BEYOND CORRELATION AND SHARED SPACES: CONDITIONAL AND ADAPTIVE GRAPH IMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A column-centric graph provides a natural representation for tabular imputation because its topology can be shared across rows, allowing the same structural information to support both in-sample and out-of-sample imputation. However, such graph-based formulations face two fundamental challenges: pairwise associations may not reflect direct conditional dependencies, and shared message transformations may overlook the heterogeneous representation spaces of different columns. This paper proposes CAGI (Conditional and Adaptive Graph Imputation) to address these challenges. CAGI constructs a conditional dependency graph by estimating latent correlations with a rank-based estimator and deriving a sparse graph from the resulting partial correlations, and performs column-adaptive graph diffusion through layer-wise source and target transformations. A theoretical analysis establishes when conditional neighborhoods provide sufficient information for reconstruction and when graph diffusion can preserve the information required by the Bayes-optimal imputation function. An excess-risk decomposition further characterizes the effects of information loss in diffusion and approximation error. Experiments on tabular imputation tasks demonstrate the effectiveness of CAGI.

1 INTRODUCTION

Missing values are ubiquitous in tabular data and can substantially affect downstream analysis and prediction (Fan, 2025). Since a missing entry is often related to other variables within the same sample, exploiting dependencies among columns provides a natural basis for imputation (van Buuren & Groothuis-Oudshoorn, 2011; You et al., 2020). A column-centric graph is appealing because its structure can be shared across samples, supporting both in-sample and out-of-sample imputation. However, effectively exploiting such a graph requires addressing two challenges: identifying meaningful dependencies among columns and propagating information across heterogeneous variables.

Existing graph-based imputation methods commonly construct edges from marginal associations, such as pairwise correlations (Zhang et al., 2025c). However, marginal associations can mix direct dependencies with associations mediated by other variables, potentially introducing redundant or misleading connections into the imputation graph (Fan et al., 2017). Figure 1 provides an intuitive example. Meanwhile, propagation-based methods, including Graph Neural Networks and Transformers, typically rely on shared transformations (You et al., 2020; Somepalli et al., 2021), implicitly assuming that representations of different columns can be processed within a common space. This assumption is restrictive for tabular data, where columns may have substantially different semantics and statistical characteristics (Jiang et al., 2026), making a universal transformation inadequate for aligning information across heterogeneous variables.

To address these challenges, we propose CAGI (Conditional and Adaptive Graph Imputation), a graph-based imputation framework that separates the roles of graph structure and information transformation. For graph construction, CAGI estimates the latent dependency structure among columns (Fan et al., 2017) and derives a sparse graph from their conditional dependencies. Unlike marginal associations, these dependencies characterize the relationship between two columns after accounting for the effects of the remaining variables, allowing the graph to distinguish direct dependencies from associations mediated by other columns. This provides a structural representation that is more closely aligned with the information dependencies relevant to conditional reconstruction. For information propagation, CAGI replaces a universal shared transformation with column-specific

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

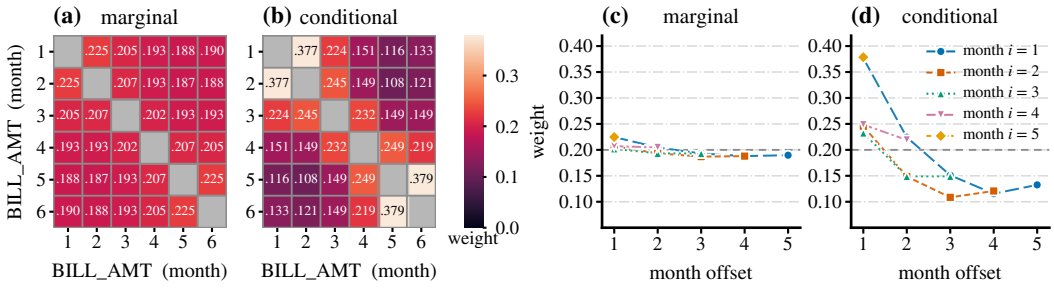


Figure 1: Comparison between marginal correlation and conditional dependence among the six monthly bill-amount columns (`BILL_AMT1` - `BILL_AMT6`) in the `default` dataset. (a) Pairwise marginal correlations and (b) pairwise conditional dependencies shown as matrices. (c) Marginal correlation and (d) conditional dependence plotted against the number of months separating two columns. Marginal correlations remain relatively similar across month pairs, whereas conditional dependencies exhibit a clear decay with increasing temporal distance, revealing stronger direct dependencies between neighboring months and weaker dependencies between distant months.

source and target transformations (Ribeiro et al., 2026). Specifically, each column is associated with transformations that determine how its representation is projected when information is sent from or received by that column, enabling the propagation process to account for the heterogeneous representation spaces of different variables. These transformations are adaptive across graph layers, allowing the representation alignment to evolve with the propagated information. The transformation mechanism is agnostic to the specific graph propagation operator and can be combined with general message-passing architectures. In CAGI, these two components play complementary roles: the conditional dependency graph determines *which information should flow*, while the column-specific transformations determine *how that information should be aligned and communicated.*

We further provide a theoretical analysis that connects these two design choices to the imputation objective. Our results characterize when conditional neighborhoods preserve the information required for reconstruction and when column-adaptive propagation can retain sufficient information for Bayes-optimal imputation, providing a principled perspective on the role of graph structure and information alignment in tabular imputation. The main contributions are summarized as follows:

- **Conditional dependency modeling**: We introduce a graph construction strategy based on conditional dependencies, providing a more informative structural representation.
- **Adaptive information propagation across heterogeneous columns**: We enable column-aware transformation during propagation, relaxing the common-space assumption.
- **Theoretical analysis:** We establish theoretical results on the conditional neighborhoods and adaptive propagation, and derive conditions to approximate Bayes-optimal imputation.
- **Empirical evaluation**: We conduct extensive experiments on tabular imputation tasks to demonstrate the effectiveness of CAGI.

