# Variational Graph Inference for Structural Missingness in Heterogeneous Data

Youran Zhou(✉), Mohamed Reda Bouadjenek, and Sunil Aryal

School of Information Technology, Deakin University, Geelong, Victoria, Australia
{echo.zhou, reda.bouadjenek, sunil.aryal}@deakin.edu.au

**Abstract.** Learning from incomplete heterogeneous tabular data is challenging because missingness often exhibits structural dependencies across samples and features rather than occurring independently. Viewing tabular data as a partially observed bipartite graph, missing entries correspond to removed edges and structured missingness manifests as sparsification patterns in the sample–feature interaction graph. To address this problem, we propose **IVGAE**, a variational graph framework that jointly models latent data representations and missingness mechanisms. IVGAE performs variational inference over the bipartite graph to capture structural dependencies among heterogeneous features while modeling missingness as a graph-level generative process. A dual-decoder architecture simultaneously reconstructs feature values and infers missingness patterns under MCAR, MAR, and MNAR regimes. To effectively encode heterogeneous attributes, we introduce a Transformer-based feature embedding module that preserves categorical semantics without relying on high-dimensional one-hot representations. Experiments on 16 real-world datasets demonstrate that IVGAE consistently improves reconstruction accuracy and downstream prediction performance compared with strong imputation baselines across diverse missing regimes. The codes are available at: `https://github.com/echoid/IVGAE`

**Keywords:** Incomplete data · Heterogeneous data · Missing Mechanism · Graph representation learning · Variational Autoencoder.

## 1 Introduction

Incomplete data is a persistent challenge in real-world machine learning applications, arising from diverse factors such as sensor failures, human errors, and privacy-driven data omissions. Missing values not only reduce the amount of available information but can also introduce statistical bias and degrade the reliability of learned models. This problem is particularly pronounced in *heterogeneous tabular data*, where numerical and categorical attributes coexist and interact in complex ways across samples and features. Such heterogeneity makes modeling feature dependencies more difficult and often limits the effectiveness of conventional imputation techniques.

A key difficulty in learning from incomplete data lies in the underlying *missingness mechanism*. In statistical literature, missing data are commonly categorized as missing completely at random (MCAR), missing at random (MAR), or

missing not at random (MNAR). These mechanisms describe how the probability of missingness depends on observed or unobserved variables and fundamentally determine the feasibility and bias of imputation. In practice, however, missingness rarely occurs independently. Instead, missing entries often exhibit *structural patterns* driven by correlations among features, data collection processes, or latent factors. Such structured missingness implies that the absence of a value may carry informative signals about both the data distribution and the missingness process itself.

Traditional imputation approaches, including statistical heuristics and classical machine learning methods, often fail to capture higher-order dependencies or rely on rigid distributional assumptions. For instance, mean or mode [8, 21] imputation operates independently per feature, while $k$-nearest neighbors [33], MICE [4], Regression [3], Optimal Transport [27, 17] and matrix factorization [16] impose linearity or independence assumptions that generalize poorly to unseen data.

Graph-based learning has recently emerged as a promising paradigm for modeling structured relationships in tabular data. By representing tabular datasets as bipartite graphs connecting samples and features, graph-based models can naturally capture relational dependencies and propagate information through the data structure [31, 34]. Graph Neural Networks (GNNs) have therefore been explored for missing value imputation [6, 22, 20, 29, 7]. Nevertheless, most existing graph-based imputers primarily focus on reconstructing missing feature values and often overlook the structured nature of missingness itself. As a result, their robustness may degrade under more complex missing regimes such as MAR or MNAR.

To address these challenges, we propose **IVGAE**, an **I**ncomplete **V**ariational **G**raph **A**uto**E**ncoder designed for heterogeneous tabular data with structural missingness. IVGAE models tabular data as a partially observed bipartite graph and performs variational inference over this graph structure to learn latent representations that capture both feature dependencies and missingness patterns. The model adopts a *dual-decoder architecture*, where one decoder reconstructs feature values while the other estimates missingness masks as edge probabilities in the graph. This design enables IVGAE to jointly model data reconstruction and missingness inference, improving robustness across different missingness mechanisms. In addition, we introduce a Transformer-based heterogeneous embedding module to encode categorical features without relying on high-dimensional one-hot representations.

Extensive experiments on 16 real-world datasets demonstrate that IVGAE consistently outperforms existing baselines across multiple missingness settings. The proposed framework achieves significant improvements in reconstruction accuracy and downstream predictive performance, highlighting the effectiveness of modeling structural dependencies in incomplete heterogeneous data.

Our contributions are summarized as follows:

- We propose a **variational graph inference framework** for incomplete heterogeneous tabular data by modeling tabular datasets as bipartite graphs that capture structural dependencies between samples and features.
- We design a **dual-decoder variational graph autoencoder** that jointly reconstructs feature values and infers missingness patterns, enabling mechanism-aware learning under MCAR, MAR, and MNAR settings.
- We introduce a **Transformer-based heterogeneous embedding module** that efficiently encodes categorical attributes without high-dimensional one-hot expansion.
- We conduct **extensive experiments on 16 real-world datasets**, demonstrating consistent improvements in both imputation accuracy and downstream predictive tasks.

## 2 Related Work

*Statistical Imputation Methods.* Early imputation techniques rely on statistical heuristics such as mean, mode, or $k$-nearest neighbors (KNN) [10]. More advanced approaches include Multiple Imputation by Chained Equations (MICE) [4] and MissForest [24], which iteratively model feature relationships through regression or ensemble learning. Low-rank EM-based methods further assume that the complete data matrix lies in a low-dimensional subspace [23]. While these methods are effective in relatively simple settings, they typically rely on strong distributional assumptions and often treat missing entries independently, limiting their ability to capture complex dependencies under realistic missingness mechanisms. Domain-specific approaches developed for recommender systems [14, 28] partially address MNAR bias but are tailored to particular application domains.

*Deep Generative Imputation Models.* Recent work has explored deep generative models for imputing incomplete data by learning flexible data distributions. Adversarial approaches such as GAIN [30] and MisGAN [1] employ generator–discriminator frameworks to estimate missing values, while variational methods including MIWAE [15] and NOT-MIWAE [9] learn latent representations consistent with observed entries. HIVAE [18] further extends variational autoencoders to heterogeneous tabular data. Although these models provide greater modeling flexibility than traditional methods, many still require careful preprocessing of categorical attributes and do not explicitly model the missingness process [5, 25].

*Graph-Based Imputation.* Graph-based learning has recently emerged as an effective paradigm for modeling relational dependencies in incomplete data. Early work such as GC-MC [26] and IGMC [32] applied graph neural networks to matrix completion problems. More recent approaches, including GRAPE [31] and IGRM [34], represent tabular datasets as bipartite graphs to capture interactions between samples and features. By propagating information through graph structures, these models can exploit relational patterns that are difficult to capture

with traditional methods. However, most existing graph-based imputers primarily focus on reconstructing missing feature values and do not explicitly model the missingness process itself, which can limit their robustness under complex missingness mechanisms such as MAR and MNAR.

In contrast to these approaches, our proposed **IVGAE** performs variational graph inference on bipartite sample–feature graphs and employs a dual-decoder architecture to jointly reconstruct feature values and infer missingness patterns, enabling mechanism-aware learning for heterogeneous tabular data.

