of which MCAR is a special case.

10 imputations of the test set are generated for each missing percentage and method, and RMSE is computed based on imputed vs. true feature values. On a ml.m5.24xlarge AWS instance, each GAN replicate is trained for 1000 epochs using default Adam optimizer learning rate (0.001) and mini-batch size 256. Data is again standardized to zero mean and unit variance. The generator-discriminator architecture follows a standard CGAN architecture with 3 fully connected layers, batch normalization, and leaky-relu activation. Imputation results generated by GAIN is included as a reference.

Figure 6 charts the increase in RMSE as missingness level is increased from 10%; a good initial missing proportion would be 30% or lower. The Conditional Imputation GAN results in lower RMSE across the board by better learning the disparate underly-

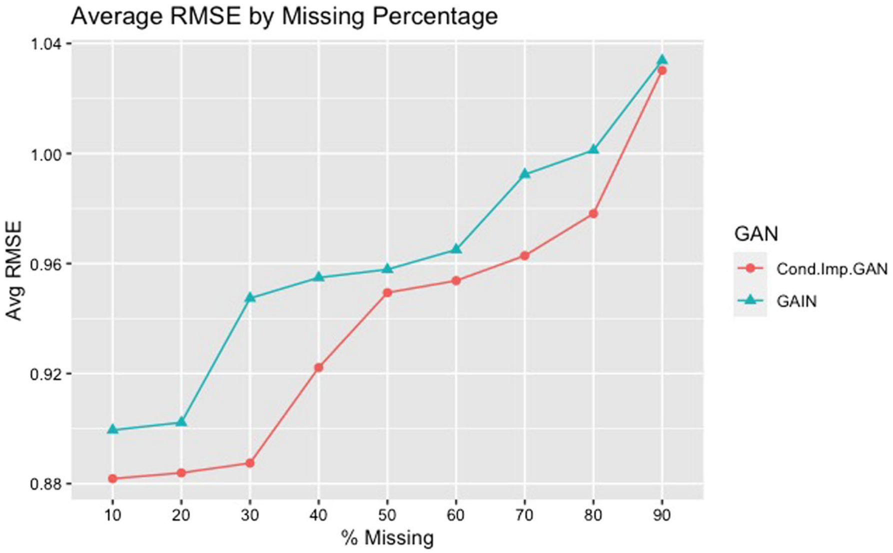

Fig. 6 Amazon results—imputation quality (RMSE) by missingness level. Conditional Imputation GAN imputations have lower errors than GAIN on average at all missingness levels

ing distributions conditional on auxiliary columns, and is a testimony to the flexibility of GAN-generated imputations overall. In addition, our method also demonstrates comparable performance at higher missingness levels; the RMSE for 50% missing is equivalent to that of GAIN at just 30% missing.

5.3 Ranking quality: ranking models with imputed data

To illustrate the downstream effect of imputed training data on ranking quality, we compare NDCG and MRR for standard ranking models trained on imputed versus baseline ground-truth models. We sample 500K query-groups each for training, testing, and validation sets. A standard model choice is the pairwise LambdaMart (Burges 2010) model trained through LightGBM (Ke et al. 2017), a computationally efficient gradient boosting framework.

For the training set, we fix always observed columns and randomly mask 20% of feature values as missing. 30 imputations of the ranking dataset are generated using the Conditional Imputation GAN with the same hyperparameters (learning rate is 0.001, mini-batch size is 256, etc.) and generator-discriminator set-up as the previous experiment; 30 imputations are generated via GAIN as a reference point. We generate 30 imputations in order to ensure sufficient sample size for independent two-sample t-tests that will later be used to compare ranking quality metrics NDCG and MRR. Given the large-scale dataset and longer training time, we also optionally included dropout layers for regularization purposes. Other methods such as MissForest do not scale in this case due to computational cost; a 500K query-group dataset with 200+ results per group easily exceeds 100 million rows of data.