2 Notations and Problem Formulation

Consider a tabular dataset with N samples and D columns, represented by $X \in \mathbb{R}^{N\times D}$. Each column corresponds to a variable, and the n-th row X_n represents one sample. Let $\mathcal{O}_n \subseteq \{1,\dots,D\}$ denote the set of observed columns for sample n, and let $\mathcal{M}_n = \{1,\dots,D\} \setminus \mathcal{O}_n$ denote the corresponding missing columns. The goal of tabular imputation is to recover the missing entries x_{nj}, $j \in \mathcal{M}_n$, from the observed entries $X_{n,\mathcal{O}_n}$. In a probabilistic formulation, the optimal imputation under a given reconstruction loss is determined by the conditional distribution $p(X_{n,\mathcal{M}_n} \mid X_{n,\mathcal{O}_n})$. In particular, under squared error loss, the Bayes-optimal imputation is given by the conditional expectation $X^*_{n,\mathcal{M}_n} = \mathbb{E}[X_{n,\mathcal{M}_n} \mid X_{n,\mathcal{O}_n}]$. For graph-based imputation, the columns are represented as nodes of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}), |\mathcal{V}| = D$, where node $v_j \in \mathcal{V}$ corresponds to the j-th column. The graph provides a structural space over which information among columns can be propagated. Since the columns are shared across samples, the same graph structure can be instantiated for different rows, while the node features vary according to the corresponding sample.

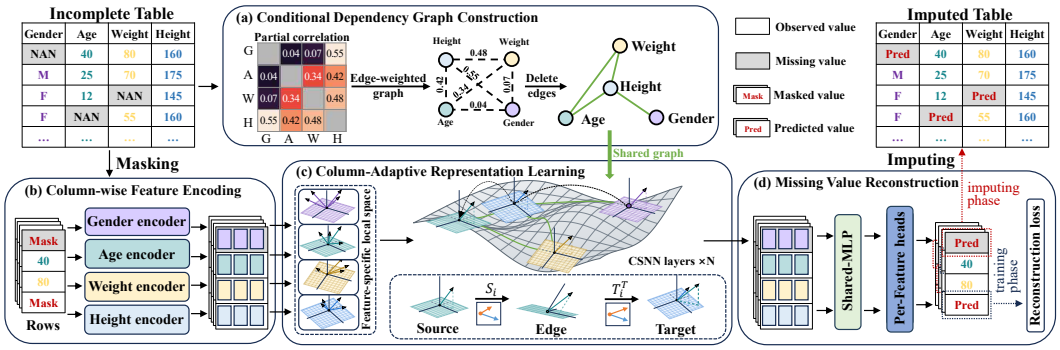


Figure 2: Overview of CAGI. The framework consists of 4 stages: **(a)** conditional dependency graph construction, which determines the shared graph structure; **(b)** column-wise feature encoding, which maps numerical and categorical values into column-specific representation spaces; **(c)** column-adaptive representation learning, which alternates graph propagation and column-specific source/target transformations to align and propagate information across heterogeneous columns; and **(d)** missing value reconstruction, which decodes the learned representations through a shared decoder and column-specific ones. Different colors denote different columns, i.e., `gender`, `age`, `weight`, and `height`, and are consistently used for values, encoders, and representation spaces.

3 CAGI: Conditional and Adaptive Graph Imputation

Motivation: A key question in graph-based tabular imputation is how to determine which columns should be connected. Marginal associations, such as pairwise correlations, can mix direct dependencies with associations mediated by other variables. A representative example is provided by the `default` dataset, where `BILL_AMT1` - `BILL_AMT6` record the bill amounts from the first to the sixth month. Intuitively, consecutive monthly bills should be more strongly related than distant months, suggesting a near-Markov dependency structure. Figure 1 compares the two types of dependencies: (a) and (b) show the dependency matrices based on marginal correlation and conditional dependence, while (c) and (d) plot their weights against the number of months separating each pair.

The contrast is clear. Marginal correlations remain relatively similar across the 15 possible edges, whereas conditional dependencies exhibit a clear decay with increasing temporal distance, with neighboring months showing stronger dependencies than distant ones. This illustrates that marginal correlation can capture mediated associations, while conditional dependence better reveals direct relationships. Motivated by this observation, CAGI constructs the graph from conditional dependencies to better align graph connectivity with the information structure underlying reconstruction.

Overview: Conditional and Adaptive Graph Imputation (CAGI) follows the column-centric graph formulation for tabular imputation, in which the columns of a table are represented as nodes of a shared graph and the relationships between columns define its edges. Each row then instantiates this shared graph with row-specific node features, where the feature of each node represents the value of the corresponding column in that row. During training, following the masked reconstruction strategy of ReMasker, we randomly mask a subset of observed cells in addition to the originally missing entries and train the model to reconstruct the masked values. This allows the model to learn imputation from observed entries while using the same procedure for training and inference.

As shown in Figure 2, CAGI consists of four main components. **(a) Conditional Dependency Graph Construction** builds a shared graph over columns based on their conditional dependencies, providing the structural information for subsequent information propagation (Section 3.1). **(b) Column-wise Feature Encoding** maps the values of different columns into their respective representation spaces. Numerical values and categorical values are encoded using column-specific encoders following the general design of FT-Transformer, while preserving the distinction between heterogeneous columns. **(c) Column-Adaptive Representation Learning** propagates information over the shared graph while applying column-specific source and target transformations (Section 3.2). The propagation and alignment operations are alternated to progressively obtain representations for the target cells, including missing values. **(d) Missing Value Reconstruction** decodes the learned representations using a shared decoder followed by column-specific decoders to produce the reconstructed values (Section H).

3.1 CONDITIONAL DEPENDENCY GRAPH CONSTRUCTION

Rank-Based Latent Correlation Estimation: Let $Z = (Z_1, \ldots, Z_d)^\top$ denote latent continuous variables associated with the d observed columns. We assume that the latent variables admit a Gaussian copula representation (Fan et al., 2017) with correlation matrix $\Sigma = (\Sigma_{jk})_{d\times d}$. The observed variables need not themselves be Gaussian, since the latent formulation only specifies the dependence structure after suitable monotone transformations. This property is particularly useful for heterogeneous tabular data containing both continuous and discrete variables. Since Z is unobserved, its correlation matrix cannot be estimated by directly computing the sample covariance of the latent variables. Instead, we estimate $\widehat{R} = (\widehat{R}_{jk})_{d\times d}$ from the observed samples, using an estimator matched to the pair of column types involved. For a numeric column, we first apply the monotone marginal transformation $\widehat{Z}_{rj} = \Phi^{-1}\left(\frac{\widehat{\mathrm{rank}}(X_{rj})-1/2}{n_j}\right), \quad r \in \mathcal{J}_j$, where $\mathcal{J}_j$ is the set of rows in which column j is observed, $n_j = |\mathcal{J}_j|$, and $\widehat{\mathrm{rank}}(X_{rj})$ is the rank of X_{rj} among the observed values of column j. The transformation maps each column onto the standard normal scale and is monotone within the column. For a numeric-numeric pair, $\widehat{R}_{jk}$ is the Pearson correlation of the transformed values over the jointly observed rows,

$$\widehat{R}_{jk} = \mathrm{corr}\big\{\widehat{Z}_{rj}, \widehat{Z}_{rk} : r \in \mathcal{J}_j \cap \mathcal{J}_k\big\}, \tag{1}$$

which is the nonparanormal estimator of the latent correlation (Liu et al., 2009) and requires no tuning parameter.

