M^3-Impute: Mask-guided Representation Learning for Missing Value Imputation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Missing values are a common problem that poses significant challenges to data analysis and machine learning. This problem necessitates the development of an effective imputation method to fill in the missing values accurately, thereby enhancing the overall quality and utility of the datasets. Existing imputation methods, however, fall short of considering the 'missingness' information in the data during initialization and modeling the entangled feature and sample correlations explicitly during the learning process, thus leading to inferior performance. We propose M^3-Impute, which aims to leverage the missingness information and such correlations with novel masking schemes. M^3-Impute first models the data as a bipartite graph and uses an off-the-shelf graph neural network, equipped with a refined initialization process, to learn node embeddings. They are then optimized through M^3-Impute's novel feature correlation unit (**FCU**) and sample correlation unit (**SCU**) that enable explicit consideration of feature and sample correlations for imputation. Experiment results on 15 benchmark datasets under three different missing patterns show the effectiveness of M^3-Impute by achieving 13 best and 2 second-best MAE scores on average.

1 Introduction

Missing values in a dataset are a pervasive issue in real-world data analysis. They arise for various reasons, ranging from the limitations of data collection methods to errors during data transmission and storage. Since many data analysis algorithms cannot directly handle missing values, the most common way to deal with them is to discard the corresponding samples or features with missing values, which would compromise the quality of data analysis. To tackle this problem, missing value imputation algorithms have been proposed to preserve all samples and features by imputing missing values with estimated ones based on the observed values in the dataset, so that the dataset can be analyzed as a complete one without losing any information.

The imputation of missing values usually requires modeling of correlations between different features and samples. Feature-wise correlations help predict missing values from other observed features in the same sample, while sample-wise correlations help predict them in one sample from other similar samples. It is thus important to jointly model the feature-wise and sample-wise correlations in the dataset. In addition, the prediction of missing values also largely depends on the 'missingness' of the data, i.e., whether a certain feature value is observed or not in the dataset. Specifically, the missingness information directly determines which observed feature values can be used for imputation. For example, even if two samples are closely related, it may be less effective to use them for imputation if they have missing values in exactly the same features. It still remains a challenging problem how to jointly model feature-wise and sample-wise correlations with such data missingness.

Among existing methods for missing value imputation, statistical methods [4, 9, 14, 16, 18, 19, 22, 28, 30, 31, 37, 43] extract data correlations with statistical models, which are generally not flexible

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

in handling mixed data types and struggles to scale up to large datasets. Learning-based imputation methods [10, 24, 27, 29, 33, 42, 50, 51, 53], instead, take advantage of the strong expressiveness and scalability of machine/deep learning algorithms to model data correlations. However, most of them are still built upon the raw tabular data structure as is, which greatly restricts them from jointly modeling the feature-wise and sample-wise correlations. In light of this, graph-based methods [52, 54] have been proposed to model the raw data as a bipartite graph, with samples and features being two different types of nodes. A sample node and a feature node are connected if the feature value is observed in that sample. The missing values are then predicted as the inner product between the embeddings of the corresponding sample and feature nodes. However, this simple prediction does not consider the specific missingness information as mentioned above. For instance, the target feature to impute may have different correlations with features in the samples which have different kinds of missingness; however, the *same* feature-node embedding is still used for their imputation. A similar issue also arises for sample-node embeddings.

In this work, we address these problems by proposing M^3-Impute, a mask-guided representation learning method for missing value imputation. The key idea behind M^3-Impute is to explicitly utilize the data-missingness information as model input with our proposed novel masking schemes so that it can accurately learn feature-wise and sample-wise correlations in the presence of different kinds of data missingness. M^3-Impute first builds a bipartite graph from the data as used in [52]. In the embedding initialization for graph representation learning, however, we not only use the the relationships between samples and their associated features but also the missingness information so as to initialize the embeddings of samples and features jointly and effectively. We then propose novel feature correlation unit (**FCU**) and sample correlation unit (**SCU**) in M^3-Impute to explicitly take feature-wise and sample-wise correlations into account for imputation. **FCU** learns the correlations between the target missing feature and observed features within each sample, which are then further updated via a soft mask on the sample missingness information. **SCU** then computes the sample-wise correlations with another soft mask on the missingness information for each pair of samples that have values to impute. We then integrate the output embeddings of **FCU** and **SCU** to estimate the missing values in a dataset. We carry out extensive experiments on 15 open datasets. The results show that M^3-Impute outperforms state-of-the-art methods in 13 of the 15 datasets on average under three different settings of missing value patterns, achieving up to 11.47% improvement in MAE compared to the second-best method.

2 Related Work

Statistical methods: These imputation approaches include joint modeling with expectation-maximization (EM) [9, 16, 22], k-nearest neighbors (kNN) [14, 43], and matrix completion [5, 6, 18, 32]. However, joint modeling with EM and matrix completion often lack the flexibility to handle data with mixed modalities, while kNN faces scalability issues due to its high computational complexity. In contrast, M^3-Impute is scalable and adaptive to different data distributions.

Learning-based methods: Iterative imputation frameworks [1, 2, 15, 20, 23, 24, 35, 41, 44, 45], such as MICE [45] and HyperImpute [23], have been extensively studied. These iterative frameworks apply different imputation methods for each feature and iteratively estimate missing values until convergence. In addition, for deep neural network learners, both generative models [27, 29, 36, 50, 51, 53], such as GAIN [50] and MIWAE [29], and discriminative models [10, 24, 48], such AimNet [48], have also been proposed. However, these methods are built upon raw tabular data structures, which fall short of capturing the complex correlations in features, samples, and their combination [54]. In contrast, M^3-Impute is based on the bipartite graph modeling of the data, which is more suitable for learning the data correlations for imputation.

Graph neural network-based methods: GNN-based methods [40, 52, 54] are proposed to address the drawbacks mentioned above due to their effectiveness in modeling complex relations between entities. Among them, GRAPE [52] transforms tabular data into a bipartite graph where features are one type of node and samples are the other. A sample node is connected to a feature node only if the corresponding feature value is present. This transformation allows the imputation task to be framed as a link prediction problem, where the inner product of the learned node embeddings is computed as the predicted values. IGRM [54] further enhances the bipartite graph by explicitly introducing linkages between sample nodes to facilitate message propagation between samples. However, these methods do not effectively encode the missingness information of different samples and features into

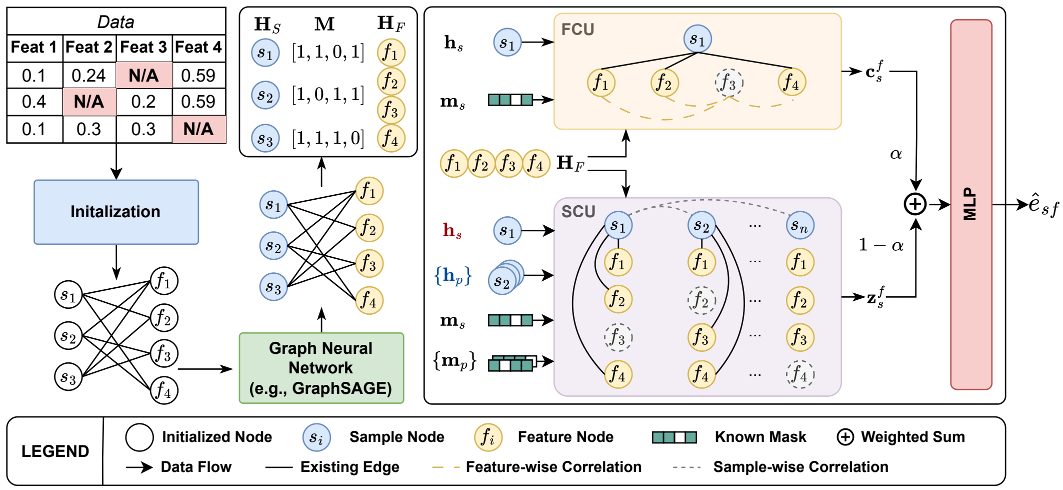

Figure 1: Overview of the M^3-Impute model.

the imputation process, which can impair their imputation accuracy. In contrast, M^3-Impute enables explicit modeling of missingness information through novel masking schemes so that feature-wise and sample-wise correlations can be accurately captured in the imputation process.

3 M^3-Impute

3.1 Overview

We here provide an overview of M^3-Impute to impute the missing value of feature f for a given sample s, as depicted in Figure 1. Initially, the data matrix with missing values is modeled as an undirected bipartite graph, and the missing value is imputed by predicting the edge weight $\hat{e}_{sf}$ of its corresponding missing edge (Section 3.2). M^3-Impute next employs a GNN model, such as GraphSAGE [17], on the bipartite graph to learn the embeddings of samples and features. These embeddings, along with the known masks of the data matrix (used to indicate which feature values are available in each sample), are then input into our novel feature correlation unit (**FCU**) and sample correlation unit (**SCU**), which shall be explained in Section 3.3 and Section 3.4, to obtain feature-wise and sample-wise correlations, respectively. Finally, M^3-Impute takes the feature-wise and sample-wise correlations into a multi-layer perceptron (MLP) to predict the missing feature value $\hat{e}_{sf}$ (Section 3.5). The whole process, including the embedding generation, is trained in an end-to-end manner.

3.2 Initialization Unit

Let $\mathbf{A} \in \mathbb{R}^{n\times m}$ be an $n \times m$ matrix that consists of n data samples and m features, where $\mathbf{A}_{ij}$ denotes the j-th feature value of the i-th data sample. We introduce an $n \times m$ mask matrix $\mathbf{M} \in \{0,1\}^{n\times m}$ for $\mathbf{A}$ to indicate that the value of $\mathbf{A}_{ij}$ is *observed* when $\mathbf{M}_{ij} = 1$. In other words, the goal of imputation here is to predict the missing feature values $\mathbf{A}_{ij}$ for i and j such that $\mathbf{M}_{ij} = 0$. We define the *masked* data matrix $\mathbf{D}$ to be $\mathbf{D} = \mathbf{A} \odot \mathbf{M}$, where $\odot$ is the Hadamard product, i.e., the element-wise multiplication of two matrices.