The imputed ranking datasets are then used to train a standard ranking model using LightGBM, a total of 60 models with 30 for each GAN architecture. All models share the same binary target with the same testing and validation ranking set. Final ranking model metrics are directly comparable; specific computations of NDCG and MRR are discussed below.

5.3.1 NDCG and MRR

Ranking quality will be measured by two metrics, Normalized Discounted Cumulative Gain and Mean Reciprocal Rank. As a key information retrieval metric, Normalized Discounted Cumulative Gain (NDCG) (Valizadegan et al. 2009) measures ranking quality by summarizing the gains from a particular ranking order. It is standardized by position and is between [0, 1]. We first define Discounted Cumulative Gain (DCG) at position p, that is, for the top p results returned by a ranking model as:

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

where rel_i is the ranking score of result at position i for query q, as predicted by an arbitrary ranking model. Greater penalty is given for relevant results ranked in lower positions. DCG is divided by Ideal Discounted Cumulative Gain (IDCG), the maximum possible DCG through position p. If ranking model orders a set of results in the optimal order possible, the NDCG will be equal to 1. That is,

$$\mathrm{NDCG_p} = \frac{DCG_p}{IDCG_p}, \quad \mathrm{IDCG_p} = \sum_{i=1}^{|rel_p|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

and $|rel_p|$ represents the optimal order of search results up to position p. This optimal order is usually known via the target column. The final NDCG given by a specific ranking model or search engine algorithm can be computed as the average of the NDCG for each query-group in the testing ranking dataset, and is directly comparable across different models or algorithms.

The second metric, Mean Reciprocal Rank (MRR) (Radev et al. 2002), measure ranking quality by evaluating the probability of the first correct answer for a given query. Specifically

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\mathrm{rank}_q}$$

where rank_i is the rank position of the first relevant result for query q in a dataset with Q query-groups.

Table 4 Amazon results—imputation quality (RMSE)

GAN structure	Test RMSE	Std. Err.
Cond. Imp. GAN w/Dropout	**0.881**	0.00004
Cond. Imp. GAN	0.930	0.00004
GAIN w/Dropout	0.963	0.00004
GAIN	0.985	0.00003

Bold value indicates that the associated results (imputation as measured by RMSE, etc.) are the "best" out of all benchmarks/method comparisons

Table 5 Amazon results—ranking model metrics: conditional imputation GAN versus GAIN w/Dropout

	Mean diff	t-stat	df*	p-val
NDCG_{10}	0.001	5.967	39.769	0
NDCG Gain	0.172	5.970	39.800	0
MRR_{10}	0.001	5.336	40.091	0
MRR Gain	0.185	5.340	40.100	0

*Satterthwaite approximation for degrees of freedom

5.3.2 Results

From Table 4, imputations generated by a Conditional Imputation GAN with dropout layers resulted in the lowest RMSE on average and is 8.5% lower ($p < 0.05$) than the benchmark GAIN equivalent. The additional improvement from dropout illustrates that other neural net training techniques can further optimize imputation based on data type.

We now compare ranking models trained on the imputed datasets, as measured by four performance metrics that indicate ranking quality. NDCG_{10} and MRR_{10} are defined in Sect. 5.3.1 and computed based on the first 10 results per query-group as returned by the ranking model. NDCG and MRR Gain refer to percentage-wise improvement compared the baseline ranking model. In terms of all four metrics, the ranking models trained on imputations from Conditional GANs demonstrated statistically significant gains than those from the vanilla GAIN; see Table 5. This result holds even after controlling for false discovery rates using Benjamini-Hochberg procedure. Unequal variances are assumed and Satterthwaite approximation for degrees of freedom (df*) is used. We emphasize here that given the volume of queries, even a minor but statistically significant improvement in performance is important for impact on downstream applications.