## 3 IVGAE: The Proposed Method

### 3.1 Problem Formulation

**Missing Data.** We consider a dataset represented by a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, where $n$ and $p$ denote the numbers of samples and features. Let $X_{ij}$ denote the value of feature $v_j$ for sample $x_i$. Missingness is represented by a binary mask matrix $\mathbf{M} \in \{0,1\}^{n \times p}$, where $M_{ij} = 1$ indicates an observed entry and $M_{ij} = 0$ a missing one. The dataset can therefore be decomposed into observed entries $\mathbf{X}^o$ and missing entries $\mathbf{X}^m$. In practice, missing entries often exhibit structured patterns across samples and features rather than occurring independently. Following Rubin's taxonomy [19], missingness mechanisms are categorized as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR), depending on whether the probability of missingness depends on neither the data, the observed data, or the unobserved data. The goal of imputation is to estimate the missing entries $\mathbf{X}^m$ given the observed data $\mathbf{X}^o$ and the mask $\mathbf{M}$, while capturing the underlying structure of missingness patterns.

**Bipartite Graph Representation.** We represent the dataset as an undirected bipartite graph $\mathcal{G} = (\mathcal{X}, \mathcal{V}, \mathcal{E})$, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ denotes sample nodes and $\mathcal{V} = \{v_1, \ldots, v_p\}$ feature nodes. An edge $(x_i, v_j) \in \mathcal{E}$ exists if the entry $X_{ij}$ is observed (i.e., $M_{ij} = 1$), with edge weight equal to the observed value $X_{ij}$.

This representation transforms the tabular dataset into a sample–feature interaction graph, enabling the model to capture dependencies among features and structured missingness patterns. The resulting adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ serves as the input structure for subsequent graph inference.

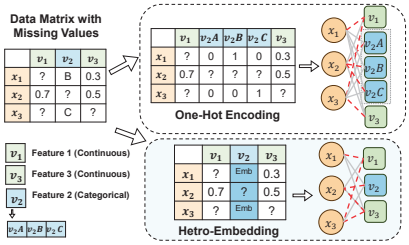



Fig. 1: Encoding strategies for heterogeneous data in a bipartite graph. **Top:** One-hot encoding expands categorical features into multiple binary nodes. **Bottom:** The proposed embedding learns compact representations while preserving feature relationships.

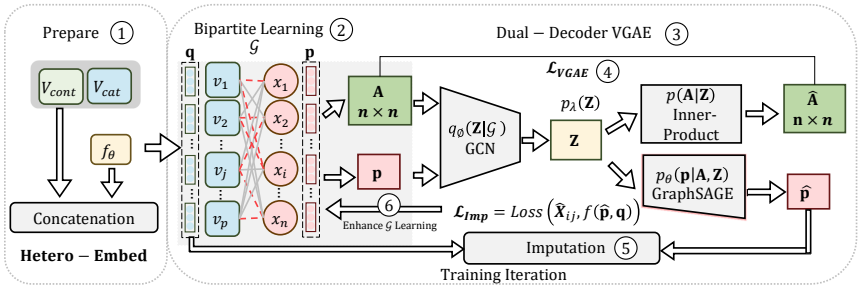



Fig. 2: Overview of `IVGAE`. The model encodes a bipartite graph, learns latent node embeddings via variational inference, and reconstructs feature values $\hat{\mathbf{X}}$ and adjacency $\hat{\mathbf{A}}$ through a dual-decoder mechanism for mechanism-aware imputation.

### 3.2 Heterogeneous Feature Encoding

To obtain node representations for graph construction, we encode each sample into a unified feature embedding. Given a sample $x_i$, let $\mathbf{v}_{\text{cont}}$ denote its numerical features and $\mathbf{v}_{\text{cat}}$ its categorical features. Categorical attributes are first mapped to token embeddings and then processed by a Transformer encoder that models interactions among feature tokens. Numerical features are normalized and projected through a linear transformation. The resulting heterogeneous embedding is defined as $\mathbf{h}_i = g_\phi(\mathbf{v}_{\text{cont}}) \parallel f_\theta(\mathbf{v}_{\text{cat}})$ where $f_\theta(\cdot)$ denotes the Transformer-based embedding function for categorical attributes and $g_\phi(\cdot)$ the projection of numerical features. The embedding $\mathbf{h}_i$ serves as the node representation of sample $x_i$ in the bipartite graph.

### 3.3 Graph Representation Learning

**Network Architecture.** We adopt the Variational Graph Autoencoder (VGAE) framework [12] to perform imputation as an edge prediction task on the bipartite graph. As illustrated in Figure 2, the proposed `IVGAE` consists of two components: (i) a **bipartite graph encoder** that learns latent representations for sample and feature nodes, and (ii) a **dual-decoder module** that reconstructs both feature values and missingness patterns. Existing graph-based imputers [31] typically estimate missing values through deterministic edge prediction

$$\hat{X}_{ij} = \mathcal{F}(\mathcal{G}), \tag{1}$$

where $\mathcal{F}$ denotes a reconstruction function over graph $\mathcal{G}$. In contrast, `IVGAE` introduces a **dual-decoder mechanism** that jointly reconstructs refined sample embeddings $\mathbf{p}$ and the adjacency matrix $\mathbf{A}$, enabling mechanism-aware modeling of missingness.

**Node and Edge Updates.** Let $\mathbf{p}_i^{(l)}$ and $\mathbf{q}_j^{(l)}$ denote the sample and feature node embeddings at layer $l$, respectively. Given the bipartite graph $\mathcal{G} = (\mathcal{X}, \mathcal{V}, \mathcal{E})$,

IVGAE learns node representations through multi-layer message passing. At layer $l$, the sample node aggregates messages from its neighboring feature nodes:

$$h_i^{(l)} = \text{Mean}_{j \in \mathcal{N}(i)} \left( \delta \left( \mathbf{W}^{(l)} \cdot \text{Concat}(\mathbf{q}_j^{(l-1)}, e_{ij}^{(l-1)}) \right) \right), \tag{2}$$

where $\mathcal{N}(i)$ denotes the set of feature nodes connected to sample $x_i$, $\mathbf{W}^{(l)}$ is a learnable weight matrix, $e_{ij}^{(l-1)}$ denotes the edge embedding, and $\delta(\cdot)$ is a non-linear activation. Node and edge representations are jointly refined at each layer through bidirectional message passing:

$$\begin{aligned} \mathbf{p}_i^{(l)} &= \delta \left( \widehat{\mathbf{W}}^{(l)} \cdot \text{Concat}(\mathbf{p}_i^{(l-1)}, h_i^{(l)}) \right), \\ \mathbf{q}_j^{(l)} &= \delta \left( \mathbf{Q}^{(l)} \cdot \text{Concat}(\mathbf{q}_j^{(l-1)}, h_j^{(l)}) \right), \\ e_{ij}^{(l)} &= \delta \left( \mathbf{U}^{(l)} \cdot \text{Concat}(e_{ij}^{(l-1)}, \mathbf{p}_i^{(l)}, \mathbf{q}_j^{(l)}) \right), \end{aligned} \tag{3}$$

$\widehat{\mathbf{W}}^{(l)}$, $\mathbf{Q}^{(l)}$, and $\mathbf{U}^{(l)}$ are learnable transformation matrices for updating sample nodes, feature nodes, and edge representations, respectively.

### 3.4 Latent Representation Learning

To learn expressive representations for imputation, IVGAE adopts the Variational Graph Autoencoder [12] as the backbone to model latent node embeddings conditioned on graph structure. Unlike deterministic graph autoencoders, VGAE employs variational inference to capture uncertainty in the latent space, which improves robustness under incomplete or noisy observations. Building on this framework, IVGAE introduces a **dual reconstruction strategy** that simultaneously models relational structure and feature representations. Given the graph encoder described above, the model reconstructs two complementary components: A reconstructed **adjacency matrix** $\hat{\mathbf{A}}$, which captures latent relational dependencies among samples and reflects structured missingness patterns and a refined **sample embeddings** $\hat{\mathbf{p}}$, which encode high-order feature interactions informed by the learned graph topology.

Jointly reconstructing $\hat{\mathbf{A}}$ and $\hat{\mathbf{p}}$ enables IVGAE to integrate relational priors with feature-level semantics, improving the consistency and accuracy of downstream imputation.