As used in recent studies [52, 54], we model the masked data matrix $\mathbf{D}$ as a bipartite graph and tackle the missing value imputation problem as a link prediction task on the bipartite graph. Specifically, $\mathbf{D}$ is modeled as an undirected bipartite graph $\mathcal{G} = (\mathcal{S} \cup \mathcal{F}, \mathcal{E})$, where $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ is the set of 'sample' nodes and $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ is the set of 'feature' nodes. Also, $\mathcal{E}$ is the set of edges that only exist between sample node s and feature node f when $\mathbf{D}_{sf} \neq 0$, and each edge $(s,f) \in \mathcal{E}$ is associated with edge weight e_{sf}, which is given by $e_{sf} = \mathbf{D}_{sf}$. Then, the missing value imputation problem becomes, for any missing entries in $\mathbf{D}$ (where $\mathbf{D}_{sf}$ = 0), to predict their corresponding edge weights by developing a learnable mapping $F(\cdot)$, i.e.,

$$\hat{e}_{sf} = F(\mathcal{G}, (s,f) \notin \mathcal{E}). \tag{1}$$

The recent studies that use the bipartite graph modeling [52, 54] initialize all sample node embeddings as all-one vectors and feature node embeddings as one-hot vectors, which have a value 1 in the positions representing their respective features and 0's elsewhere. We observe, however, that such an initialization does not effectively utilize the information from the masked data matrix, which leads to inferior imputation accuracy, as shall be demonstrated in Section 4.3. Thus, in M^3-Impute, we propose to initialize each sample node embedding based on its associated (initial) feature embeddings instead of initializing them separately. While the feature embeddings are randomly initialized, the sample node embeddings are initialized in a way that reflects the embeddings of the features whose values are available in their corresponding samples.

Let $\mathbf{h}_f^0$ be the initial embedding of feature f, which is a randomly initialized d-dimensional vector, and define $\mathbf{H}_F^0 = [\mathbf{h}_{f_1}^0 \mathbf{h}_{f_2}^0 \dots \mathbf{h}_{f_m}^0] \in \mathbb{R}^{d\times m}$. Also, let $\mathbf{d}_s \in \mathbb{R}^m$ be the s-th column vector of $\mathbf{D}^\top$, which is a vector of the feature values of sample s, and let $\mathbf{m}_s \in \mathbb{R}^m$ be its corresponding mask vector, i.e., $\mathbf{m}_s = \mathrm{col}_s(\mathbf{M}^\top)$, where $\mathrm{col}_s(\cdot)$ denotes the s-th column vector of the matrix. We then initialize the embedding $\mathbf{h}_s^0$ of each sample node s as follows:

$$\mathbf{h}_s^0 = \phi\Big(\mathbf{H}_F^0\big[\mathbf{d}_s + \epsilon(\mathbb{1} - \mathbf{m}_s)\big]\Big), \tag{2}$$

where $\mathbb{1} \in \mathbb{R}^m$ is an all-one vector, and $\phi(\cdot)$ is an MLP. Note that the term $\mathbf{d}_s + \epsilon(\mathbb{1} - \mathbf{m}_s)$ indicates a vector that consists of observable feature values of s and some small positive values ϵ in the places where the feature values are unavailable (masked out).

3.3 Feature Correlation Unit

To improve the accuracy of missing value imputation, we aim to fully exploit feature correlations which often appear in the datasets. While the feature correlations are naturally captured by GNNs, we observe that there is still room for improvement. We propose **FCU** as an integral component of M^3-Impute to fully exploit the feature correlations.

To impute the missing value of feature f for a given sample s, **FCU** begins by computing the feature 'context' vector of sample s in the embedding space that reflects the correlations between the target missing feature f and observed features. Let $\mathbf{h}_f \in \mathbb{R}^d$ be the learned embedding vector of feature f from the GNN, and let $\mathbf{H}_F$ be the $d \times m$ matrix that consists of all the learned feature embedding vectors. We first obtain dot-product similarities between feature f and all the features in the embedding space, i.e., $\mathbf{H}_F^\top \mathbf{h}_f$. We then mask out the similarity values with respect to *non-observed* features in sample s. Here, instead of applying the mask vector $\mathbf{m}_s$ of sample s directly, we use a learnable 'soft' mask vector, denoted by $\mathbf{m}'_s$, which is defined to be $\mathbf{m}'_s = \sigma_1(\mathbf{m}_s) \in \mathbb{R}^m$, where $\sigma_1(\cdot)$ is an MLP with the GELU activation function [21]. In other words, we obtain feature-wise similarities with respect to sample s, denoted by $\mathbf{r}_s^f$, as follows:

$$\mathbf{r}_s^f = \sigma_2\left((\mathbf{H}_F^\top \mathbf{h}_f) \odot \mathbf{m}'_s\right) \in \mathbb{R}^d, \tag{3}$$

where $\sigma_2(\cdot)$ denotes another MLP with the GELU activation function. **FCU** next obtains the Hadamard product between the learned embedding vector of sample s, $\mathbf{h}_s$, and the feature-wise similarities with respect to sample s, $\mathbf{r}_s^f$, to learn their joint representations in a multiplicative manner. Specifically, **FCU** obtains the feature context vector of sample s, denoted by $\mathbf{c}_s^f$, as follows:

$$\mathbf{c}_s^f = \sigma_3\left(\mathbf{h}_s \odot \mathbf{r}_s^f\right) \in \mathbb{R}^d, \tag{4}$$

where $\sigma_3(\cdot)$ is also an MLP with the GELU activation function. That is, **FCU** fuses the representation vector of s and the vector that has embedding similarity values between the target feature f and the available features in s through the effective use of the soft mask $\mathbf{m}'_s$. From (3) and (4), the operations of **FCU** can be written as

$$\mathbf{c}_s^f = \mathbf{FCU}(\mathbf{h}_s, \mathbf{m}_s, \mathbf{H}_F) = \sigma_3\left(\mathbf{h}_s \odot \sigma_2\left((\mathbf{H}_F^\top \mathbf{h}_f) \odot \sigma_1(\mathbf{m}_s)\right)\right). \tag{5}$$

3.4 Sample Correlation Unit

To measure similarities between s and other samples, a common approach would be to use the dot product or cosine similarity between their embedding vectors. This approach, however, fails to take into account the observability or availability of each feature in a sample. It also does not capture the fact that different observed features are of different importance to the target feature to impute when it comes to measuring the similarities. We introduce **SCU** as another integral component of M^3-Impute to compute the sample 'context' vector of sample s by incorporating the embedding vectors of its similar samples as well as different weights of observed features. **SCU** works based on the two novel masking schemes, which shall be explained shortly.

Suppose we are to impute the missing value of feature f for a given sample s. **SCU** aims to leverage the information from the samples that are similar to s. As a first step to this end, we create a subset of samples $\mathcal{P} \subset \mathcal{S}$ that are similar to s. Specifically, we randomly choose and put a sample into $\mathcal{P}$ with probability that is proportional to the cosine similarity between s and the sample. This operation is repeated without replacement until $\mathcal{P}$ reaches a given size.

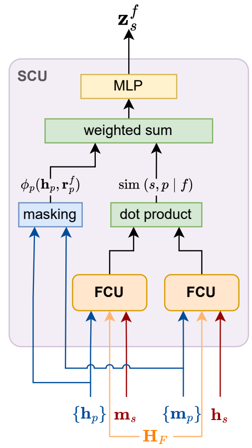

Figure 2: **SCU**.

Mutual Sample Masking: Given a subset of samples $\mathcal{P}$ that include s, we first compute the pairwise similarities between s and other samples in the subset $\mathcal{P}$. While they are computed in a similar way to **FCU**, we only consider the commonly observed features (or the common ones that have feature values) in both s and its peer $p \in \mathcal{P} \setminus \{s\}$, to calculate their pairwise similarity in the sense that the missing value of feature f is inferred. Specifically, we compute the pairwise similarity between s and $p \in \mathcal{P} \setminus \{s\}$, which is denoted by $\text{sim}(s, p \mid f)$, as follows:

$$\text{sim}(s, p \mid f) = \mathbf{FCU}(\mathbf{h}_s, \mathbf{m}_p, \mathbf{H}_f) \cdot \mathbf{FCU}(\mathbf{h}_p, \mathbf{m}_s, \mathbf{H}_f) \in \mathbb{R}, \quad (6)$$

where $\mathbf{h}_s$ and $\mathbf{h}_p$ are the learned embedding vectors of samples s and p from the GNN, respectively, and $\mathbf{m}_s$ and $\mathbf{m}_p$ are their respective mask vectors. Note that the multiplication in the RHS of (6) is the dot product.