For a discrete column we treat the categories as ordered and fix the latent thresholds at the empirical marginal frequencies. Let K_j be the number of categories of column j, let $c_{j1} < \cdots < c_{jK_j}$ be its ordered categories, and let $\widehat{F}_j$ be its empirical distribution function. We set $\widehat{\Gamma}_{j\ell} = \Phi^{-1}\big(\widehat{F}_j(c_{j\ell})\big)$ for $\ell = 1, \ldots, K_j - 1$, with $\widehat{\Gamma}_{j0} = -\infty$ and $\widehat{\Gamma}_{jK_j} = +\infty$. With the thresholds fixed in this way, each entry of $\widehat{R}$ is obtained by maximizing a one-dimensional likelihood in the correlation. For a discrete-numeric pair, with ℓ_{jr} the category of column j in row r,

$$\widehat{R}_{jk} = \arg\max_{\rho\in(-1,1)} \sum_{r\in\mathcal{J}_j\cap\mathcal{J}_k} \log\left[\Phi\left(\frac{\widehat{\Gamma}_{j,\ell_{jr}} - \rho\,\widehat{Z}_{rk}}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{\widehat{\Gamma}_{j,\ell_{jr}-1} - \rho\,\widehat{Z}_{rk}}{\sqrt{1-\rho^2}}\right)\right], \tag{2}$$

which is the two-step polyserial estimator (Olsson et al., 1982). For a discrete-discrete pair, the same construction applied to the bivariate normal probabilities of the observed cell counts gives the two-step polychoric estimator (Olsson, 1979). A pair with fewer than ten jointly observed rows is left at $\widehat{R}_{jk} = 0$, and $\widehat{R}_{jj} = 1$ throughout. The resulting matrix $\widehat{R}$ provides an estimate of the latent correlation structure while avoiding explicit estimation of the unknown marginal transformations, and it accommodates heterogeneous column types without requiring a common marginal distribution.

Sparse Precision Estimation: Although $\widehat{R}$ captures latent pairwise dependence, directly using it as the adjacency matrix would still represent marginal rather than conditional relationships. We therefore derive the graph from the partial correlations implied by an estimated precision matrix $\widehat{\Omega} \approx \Sigma^{-1}$. Because the estimated latent correlation matrix is not necessarily positive semidefinite in finite samples, we symmetrize it, truncate its spectrum, and rescale the result to unit diagonal,

$$\widehat{R}_{\mathrm{psd}} = \mathrm{diag}(\Xi)^{-1/2}\,\Xi\,\mathrm{diag}(\Xi)^{-1/2}, \quad \Xi = U\max(\Lambda, 10^{-8})U^\top, \tag{3}$$

where $U\Lambda U^\top$ is the eigen-decomposition of the symmetrized $\widehat{R}$; the truncation removes the negative eigenvalues and the rescaling restores unit diagonal. We then invert the projected matrix directly,

$$\widehat{\Omega} = \big(\widehat{R}_{\mathrm{psd}} + 10^{-8}I\big)^{-1}, \tag{4}$$

where the ridge term $10^{-8}I$ keeps the inverse well defined when $\widehat{R}_{\mathrm{psd}}$ is close to singular. Conditional dependence is read off from the partial correlation $C_{ij} = -\widehat{\Omega}_{ij}/\sqrt{\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}}$, which is scale-free and lies in $[-1, 1]$. Because the three blocks of the matrix are on different scales, edges are selected within each block separately:

$$(i,j) \in E \quad \Longleftrightarrow \quad |C_{ij}| > \tau_{b(i,j)}, \tag{5}$$

where $b(i,j)$ is the block containing the pair (i,j) and τ_b is the empirical quantile of the off-diagonal entries $\{|C_{ij}|\}$ of block b at a level that we set to 0.5 by default, which retains the upper half of each block, and pairs tied with the quantile are excluded. A column left isolated by the threshold is connected to the column with which it has the strongest partial correlation. The components left by the threshold are then joined by the edges of a maximum spanning forest over $\{|C_{ij}|\}$, so that the graph remains connected at any threshold level. The retained partial correlation, relative to its average over the retained edges, is used as the edge strength, $a_{ij} = |C_{ij}| / \operatorname{mean}_E |C|$. The resulting graph is intended to encode direct conditional dependencies rather than all marginal associations. Algorithm 1 in Appendix F shows this process.

3.2 Column-Adaptive Graph Diffusion

Column-Specific Source and Target Transformations: Let $H_r^{(\ell)} = \left[H_{r,1}^{(\ell)}, \ldots, H_{r,d}^{(\ell)}\right]^\top$ denote the hidden representation of row r at layer ℓ, where each node feature is represented in a q-dimensional column-specific coordinate space. Instead of applying one shared transformation to all messages, we assign each column i two learnable linear transformations: $S_i^{(\ell)}, T_i^{(\ell)} \in \mathbb{R}^{q\times q}$. The source transformation $S_i^{(\ell)}$ determines how information is projected when it leaves column i, whereas the target transformation $T_i^{(\ell)}$ determines how information is aligned when it arrives at column i (Ribeiro et al., 2026). Consequently, a message sent from column j to column i has the form

$$m_{j\to i}^{(\ell)} = a_{ij}\big(T_i^{(\ell)}\big)^\top S_j^{(\ell)} H_{r,j}^{(\ell)}. \tag{6}$$

The induced source-target transformation is therefore $A_{ij}^{(\ell)} = \big(T_i^{(\ell)}\big)^\top S_j^{(\ell)}$. We further parameterize the transformations as conformal maps,

$$S_i^{(\ell)} = c_i^{S,(\ell)} Q_i^{S,(\ell)}, \quad T_i^{(\ell)} = c_i^{T,(\ell)} Q_i^{T,(\ell)}, \tag{7}$$

where $c_i^{S,(\ell)}, c_i^{T,(\ell)} > 0$ are learnable scalar factors and $Q_i^{S,(\ell)}, Q_i^{T,(\ell)}$ are orthogonal transformations. The orthogonal components are implemented using a product of Householder reflections, providing a stable and efficient parameterization of the column-specific transformations (Mhammedi et al., 2017). Hence,

$$\left(T_i^{(\ell)}\right)^\top S_j^{(\ell)} = c_i^{T,(\ell)} c_j^{S,(\ell)} \left(Q_i^{T,(\ell)}\right)^\top Q_j^{S,(\ell)}. \tag{8}$$

The interaction between two columns is decomposed into a magnitude component and a relative coordinate transformation. This enables the model to explicitly align heterogeneous column representations before aggregating their information.