**Adjacency Reconstruction and Missingness Modeling.** The adjacency matrix $\mathbf{A}$ encodes latent similarity between samples, where $\mathbf{A}_{ij}$ reflects whether samples $x_i$ and $x_j$ exhibit similar feature patterns [34, 12]. Reconstructing $\hat{\mathbf{A}}$ allows IVGAE to capture structural relationships in the data and propagate reliable information across similar samples.

This mechanism is particularly beneficial under non-random missingness. For **MNAR** settings, where missingness depends on unobserved attributes, samples with similar profiles often exhibit similar missing patterns. By learning $\hat{\mathbf{A}}$, IVGAE enables information transfer across such neighborhoods, mitigating biases

caused by systematic data omission. For **MAR** scenarios, where missingness correlates with observed features, the reconstructed adjacency structure ensures that inferred values remain consistent with observed relational patterns.

Through this mechanism-aware reconstruction, IVGAE bridges sample-level similarity and feature-level dependencies, enabling robust imputation across diverse missingness mechanisms.

**Variational Inference.** Within IVGAE, the encoder learns probabilistic latent embeddings for each node conditioned on the observed graph structure. Following the Variational Graph Autoencoder (VGAE) formulation [12], the posterior distribution over latent variables is defined as

$$q_\phi(\mathbf{Z}|\mathbf{p},\mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)), \tag{4}$$

where $N$ denotes the number of nodes in the graph. Each latent embedding $\mathbf{z}_i$ follows a Gaussian distribution parameterized by mean $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\sigma}_i^2$. These parameters are generated by two Graph Convolutional Networks (GCNs):

$$\boldsymbol{\mu} = \mathrm{GCN}_\mu(\mathbf{p},\mathbf{A}), \qquad \log \boldsymbol{\sigma} = \mathrm{GCN}_\sigma(\mathbf{p},\mathbf{A}).$$

The GCN operation is defined as

$$\mathrm{GCN}(\mathbf{p},\mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{p}\mathbf{W}_0)\mathbf{W}_1, \tag{5}$$

where $\tilde{\mathbf{A}}$ denotes the symmetrically normalized adjacency matrix and $\mathbf{W}_0, \mathbf{W}_1$ are learnable weight matrices. Through neighborhood aggregation, the encoder captures structural dependencies among nodes and produces stochastic latent representations. The variational formulation further models epistemic uncertainty arising from incomplete observations.

**Graph Reconstruction.** The decoder reconstructs both the graph structure and node representations.

The adjacency matrix $\hat{\mathbf{A}}$ is reconstructed using an inner-product decoder:

$$p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{N} p(\mathbf{A}_{ij}|\mathbf{z}_i,\mathbf{z}_j), \qquad p(\mathbf{A}_{ij}=1|\mathbf{z}_i,\mathbf{z}_j) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j), \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function. This reconstruction performs link prediction to recover latent relational structures in the graph. In addition to adjacency reconstruction, IVGAE refines node embeddings using a GraphSAGE decoder [7]:

$$p_\theta(\mathbf{p}|\mathbf{A},\mathbf{Z}) = \mathrm{GraphSAGE}(\mathbf{Z},\mathbf{A}). \tag{7}$$

For node $i$, the reconstructed embedding is obtained by aggregating information from neighboring nodes:

$$\hat{\mathbf{p}}_i = \delta\left(\mathbf{\Theta} \cdot \text{Concat}\left(\mathbf{p}_i, \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{p}}_j \tilde{\mathbf{A}}_{ij}\right)\right), \tag{8}$$

where $\mathbf{\Theta}$ is a learnable weight matrix, $\delta(\cdot)$ denotes a nonlinear activation, and $\mathcal{N}(i)$ is the set of neighbors of node $i$.

Equations 4–8 correspond to Step 3 in Figure 2, illustrating how the encoder–decoder pipeline jointly learns probabilistic latent embeddings while reconstructing both graph topology and node features. Together, these processes enable IVGAE to capture structural priors and uncertainty for mechanism-aware data imputation.

**Algorithm 1:** Training Procedure of `IVGAE`

**Input:** Heterogeneous data $\mathbf{V}_{\text{cont}}, \mathbf{V}_{\text{cat}}$, initial adjacency $\mathbf{A}$, iterations $T$
**Output:** Imputed matrix $\hat{\mathbf{X}}$, predicted adjacency $\hat{\mathbf{A}}$

$\mathbf{X} \leftarrow \text{Concat}(f_\theta(\mathbf{V}_{\text{cat}}), \mathbf{V}_{\text{cont}})$;
**for** $t \leftarrow 1$ **to** $T$ **do**
  Construct bipartite graph $\mathcal{G}^{(t)}$ from $\mathbf{X}$;
  Compute embeddings $\mathbf{p}^{(t)}, \mathbf{q}^{(t)}$ via Eq. (3);
  Infer $\mathbf{Z} \sim q_\phi(\mathbf{Z} \mid \mathbf{p}^{(t)}, \mathbf{A})$ via Eq. (4)–(5);
  Reconstruct adjacency $\hat{\mathbf{A}}^{(t)}$ via Eq. (6);
  Reconstruct embeddings $\hat{\mathbf{p}}^{(t)}$ via Eq. (7)–(8);
  Predict missing values $\hat{\mathbf{X}}$ via Eq. (10);
  Compute $\mathcal{L}_{\text{imp}}$ via Eq. (11);
  Update graph using $\hat{\mathbf{p}}^{(t)}$;
**return** $\hat{\mathbf{X}}, \hat{\mathbf{A}}$

### 3.5 Training Objective

The parameters of IVGAE are learned by maximizing the evidence lower bound (ELBO), which approximates the marginal likelihood of the observed graph structure:

$$\mathcal{L}_{\text{VGAE}} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{p},\mathbf{A})}\left[\log p_\theta(\mathbf{A}|\mathbf{Z})\right] - \text{KL}\left(q_\phi(\mathbf{Z}|\mathbf{p},\mathbf{A}) \parallel p(\mathbf{Z})\right), \tag{9}$$

where the first term measures the reconstruction likelihood of the adjacency matrix and the second term regularizes the approximate posterior toward the prior distribution $p(\mathbf{Z})$. Optimization is performed using gradient descent with the reparameterization trick [11] to enable differentiable sampling of latent variables. Although IVGAE employs a dual-decoder architecture, the ELBO objective focuses on reconstructing the adjacency matrix $\mathbf{A}$, which implicitly guides missing value recovery. Since unobserved entries correspond to absent edges in the bipartite graph, accurate adjacency reconstruction captures structural priors that reflect the underlying missingness mechanism. The feature decoder then refines

node embeddings deterministically using GraphSAGE [7], enabling local feature aggregation without additional stochastic modeling.

**Edge Imputation.** After training, missing entries are imputed through edge prediction on the sample–feature bipartite graph. Given the learned sample and feature embeddings $\hat{\mathbf{p}}_i$ and $\mathbf{q}_j$, the imputed value is estimated as

$$\hat{X}_{ij} = \sigma(f(\text{Concat}(\hat{\mathbf{p}}_i, \mathbf{q}_j))), \tag{10}$$

where $f(\cdot)$ denotes a feed-forward network and $\sigma(\cdot)$ constrains the output range. This formulation treats imputation as an edge-wise regression problem on the interaction graph. To handle heterogeneous feature types, IVGAE optimizes a hybrid reconstruction loss:

$$\mathcal{L}_{\text{Imp}} = \text{MSE}(\hat{X}_{ij}, X_{ij}) + \text{CE}(\hat{X}_{ij}, X_{ij}), \tag{11}$$

where mean squared error (MSE) is applied to continuous features and cross-entropy (CE) is used for categorical variables. Equations 10–11 correspond to the edge prediction stage illustrated in Fig. 2.