Irrelevant Feature Masking: After we obtain the pairwise similarities between s and other samples in $\mathcal{P}$, it would be natural to consider a weighted sum of their corresponding embedding vectors, i.e., $\sum_{p \in \mathcal{P} \setminus \{s\}} \text{sim}(s, p \mid f)\, \mathbf{h}_p$, in imputing the value of the target feature f. However, we observe that $\mathbf{h}_p$ contains the information from the features whose values are available in p as well as possibly other features as it is learned via the so-called neighborhood aggregation mechanism that is central to GNNs, but some of the features may be irrelevant in inferring the value of feature f. Thus, instead of using $\{\mathbf{h}_p\}$ directly, we introduce a d-dimensional mask vector $\mathbf{r}_p^f$ for $\mathbf{h}_p$, which is to mask out potentially irrelevant feature information in $\mathbf{h}_p$, when it comes to imputing the value of feature f. Specifically, it is defined by

$$\mathbf{r}_p^f = \sigma_4\left([\mathbf{m}_p; \overline{\mathbf{m}}_f]\right) \in \mathbb{R}^d, \quad (7)$$

where $\overline{\mathbf{m}}_f$ is an m-dimensional one-hot vector that has a value 1 in the place of feature f and 0's elsewhere, $[\cdot\,;\cdot]$ denotes the vector concatenation operation, and $\sigma_4(\cdot)$ is an MLP with the GELU activation function. Note that the rationale behind the design of $\mathbf{r}_p^f$ is to embed the information on the features whose values are present in p as well as the information on the target feature f to impute. The mask $\mathbf{r}_p^f$ is then applied to $\mathbf{h}_p$ to obtain the masked embedding vector of p as follows:

$$\phi_p(\mathbf{h}_p, \mathbf{r}_p^f) = \sigma_5\left(\mathbf{h}_p \odot \mathbf{r}_p^f\right) \in \mathbb{R}^d, \quad (8)$$

where $\sigma_5(\cdot)$ is also an MLP with the GELU activation function. Once we have the masked embedding vectors of samples (excluding s) in $\mathcal{P}$, we finally compute the sample context vector of sample s, denoted by $\mathbf{z}_s^f$, which is a weighted sum of the masked embedding vectors with weights being the pairwise similarity values, i.e.,

$$\mathbf{z}_s^f = \sigma_6\left(\sum_{p \in \mathcal{P} \setminus \{s\}} \text{sim}(s, p \mid f)\; \phi_p(\mathbf{h}_p, \mathbf{r}_p^f)\right) \in \mathbb{R}^d, \quad (9)$$

where $\sigma_6(\cdot)$ is again an MLP with the GELU activation function. From (6)–(9), the operations of **SCU** can be written as

$$\mathbf{z}_s^f = \mathbf{SCU}(\mathbf{H}_\mathcal{P}, \mathbf{M}_\mathcal{P}, \mathbf{H}_F) = \sigma_6\left(\sum_{p \in \mathcal{P} \setminus \{s\}} \text{sim}(s, p \mid f)\; \sigma_5\left(\mathbf{h}_p \odot \sigma_4\left([\mathbf{m}_p; \overline{\mathbf{m}}_f]\right)\right)\right), \quad (10)$$

where $\mathbf{H}_\mathcal{P} = \{\mathbf{h}_p, p \in \mathcal{P}\}$ and $\mathbf{M}_\mathcal{P} = \{\mathbf{m}_p, p \in \mathcal{P}\}$.

Algorithm 1 Forward computation of M^3-Impute to impute the value of feature f for sample s.

1: **Input:** Bipartite graph $\mathcal{G}$, initial feature node embeddings $\mathbf{H}_F^0$, GNN model (e.g., GraphSAGE) $\mathbf{GNN}(\cdot)$, known mask matrix $\mathbf{M}$, and a subset of samples $\mathcal{P} \subset \mathcal{S}$.
2: **Output:** Predicted missing feature value $\hat{e}_{sf}$.
3: Obtain initial sample node embeddings $\mathbf{H}_S^0$ according to Equation (2).
4: $\mathbf{H}_S, \mathbf{H}_F = \mathbf{GNN}(\mathbf{H}_S^0, \mathbf{H}_F^0, \mathcal{G})$. ▷ Perform graph representation learning
5: $\mathbf{c}_s^f = \mathbf{FCU}(\mathbf{h}_s, \mathbf{m}_s, \mathbf{H}_F)$.
6: $\mathbf{z}_s^f = \mathbf{SCU}(\mathbf{H}_\mathcal{P}, \mathbf{M}_\mathcal{P}, \mathbf{H}_F)$.
7: Predict the missing feature value $\hat{e}_{sf}$ using Equation (11).

3.5 Imputation

For a given sample s, to impute the missing value of feature f, M^3-Impute obtains its feature context vector $\mathbf{c}_s^f$ and sample context vector $\mathbf{z}_s^f$ through **FCU** and **SCU**, respectively, which are then used for imputation. Specifically, it is done by predicting the corresponding edge weight $\hat{e}_{sf}$ as follows:

$$\hat{e}_{sf} = \phi_\alpha\left((1-\alpha)\mathbf{c}_s^f + \alpha\mathbf{z}_s^f\right), \tag{11}$$

where $\phi_\alpha(\cdot)$ denotes an MLP with a non-linear activation function (i.e., ReLU for continuous values and softmax for discrete ones), and α is a learnable scalar parameter. This scalar parameter α is introduced to strike a balance between leveraging feature-wise correlation and sample-wise correlation. It is necessary because the quality of $\mathbf{z}_s^f$ relies on the quality of the samples chosen in $\mathcal{P}$, so overly relying on $\mathbf{z}_s^f$ would backfire if their quality is not as desired. To address this problem, instead of employing a fixed weight α, we make α learnable and adaptive in determining the weights for $\mathbf{c}_s^f$ and $\mathbf{z}_s^f$. Note that this kind of learnable parameter approach has been widely adopted in natural language processing [26, 34, 38, 46] and computer vision [8, 55, 56], showing superior performance to its fixed counterpart. In M^3-Impute, the scalar parameter α is learned based on the similarity values between s and its peer samples $p \in \mathcal{P} \setminus \{s\}$ as follows:

$$\alpha = \phi_\gamma\Big(\mathop{\Big\|}_{p\in\mathcal{P}\setminus\{s\}} \mathrm{sim}\left(s, p \mid f\right)\Big), \tag{12}$$

where $\|$ represents the concatenation operation, and $\phi_\gamma(\cdot)$ is an MLP with the activation function $\gamma(x) = 1 - 1\,/\,e^{|x|}$. The overall operation of M^3-Impute is summarized in Algorithm 1. To learn network parameters, we use cross-entropy loss and mean square error loss for imputing discrete and continuous feature values, respectively.

4 Experiments

4.1 Experiment Setup

Datasets: We conduct experiments on 15 open datasets. These real-world datasets consist of mixed data types with both continuous and discrete values and cover different domains including civil engineering (CONCRETE, ENERGY), physics and chemistry (YACHT), thermal dynamics (NAVAL), etc. Since the datasets are fully observed, we introduce missing values by applying a randomly generated mask to the data matrix. Specifically, as used in prior studies [23, 24], we apply three masking generation schemes, namely missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).[1] We use MCAR with a missing ratio of 30%, unless otherwise specified. We follow the preprocessing steps adopted in [52, 54] to scale feature values to [0, 1] with a MinMax scaler [25]. Due to the space limit, we below present the results of eight datasets that are used in Grape [52] and report the other results in Appendix.

Baseline models: M^3-Impute is compared against popular and state-of-the-art imputation methods, including statistical methods, deep generative methods, and graph-based methods listed as follows: **MEAN**: It imputes the missing value $\hat{e}_{sf}$ as the mean of observed values in feature f from all the samples. K-nearest neighbors (**kNN**) [43]: It imputes the missing value $\hat{e}_{sf}$ using the kNNs that have observed values in feature f with weights that are based on the Euclidean distance to sample s. Multivariate imputation by chained equations (**Mice**) [45]: This method runs multiple regressions where each missing value is modeled upon the observed non-missing values. Iterative

[1]More details about the datasets and mask generation for missing values can be found in Appendix.

Table 1: Imputation accuracy in MAE. MAE scores are enlarged by 10 times.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Mean	2.09	0.98	1.79	1.85	3.10	2.31	2.50	1.68
Svd [18]	2.46	0.92	1.94	1.53	2.24	0.50	3.67	2.33
Spectral [30]	2.64	0.91	1.98	1.46	2.26	0.41	2.80	2.13
Mice [45]	1.68	0.77	1.34	1.16	1.53	0.20	2.50	1.16
kNN [43]	1.67	0.72	1.16	0.95	1.81	0.10	2.77	1.38
Gain [50]	2.26	0.86	1.67	1.23	1.99	0.46	2.70	1.31
Miwae [29]	4.68	1.00	1.81	3.81	2.79	2.37	2.57	1.74
Grape [52]	1.46	**0.60**	0.75	0.64	1.36	0.07	2.50	1.00
Miracle [24]	42.97	1.13	1.71	42.23	41.43	0.17	**2.49**	1.15
HyperImpute [23]	1.76	0.67	0.84	0.82	**1.32**	**0.04**	2.58	1.06
M^3-Impute	**1.33**	**0.60**	**0.71**	**0.60**	**1.32**	0.06	2.50	**0.99**

SVD (**Svd**) [18]: It imputes missing values by solving a matrix completion problem with iterative low-rank singular value decomposition. Spectral regularization algorithm (**Spectral**) [30]: This matrix completion algorithm uses the nuclear norm as a regularizer and imputes missing values with iterative soft-thresholded SVD. **Miwae** [29]: It works based on an autoencoder generative model trained to maximize a potentially tight lower bound of the log-likelihood of the observed data and Monte Carlo techniques for imputation. **Miracle** [24]: It uses the imputation results from naive methods such as Mean and refines them iteratively by learning a missingness graph (m-graph) and regularizing an imputation function. **Gain** [50]: This method trains a data imputation generator with a generalized generative adversarial network in which the discriminator aims to distinguish between real and imputed values. **Grape** [52]: It models the data as a bipartite graph and imputes missing values by predicting the weights of the missing edges, each of which is done based on the inner product between the embeddings of its corresponding sample and feature nodes. **HyperImpute** [23]: HyperImpute is a framework that conducts an extensive search among a set of imputation methods, selecting the optimal imputation method with fine-tuned parameters for each feature in the dataset.

Model configurations: Parameters of M^3-Impute are updated by the Adam optimizer with a learning rate of 0.001 for 40,000 epochs. For graph representation learning, we use a variant of GraphSAGE [17], which not only learns node embeddings but also edge embeddings via the neighborhood aggregation mechanism, as similarly used in [52]. We consider its three-layer GNN model. We employ mean-pooling as the aggregation function and use ReLU as the activation function for the GNN layers. We set the embedding dimension d to 128. It is known that randomly dropping out a subset of observable edges during training improves the model's generalization ability. We also leverage the observation and randomly drop 50% of observable edges during training. For each experiment, we conduct five runs with different random seeds and report the average results.