Remark: The proposed source and target transformations are *agnostic* to the specific graph propagation mechanism and can be integrated with general message-passing or diffusion-based graph neural networks. In this work, we instantiate them with a bidirectional graph diffusion operator, where the learned transformations determine how information is aligned between source and target columns during propagation. The resulting propagation procedure is described in Appendix G.

4 Theoretical Analysis

This section gives a theoretical analysis of the two parts of CAGI, with proofs in the Appendix B.

Conditional Dependency Reveals Direct Relationships: We first characterize the relationship between marginal and conditional dependencies.

Proposition 1 (Marginal versus Conditional Dependency)**.** *Consider the Gaussian first-order Markov chain $Z_1 \to Z_2 \to \cdots \to Z_d$, generated by $Z_1 \sim \mathcal{N}(0,1), Z_t = \rho Z_{t-1} + \varepsilon_t, \quad t = 2,\ldots,d$, where $|\rho| < 1$ and $\varepsilon_t \sim \mathcal{N}(0, 1-\rho^2)$ are mutually independent. Then $\operatorname{Corr}(Z_i, Z_j) = \rho^{|i-j|}$, while the precision matrix $\Omega = \Sigma^{-1}$ satisfies $\Omega_{ij} = 0, |i-j| > 1$. Hence, whenever $\rho \neq 0$, non-adjacent variables have non-zero marginal correlations but are conditionally independent given the remaining variables.*

The proposition demonstrates that marginal correlation can encode indirect associations mediated by other variables, whereas the precision matrix characterizes direct conditional dependencies. We

next formalize the effect of estimating the precision matrix. Let $\widehat{\Omega}$ denote the estimated precision matrix and let $\widehat{E}_\tau = \left\{(i,j) : i \neq j,\ |\widehat{\Omega}_{ij}| \geq \tau\right\}$ denote the thresholded edge set.

Lemma 2 (Support Recovery under Uniform Precision Error)**.** *Suppose $\|\widehat{\Omega} - \Omega\|_{\max} \leq \epsilon$, and define $\omega_{\min} = \min_{(i,j)\in E^\star} |\Omega_{ij}|$, where $E^\star$ denotes the support of the off-diagonal entries of Ω. If $\epsilon < \tau < \omega_{\min} - \epsilon$, then $\widehat{E}_\tau = E^\star$.*

This result shows that, provided the precision estimator is sufficiently accurate and the non-zero conditional dependencies are separated from zero, the thresholded precision matrix recovers the conditional dependency structure.

Conditional Neighborhoods for Imputation: We next establish the statistical relevance of the conditional neighborhood for reconstruction. Let $\mathcal{N}_i = \mathcal{N}(i)$ denote the conditional neighborhood of column i introduced in Section 3.1.

Theorem 3 (Conditional Neighborhood Sufficiency)**.** *Suppose the joint distribution of the latent variables is strictly positive and Markov with respect to the conditional dependency graph. Then, for every node i, $p(Z_i \mid Z_{-i}) = p(Z_i \mid Z_{\mathcal{N}_i})$. Consequently, the conditional distribution required for optimal reconstruction of Z_i depends only on its conditional neighborhood.*

The Gaussian case gives an explicit form of this conditional predictor. Therefore, the conditional neighborhood identifies the variables that are directly relevant to conditional reconstruction.

Column-Adaptive Diffusion Strictly Generalizes Shared Propagation: We now analyze the expressive capacity of the source-target transformations introduced in Section 3.2. For a fixed layer, suppress the layer superscript when no ambiguity arises. Note that for any edge $(i,j) \in E$, the pairwise transformation used by CAGI is $A_{ij} = T_i^\top S_j$.

Definition 4 (Shared and Column-Adaptive Propagation Classes)**.** *For a fixed graph G, define the shared propagation class as $\mathcal{F}_{\text{shared}} = \{A_{ij} = W : W \in \mathbb{R}^{q\times q},\ \forall (i,j) \in E\}$, and the column-adaptive propagation class as $\mathcal{F}_{\text{adapt}} = \left\{A_{ij} = T_i^\top S_j : S_j, T_i \in \mathbb{R}^{q\times q},\ \forall (i,j) \in E\right\}$.*

Theorem 5 (Strict Expressiveness of Column-Adaptive Propagation)**.** *For any graph G, $\mathcal{F}_{\text{shared}} \subseteq \mathcal{F}_{\text{adapt}}$. Moreover, if G contains two edges $(j,i_1),(j,i_2), i_1 \neq i_2$, then $\mathcal{F}_{\text{shared}} \subsetneq \mathcal{F}_{\text{adapt}}$. This inclusion holds for every realization of the dynamically generated source and target transformations.*

The theorem establishes that column-adaptive propagation is not merely a reparameterization of shared message passing: it strictly enlarges the set of realizable cross-column transformations while retaining shared propagation as a special case. The geometric interpretation of the conformal parameterization introduced in Section 3.2 can be characterized as follows.

Proposition 6 (Column-Specific Coordinate Alignment)**.** *Under the conformal parameterization $S_j = c_j^S Q_j^S,\quad T_i = c_i^T Q_i^T$, where $c_j^S, c_i^T > 0$ and Q_j^S, Q_i^T are orthogonal, the induced transformation satisfies $A_{ij} = c_i^T c_j^S (Q_i^T)^\top Q_j^S$. Consequently, $R_{ij} = (Q_i^T)^\top Q_j^S$ is orthogonal and $\|A_{ij}x\|_2 = c_i^T c_j^S \|x\|_2$ for every $x \in \mathbb{R}^q$.*

Hence, the adaptive transformation separates cross-column communication into a relative coordinate alignment and a scalar magnitude, allowing different columns to retain distinct representation spaces while communicating through an explicit alignment.

Bayes-Optimal Imputation: We finally connect the conditional graph and adaptive diffusion to the imputation objective. For a target column i, let M denote the masking pattern and let $\mathcal{O}_i(M) = (G, M, \{X_j : M_j = 1,\ j \neq i\})$ denote the information available for reconstructing the target cell. For a fixed row, we write the final node representation introduced in Section 3.2 as $H_i^{(L)}$ by suppressing the row index. For squared reconstruction loss, define $\mathcal{R}_i(f) = \mathbb{E}\big[\big(X_i - f(H_i^{(L)})\big)^2\big]$, and let $\mathcal{R}_i^\star = \inf_f \mathbb{E}\big[\left(X_i - f(\mathcal{O}_i(M))\right)^2\big]$ denote the Bayes risk based on all information available after masking. The corresponding Bayes decision is $\eta_i(\mathcal{O}_i(M)) = \mathbb{E}\left[X_i \mid \mathcal{O}_i(M)\right]$.