## 4 Experiments

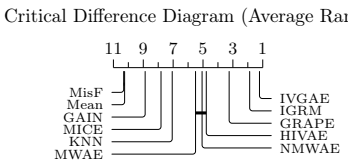



Fig. 3: Critical difference diagram of average `AvgErr` ranks. Lower is better; connected methods are not significantly different at $\alpha = 0.05$.

**Datasets and Missing Data Simulation.** We evaluate `IVGAE` on sixteen publicly available tabular datasets from the UCI Machine Learning Repository and Kaggle, covering diverse domains and heterogeneous feature types. Detailed dataset statistics are provided in the supplementary material. To simulate incomplete data, we inject artificial missingness into originally observed entries at rates of 10%, 30%, 50%, and 70% under three standard mechanisms: **MCAR**, where entries are removed uniformly at random; **MAR**, where the missingness of a feature $x_k$ depends on another observed feature $x_{k'}$ via threshold-based removal [17]; and **MNAR**, where missingness depends on the feature value itself, with extreme continuous values or selected categorical levels assigned higher missingness probabilities [35]. Model performance is evaluated from two perspectives: (i) **imputation accuracy** and (ii) **downstream predictive performance**. Datasets without labels are used only for reconstruction evaluation. Further details of the missingness generation procedures are provided in the supplementary material.

**Baseline Methods.** We compare `IVGAE` with representative imputation baselines spanning statistical, deep generative, and graph-based paradigms. Statistical methods include `Mean/Mode`, `KNN` [33], `MICE` [4], and `MissForest` [24].

Deep generative models include GAIN [30], MWAE [15], NMWAE [9], and HI-VAE [18]. Graph-based approaches include GRAPE [31] and IGRM [34]. For methods that do not natively support categorical inputs, categorical features are converted using one-hot encoding to ensure consistent evaluation.

**Evaluation Metrics.** We evaluate imputation performance using the **Average Error (AvgErr)** across variables, which measures the discrepancy between ground-truth and imputed values while accounting for heterogeneous feature types. Detailed formulations for continuous and categorical variables are provided in the supplementary material. To assess the utility of imputed data for downstream tasks, we train an XGBoost classifier on each imputed dataset and report the **F1-score** using five-fold cross-validation with hyperparameter tuning. Together, AvgErr and F1-score evaluate both reconstruction quality and predictive utility.

Table 1: AvgErr comparison for imputation under 30% MCAR, MAR, and MNAR. Best results are **bolded** and second-best are underlined. Lower is better.

| MCAR | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Conc.** | **Diab.** | **DOW.** | **Ecom.** | **Heart** | **Hous.** | **Sonar** | **Spam** | **Stud.** | **Wine** | **Yacht** |
| Mean | 0.245 | 0.261 | 0.172 | 0.280 | 0.233 | 0.183 | 0.437 | 0.153 | 0.251 | 0.233 | 0.182 | 0.321 | 0.295 | 0.312 | 0.098 | 0.214 |
| KNN | 0.189 | 0.175 | 0.133 | 0.219 | 0.190 | 0.125 | 0.257 | 0.040 | 0.276 | 0.181 | 0.096 | 0.275 | 0.243 | 0.251 | 0.080 | 0.218 |
| MICE | 0.192 | 0.188 | 0.145 | 0.228 | 0.202 | 0.136 | 0.332 | 0.059 | 0.243 | 0.204 | 0.116 | 0.281 | 0.250 | 0.259 | 0.076 | 0.209 |
| MisF | 0.275 | 0.233 | 0.213 | 0.301 | 0.278 | 0.215 | 0.364 | 0.094 | 0.275 | 0.243 | 0.144 | 0.320 | 0.285 | 0.306 | 0.108 | 0.280 |
| GAIN | 0.210 | 0.195 | 0.155 | 0.250 | 0.211 | 0.152 | 0.340 | 0.123 | 0.261 | 0.220 | 0.135 | 0.302 | 0.275 | 0.289 | 0.092 | 0.228 |
| NMWAE | 0.176 | 0.163 | 0.124 | 0.199 | 0.175 | 0.101 | 0.251 | 0.045 | 0.245 | 0.174 | 0.075 | 0.268 | 0.232 | 0.241 | 0.078 | 0.175 |
| MWAE | 0.178 | 0.162 | 0.122 | 0.205 | 0.182 | 0.118 | 0.250 | 0.044 | 0.257 | 0.182 | 0.072 | 0.249 | 0.237 | 0.247 | 0.072 | 0.180 |
| HIVAE | 0.168 | 0.160 | 0.117 | 0.195 | 0.165 | 0.099 | **0.237** | 0.035 | 0.244 | 0.182 | 0.091 | 0.235 | 0.218 | 0.237 | 0.074 | 0.175 |
| GRAPE | 0.162 | 0.143 | 0.108 | 0.195 | 0.169 | 0.086 | 0.249 | 0.020 | 0.235 | 0.155 | 0.076 | 0.220 | 0.220 | 0.230 | 0.063 | 0.161 |
| IGRM | 0.155 | 0.131 | 0.099 | 0.183 | 0.161 | 0.074 | 0.241 | 0.018 | **0.227** | 0.151 | 0.069 | 0.218 | 0.207 | 0.225 | 0.062 | **0.151** |
| IVGAE | **0.150** | **0.129** | **0.095** | **0.178** | **0.155** | **0.073** | 0.240 | **0.015** | **0.227** | **0.150** | **0.067** | **0.215** | **0.205** | **0.222** | **0.060** | **0.151** |
| **MAR** | | | | | | | | | | | | | | | | |
| **Method** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Conc.** | **Diab.** | **DOW.** | **Ecom.** | **Heart** | **Hous.** | **Sonar** | **Spam** | **Stud.** | **Wine** | **Yacht** |
| Mean | 0.263 | 0.277 | 0.191 | 0.298 | 0.251 | 0.205 | 0.459 | 0.175 | 0.269 | 0.247 | 0.197 | 0.341 | 0.311 | 0.327 | 0.113 | 0.233 |
| KNN | 0.208 | 0.191 | 0.151 | 0.237 | 0.218 | 0.141 | 0.276 | 0.059 | 0.295 | 0.186 | 0.119 | 0.295 | 0.256 | 0.275 | 0.095 | 0.238 |
| MICE | 0.211 | 0.255 | 0.163 | 0.242 | 0.221 | 0.150 | 0.351 | 0.077 | 0.261 | 0.221 | 0.135 | 0.304 | 0.274 | 0.280 | 0.097 | 0.231 |
| MisF | 0.213 | 0.216 | 0.178 | 0.315 | 0.276 | 0.238 | 0.378 | 0.109 | 0.290 | 0.257 | 0.159 | 0.307 | 0.304 | 0.324 | 0.135 | 0.297 |
| GAIN | 0.210 | 0.212 | 0.172 | 0.268 | 0.222 | 0.170 | 0.362 | 0.144 | 0.276 | 0.235 | 0.152 | 0.321 | 0.291 | 0.305 | 0.107 | 0.212 |
| NMWAE | 0.193 | 0.181 | **0.131** | 0.201 | 0.193 | 0.115 | 0.267 | 0.059 | 0.263 | 0.192 | 0.091 | 0.286 | 0.246 | 0.256 | 0.101 | **0.171** |
| MWAE | 0.193 | 0.181 | 0.135 | 0.213 | 0.185 | 0.141 | 0.264 | 0.064 | 0.277 | 0.199 | 0.090 | 0.262 | 0.252 | 0.268 | 0.088 | 0.178 |
| HIVAE | 0.190 | 0.178 | 0.137 | 0.226 | 0.197 | 0.112 | 0.258 | 0.052 | 0.263 | 0.204 | 0.107 | 0.257 | 0.231 | 0.259 | 0.094 | 0.196 |
| GRAPE | 0.181 | 0.161 | 0.135 | 0.207 | 0.188 | 0.105 | 0.263 | 0.041 | 0.257 | 0.176 | 0.091 | 0.240 | 0.235 | 0.253 | **0.077** | 0.179 |
| IGRM | 0.172 | 0.149 | 0.132 | 0.201 | 0.185 | 0.096 | 0.259 | **0.033** | **0.247** | 0.176 | **0.086** | 0.236 | 0.223 | 0.245 | 0.079 | 0.172 |
| IVGAE | **0.169** | **0.147** | **0.131** | **0.199** | **0.183** | **0.091** | **0.255** | 0.036 | **0.247** | **0.167** | **0.086** | **0.229** | **0.222** | **0.242** | 0.082 | **0.171** |
| **MNAR** | | | | | | | | | | | | | | | | |
| **Method** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Conc.** | **Diab.** | **DOW.** | **Ecom.** | **Heart** | **Hous.** | **Sonar** | **Spam** | **Stud.** | **Wine** | **Yacht** |
| Mean | 0.266 | 0.276 | 0.194 | 0.297 | 0.256 | 0.204 | 0.461 | 0.179 | 0.272 | 0.248 | 0.194 | 0.342 | 0.313 | 0.329 | 0.112 | 0.235 |
| KNN | 0.212 | 0.190 | 0.155 | 0.233 | 0.223 | 0.141 | 0.277 | 0.056 | 0.296 | 0.184 | 0.121 | 0.293 | 0.262 | 0.267 | 0.095 | 0.234 |
| MICE | 0.209 | 0.259 | 0.161 | 0.236 | 0.223 | 0.148 | 0.348 | 0.075 | 0.259 | 0.225 | 0.132 | 0.301 | 0.271 | 0.279 | 0.098 | 0.229 |
| MisF | 0.216 | 0.221 | 0.176 | 0.317 | 0.271 | 0.243 | 0.373 | 0.112 | 0.294 | 0.259 | 0.157 | 0.342 | 0.308 | 0.323 | 0.129 | 0.301 |
| GAIN | 0.213 | 0.209 | 0.173 | 0.264 | 0.225 | 0.172 | 0.362 | 0.143 | 0.276 | 0.233 | 0.153 | 0.318 | 0.292 | 0.287 | 0.107 | 0.215 |
| NMWAE | 0.197 | 0.182 | 0.134 | 0.205 | 0.193 | 0.117 | 0.262 | 0.057 | 0.257 | 0.196 | 0.087 | 0.287 | 0.251 | 0.260 | 0.106 | **0.172** |
| MWAE | 0.193 | 0.184 | 0.131 | 0.208 | 0.182 | 0.121 | 0.269 | 0.068 | 0.262 | 0.196 | 0.095 | 0.265 | 0.248 | 0.264 | 0.089 | 0.175 |
| HIVAE | 0.187 | 0.183 | 0.141 | 0.228 | 0.199 | 0.111 | **0.255** | 0.055 | 0.262 | 0.204 | 0.105 | **0.261** | 0.229 | 0.257 | 0.098 | 0.198 |
| GRAPE | 0.184 | 0.163 | 0.132 | 0.205 | 0.189 | 0.102 | 0.266 | 0.038 | 0.253 | 0.174 | 0.089 | **0.261** | 0.234 | 0.255 | 0.078 | 0.177 |
| IGRM | 0.175 | **0.147** | 0.127 | **0.199** | **0.181** | 0.095 | 0.259 | **0.034** | **0.243** | **0.166** | **0.085** | 0.265 | 0.226 | 0.244 | **0.074** | 0.177 |
| IVGAE | **0.172** | **0.147** | **0.126** | **0.199** | 0.182 | **0.088** | **0.255** | 0.036 | **0.243** | 0.168 | 0.087 | **0.261** | **0.222** | **0.242** | **0.074** | **0.172** |