4.2 Overall Performance

We first compare the feature imputation performance of M^3-Impute with popular and state-of-the-art imputation methods. As shown in Table 1, M^3-Impute achieves the lowest imputation MAE for six out of the eight examined datasets and the second-best MAE scores in the other two, which validates the effectiveness of M^3-Impute. For Kin8nm dataset, M^3-Impute underperforms Miracle. It is mainly because each feature in Kin8nm is independent of the others, so none of the observed features can help impute missing feature values. For Naval dataset, the only model that outperforms M^3-Impute is HyperImpute [23]. In the Naval dataset, nearly every feature exhibits a strong linear correlation with the other features, i.e., every pair of features has correlation coefficient close to one. This allows HyperImpute to readily select a linear model from its model pool for each feature to impute. Nonetheless, M^3-Impute exhibits overall superior performance to the baselines as it can be well adapted to each dataset that possesses different amounts of correlations over features and samples. In other words, M^3-Impute benefits from explicitly incorporating feature-wise and sample-wise correlations together with our carefully designed mask schemes. Furthermore, we evaluate the performance of M^3-Impute under MAR and MNAR settings. We observe that M^3-Impute consistently outperforms all the baselines under all datasets and achieves a larger margin in the improvement compared to the case with MCAR setting. This implies that M^3-Impute is also effective in handling different patterns of missing values in the input data. Comprehensive results are provided in Appendix.

Table 2: Ablation study. M^3-Uniform stands for M^3-Impute with the uniform sampling strategy.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
HyperImpute	1.76 $\pm$ .03	0.67 $\pm$ .01	0.84 $\pm$ .02	0.82 $\pm$ .01	1.32 $\pm$ .02	**0.04** $\pm$ .00	$\underline{2.58}$ $\pm$ .05	1.06 $\pm$ .01
Grape	1.46 $\pm$ .01	**0.60** $\pm$ .00	0.75 $\pm$ .01	0.64 $\pm$ .01	1.36 $\pm$ .01	0.07 $\pm$ .00	**2.50** $\pm$ .00	$\underline{1.00}$ $\pm$ .00
Architecture								
Init Only	1.43 $\pm$ .01	**0.60** $\pm$ .00	0.74 $\pm$ .00	0.63 $\pm$ .01	1.35 $\pm$ .01	$\underline{0.06}$ $\pm$ .00	**2.50** $\pm$ .00	**0.99** $\pm$ .00
Init+**FCU**	1.35 $\pm$ .01	$\underline{0.61}$ $\pm$ .00	$\underline{0.72}$ $\pm$ .03	$\underline{0.61}$ $\pm$ .02	1.32 $\pm$ .00	$\underline{0.07}$ $\pm$ .01	**2.50** $\pm$ .00	**0.99** $\pm$ .00
Init+**SCU**	1.37 $\pm$ .01	**0.60** $\pm$ .00	0.73 $\pm$ .00	0.63 $\pm$ .01	**1.30** $\pm$ .00	0.09 $\pm$ .01	**2.50** $\pm$ .00	$\underline{1.00}$ $\pm$ .00
M^3-Impute	**1.33** $\pm$ .04	**0.60** $\pm$ .00	**0.71** $\pm$ .01	**0.60** $\pm$ .00	1.32 $\pm$ .01	$\underline{0.06}$ $\pm$ .00	**2.50** $\pm$ .00	**0.99** $\pm$ .00
Sampling Strategy								
M^3-Uniform	$\underline{1.34}$ $\pm$ .01	**0.60** $\pm$ .00	0.73 $\pm$ .01	$\underline{0.61}$ $\pm$ .00	$\underline{1.31}$ $\pm$ .00	$\underline{0.06}$ $\pm$ .00	**2.50** $\pm$ .00	**0.99** $\pm$ .00

4.3 Ablation Study

To study the effectiveness of three integral components of M^3-Impute, we consider three variants of M^3-Impute, each with a subset of the components, namely initialization only (Init Only), initialization + **FCU** (Init + **FCU**), and initialization + **SCU** (Init + **SCU**). The performance of these variants are evaluated against the top-performing imputation baselines such as Grape and HyperImpute. As shown in Table 2, the three variants derived from M^3-Impute achieve lower MAE values than both baselines in most datasets, demonstrating the effectiveness of our novel components in M^3-Impute.

Specifically, for initialization only, the key difference between M^3-Impute and Grape lies in our refined initialization process of feature-node and sample-node embeddings. The reduced MAE values observed by the Init Only variant demonstrate that our proposed initialization process is more effective in utilizing information between samples and their associated features, including missing ones, as compared to the basic initialization used in [52]. In addition, we observe that when **FCU** or **SCU** is incorporated, MAE values are further reduced for most datasets. This validates that explicitly modeling feature-wise or sample-wise correlations through our novel masking schemes can improve imputation accuracy. When all the three components are combined together as in M^3-Impute, they work synergistically to lower MAE values, validating the efficacy of explicit consideration of both sample-wise and feature-wise correlations (in addition to the refined initialization process) for missing data imputation.

4.4 Robustness

Missing ratio: In practice, datasets may possess different missing ratios. To validate the model's robustness under such circumstances, we evaluate the performance of M^3-Impute and other baseline models with varying missing ratios, i.e., 0.1, 0.3, 0.5, and 0.7. Figure 3 shows their performance. We use the MAE of HyperImpute (HI) as the reference performance and offset the performance of each model by $\text{MAE}_x - \text{MAE}_{HI}$, where x represents the considered model. For clarity, we here only report the results of four top-performing models. As shown in Figure 3, M^3-Impute outperforms other baseline models for almost all the cases, especially under YACHT, CONCRETE, ENERGY, and HOUSING datasets. It is worth noting that modeling feature correlations in these datasets is particularly challenging due to the presence of considerable amounts of weakly correlated features, along with a few strongly correlated ones. Nonetheless, **FCU** and **SCU** in M^3-Impute were able to better capture such correlations with our efficient masking schemes, thereby resulting in a large improvement in imputation accuracy. In addition, for KIN8NM dataset, M^3-Impute ties with the second-best model, Grape. As mentioned in Section 4.2, each feature in KIN8NM is independent of the others, so none of the observed features can help impute missing feature values. For NAVAL dataset, where each feature strongly correlates with the others, M^3-Impute surpasses Grape but falls short of HyperImpute, due to the same reason as discussed above. Overall, M^3-Impute is robust to various missing ratios. Comprehensive results for all the baseline models can be found in Appendix.

Sampling strategy in SCU: While **SCU** uses a sampling strategy based on pairwise cosine similarities to construct a subset of samples $\mathcal{P}$, the simplest sampling strategy to build $\mathcal{P}$ would be to choose samples uniformly at random without replacement (M^3-Uniform). Intuitively, this approach cannot identify similar peer samples accurately and thus would lead to inferior performance. Nonetheless, as shown in Table 2, even with this naive uniform sampling strategy, M^3-Uniform still outperforms the two leading imputation baselines.

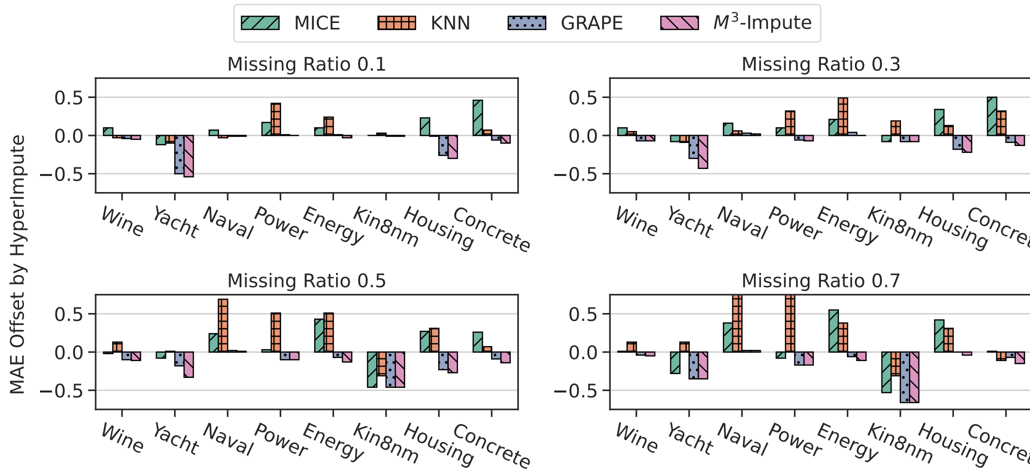

Figure 3: Model performance vs. missing ratios. MAE scores are offset by HyperImpute [23].

Size of $\mathcal{P}$ in SCU: Intuitively, neither an excessively small nor overly large size of the sample subset $\mathcal{P}$ is optimal. Too few peer samples leave **SCU** with insufficient information to learn sample-wise correlations, while too many peer samples may include quite a few dissimilar ones, which may introduce significant noise to the computation of **SCU** and thus degrade the performance. Table 3 shows the performance of M^3-Impute with varying numbers of peer samples. In general, the trends agree with our intuition. Although the optimal size varies across different datasets, we observe that having the number of peer samples to be 5 to 10 achieves the overall best imputation accuracy.