Definition 7 (Mask-Aware Predictive Neighborhood)**.** *For target column i and masking pattern M, a set $\mathcal{B}_i(M) \subseteq V \setminus \{i\}$ is called a predictive neighborhood if $X_i \perp \{X_j : M_j = 1,\ j \notin \mathcal{B}_i(M)\} \mid \{X_j : M_j = 1,\ j \in \mathcal{B}_i(M)\}, G, M$. Equivalently, $p(X_i \mid \mathcal{O}_i(M)) = p\left(X_i \mid G, M, \{X_j : M_j = 1,\ j \in \mathcal{B}_i(M)\}\right)$.*

Table 1: Cross-dataset averages under each of the three 30% missingness mechanisms (MCAR, MAR and MNAR). Each cell holds two values, in-sample | out-of-sample, each averaged with equal weight over the 9 datasets. The best value of each side within a mechanism is in bold and the second best underlined. The values are point averages and no dispersion is reported.

Method	MAE ↓						RMSE ↓						R^2 ↑					
	MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Mean	0.651	0.653	0.666	0.674	0.672	0.656	0.983	0.980	1.012	1.003	1.039	0.985	-0.001	-0.001	-0.094	-0.069	-0.072	-0.040
MissForest	0.447	0.449	0.477	0.481	0.457	0.455	0.783	0.785	0.847	0.817	0.843	0.803	0.015	0.042	0.162	0.171	0.099	-0.158
TabCSDI	0.659	0.660	0.653	0.681	0.670	0.666	0.988	0.985	0.993	1.005	1.030	0.988	-0.011	-0.012	-0.040	-0.064	-0.051	-0.050
MissingDataOT	0.529	0.487	0.557	0.541	0.548	0.499	0.894	0.851	0.935	0.910	0.951	0.864	0.162	0.232	0.091	0.145	0.124	0.203
GRAPE	0.467	0.558	0.494	0.585	0.457	0.558	0.781	0.877	0.819	0.886	0.793	0.877	-0.009	-0.169	-0.221	-0.004	0.106	-1.063
HyperImpute	0.380	0.391	0.396	0.447	0.381	0.397	0.733	0.715	0.760	0.803	0.793	0.743	0.359	0.419	0.289	0.111	-2.440	-2.349
ReMasker	0.382	0.392	0.405	0.432	0.381	0.400	0.704	0.709	0.747	0.751	0.728	0.726	0.344	0.323	0.344	0.200	0.401	0.317
DiffPuter	0.430	0.460	0.461	0.513	0.497	0.497	0.727	0.710	0.909	1.116	0.974	0.959	0.417	0.437	-1.107	-0.953	0.334	-0.280
CACTI	0.345	0.345	0.372	0.395	0.350	0.358	0.660	0.658	0.754	0.727	0.749	0.705	0.489	0.496	0.374	0.380	0.496	0.464
Ours	**0.330**	**0.329**	**0.353**	**0.371**	**0.336**	**0.344**	**0.636**	**0.630**	**0.714**	**0.687**	**0.722**	**0.678**	**0.545**	**0.554**	**0.489**	**0.489**	**0.550**	**0.510**

When the relevant conditional neighbors remain observed, the conditional neighborhood $\mathcal{N}_i$ provides a natural choice of $\mathcal{B}_i(M)$. When some conditional neighbors are masked, however, additional observed variables may become informative after marginalizing over the missing variables. The mask-aware definition therefore captures the information actually required by the imputation.

Theorem 8 (Bayes Sufficiency of Graph Diffusion)**.** *Suppose that, almost surely with respect to the masking distribution, the following conditions hold for target column i: **1. Predictive locality:** there exists a predictive neighborhood $\mathcal{B}_i(M)$ such that $p(X_i \mid \mathcal{O}_i(M)) = p\left(X_i \mid G, M, X_{\mathcal{B}_i(M)}\right)$; **2. Information accessibility:** there exists a measurable function r_i such that $X_{\mathcal{B}_i(M)} = r_i(H_i^{(L)}, G, M)$; **3. Decoder realizability:** the reconstruction function class $\mathcal{F}$ contains a function g_i satisfying $\eta_i(\mathcal{O}_i(M)) = g_i(H_i^{(L)}, G, M)$; Then $\inf_{f\in\mathcal{F}} \mathcal{R}_i(f) = \mathcal{R}_i^\star$. If $\mathcal{F}$ is dense in the corresponding L_2 function space, then for every $\varepsilon > 0$ there exists $f \in \mathcal{F}$ such that $\mathcal{R}_i(f) \le \mathcal{R}_i^\star + \varepsilon$.*

The theorem shows that graph diffusion need not preserve the entire observed table. It is sufficient for the representation to retain the information required by the Bayes decision. We next quantify the performance degradation when the diffusion representation loses Bayes-relevant information. Define $\eta_i^O = \mathbb{E}\left[X_i \mid \mathcal{O}_i(M)\right], \quad \eta_i^H = \mathbb{E}\left[X_i \mid H_i^{(L)}, G, M\right]$.

Theorem 9 (Exact Excess-Risk Decomposition)**.** *Suppose that $H_i^{(L)}$ is a measurable function of $\mathcal{O}_i(M)$. Under squared reconstruction loss, for any measurable reconstruction function f_i, $\mathcal{R}_i(f_i) - \mathcal{R}_i^\star = \mathcal{E}_{\text{rep}} + \mathcal{E}_{\text{dec}}$, where $\mathcal{E}_{\text{rep}} = \mathbb{E}\big[\big(\eta_i^O - \eta_i^H\big)^2\big]$, and $\mathcal{E}_{\text{dec}} = \mathbb{E}\big[\big(f_i(H_i^{(L)}) - \eta_i^H\big)^2\big]$. Moreover, $\mathcal{E}_{\text{rep}} = \mathbb{E}\big[\operatorname{Var}\big(\eta_i^O \mid H_i^{(L)}, G, M\big)\big]$.*

The decomposition separates two fundamentally different sources of excess imputation risk: Thus, improving the reconstruction head alone cannot eliminate the first term; the graph diffusion must preserve the information required for the conditional imputation task.