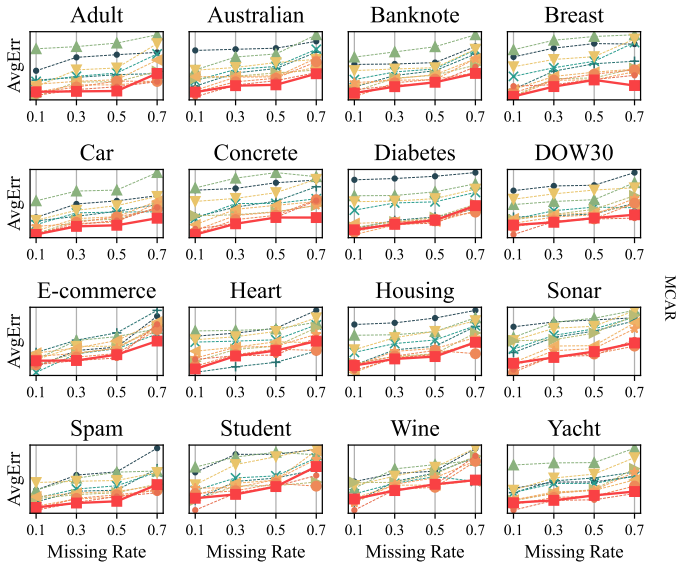



Fig. 4: Comparison of AvgErr under the MCAR mechanism across different missing rates. Lower values indicate better imputation performance.

## 5 Results and Analysis

### 5.1 Qualitative Analysis

Figures 3, 4, 5, and 6 compare imputation performance under MCAR, MAR, and MNAR across varying missing rates. As expected, reconstruction accuracy decreases as missingness increases. Nevertheless, `IVGAE` consistently achieves lower `AvgErr` and shows smoother degradation trends, indicating stronger robustness to data sparsity. The Critical Difference (CD) diagram in Fig. 3 further confirms this observation. `IVGAE` achieves the best overall average rank across all datasets and missingness mechanisms, with statistically significant improvements over most baselines at $\alpha = 0.05$.

Table 2: MSE on the Yahoo! R3 dataset (lower is better).

| Type | Model | MSE |
|---|---|---|
| Stat. | Mean | 2.571 ± 0.001 |
| | KNN | 2.124 ± 0.002 |
| | MICE | 2.028 ± 0.004 |
| | MisForest | 1.987 ± 0.004 |
| Gen. | GAIN | 1.157 ± 0.007 |
| | MIWAE | 2.055 ± 0.001 |
| | NMWAE | 0.939 ± 0.007 |
| | HIVAE | 0.997 ± 0.004 |
| Graph | GRAPE | 2.001 ± 0.007 |
| | IGRM | 0.957 ± 0.005 |
| Ours | IVGAE | **0.937 ± 0.007** |

Table 1 reports numerical results at a representative 30% missing rate. Under this setting, `IVGAE` outperforms most baselines on heterogeneous datasets such as `Adult`, `Heart`, and `Student`. Minor fluctuations on smaller datasets (e.g., `Wine` and `Yacht`) likely arise from limited sample sizes and weaker relational structures.

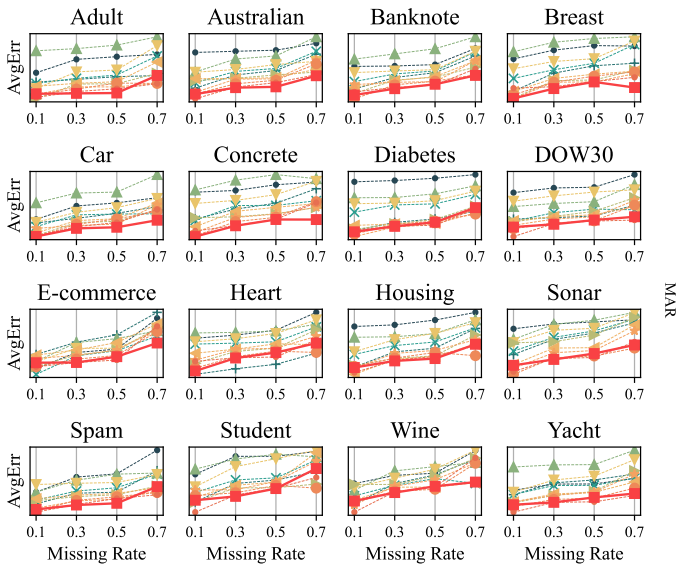



Fig. 5: Comparison of AvgErr under the MAR mechanism across different missing rates. Lower values indicate better imputation performance.