Table 3: MAE scores for varying peer-sample size ($|\mathcal{P}|-1$) and different values of ϵ.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Peer $=1$	1.34 ± .00	**0.60** ± .00	0.73 ± .00	0.61 ± .01	1.32 ± .00	**0.06** ± .00	**2.5** ± .00	**0.99**± .00
Peer $=2$	1.35 ± .01	$\underline{0.61}$ ± .00	$\underline{0.72}$ ± .01	**0.59** ± .01	$\underline{1.32}$ ± .00	**0.06** ± .00	**2.5** ± .00	$\underline{1.00}$ ± .00
Peer $=5$	**1.33** ± .04	**0.60** ± .00	**0.71** ± .01	$\underline{0.60}$ ± .00	$\underline{1.32}$ ± .01	**0.06** ± .00	**2.5** ± .00	**0.99**± .00
Peer $=10$	**1.33** ± .01	$\underline{0.61}$ ± .00	**0.71** ± .01	$\underline{0.60}$ ± .01	**1.31** ± .01	$\underline{0.07}$ ± .00	**2.5** ± .00	$\underline{1.00}$ ± .00
Peer $=15$	$\underline{1.34}$ ± .00	$\underline{0.61}$ ± .00	$\underline{0.72}$ ± .01	$\underline{0.60}$ ± .00	**1.31** ± .00	$\underline{0.07}$ ± .00	**2.5** ± .00	**0.99** ± .00
Peer $=20$	$\underline{1.34}$ ± .04	$\underline{0.61}$ ± .00	$\underline{0.72}$ ± .01	$\underline{0.60}$ ± .01	**1.31** ± .00	$\underline{0.07}$ ± .00	**2.5** ± .00	$\underline{1.00}$ ± .00
$\epsilon = 0$	1.34 ± .01	$\underline{0.61}$ ± .00	**0.71** ± .01	**0.60** ± .01	**1.30** ± .00	**0.06** ± .00	**2.50** ± .00	**0.99** ± .00
$\epsilon = 10^{-5}$	**1.31** ± .01	$\underline{0.61}$ ± .00	**0.71** ± .00	**0.60** ± .01	**1.30** ± .00	$\underline{0.07}$ ± .00	**2.50** ± .00	$\underline{1.00}$ ± .00
$\epsilon = 10^{-4}$	$\underline{1.33}$ ± .04	**0.60** ± .00	**0.71** ± .01	**0.60** ± .00	**1.30** ± .00	**0.06** ± .00	**2.50** ± .00	**0.99** ± .00
$\epsilon = 10^{-3}$	$\underline{1.33}$ ± .04	**0.60** ± .00	$\underline{0.72}$ ± .01	**0.60** ± .01	**1.30** ± .00	$\underline{0.07}$ ± .01	**2.50** ± .00	**0.99** ± .00

Initialization parameter ϵ: We also evaluate whether a non-zero value of ϵ in the initialization process of M^3-Impute indeed lead to an improvement in imputation accuracy. As shown in Table 3, for YACHT and WINE datasets, the introduction of a non-zero value of ϵ results in lower MAE scores. Another insight that we have from Table 3 is that ϵ should not be set too large, as a large value of ϵ might impose incorrect weights to the features with missing values. We observe that it is an overall good choice to set ϵ to 1×10^{-5} or 1×10^{-4}.

5 Conclusion

We have presented M^3-Impute, a mask-guided representation learning for missing data imputation. M^3-Impute improved the initialization process by considering the relationships between samples and their associated features (including missing ones) even in initializing the embeddings. In addition, for more effective representation learning, we introduced two novel components in M^3-Impute – **FCU** and **SCU**, which learn feature-wise and sample-wise correlations, respectively, to capture data correlations explicitly and leverage them for imputation. Extensive experiment results demonstrate the effectiveness of M^3-Impute. M^3-Impute achieves overall superior performance to popular and state-of-the-art methods on 15 open datasets, with 13 best and two second-best MAE scores on average under three different settings of missing value patterns.

References

[1] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[2] Jaap Brand. *Development, implementation and evaluation of multiple imputation strategies for the statistical analysis of incomplete data sets*. 1999.

[3] Thomas Brooks, D. Pope, and Michael Marcolini. Airfoil self-noise. `https://doi.org/10.24432/C5VW2C`, March 2014.

[4] Lane F Burgette and Jerome P Reiter. Multiple imputation for missing data via sequential regression trees. *American journal of epidemiology*, 172(9):1070–1076, 2010.

[5] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[6] Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[7] Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Wine quality. `https://doi.org/10.24432/C56S3T`, October 2009.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[10] Tianyu Du, Luca Melis, and Ting Wang. Remasker: Imputing tabular data with masked autoencoding. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017.

[12] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(1):407–499, 2004.

[13] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[14] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.

[15] Andrew Gelman. Parameterization and bayesian modeling. *Journal of the American Statistical Association*, 99(466):537–545, 2004.

[16] Zoubin Ghahramani and Michael Jordan. Supervised learning from incomplete data via an em approach. *Advances in neural information processing systems*, 6, 1993.

[17] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[18] Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *J. Mach. Learn. Res.*, 16(1):33673402, jan 2015.

[19] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[20] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1(Oct):49–75, 2000.

[21] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

[22] James Honaker, Gary King, and Matthew Blackwell. Amelia ii: A program for missing data. *Journal of statistical software*, 45:1–47, 2011.

[23] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9916–9937. PMLR, 2022.

[24] Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. MIRACLE: causally-aware imputation via learning missing data mechanisms. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23806–23817, 2021.

[25] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets, 2nd Ed*. Cambridge University Press, 2014.

[26] Mingzhe Li, Xiuying Chen, Shen Gao, Zhangming Chan, Dongyan Zhao, and Rui Yan. VMSMO: Learning to generate multimodal summary for video-based news articles. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9360–9369, Online, November 2020. Association for Computational Linguistics.

[27] Steven Cheng-Xian Li, Bo Jiang, and Benjamin M. Marlin. Misgan: Learning from incomplete data with generative adversarial networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[28] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[29] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4413–4423. PMLR, 2019.

[30] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, 11:22872322, aug 2010.

[31] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[32] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[33] Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7130–7140. PMLR, 13–18 Jul 2020.

[34] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

[35] Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk, Peter Solenberger, et al. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey methodology*, 27(1):85–96, 2001.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[37] Joseph L Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[38] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[39] V. Sigillito, S. Wing, L. Hutton, and K. Baker. Ionosphere. `https://doi.org/10.24432/C5W01B`, December 1988.

[40] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. Missing data imputation with adversarially-trained graph convolutional networks. *Neural Networks*, 129:249–260, 2020.

[41] Daniel J Stekhoven and Peter Bühlmann. Missforestnon-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[42] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24804–24816, 2021.

[43] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[44] Stef Van Buuren, Jaap PL Brand, Catharina GM Groothuis-Oudshoorn, and Donald B Rubin. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, 76(12):1049–1064, 2006.

[45] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):167, 2011.

[46] Wenbo Wang, Yang Gao, Heyan Huang, and Yuxiang Zhou. Concept pointer network for abstractive summarization. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3076–3085, Hong Kong, China, November 2019. Association for Computational Linguistics.

[47] William H. Wolberg, Olvi L. Mangasarian, and W. Nick Street. Breast cancer wisconsin (diagnostic). `https://doi.org/10.24432/C5DW2B`, October 1995.

[48] Richard Wu, Aoqian Zhang, Ihab Ilyas, and Theodoros Rekatsinas. Attention-based learning for missing data imputation in holoclean. *Proceedings of Machine Learning and Systems*, 2:307–325, 2020.

[49] I-Cheng Yeh. Blood transfusion service center. `https://doi.org/10.24432/C5GS39`, October 2008.

[50] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: missing data imputation using generative adversarial nets. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5675–5684. PMLR, 2018.

[51] Seongwook Yoon and Sanghoon Sull. GAMIN: generative adversarial multiple imputation network for highly missing data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8453–8461. Computer Vision Foundation / IEEE, 2020.

[52] Jiaxuan You, Xiaobai Ma, Daisy Yi Ding, Mykel J. Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[53] Shuhan Zheng and Nontawat Charoenphakdee. Diffusion models for missing value imputation in tabular data. *CoRR*, abs/2210.17128, 2022.

[54] Jiajun Zhong, Ning Gui, and Weiwei Ye. Data imputation with iterative graph reconstruction. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 11399–11407. AAAI Press, 2023.

[55] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[56] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9300–9308, 2019.

A Appendix

Table 4: Overview of Datasets.

	Concrete	Housing	Wine	Yacht	Energy	Kin8nm	Naval	Power
# Samples	1030	506	1599	308	768	8192	11934	9568
# Features	8	13	11	6	8	8	16	4

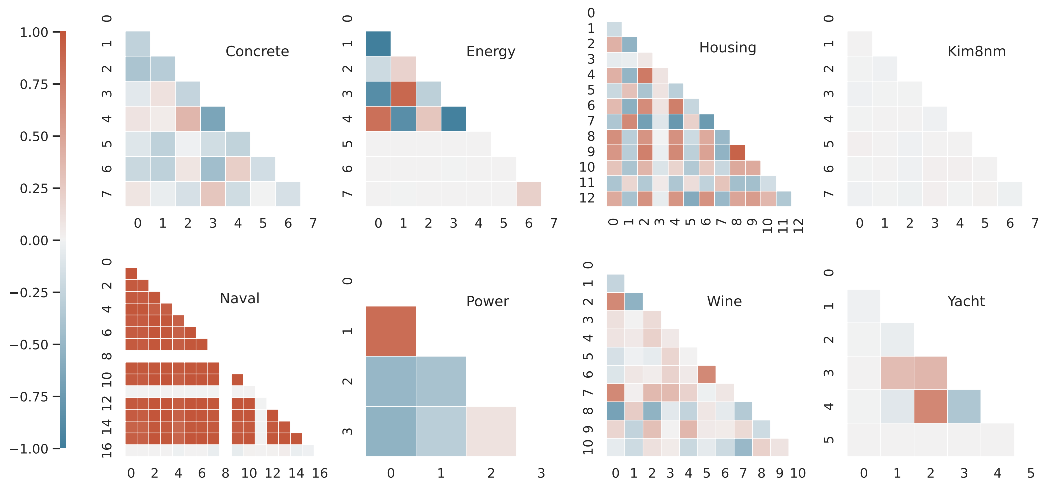

Figure 4: Pearson correlation coefficients of UCI datasets.

In this section, we discuss further experimental details. We first give an overview of the dataset details in Section A.1, followed by the implementation of different missing types and present corresponding imputation performance under MAR and MNAR settings (Section A.2). We then provide the comprehensive results of the robustness experiments (Section A.3). Finally, we extend our evaluation of M^3-Impute to seven additional datasets (Section A.4) and elaborate on the computational resources in Section A.5.

A.1 Dataset Details

Table 4 presents the statistics of the eight UCI datasets [11] used throughout Section 4. Figure 4 illustrates the Pearson correlation coefficients among the features. In the Kin8nm dataset, all features are linearly independent, whereas the Naval dataset exhibits strong correlations among its features. Under the MCAR setting, M^3-Impute performs comparably to the baseline imputation methods on these two datasets (shown in Table 1). However, in real-world scenarios, features are not always entirely independent or strongly correlated. In the other six datasets, we observe a mix of weakly correlated features along with a few that are strongly correlated. In these cases, M^3-Impute consistently outperforms all baseline methods.