Corollary 10 (Approximate Bayes Optimality)**.** *Under the conditions of the preceding theorem, suppose $\mathcal{E}_{\text{rep}} \le \varepsilon_{\text{rep}}$ and $\inf_{f_i\in\mathcal{F}_i} \mathbb{E}\left[\left(f_i(H_i^{(L)}) - \eta_i^H\right)^2\right] \le \varepsilon_{\text{dec}}$. Then*

$$\inf_{f_i\in\mathcal{F}_i} \mathcal{R}_i(f_i) \le \mathcal{R}_i^\star + \varepsilon_{\text{rep}} + \varepsilon_{\text{dec}}. \tag{9}$$

The corollary gives a quantitative interpretation of CAGI: the conditional dependency graph determines which variables are statistically relevant for reconstruction, whereas the column-adaptive diffusion determines whether the corresponding information can be preserved across heterogeneous representation spaces. When the Bayes-relevant information is preserved and the reconstruction family is sufficiently expressive, the resulting imputation risk can approach the Bayes-optimal risk.

5 Experiments

The datasets, the missingness settings, the baselines and the evaluation protocol are detailed in Appendix D, together with the shared input representation and the evaluation scale.

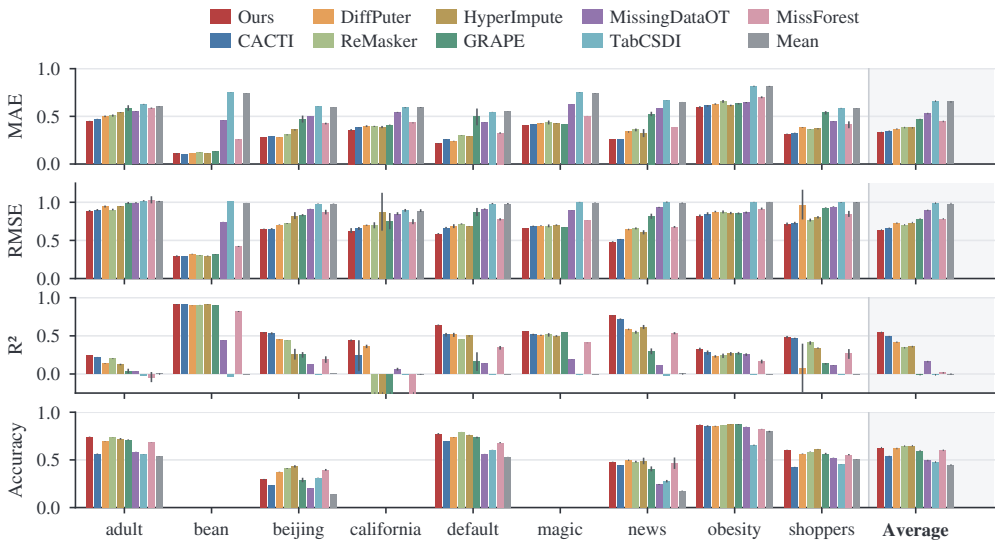


Figure 3: In-sample per-dataset results under MCAR (missing rate 30%). The four panels from top to bottom are MAE, RMSE, R^2 and Accuracy. Lower MAE and RMSE are better; higher R^2 and Accuracy are better. Datasets without discrete features have no accuracy metric.

Main Imputation Results: The main results are summarized in Table 1, which gives the three numerical metrics under each of the three 30% missingness mechanisms, and Figure 5 gives the accuracy on the categorical columns. Within each mechanism and side, Table 1 averages with equal weight over the 9 datasets, and Figure 5 pools in addition over the 10 settings (5 settings $\times$ 2 sides), so the two are not read cell by cell. The comparison reference is the second-best method on the cross-dataset average of each metric. The per-dataset detail behind these averages is given for all 10 settings in Appendix E.

On numerical columns, our method ranks first in all 18 cells of Table 1 (3 metrics $\times$ 3 mechanisms $\times$ 2 sides), and it ranks first in the cross-dataset average of MAE, RMSE and R^2 in all 10 groups (5 settings $\times$ 2 sides), with consistent conclusions in-sample and out-of-sample. Relative to the reference above, MAE decreases by 4.1%–6.0% (in-sample) and 3.9%–6.2% (out-of-sample), RMSE decreases by 0.8%–5.5%, and R^2 increases by 0.04–0.13. The reference for MAE is CACTI in all 10 groups (Gorla et al., 2025). Among the 90 (9 datasets $\times$ 5 settings $\times$ 2 sides) cells, the proportion of best results (ties included) is MAE $77/90$, RMSE $72/90$ and R^2 $81/90$. The per-setting, per-side numbers from which these counts are computed are shown in the per-setting figures of Appendix E.

Downstream Prediction: Imputation quality is ultimately reflected in its value for downstream tasks. We fit a fixed XGBoost predictor (Chen & Guestrin, 2016) on the imputed training set and evaluate it on the test set. The 6 classification datasets report AUC, and beijing, california and news report RMSE on the log scale. Downstream labels take no part in the imputation training of this paper. Table 3 gives the results under MCAR 30%.

Table 2: In-sample average error difference under MCAR 30% of the three graph variants relative to our method.

	MAE		RMSE	
Graph	Δ	Rel.	Δ	Rel.
Marginal	0.0032	1.16%	0.0042	0.72%
Complete	0.0108	4.94%	0.0109	2.40%
Random	0.0157	7.29%	0.0206	4.19%

What Does the Conditional-Dependence Graph Contribute? On the real imputation task, the conditional-dependence graph is compared with a marginal-correlation graph of the same edge count, a random graph and a complete graph (Telyatnikov & Scardapane, 2023). Better structural recovery does not by itself imply better prediction, so we use imputation error rather than structural recovery metrics as the criterion.

Table 2 summarizes the average error difference of the three graph variants relative to our method. The averages are taken with equal weight over the 9 datasets, Δ is the variant minus our method, so that a positive value means the variant is worse, and Rel. is the equal-weight average of the per-dataset ratio of Δ to the error of our method.

Table 3: Downstream prediction under MCAR (missing rate 30%).