6 Conclusion

We have demonstrated a novel Conditional Imputation GAN for extended missing mechanisms in ranking applications. Theoretical analysis showed compatible imputation guarantees for EMAR and EAMAR mechanisms that encompass a broader collection of missing models and datasets. Using a variety of ranking datasets, we showcase the superior imputation quality of our method against standard benchmarks. Experiment results illustrate the flexibility of the method, which generalizes well across a range of distributions and heterogeneous subgroups specified by always observed columns. GAN-based imputation approaches also scale computationally for very large datasets compared to the random-forest based MissForest, and generalizes better for complex missing mechanisms compared to the MCAR assumptions of MICE.

In particular, simulations with five different distributions show that Conditional Imputation GAN outperformed traditional imputation methods such as MICE and MissForest, especially for non-Gaussian distributions. Furthermore, our method's imputations had lower standard errors overall which attests to the robustness across multiple imputations. Results using the open-source MSR ranking dataset confirm that Conditional Imputation GAN adapts well to multi-modal distributions that vary significantly conditional on auxiliary information, which other benchmarks fail to capture. Finally, experiments with the proprietary 1.8 million query-group Amazon ranking dataset demonstrate that downstream ranking models trained on imputed data also perform well as measured by NDCG and MRR.

Future work can explore whether GAN imputation optimality is possible under challenging missing mechanisms such as MNAR, and whether more complex GAN architecture (e.g. multi-task multi-label) could also benefit GAN imputation quality.

Acknowledgements The authors gratefully acknowledge financial support from the National Science Foundation Awards 1934985, 1940124, 1940276, and 2114143.

A Additional MSR experiments

To further explore the effect of missingness levels on accuracy for the MSR dataset, we repeat experiments in Sect. 4 with additional missingness levels from 40 to 80%. The exact same data standardization and training hyperparameters in Sect. 4 are used for both Conditional Imputation GAN and GAIN; default settings for MICE and MissForest are also used in their respective `R` packages. Imputation quality measured by average RMSE over 10 replicates with corresponding standard errors are reported in Table 6; lower missing percentage results are reported in Table 3. We note that Conditional Imputation GAN performs best for all missingness levels up to 70% and consistently outperforms benchmark GAIN, which is evidence that the auxiliary information provided by Query Class labels augments the imputation process. Although MissForest has slightly lower error than Conditional Imputation GAN for 80% missing, we note that these very high missingness levels are subject to greater imputation variability and that in terms of computational cost, GAN-based methods would scale much better for very large datasets. MICE consistently performs lowest out of all imputation methods.

Table 6 MSR results for high missingness—imputation quality (RMSE)

Method	40% Missing	50% Missing	60% Missing	70% Missing	80% Missing
Cond. Imp. GAN	**0.244**	**0.245**	**0.247**	**0.254**	0.267
	(0.00001)	(0.00001)	(0.00001)	(0.00001)	(0.00002)
GAIN	0.252	0.263	0.271	0.272	0.299
	(0.00001)	(0.00001)	(0.00001)	(0.00002)	(0.00002)
MICE	0.315	0.316	0.318	0.320	0.322
	(0.00043)	(0.00043)	(0.00035)	(0.00024)	(0.00024)
MissForest	0.250	0.252	0.257	0.257	**0.264**
	(0.00367)	(0.00113)	(0.00502)	(0.007)	(0.00549)

Bold values indicate that the associated results (imputation as measured by RMSE, etc.) are the "best" out of all benchmarks/method comparisons

References

Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning, PMLR, pp 214–223

Burges CJ (2010) From RankNet to LambdaRank to LambdaMART: an overview. Learning 11(23–581):81

Camino RD, Hammerschmidt CA, State R (2019) Improving missing data imputation with deep generative models. arXiv preprint arXiv:1902.10666

Doretti M, Geneletti S, Stanghellini E (2018) Missing data: a unified taxonomy guided by conditional independence. Int Stat Rev 86(2):189–204

Goodfellow I, Pouget-Abadie J, Mirza M, et al (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680