A.2 Detailed Results of Different Missing Types

We adopt the same procedure outlined in [52, 54] to generate missing values under different settings.

- **MCAR**: A $n \times m$ matrix is sampled from a uniform distribution. Positions with values no greater than the ratio of missingness are viewed as missing and the remaining positions are observable.
- **MAR**: First, a subset of features is randomly selected to be fully observed. Then, these remaining features have values removed according to a logistic model with random weights, using the fully observed feature values as input. The desired rate of missingness is achieved by adjusting the bias term.
- **MNAR**: This is done by first apply the MAR mechanism above. Then, the remaining feature values are masked out by the MCAR mechanism.

Table 5: MAE scores under MAR setting.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Mean	2.20	1.09	1.79	2.02	3.26	2.75	**2.49**	1.81
Svd [18]	2.64	1.04	2.32	1.71	3.68	0.52	2.69	2.37
Spectral [30]	3.06	0.91	2.12	1.84	2.88	1.29	3.56	3.37
Mice [45]	1.79	0.79	1.27	1.22	1.12	<u>0.27</u>	<u>2.51</u>	1.16
Knn [43]	1.69	<u>0.66</u>	<u>0.89</u>	0.89	1.61	**0.07**	2.94	1.11
Gain [50]	2.07	1.13	1.87	0.92	2.26	0.91	2.93	1.42
Miwae [29]	3.47	1.04	1.87	3.79	3.82	3.78	2.57	2.07
Grape [52]	<u>1.20</u>	**0.60**	**0.77**	<u>0.66</u>	<u>1.05</u>	**0.07**	**2.49**	<u>1.06</u>
Miracle [24]	44.33	1.70	3.08	48.63	38.20	48.77	2.82	0.86
HyperImpute [23]	2.06	0.78	1.30	1.05	1.11	1.01	3.07	1.07
M^3-Impute	**1.09**	**0.60**	**0.77**	**0.60**	**0.98**	**0.07**	**2.49**	**1.01**

Table 6: MAE scores under MNAR setting.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Mean	2.18	1.04	1.80	1.95	3.17	2.60	<u>2.49</u>	1.76
Svd [18]	2.61	1.06	2.24	1.58	3.55	0.53	2.69	2.27
Spectral [30]	2.75	1.01	1.86	1.60	2.50	1.35	3.34	3.14
Mice [45]	1.91	0.77	1.37	1.22	1.57	0.21	2.50	1.08
Knn [43]	1.92	0.75	1.15	0.95	1.96	**0.08**	3.06	1.65
Gain [50]	2.34	0.92	1.80	1.08	1.92	1.12	2.78	1.22
Miwae [29]	3.77	1.02	1.86	3.80	2.74	3.79	2.58	1.93
Grape [52]	<u>1.23</u>	<u>0.61</u>	<u>0.73</u>	<u>0.61</u>	<u>1.16</u>	**0.08**	**2.46**	<u>1.02</u>
Miracle [24]	43.57	1.03	2.15	46.17	39.37	46.50	2.64	1.06
HyperImpute [23]	1.95	0.72	0.88	0.85	1.19	0.85	2.71	1.09
M^3-Impute	**1.15**	**0.60**	**0.68**	**0.54**	**1.09**	**0.08**	**2.46**	**1.00**

In addition to the results for MCAR setting presented in Table 4.2, Table 5 and Table 6 present the MAE scores under MAR and MNAR settings, respectively. M^3-Impute consistently outperforms all baseline methods in both scenarios.

A.3 Robustness against Various Ratios of Missingness

Table 8 presents the performance of various imputation methods across different ratios of missingness. M^3-Impute achieves the lowest MAE scores in most cases and the second-best MAE scores in the remaining ones.

A.4 Further Evaluation on Seven Additional Datasets

Table 7: Overview of seven additional datasets.

	airfoil	blood	wine-white	ionosphere	breast	iris	diabetes
# Samples	1503	748	4899	351	569	150	442
# Features	6	4	12	34	30	4	10

In this experiment, we further evaluate M^3-Impute on seven datasets: Airfoil [3], Blood [49], Wine-White [7], Ionosphere [39], Breast Cancer [47], Iris [13], and Diabetes [12]. An overview of dataset details is provided in Table 7, and feature correlations are illustrated in Figure 5. We simulate missingness in data under MCAR, MAR, and MNAR conditions, each with a missing ratio of 0.3. Results are demonstrated in Table 9. Across all three types of missingness, M^3-Impute achieves five best and two second-best MAE scores on average.

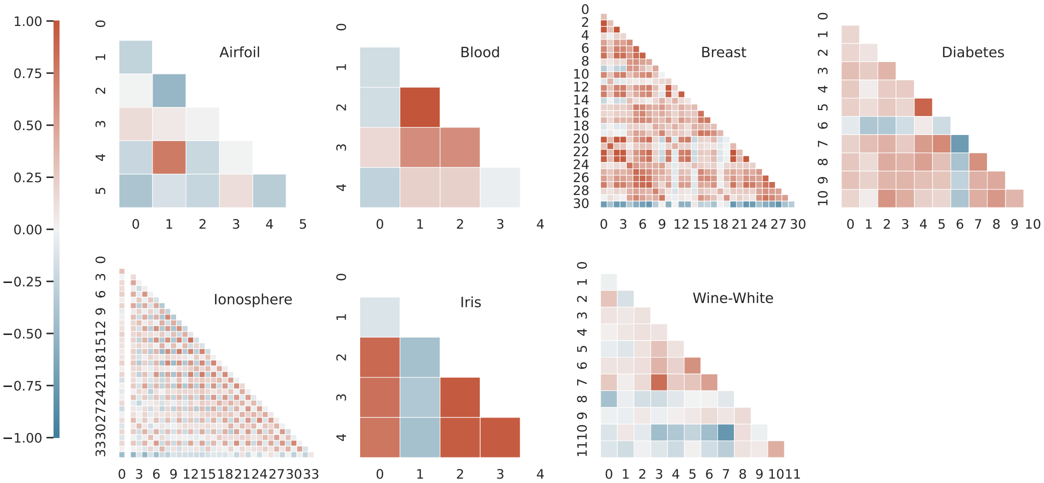

Figure 5: Pearson correlation coefficient of 7 extra datasets.

A.5 Computational Resources

All our experiments are conducted on a GPU server running Ubuntu 22.04, with PyTorch 2.1.0 and CUDA 12.1. We train and test M^3-Impute using a single NVIDIA A100 80G GPU. With the experimental setup described in Section 4.1, the total runtime (including both training and testing) for each of the five repeated runs ranged from 1 to 5 hours, depending on the scale of the datasets.

Table 8: MAE scores across different levels of missingness.