Method	AUC ↑ (classification)						RMSE(log) ↓ (regression)		
	adult	bean	default	magic	obesity	shoppers	beijing	california	news
Mean	0.8996 ±0.0016	0.9931 ±0.0000	0.7444 ±0.0036	0.8828 ±0.0041	**0.9695** ±0.0018	**0.8674** ±0.0029	0.7653 ±0.0023	0.2823 ±0.0028	0.8277 ±0.0011
MissForest	0.9003 ±0.0013	0.9935 ±0.0002	0.7462 ±0.0020	0.8844 ±0.0060	0.9634 ±0.0046	0.8566 ±0.0041	0.7706 ±0.0005	0.2830 ±0.0015	0.8240 ±0.0019
TabCSDI	0.5403 ±0.0056	0.5125 ±0.0012	0.5326 ±0.0103	0.5048 ±0.0012	0.5037 ±0.0070	0.5033 ±0.0117	1.0588 ±0.0116	0.5042 ±0.0018	0.9051 ±0.0015
MissingDataOT	0.8938 ±0.0016	0.9929 ±0.0004	0.7451 ±0.0023	0.8747 ±0.0031	0.9599 ±0.0074	0.8636 ±0.0021	0.8019 ±0.0017	0.3039 ±0.0004	0.8294 ±0.0022
GRAPE	0.8790 ±0.0033	0.9935 ±0.0002	0.6998 ±0.0064	0.8478 ±0.0062	0.9535 ±0.0090	0.8294 ±0.0136	0.7867 ±0.0075	0.4225 ±0.0102	0.8765 ±0.0257
HyperImpute	0.8958 ±0.0012	0.9944 ±0.0002	0.7486 ±0.0011	0.8827 ±0.0024	0.9471 ±0.0017	0.8491 ±0.0036	0.7678 ±0.0025	0.2918 ±0.0020	0.8211 ±0.0010
ReMasker	0.8879 ±0.0046	0.9944 ±0.0000	0.7443 ±0.0030	0.8832 ±0.0056	0.9490 ±0.0045	0.8556 ±0.0072	0.7616 ±0.0024	0.3484 ±0.0213	0.8440 ±0.0075
DiffPuter	0.8954 ±0.0012	0.9940 ±0.0001	0.7473 ±0.0009	0.8831 ±0.0019	0.9595 ±0.0042	0.8571 ±0.0025	0.7565 ±0.0017	0.3007 ±0.0120	0.8232 ±0.0008
CACTI	0.8987 ±0.0010	**0.9946** ±0.0001	0.7446 ±0.0011	0.8880 ±0.0016	0.9626 ±0.0035	0.8636 ±0.0005	**0.7401** ±0.0029	0.2831 ±0.0018	0.8211 ±0.0018
Ours	**0.9004** ±0.0020	**0.9946** ±0.0001	**0.7517** ±0.0037	**0.8883** ±0.0020	0.9640 ±0.0022	0.8647 ±0.0007	0.7501 ±0.0015	**0.2793** ±0.0017	**0.8197** ±0.0002

What Does the Orthogonal Factor Constrain? For any orthogonal matrix Q we have $\|Qx\|_2 = \|x\|_2$ and $\langle Qx, Qy\rangle = \langle x, y\rangle$. The orthogonal factor only rotates or reflects the coordinates and applies no anisotropic scaling on its own. The product of the two endpoints' orthogonal maps is still orthogonal, and the change in magnitude during message passing comes from the scales, the fixed edge weights and the subsequent operations.

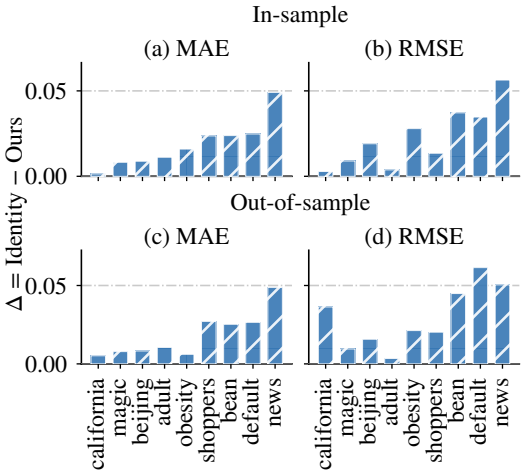


Figure 4: Change in error when the conformal maps are replaced by identity matrices.

Figure 4 gives the measured result of replacing the conformal maps by identity matrices, with the graph and the training protocol held fixed. Positive value means the error grows when the maps are replaced. The ablation sets the conformal maps to the identity at once, $\mathbf{S}_i = \mathbf{T}_i = I$ at every node, so the sending and receiving passes coincide (Ribeiro et al., 2026). The comparison covers all nine datasets of Table 8. All 36 cells are positive. All 36 cells are positive. In magnitude, averaged with equal weight over the nine datasets, setting the conformal maps to the identity raises the in-sample MAE and RMSE by 0.0185 and 0.0227 and the out-of-sample ones by 0.0184 and 0.0292. Relative to the identity-map variant, the full model lowers the error by 5.4%, 3.5%, 5.3% and 4.5% in the same order.

6 Conclusion

This work revisits graph-based tabular imputation from the perspective of information structure. CAGI separates graph-based imputation into two aspects: identifying relevant dependencies and communicating information across heterogeneous variables. Conditional dependencies provide a graph structure that better reflects the relationships underlying reconstruction, while column-specific source and target transformations adapt information alignment to different columns. Our theoretical analysis further shows that effective graph-based imputation depends on both exposing sufficient information for reconstruction and preserving it during propagation. This perspective highlights that effective imputation requires not only expressive predictors, but also appropriate information structures and mechanisms for communicating heterogeneous representations.

AI USE STATEMENT

Generative AI tools were used to help develop the theoretical framework of this work, to implement the method, to help interpret the experimental results, and to assist with translation. They were also used to assist with language editing and manuscript writing, including improving grammar, clarity, phrasing, and overall presentation, to help summarise and analyse the existing literature, and to make minor visual edits to a scientific figure. They were not used to formulate mathematical claims, to provide ingredients for or assist with proofs, to propose or refine research hypotheses, to design the methodology or experiments, to generate synthetic datasets, or to clean or reorder datasets; qualitative or thematic data analysis is not applicable to this work. All mathematical claims and proofs, the experimental protocol, and every reported result were developed and verified by the authors. The authors reviewed all AI-assisted edits and take full responsibility for the final manuscript.

ETHICS STATEMENT

This work studies imputation for tabular data and is developed for research use. All experiments are conducted on publicly available benchmark datasets, and we follow the usage terms and licences of the benchmark suite from which they are taken; the authors collected no data from human subjects. Tabular imputation can feed into downstream decisions about individuals, so we emphasise that the method should not be used for profiling or for decisions about individuals without task-specific validation and an assessment of the consequences of imputation errors. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

We provide the complete implementation details of the method. The datasets, train/test partitions and missing masks are described in Appendix D; the model configuration and training settings are given in Appendix E and Table 9, and the graph construction procedure is specified in Appendix F. All hyperparameters are stated in the paper or in these appendices, and the configuration used for every baseline is recorded in Table 10. The theoretical claims are stated together with their assumptions and proved in Appendix B. The source code is publicly available at `https://anonymous.4open.science/r/CAGI-14AC`.

REFERENCES

Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Lió, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *NeurIPS*, 2022.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *KDD*, pp. 785–794. ACM, 2016.