### 5.2 Downstream Task Evaluation

**Supervised Learning Classification.** To evaluate the utility of imputed data for downstream prediction, we measure classification performance using the F1-score at a representative missing rate of 30% (Table 3). This setting reflects a moderate level of missingness commonly used in imputation benchmarks. Overall, `IVGAE` achieves superior or competitive performance across datasets, outperforming both conventional and generative baselines in most cases. In particular, it improves the F1-score by up to **5%** compared with graph-based methods such as `GRAPE` and `IGRM`. These results suggest that mechanism-aware graph modeling helps preserve relational information that is beneficial for downstream predictive tasks. Slight underperformance on the `Banknote` dataset (MAR) can be attributed to its low feature dimensionality and weak relational structure, where graph-based modeling provides limited advantage.

**Yahoo! R3 Recommendation Task.** We further evaluate imputation performance on the Yahoo! R3 dataset [13], which contains *MNAR* training ratings and *MCAR* test ratings. As shown in Table 2, deep generative models outperform traditional baselines, reflecting their ability to capture non-linear user–item interactions. Among all methods, `IVGAE` achieves the lowest MSE, demonstrating its effectiveness in reconstructing unobserved user preferences. This result suggests that mechanism-aware graph modeling can better capture structural dependencies in recommendation data, where missingness is driven by user behavior.

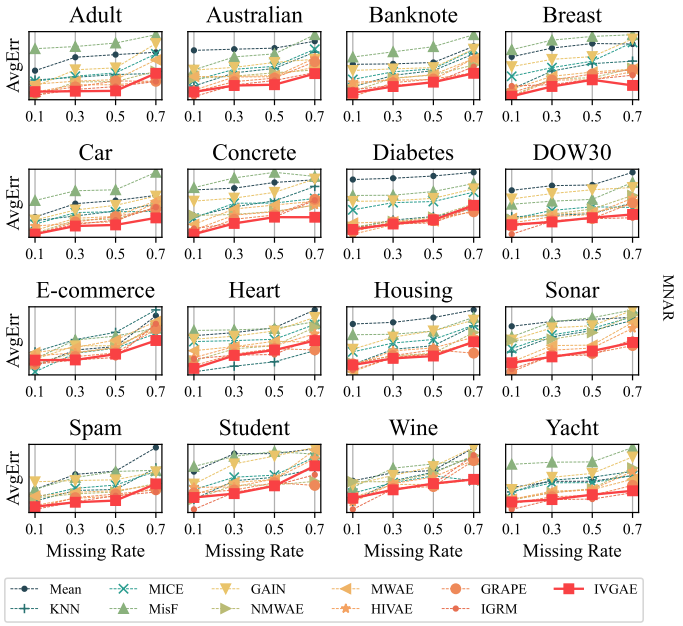



Fig. 6: Comparison of AvgErr under the MNAR mechanism across different missing rates. Lower values indicate better imputation performance.

### 5.3 Scalability

We analyze the runtime scalability of `IVGAE` on synthetic datasets by varying (i) the number of samples ($n$, with $d$=30 and missing rate 30%), (ii) feature dimensionality ($d$, with $n$=1000 and missing rate 30%), and (iii) the missing rate (with $n$=1000 and $d$=30). Results are reported in Fig. 7a, where the runtime axis is plotted on a logarithmic scale. `IVGAE` exhibits near-linear growth as the number of samples increases, while deep generative models such as `GAIN` and `MIWAE` incur significantly higher runtime despite showing similar scaling trends. As feature dimensionality increases, graph-based methods (`GRAPE`, `IGRM`, and `IVGAE`) demonstrate comparable scaling behavior, indicating that message passing dominates the computational cost. Across different missing rates, the runtime of `IVGAE` remains nearly constant, suggesting robustness to increasing sparsity. Several baselines did not converge within the time limit at higher missing ratios and are therefore omitted from the plot.

### 5.4 Ablation Study

**Effect of the Dual-Decoder VGAE.** We evaluate the impact of variational inference and graph structure by comparing `IVGAE` with two reduced variants: (i) a deterministic Graph Autoencoder (GAE) [12], and (ii) a structure-agnostic MLP imputer. As shown in Table 4, replacing the variational module with GAE

Table 3: F1-score comparison under 30% missing rate on classification tasks. Best results are **bolded** and second-best are underlined. Higher is better.

| MCAR | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Heart** | **Sonar** | **Spam** | **Stud.** | **Wine** |
| Mean | 0.245 | 0.613 | 0.762 | 0.428 | 0.438 | 0.177 | 0.691 | 0.698 | 0.199 | 0.796 |
| KNN | 0.241 | 0.652 | 0.772 | 0.412 | 0.404 | 0.181 | 0.721 | 0.725 | 0.198 | 0.872 |
| MICE | 0.218 | 0.612 | 0.772 | 0.451 | 0.428 | 0.277 | 0.741 | 0.693 | 0.213 | 0.856 |
| EM | 0.227 | 0.480 | 0.771 | 0.451 | 0.374 | 0.281 | 0.663 | 0.515 | 0.174 | 0.690 |
| MisF | 0.242 | 0.651 | 0.711 | 0.416 | 0.374 | 0.228 | 0.698 | 0.569 | 0.166 | 0.823 |
| GAIN | 0.242 | 0.655 | 0.761 | 0.413 | 0.369 | 0.281 | 0.605 | 0.835 | 0.178 | 0.846 |
| NMWAE | 0.245 | 0.658 | 0.735 | 0.424 | 0.206 | 0.205 | 0.621 | 0.794 | 0.220 | 0.883 |
| MWAE | 0.244 | 0.517 | **0.810** | 0.451 | 0.352 | 0.282 | **0.762** | 0.650 | 0.229 | 0.861 |
| HIVAE | 0.244 | 0.587 | 0.773 | 0.447 | 0.379 | 0.243 | 0.691 | 0.722 | 0.224 | 0.872 |
| GRAPE | 0.248 | 0.574 | 0.777 | 0.441 | 0.412 | **0.289** | 0.742 | 0.784 | 0.220 | 0.895 |
| IGRM | 0.239 | 0.649 | 0.770 | 0.451 | 0.445 | 0.282 | 0.734 | 0.795 | 0.229 | 0.762 |
| IVGAE | **0.250** | **0.658** | 0.781 | **0.461** | **0.460** | **0.289** | 0.754 | **0.857** | **0.236** | **0.912** |