	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Missing 10%								
Mean	2.22 ± 0.05	0.96 ± 0.02	1.81 ± 0.02	1.84 ± 0.01	3.09 ± 0.07	2.30 ± 0.01	2.50 ± 0.01	1.68 ± 0.00
Svd	1.92 ± 0.16	0.88 ± 0.03	2.04 ± 0.04	1.69 ± 0.11	1.75 ± 0.10	0.34 ± 0.00	5.04 ± 0.06	2.26 ± 0.04
Spectral	2.24 ± 0.12	0.76 ± 0.02	1.84 ± 0.05	1.28 ± 0.04	1.76 ± 0.08	0.38 ± 0.01	2.71 ± 0.02	1.77 ± 0.02
Mice	1.38 ± 0.13	0.62 ± 0.01	0.97 ± 0.04	0.98 ± 0.04	1.28 ± 0.07	0.13 ± 0.00	2.50 ± 0.01	1.01 ± 0.01
Knn	1.40 ± 0.17	0.49 ± 0.01	0.58 ± 0.05	0.74 ± 0.04	1.42 ± 0.05	**0.03** ± 0.00	2.53 ± 0.01	1.26 ± 0.00
Gain	2.30 ± 0.04	0.83 ± 0.04	1.62 ± 0.05	1.16 ± 0.05	1.95 ± 0.05	0.45 ± 0.01	2.74 ± 0.02	1.22 ± 0.00
Miwae	4.57 ± 0.09	0.98 ± 0.01	1.85 ± 0.03	3.78 ± 0.10	2.77 ± 0.16	2.36 ± 0.00	2.56 ± 0.00	1.74 ± 0.00
Grape	1.00 ± 0.00	0.48 ± 0.00	0.45 ± 0.01	0.49 ± 0.00	1.19 ± 0.00	0.05 ± 0.00	2.49 ± 0.00	0.85 ± 0.03
Miracle	44.77 ± 0.05	0.97 ± 0.19	1.91 ± 0.07	43.90 ± 0.33	41.43 ± 0.34	0.12 ± 0.00	**2.48** ± 0.00	1.07 ± 0.05
HyperImpute	1.50 ± 0.11	0.52 ± 0.00	0.51 ± 0.04	0.75 ± 0.04	1.18 ± 0.05	0.06 ± 0.04	2.50 ± 0.00	**0.84**± 0.00
M^3-Impute	**0.96** ± 0.00	**0.47** ± 0.01	**0.41** ± 0.01	**0.45** ± 0.00	**1.15** ± 0.00	0.05 ± 0.00	2.49 ± 0.00	**0.84** ± 0.01
	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Missing 30%								
Mean	2.09 ± 0.04	0.98 ± 0.01	1.79 ± 0.01	1.85 ± 0.00	3.10 ± 0.04	2.31 ± 0.00	2.50 ± 0.00	1.68 ± 0.00
Svd	2.46 ± 0.16	0.92 ± 0.01	1.94 ± 0.02	1.53 ± 0.03	2.24 ± 0.06	0.50 ± 0.00	3.67 ± 0.06	2.33 ± 0.01
Spectral	2.64 ± 0.11	0.91 ± 0.01	1.98 ± 0.04	1.46 ± 0.03	2.26 ± 0.09	0.41 ± 0.00	2.80 ± 0.01	2.13 ± 0.01
Mice	1.68 ± 0.05	0.77 ± 0.00	1.34 ± 0.01	1.16 ± 0.03	1.53 ± 0.04	0.20 ± 0.01	2.50 ± 0.00	1.16 ± 0.01
Knn	1.67 ± 0.02	0.72 ± 0.00	1.16 ± 0.03	0.95 ± 0.01	1.81 ± 0.03	0.10 ± 0.00	2.77 ± 0.01	1.38 ± 0.01
Gain	2.26 ± 0.11	0.86 ± 0.00	1.67 ± 0.03	1.23 ± 0.02	1.99 ± 0.03	0.46 ± 0.02	2.70 ± 0.00	1.31 ± 0.05
Miwae	4.68 ± 0.16	1.00 ± 0.00	1.81 ± 0.01	3.81 ± 0.04	2.79 ± 0.04	2.37 ± 0.00	2.57 ± 0.00	1.74 ± 0.00
Grape	1.46 ± 0.01	**0.60** ± 0.00	0.75 ± 0.01	0.64 ± 0.01	1.36 ± 0.01	0.07 ± 0.00	2.50 ± 0.00	1.00 ± 0.00
Miracle	42.97 ± 0.53	1.13 ± 0.00	1.71 ± 0.05	42.23 ± 0.31	41.43 ± 0.34	0.17 ± 0.00	**2.49** ± 0.00	1.15 ± 0.01
HyperImpute	1.76 ± 0.03	0.67 ± 0.01	0.84 ± 0.02	0.82 ± 0.01	**1.32** ± 0.02	**0.04** ± 0.00	2.58 ± 0.05	1.06 ± 0.01
M^3-Impute	**1.33** ± 0.04	**0.60** ± 0.00	**0.71** ± 0.01	**0.60** ± 0.00	**1.32** ± 0.01	0.06 ± 0.00	2.50 ± 0.00	**0.99** ± 0.00
	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Missing 50%								
Mean	2.12 ± 0.02	0.98 ± 0.01	1.81 ± 0.01	1.84 ± 0.01	3.08 ± 0.02	2.31 ± 0.00	**2.50** ± 0.00	1.67 ± 0.00
Svd	3.00 ± 0.11	1.18 ± 0.00	2.19 ± 0.01	1.88 ± 0.01	2.88 ± 0.04	0.87 ± 0.00	3.30 ± 0.01	2.92 ± 0.02
Spectral	3.17 ± 0.13	1.13 ± 0.00	2.31 ± 0.01	1.76 ± 0.03	3.03 ± 0.02	0.46 ± 0.00	3.02 ± 0.00	2.98 ± 0.02
Mice	1.99 ± 0.08	0.83 ± 0.00	1.59 ± 0.03	1.33 ± 0.02	2.13 ± 0.12	0.31 ± 0.01	**2.50** ± 0.00	1.32 ± 0.01
Knn	2.08 ± 0.02	0.98 ± 0.01	1.40 ± 0.02	1.37 ± 0.01	2.21 ± 0.01	0.76 ± 0.01	2.65 ± 0.00	1.80 ± 0.01
Gain	2.33 ± 0.03	1.18 ± 0.15	2.20 ± 0.17	1.43 ± 0.09	2.58 ± 0.09	0.56 ± 0.03	2.86 ± 0.06	1.36 ± 0.00
Miwae	4.57 ± 0.06	1.01 ± 0.01	1.85 ± 0.02	3.79 ± 0.01	2.83 ± 0.05	2.38 ± 0.00	2.58 ± 0.00	1.73 ± 0.00
Grape	1.89 ± 0.02	0.75 ± 0.01	1.24 ± 0.00	0.83 ± 0.01	1.63 ± 0.01	0.09 ± 0.00	**2.50** ± 0.00	**1.19** ± 0.00
Miracle	40.77 ± 0.34	1.08 ± 0.00	2.00 ± 0.08	39.40 ± 0.33	37.40 ± 0.22	0.24 ± 0.00	2.82 ± 0.06	1.29 ± 0.00
HyperImpute	2.07 ± 0.11	0.85 ± 0.00	1.33 ± 0.08	1.06 ± 0.11	1.70 ± 0.05	**0.07** ± 0.00	2.96 ± 0.04	1.29 ± 0.01
M^3-Impute	**1.74** ± 0.01	**0.74** ± 0.00	**1.19** ± 0.02	**0.79** ± 0.01	**1.57** ± 0.00	0.08 ± 0.00	**2.50** ± 0.00	**1.19** ± 0.00
	Yacht	Wine	Concrete	Housing	Energy	Naval	Kin8nm	Power
Missing 70%								
Mean	2.16 ± 0.06	0.99 ± 0.00	1.81 ± 0.01	1.83 ± 0.02	3.08 ± 0.01	2.31 ± 0.00	2.50 ± 0.00	1.67 ± 0.00
Svd	3.78 ± 0.06	1.63 ± 0.02	2.53 ± 0.03	2.58 ± 0.07	3.65 ± 0.09	1.56 ± 0.00	3.58 ± 0.00	3.88 ± 0.01
Spectral	4.17 ± 0.10	1.67 ± 0.02	2.75 ± 0.01	2.59 ± 0.05	4.00 ± 0.03	1.04 ± 0.00	3.73 ± 0.01	4.33 ± 0.01
Mice	2.21 ± 0.10	0.93 ± 0.01	1.72 ± 0.02	1.54 ± 0.04	2.71 ± 0.15	0.53 ± 0.00	2.62 ± 0.08	1.46 ± 0.00
Knn	2.62 ± 0.08	1.05 ± 0.00	1.60 ± 0.01	1.43 ± 0.02	2.54 ± 0.04	1.08 ± 0.00	2.84 ± 0.01	2.73 ± 0.00
Gain	3.07 ± 0.08	1.61 ± 0.15	2.84 ± 0.04	3.09 ± 0.04	3.83 ± 0.15	1.07 ± 0.02	3.31 ± 0.21	1.51 ± 0.05
Miwae	4.56 ± 0.07	1.02 ± 0.00	1.84 ± 0.01	3.78 ± 0.02	3.02 ± 0.07	2.38 ± 0.00	2.58 ± 0.00	1.72 ± 0.00
Grape	**2.14** ± 0.01	0.88 ± 0.01	1.64 ± 0.02	1.12 ± 0.01	2.10 ± 0.01	0.17 ± 0.00	**2.49** ± 0.00	**1.37** ± 0.00
Miracle	38.37 ± 0.38	1.03 ± 0.00	2.45 ± 0.21	36.23 ± 0.21	33.93 ± 0.17	0.53 ± 0.00	3.09 ± 0.02	1.92 ± 0.04
HyperImpute	2.49 ± 0.08	0.92 ± 0.02	1.71 ± 0.01	1.12 ± 0.13	2.16 ± 0.06	**0.15** ± 0.00	3.15 ± 0.03	1.54 ± 0.02
M^3-Impute	**2.14** ± 0.00	**0.87** ± 0.00	**1.56** ± 0.01	**1.08** ± 0.00	**2.05** ± 0.00	0.17 ± 0.00	**2.49** ± 0.00	**1.37** ± 0.00