Guo Z, Wan Y, Ye H (2019) A data imputation method for multivariate time series based on generative adversarial network. Neurocomputing 360:185–197

Heitjan DF, Basu S (1996) Distinguishing missing at random and missing completely at random. Am Stat 50(3):207–213

Ke G, Meng Q, Finley T, et al (2017) LightGBM: a highly efficient gradient boosting decision tree. In: Advances in neural information processing systems, pp 3146–3154

Kim J, Tae D, Seok J (2020) A survey of missing data imputation using generative adversarial networks. In: 2020 International conference on artificial intelligence in information and communication (ICAIIC), IEEE, pp 454–456

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: ICLR (Poster)

Lee D, Kim J, Moon WJ, et al (2019) CollaGAN: Collaborative gan for missing image data imputation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2487–2496

Li H (2011) A short introduction to learning to rank. IEICE Trans Inf Syst 94(10):1854–1862

Li SCX, Jiang B, Marlin B (2018) MisGAN: learning from incomplete data with generative adversarial networks. In: International conference on learning representations

Little RJ, Rubin DB (2019) Statistical analysis with missing data, vol 793. Wiley

Luo Y, Cai X, Zhang Y, et al (2018) Multivariate time series imputation with generative adversarial networks. In: Advances in neural information processing systems, pp 1596–1607

Marlin BM, Zemel RS (2009) Collaborative prediction and ranking with non-random missing data. In: Proceedings of the third ACM conference on recommender systems, pp 5–12

Mealli F, Rubin DB (2015) Clarifying missing at random and related definitions, and implications when coupled with exchangeability. Biometrika 102(4):995–1000

Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784

Oza M, Vaghela H, Srivastava K (2020) Progressive generative adversarial binary networks for music generation. In: International conference on innovative computing and communications. Springer, pp 181–192

Qin T, Liu T (2013) Introducing LETOR 4.0 datasets. CoRR abs/1306.2597. arXiv:1306.2597

Radev DR, Qi H, Wu H, et al (2002) Evaluating web-based question answering systems. In: LREC, Citeseer

Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: 4th International conference on learning representations, ICLR 2016, conference track proceedings, San Juan, Puerto Rico

Salimans T, Goodfellow I, Zaremba W, et al (2016) Improved techniques for training gans. In: Advances in neural information processing systems, pp 2234–2242

Seaman S, Galati J, Jackson D, et al (2013) What is meant by "missing at random"? Stat Sci 28(2):257–268

Sheng L, Pan J, Guo J, et al (2019) Unsupervised bi-directional flow-based video generation from one snapshot. arXiv preprint arXiv:1903.00913

Stekhoven DJ, Bühlmann P (2012) MissForest-non-parametric missing value imputation for mixed-type data. Bioinformatics 28(1):112–118

Tang F, Ishwaran H (2017) Random forest missing data algorithms. Stat Anal Data Min ASA Data Sci J 10(6):363–377

Thanh-Tung H, Tran T (2020) Catastrophic forgetting and mode collapse in GANs. In: 2020 International joint conference on neural networks (IJCNN), IEEE, pp 1–10

Valizadegan H, Jin R, Zhang R, et al (2009) Learning to rank by optimizing NDCG measure. In: Advances in neural information processing systems, pp 1883–1891

Van Buuren S, Groothuis-Oudshoorn K (2011) mice: multivariate imputation by chained equations in R. J Stat Softw 45:1–67

Van Buuren S (2018) Flexible imputation of missing data. CRC Press

Xu B, Wang N, Chen T, et al (2015) Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853

Yoon S, Sull S (2020) GAMIN: generative adversarial multiple imputation network for highly missing data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8456–8464

Yoon J, Jordon J, Schaar M (2018) GAIN: missing data imputation using generative adversarial nets. In: International conference on machine learning, PMLR, pp 5689–5698

Zhang Y, Zhou B, Cai X et al (2021) Missing value imputation in multivariate time series with end-to-end generative adversarial networks. Inf Sci 551:67–82