| MAR | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Heart** | **Sonar** | **Spam** | **Stud.** | **Wine** |
| Mean | 0.237 | 0.512 | 0.852 | 0.478 | 0.518 | 0.185 | 0.751 | 0.576 | 0.233 | 0.778 |
| KNN | 0.235 | 0.505 | 0.847 | 0.475 | 0.452 | 0.195 | 0.795 | 0.595 | 0.235 | 0.822 |
| MICE | 0.235 | 0.524 | 0.846 | 0.471 | 0.510 | 0.216 | 0.774 | 0.635 | 0.237 | 0.852 |
| EM | 0.196 | 0.484 | 0.826 | 0.494 | 0.376 | 0.231 | 0.712 | 0.537 | 0.179 | 0.718 |
| MisF | 0.272 | 0.419 | 0.855 | 0.413 | 0.393 | 0.235 | 0.756 | 0.595 | 0.171 | 0.887 |
| GAIN | 0.279 | 0.491 | 0.811 | 0.413 | 0.416 | 0.281 | 0.689 | 0.866 | 0.184 | 0.832 |
| NMWAE | 0.294 | 0.478 | 0.815 | 0.485 | 0.356 | 0.213 | 0.721 | 0.515 | 0.219 | **0.908** |
| MWAE | 0.223 | 0.443 | 0.884 | 0.523 | 0.466 | 0.242 | 0.804 | 0.715 | **0.265** | 0.885 |
| HIVAE | 0.245 | 0.445 | 0.886 | 0.518 | 0.483 | 0.245 | 0.764 | 0.635 | 0.229 | 0.771 |
| GRAPE | 0.297 | 0.494 | 0.852 | 0.481 | 0.536 | 0.247 | 0.842 | 0.715 | 0.248 | 0.812 |
| IGRM | 0.280 | **0.535** | **0.887** | 0.501 | **0.534** | 0.276 | 0.743 | **0.871** | 0.255 | 0.872 |
| IVGAE | **0.303** | **0.535** | 0.852 | **0.536** | **0.534** | **0.297** | **0.857** | 0.827 | **0.256** | 0.902 |

| MNAR | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Adult** | **Aust.** | **Bank.** | **Breast** | **Car** | **Heart** | **Sonar** | **Spam** | **Stud.** | **Wine** |
| Mean | 0.213 | 0.666 | 0.685 | 0.413 | 0.503 | 0.155 | 0.540 | 0.771 | 0.207 | 0.724 |
| KNN | 0.221 | 0.624 | 0.684 | 0.453 | 0.453 | 0.175 | 0.580 | 0.726 | 0.188 | 0.754 |
| MICE | 0.232 | 0.670 | 0.703 | 0.441 | 0.495 | 0.210 | 0.581 | 0.730 | 0.176 | 0.727 |
| EM | 0.218 | 0.554 | 0.751 | 0.475 | 0.387 | 0.155 | 0.599 | 0.499 | 0.174 | 0.531 |
| MisF | 0.269 | 0.720 | 0.722 | 0.464 | 0.390 | 0.217 | 0.622 | 0.791 | 0.179 | 0.866 |
| GAIN | 0.277 | 0.768 | 0.771 | 0.485 | 0.312 | 0.218 | 0.641 | 0.792 | 0.181 | 0.854 |
| NMWAE | 0.201 | 0.615 | 0.686 | 0.476 | 0.215 | 0.215 | 0.745 | 0.798 | 0.219 | 0.875 |
| MWAE | 0.235 | 0.636 | 0.846 | **0.542** | 0.484 | 0.214 | 0.767 | 0.759 | 0.205 | 0.873 |
| HIVAE | 0.202 | 0.613 | 0.775 | 0.495 | 0.331 | 0.235 | 0.726 | 0.812 | 0.276 | 0.891 |
| GRAPE | 0.294 | 0.608 | 0.805 | 0.466 | 0.448 | 0.235 | 0.778 | 0.799 | 0.265 | 0.920 |
| IGRM | 0.275 | 0.786 | 0.815 | 0.459 | 0.503 | 0.259 | 0.576 | 0.583 | 0.263 | 0.879 |
| IVGAE | **0.335** | **0.802** | **0.851** | 0.511 | **0.505** | **0.297** | **0.782** | **0.819** | **0.292** | **0.936** |

increases AvgErr by up to 0.03 and reduces the F1-score by 2–3% across all missingness mechanisms. The MLP baseline performs the worst, indicating that

Table 5: Complexity comparison between encoding strategies. $N_{\text{cat}}$ and $\bar{C}$ denote the number and average cardinality of categorical features, $M$ the number of samples, and $N$ the total number of features.

| **Method** | **#Feature Nodes** | **Edge Count** |
|---|---|---|
| One-Hot Encoding | $\mathcal{O}(N_{\text{cat}} \cdot \bar{C})$ | $\mathcal{O}(M \cdot N_{\text{cat}} \cdot \bar{C})$ |
| `Hetero` | $\mathcal{O}(N)$ | $\mathcal{O}(M \cdot N)$ |

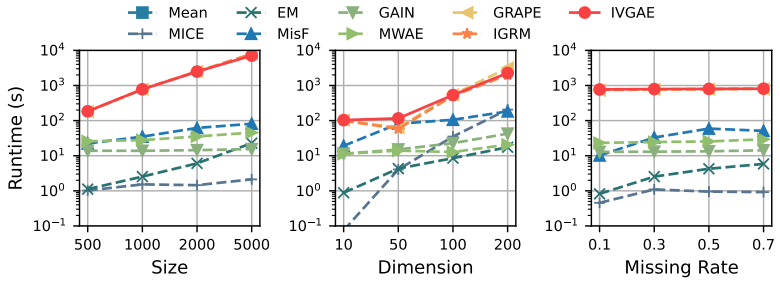
relational graph modeling is crucial for imputing heterogeneous tabular data. These results highlight the importance of combining graph structure learning with variational uncertainty modeling.

Table 4: Ablation study on key components of `IVGAE` under a 30% missing rate. We report F1-score and AvgErr under MCAR, MAR, and MNAR mechanisms.

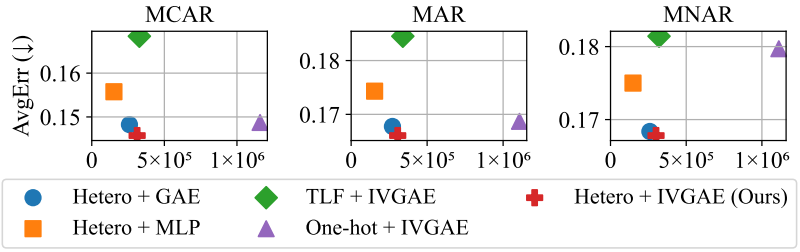
| Method | MCAR | | MAR | | MNAR | |
|---|---|---|---|---|---|---|
| | F1 (↑) | AvgErr (↓) | F1 (↑) | AvgErr (↓) | F1 (↑) | AvgErr (↓) |
| `IVGAE` | **0.5658** | **0.146** | **0.5899** | **0.166** | **0.613** | **0.168** |
| GAE | 0.5414 | 0.148 | 0.5679 | 0.168 | 0.571 | **0.168** |
| MLP | 0.5028 | 0.156 | 0.5117 | 0.174 | 0.524 | 0.175 |
| `Hetero` | **0.5658** | **0.146** | **0.5899** | **0.166** | **0.613** | **0.168** |
| TLF | 0.5214 | 0.168 | 0.5701 | 0.185 | 0.601 | 0.181 |
| One-hot | 0.5514 | 0.149 | 0.5624 | 0.169 | 0.597 | 0.180 |

**Effect of Heterogeneous Data Embedding.** We compare the proposed heterogeneous embedding (`Hetero`) with Tree-Driven Latent Factor encoding (TLF) [2] and traditional one-hot encoding. One-hot encoding expands each categorical feature into multiple binary dimensions, significantly increasing graph sparsity. For a dataset with $M$ samples and $N_{\text{cat}}$ categorical features with average cardinality $\bar{C}$, the number of edges in the bipartite graph scales as $|\mathcal{E}_{\text{one-hot}}| = M(N_{\text{num}} + N_{\text{cat}}\bar{C})$, resulting in graph complexity $\mathcal{O}(MN_{\text{cat}}\bar{C})$. In contrast, `Hetero` maps each categorical column into a compact latent representation, preserving semantic relationships while maintaining a dense graph structure. As shown in Table 4, `Hetero` consistently achieves lower AvgErr and higher F1, improving performance by up to 10% over TLF and 4% over one-hot encoding. It also reduces inference time by approximately 20–30% due to the smaller graph representation.