Table 9: MAE scores on seven additional datasets

	airfoil	blood	wine-white	ionosphere	breast	iris	diabetes
MCAR							
Mean	2.32 $\pm$ 0.05	1.14 $\pm$ 0.01	0.76 $\pm$ 0.00	2.01 $\pm$ 0.03	1.06 $\pm$ 0.00	2.15 $\pm$ 0.09	1.78 $\pm$ 0.03
Svd	2.76 $\pm$ 0.05	0.97 $\pm$ 0.04	0.87 $\pm$ 0.00	1.26 $\pm$ 0.03	0.58 $\pm$ 0.00	1.70 $\pm$ 0.07	1.76 $\pm$ 0.02
Spectral	2.30 $\pm$ 0.07	0.94 $\pm$ 0.03	0.78 $\pm$ 0.01	1.38 $\pm$ 0.02	0.38 $\pm$ 0.00	1.48 $\pm$ 0.13	1.48 $\pm$ 0.03
Mice	1.97 $\pm$ 0.04	0.69 $\pm$ 0.01	0.61 $\pm$ 0.01	1.37 $\pm$ 0.03	0.34 $\pm$ 0.01	1.07 $\pm$ 0.09	1.29 $\pm$ 0.05
Knn	2.18 $\pm$ 0.04	0.93 $\pm$ 0.01	0.64 $\pm$ 0.01	1.07 $\pm$ 0.03	0.53 $\pm$ 0.01	1.54 $\pm$ 0.22	1.71 $\pm$ 0.04
Gain	2.22 $\pm$ 0.06	1.26 $\pm$ 0.04	0.73 $\pm$ 0.01	1.50 $\pm$ 0.01	0.51 $\pm$ 0.01	1.29 $\pm$ 0.07	1.47 $\pm$ 0.06
Miracle	2.13 $\pm$ 0.05	43.17 $\pm$ 0.05	0.60 $\pm$ 0.00	37.70 $\pm$ 0.22	35.07 $\pm$ 0.41	45.13 $\pm$ 0.42	41.00 $\pm$ 0.14
Grape	$\underline{1.16}$ $\pm$ 0.02	0.68 $\pm$ 0.00	**0.52** $\pm$ 0.00	$\underline{1.08}$ $\pm$ 0.01	0.37 $\pm$ 0.00	**0.82** $\pm$ 0.00	1.31 $\pm$ 0.00
Miwae	2.36 $\pm$ 0.06	2.03 $\pm$ 0.05	0.77 $\pm$ 0.00	5.14 $\pm$ 0.06	1.89 $\pm$ 0.02	4.60 $\pm$ 0.17	5.05 $\pm$ 0.04
HyperImpute	**1.09** $\pm$ 0.02	**0.63** $\pm$ 0.02	$\underline{0.55}$ $\pm$ 0.00	1.18 $\pm$ 0.04	**0.33** $\pm$ 0.01	$\underline{1.04}$ $\pm$ 0.11	**1.17** $\pm$ 0.02
M^3-Impute	**1.09** $\pm$ 0.03	$\underline{0.67}$ $\pm$ 0.00	**0.52** $\pm$ 0.00	**1.01** $\pm$ 0.01	$\underline{0.36}$ $\pm$ 0.01	**0.82** $\pm$ 0.00	$\underline{1.29}$ $\pm$ 0.01
MAR							
Mean	2.33 $\pm$ 0.14	0.91 $\pm$ 0.02	0.87 $\pm$ 0.01	2.02 $\pm$ 0.08	1.13 $\pm$ 0.03	1.99 $\pm$ 0.25	1.74 $\pm$ 0.33
Svd	2.99 $\pm$ 0.83	0.91 $\pm$ 0.07	0.78 $\pm$ 0.05	1.40 $\pm$ 0.08	0.61 $\pm$ 0.03	1.85 $\pm$ 0.42	2.09 $\pm$ 0.02
Spectral	2.01 $\pm$ 0.60	1.22 $\pm$ 0.36	0.99 $\pm$ 0.23	1.50 $\pm$ 0.02	0.46 $\pm$ 0.04	1.62 $\pm$ 0.13	1.32 $\pm$ 0.20
Mice	2.16 $\pm$ 0.28	1.00 $\pm$ 0.40	0.63 $\pm$ 0.04	1.43 $\pm$ 0.08	0.32 $\pm$ 0.07	0.85 $\pm$ 0.09	1.33 $\pm$ 0.23
Knn	1.59 $\pm$ 0.70	0.90 $\pm$ 0.25	0.53 $\pm$ 0.02	1.09 $\pm$ 0.03	0.53 $\pm$ 0.03	0.91 $\pm$ 0.08	1.43 $\pm$ 0.23
Gain	2.29 $\pm$ 0.09	1.01 $\pm$ 0.15	0.65 $\pm$ 0.11	1.71 $\pm$ 0.10	0.69 $\pm$ 0.05	1.25 $\pm$ 0.04	1.34 $\pm$ 0.04
Miracle	2.08 $\pm$ 0.26	42.30 $\pm$ 0.22	1.05 $\pm$ 0.05	26.60 $\pm$ 0.37	39.53 $\pm$ 0.17	49.60 $\pm$ 1.14	41.83 $\pm$ 0.09
Grape	1.57 $\pm$ 0.02	$\underline{0.29}$ $\pm$ 0.01	**0.48** $\pm$ 0.00	$\underline{1.17}$ $\pm$ 0.03	0.39 $\pm$ 0.00	$\underline{0.86}$ $\pm$ 0.02	$\underline{1.12}$ $\pm$ 0.01
Miwae	2.56 $\pm$ 0.01	2.03 $\pm$ 0.03	0.69 $\pm$ 0.01	6.10 $\pm$ 0.04	2.17 $\pm$ 0.03	3.46 $\pm$ 0.13	4.26 $\pm$ 0.06
HyperImpute	**1.21** $\pm$ 0.21	0.88 $\pm$ 0.33	$\underline{0.57}$ $\pm$ 0.08	1.30 $\pm$ 0.03	**0.34** $\pm$ 0.02	1.05 $\pm$ 0.11	1.46 $\pm$ 0.10
M^3-Impute	$\underline{1.54}$ $\pm$ 0.02	**0.28** $\pm$ 0.01	**0.48** $\pm$ 0.00	**1.07** $\pm$ 0.01	$\underline{0.37}$ $\pm$ 0.01	**0.82** $\pm$ 0.03	**1.07** $\pm$ 0.00
MNAR							
Mean	2.36 $\pm$ 0.11	0.98 $\pm$ 0.05	0.82 $\pm$ 0.01	2.04 $\pm$ 0.06	1.11 $\pm$ 0.02	2.06 $\pm$ 0.09	1.77 $\pm$ 0.20
Svd	2.98 $\pm$ 0.52	0.98 $\pm$ 0.09	0.82 $\pm$ 0.04	1.36 $\pm$ 0.07	0.60 $\pm$ 0.03	1.66 $\pm$ 0.20	1.93 $\pm$ 0.02
Spectral	2.64 $\pm$ 0.18	1.40 $\pm$ 0.18	0.88 $\pm$ 0.13	1.46 $\pm$ 0.02	0.41 $\pm$ 0.03	1.35 $\pm$ 0.11	1.51 $\pm$ 0.13
Mice	2.07 $\pm$ 0.14	0.76 $\pm$ 0.17	0.62 $\pm$ 0.02	1.44 $\pm$ 0.07	0.33 $\pm$ 0.02	0.99 $\pm$ 0.11	1.27 $\pm$ 0.16
Knn	2.11 $\pm$ 0.27	1.04 $\pm$ 0.12	0.60 $\pm$ 0.02	1.12 $\pm$ 0.03	0.55 $\pm$ 0.02	1.53 $\pm$ 0.52	1.60 $\pm$ 0.17
Gain	2.21 $\pm$ 0.05	1.09 $\pm$ 0.06	0.69 $\pm$ 0.01	1.55 $\pm$ 0.03	0.62 $\pm$ 0.02	1.26 $\pm$ 0.04	1.43 $\pm$ 0.06
Miracle	1.72 $\pm$ 0.08	42.90 $\pm$ 0.14	0.59 $\pm$ 0.01	30.70 $\pm$ 0.57	37.30 $\pm$ 0.29	47.37 $\pm$ 0.90	41.60 $\pm$ 0.37
Grape	$\underline{1.46}$ $\pm$ 0.03	$\underline{0.42}$ $\pm$ 0.00	**0.49** $\pm$ 0.00	$\underline{1.15}$ $\pm$ 0.01	$\underline{0.38}$ $\pm$ 0.00	$\underline{0.89}$ $\pm$ 0.02	$\underline{1.21}$ $\pm$ 0.01
Miwae	2.47 $\pm$ 0.03	1.99 $\pm$ 0.04	0.72 $\pm$ 0.00	5.66 $\pm$ 0.02	2.05 $\pm$ 0.00	3.98 $\pm$ 0.32	4.62 $\pm$ 0.08
HyperImpute	**1.23** $\pm$ 0.04	0.82 $\pm$ 0.18	$\underline{0.58}$ $\pm$ 0.05	1.28 $\pm$ 0.02	**0.36** $\pm$ 0.03	1.07 $\pm$ 0.07	1.30 $\pm$ 0.19
M^3-Impute	$\underline{1.46}$ $\pm$ 0.01	**0.41** $\pm$ 0.00	**0.49** $\pm$ 0.00	**1.06** $\pm$ 0.02	**0.36** $\pm$ 0.01	**0.87** $\pm$ 0.00	**1.19** $\pm$ 0.00

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction sections, we clearly define the scope of this paper, focusing on missing value imputation. We propose M^3-Impute, a mask-guided imputation method designed to compute feature-wise and sample-wise correlations based on missing data patterns. A concise summary of the experimental results is provided at the end of both sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 4.2, we discussed two cases of MAE degradation for the KIN8NM and NAVAL datasets. It is mainly because 1. Each feature in KIN8NM is independent of the others, so none of the observed features can help impute missing feature values. 2. In the NAVAL dataset, nearly every feature exhibits a strong linear correlation with the other features. While it is true that M^3-Impute does not achieve the best MAE on these two datasets, our model has outperformed all the other baselines on the majority of datasets. This demonstrates the unique strengths of graph modeling in M^3-Impute over tabular data modeling in baselines like Hyperimpute. In real-world scenarios, the correlation structure of datasets is often unpredictable, and such extreme cases are relatively rare. Thus, we design a scheme to handle general cases for data imputation tasks. The empirical evidence suggests that our approach has been quite successful and exhibits overall superior performance to the baselines as it can be well adapted to each dataset that possesses different levels of correlations over features and samples.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not present theoretical results. We do not assume that the data is missing under MCAR, MAR, or MNAR conditions for M^3-Impute to be effective. Instead, M^3-Impute demonstrates robust performance across all three settings.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 3, we explain the computational pipeline of the proposed model in detail and provide a pseudo-code to better outline the methodology. The experimental setup is comprehensively described in Section 4.1. In addition, supplementary materials include our **complete codebase** to reproduce the results presented in this paper, including the model implementation, training and testing pipeline, configuration files, and execution scripts.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via

detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In the supplementary material, we provide the complete code for our model, including the scripts for experiments and evaluations, as well as the execution scripts used in the experiments. We have also included the data preprocessed by us, along with the download links for publicly available datasets. We will release the formatted codebase on GitHub following the conclusion of the anonymity period.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: Experimental setup is detailed in Section 4.1 and Appendix A.2. We also explore the hyperparameters utilized in M^3-Impute. Results are presented in Table 3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We conduct all the experiments over five runs and report the mean MAE scores, along with the standard deviations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We train and test M^3-Impute on a single Nvidia A100 80G GPU (Detailed setup described in A.5). With the experimental setup described in Section 4.1, the total running time (including training and testing) for one of the five repeated runs varies from 1 to 5 hours, depending on the scale of the datasets.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our paper adheres to the NeurIPS Code of Ethics in every respect. 1. We ensured fair wages for all human participants involved in our study, abiding by regional minimum hourly rates. 2. Our research methodology adhered to institutional protocols for human subjects and data privacy. 3. We obtained informed consent from all participants and minimized exposure of personally identifiable information. 4. The datasets used are publicly available and have not been deprecated, with all copyrights respected. 5. We have transparently communicated the societal impact of our research, considering potential misuse and its effects on discrimination, surveillance, and environmental impact. 6. We have also reflected on the biases in our models and datasets and taken steps to mitigate them. 7. Our data and models are documented and released with appropriate licenses, and we've employed secure data storage and distribution practices. 8. We ensured legal compliance and provided all necessary elements for the reproducibility of our research.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The method proposed in this work is only applicable for missing value imputation and is unlikely to have a negative social impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The model we propose does not carry the risk of misuse; the datasets were selected under fair use conditions, from publicly available sources with undisputed licenses. Therefore, our work does not require additional safeguard protections.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: In our research, we have carefully credited all the code and data used, providing explicit citations for each. The licenses for this code and data are notably permissive, including MIT, BSD 3-clause, and CC BY 4.0. In accordance with these licenses, we have properly acknowledged the contributions of the original authors.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have detailed the new datasets employed in this research in Appendix A.4. We commit to making these datasets publicly accessible following the anonymity period to foster transparency and reproducibility.

Guidelines:

- The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.