**Runtime Analysis.** Figure 7b illustrates the trade-off between runtime and imputation accuracy at a 30% missing rate. Points closer to the lower-left corner indicate a better efficiency–accuracy trade-off (lower runtime and lower error). `Hetero+IVGAE` achieves the best balance, delivering the lowest imputation error with competitive runtime. In contrast, `One-hot` and `TLF` incur higher computa-




(a) Runtime scalability under varying sample sizes, feature dimensions, and missing rates (log-scale).




(b) Accuracy–runtime trade-off under 30% missing rate across datasets.

Fig. 7: Runtime efficiency analysis of `IVGAE`.

tional cost due to expanded graph representations, while `GAE` and `MLP` are faster but substantially less accurate.

## 6 Conclusion

This paper presented `IVGAE`, a variational graph framework for imputing incomplete heterogeneous tabular data. Using a bipartite graph representation and dual-decoder architecture, `IVGAE` jointly models feature reconstruction and missingness patterns, while its Transformer-based embedding handles mixed data types without high-dimensional one-hot encoding. Experiments on 16 real-world datasets show consistent gains in imputation accuracy and downstream predictive performance over generative and graph-based baselines. These results highlight the benefit of mechanism-aware variational graph modeling. Future work will address the current reliance on a fixed graph structure through adaptive graph construction and dynamic structure learning.

## Acknowledgement

This work is supported by the Air Force Office of Scientific Research under award number FA2386-23-1-4003 and Deakin University.

## AI Usage Disclosure

Generative AI tools were used for minor language editing and proofreading. All ideas, methods, experiments, and conclusions were developed by the authors, who take full responsibility for the content of this paper.

## References


1. Al-taezi, M.A., Wang, Y., Zhu, P., Hu, Q., Al-Badwi, A.: Improved generative adversarial network with deep metric learning for missing data imputation. Neurocomputing **570**, 127062 (2024)
2. Borisov, V., Broelemann, K., Kasneci, E., Kasneci, G.: Deeptlf: robust deep neural networks for heterogeneous tabular data. International Journal of Data Science and Analytics **16**(1), 85–100 (2023)
3. Chai, Z., Zhao, C., Huang, B., Chen, H.: A deep probabilistic transfer learning framework for soft sensor modeling with missing data. IEEE Transactions on Neural Networks and Learning Systems **33**(12), 7598–7609 (2022). https://doi.org/10.1109/TNNLS.2021.3085869
4. Dutta, S., Sengupta, P.: Men and mice: relating their ages. Life sciences **152**, 244–248 (2016)
5. Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., Tabona, O.: A survey on missing data in machine learning. Journal of Big data **8**, 1–37 (2021)
6. Guo, X., Quan, Y., Zhao, H., Yao, Q., Li, Y., Tu, W.: Tabgnn: Multiplex graph neural network for tabular data prediction. arXiv preprint arXiv:2108.09127 (2021)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
8. Houari, R., Bounceur, A., Tari, A.K., Kecha, M.T.: Handling missing data problems with sampling methods. In: 2014 International conference on advanced networking distributed systems and applications. pp. 99–104. IEEE (2014)
9. Ipsen, N.B., Mattei, P.A., Frellsen, J.: not-{miwae}: Deep generative modelling with missing not at random data. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=tu29GQT0JFy`
10. Kim, J.K., Shao, J.: Statistical methods for handling incomplete data. Chapman and Hall/CRC (2021)
11. Kingma, D.P., Welling, M., et al.: Auto-encoding variational bayes (2013)
12. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
13. Marlin, B., Zemel, R.S., Roweis, S., Slaney, M.: Collaborative filtering and the missing at random assumption. arXiv preprint arXiv:1206.5267 (2012)
14. Marlin, B.M., Zemel, R.S.: Collaborative prediction and ranking with non-random missing data. In: Proceedings of the third ACM conference on Recommender systems. pp. 5–12 (2009)
15. Mattei, P.A., Frellsen, J.: Miwae: Deep generative modelling and imputation of incomplete data sets. In: International conference on machine learning. pp. 4413–4423. PMLR (2019)
16. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. Advances in neural information processing systems **20** (2007)
17. Muzellec, B., Josse, J., Boyer, C., Cuturi, M.: Missing data imputation using optimal transport. In: International Conference on Machine Learning. pp. 7130–7140. PMLR (2020)

18. Nazabal, A., Olmos, P.M., Ghahramani, Z., Valera, I.: Handling incomplete heterogeneous data using vaes. Pattern Recognition **107**, 107501 (2020)
19. Rubin, D.B.: Inference and missing data. Biometrika **63**(3), 581–592 (1976)
20. Simonovsky, M., Komodakis, N.: Graphvae: Towards generation of small graphs using variational autoencoders. In: Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27. pp. 412–422. Springer (2018)
21. Song, Q., Shepperd, M.: Missing data imputation techniques. International journal of business intelligence and data mining **2**(3), 261–291 (2007)
22. Spinelli, I., Scardapane, S., Uncini, A.: Missing data imputation with adversarially-trained graph convolutional networks. Neural Networks **129**, 249–260 (2020)
23. Sportisse, A., Boyer, C., Josse, J.: Imputation and low-rank estimation with missing not at random data. Statistics and Computing **30**(6), 1629–1643 (2020)
24. Stekhoven, D.J., Bühlmann, P.: Missforest—non-parametric missing value imputation for mixed-type data. Bioinformatics **28**(1), 112–118 (Oct 2011). https://doi.org/10.1093/bioinformatics/btr597, `http://dx.doi.org/10.1093/bioinformatics/btr597`
25. Sun, Y., Li, J., Xu, Y., Zhang, T., Wang, X.: Deep learning versus conventional methods for missing data imputation: A review and comparative study. Expert Systems with Applications **227**, 120201 (2023)
26. Van Den Berg, R., Welling, M.: Graph convolutional matrix completion
27. Wang, H., Chen, Z., Shen, Y., Zheng, H., Yang, D., Zhao, D., Liang, B.: Robust missing value imputation with proximal optimal transport for low-quality iiot data. IEEE Transactions on Neural Networks and Learning Systems **37**(1), 82–94 (2026). https://doi.org/10.1109/TNNLS.2025.3601130
28. Wang, X., Zhang, R., Sun, Y., Qi, J.: Doubly robust joint learning for recommendation on data missing not at random. In: International Conference on Machine Learning. pp. 6638–6647. PMLR (2019)
29. Xie, Z., Xu, H., Chen, W., Li, W., Jiang, H., Su, L., Wang, H., Pei, D.: Unsupervised anomaly detection on microservice traces through graph vae. In: Proceedings of the ACM Web Conference 2023. pp. 2874–2884 (2023)
30. Yoon, J., Jordon, J., Schaar, M.: Gain: Missing data imputation using generative adversarial nets. In: International conference on machine learning. pp. 5689–5698. PMLR (2018)
31. You, J., Ma, X., Ding, Y., Kochenderfer, M.J., Leskovec, J.: Handling missing data with graph representation learning. Advances in Neural Information Processing Systems **33**, 19075–19087 (2020)
32. Zhang, M., Chen, Y.: Inductive matrix completion based on graph neural networks. In: International Conference on Learning Representations (2019)
33. Zhang, S.: Nearest neighbor selection for iteratively knn imputation. Journal of Systems and Software **85**(11), 2541–2552 (2012)
34. Zhong, J., Gui, N., Ye, W.: Data imputation with iterative graph reconstruction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 11399–11407 (2023)
35. Zhou, Y.: missmecha-py: A python package for simulating and analyzing missing data mechanisms. `https://pypi.org/project/missmecha-py/` (2025), version 1.0.0