Tianyu Du, Luca Melis, and Ting Wang. ReMasker: Imputing tabular data with masked autoencoding. In *ICLR*. OpenReview.net, 2024.

Jianqing Fan, Han Liu, Yang Ning, and Hui Zou. High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(2):405–421, 03 2017. ISSN 1369-7412. doi: 10.1111/rssb.12168. URL `https://doi.org/10.1111/rssb.12168`.

Jicong Fan. An interdisciplinary and cross-task review on missing data imputation. *CoRR*, abs/2511.01196, 2025.

Konstantin Göbler, Anne Miloschewski, Mathias Drton, and Sach Mukherjee. High-dimensional undirected graphical models for arbitrary mixed data. *CoRR*, abs/2211.11700, 2022.

Aditya Gorla, Ryan Wang, Zhengtong Liu, Ulzee An, and Sriram Sankararaman. CACTI: leveraging copy masking and contextual information to improve tabular data imputation. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. HyperImpute: Generalized iterative imputation with automatic model selection. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9916–9937. PMLR, 2022.

Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(6):6488–6508, 2026.

Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph neural networks for tabular data learning: A survey with taxonomy and directions. *ACM Comput. Surv.*, 58(1):16:1–16:51, 2026.

Han Liu, John D. Lafferty, and Larry A. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.*, 10:2295–2328, 2009.

Pei-Chia Liu and Cheng-Te Li. Self-supervised bipartite graph neural networks with missing value imputation for small tabular data predictions. *ACM Trans. Intell. Syst. Technol.*, 17(2):31:1–31:31, 2026.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: deep generative modelling and imputation of incomplete data sets. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4413–4423. PMLR, 2019.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2401–2409. PMLR, 2017.

Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. Missing data imputation using optimal transport. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7130–7140. PMLR, 2020.

Ulf Olsson. Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44(4):443–460, 1979. doi: 10.1007/BF02296207.

Ulf Olsson, Fritz Drasgow, and Neil J. Dorans. The polyserial correlation coefficient. *Psychometrika*, 47(3):337–347, 1982. doi: 10.1007/BF02294164.

Yidong Ouyang, Liyan Xie, Chongxuan Li, and Guang Cheng. MissDiff: Training diffusion models on tabular data with missing values. *CoRR*, abs/2307.00467, 2023.

André Ribeiro, Ana Luiza Tenório, Juan Belieni, Amauri Souza, and Diego Mesquita. Cooperative sheaf neural networks. In *ICLR*, 2026.

Trevor W. Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A. Bernal. McFlow: Monte carlo flow models for data imputation. In *CVPR*, pp. 14193–14202. Computer Vision Foundation / IEEE, 2020.

Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. SAINT: improved neural networks for tabular data via row attention and contrastive pre-training. *CoRR*, abs/2106.01342, 2021.

Daniel J. Stekhoven and Peter Bühlmann. MissForest - non-parametric missing value imputation for mixed-type data. *Bioinform.*, 28(1):112–118, 2012.

Mihaela C. Stoian, Eleonora Giunchiglia, and Thomas Lukasiewicz. A survey on deep learning approaches for tabular data generation: Utility, alignment, fidelity, privacy, diversity, and beyond. *Trans. Mach. Learn. Res.*, 2026, 2026.

Lev Telyatnikov and Simone Scardapane. EGG-GAE: scalable graph neural networks for tabular data imputation. In *AISTATS*, volume 206 of *Proceedings of Machine Learning Research*, pp. 2661–2676. PMLR, 2023.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011. doi: 10.18637/jss.v045.i03.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: missing data imputation using generative adversarial nets. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5675–5684. PMLR, 2018.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: extending the success of self- and semi-supervised learning to tabular domain. In *NeurIPS*, 2020.

Jiaxuan You, Xiaobai Ma, Daisy Yi Ding, Mykel J. Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. In *NeurIPS*, 2020.

Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. DiffPuter: Empowering diffusion models for missing data imputation. In *ICLR*. OpenReview.net, 2025a.

Hengrui Zhang, Liancheng Fang, Qitian Wu, and Philip S. Yu. TabNAT: A continuous-discrete joint generative framework for tabular data. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b.

Zhaoyang Zhang, Hongtu Zhu, Yingjie Zhang, Hai Shu, and Ziqi Chen. Enhancing missing data imputation through combined bipartite graph and complete directed graph. *Neurocomputing*, 649: 130717, 2025c.

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

Jiajun Zhong, Ning Gui, and Weiwei Ye. Data imputation with iterative graph reconstruction. In *AAAI*, pp. 11399–11407. AAAI Press, 2023.

A Definition of Symbols

Tables 7, 4, 6 and 5 list the symbols used in this paper, in the order of first appearance.

Table 4: Symbols used for the conditional dependency graph.

Symbol	Explanation
$Z = (Z_1, \ldots, Z_d)^\top$	latent continuous variables associated with the d columns
Z_{-i}	all latent variables except Z_i
Σ	correlation matrix of the Gaussian copula representation
$\widehat{R} = (\widehat{R}_{jk})$	estimate of the latent correlation matrix
$\mathcal{J}_j, n_j$	rows in which column j is observed, and their number
$\widehat{\mathrm{rank}}(X_{rj})$	rank of X_{rj} among the observed values of column j
$\widehat{Z}_{rj}$	normal-score transform of column j at row r
K_j	number of categories of column j
$c_{j1} < \cdots < c_{jK_j}$	ordered categories of column j
$\widehat{F}_j$	empirical distribution function of column j
$\widehat{\Gamma}_{j\ell}$	latent threshold of column j at category ℓ
ℓ_{jr}	category of column j in row r
ρ	correlation parameter of the one-dimensional likelihood
$\widehat{\Omega}$	estimate of the precision matrix Σ^{-1}
$\widehat{R}_{\mathrm{psd}}$	$\widehat{R}$ after symmetrization, spectral truncation and rescaling to unit diagonal
Ξ	the spectrally truncated matrix before rescaling
$U\Lambda U^\top$	eigen-decomposition of the symmetrized $\widehat{R}$
C_{ij}	partial correlation between columns i and j
$E, \mathcal{E}$	edge set selected by thresholding the partial correlations
$b(i,j), \tau_b$	block containing the pair (i,j), and its empirical quantile level
a_{ij}	strength of the edge (i,j)
$\mathcal{N}(i), \mathcal{N}_i$	conditional neighborhood of column i, $\{j : (i,j) \in E\}$

Table 5: Symbols used for the theoretical analysis.

Symbol	Explanation
Ω	the precision matrix Σ^{-1}
$\widehat{E}_\tau$	thresholded edge set, $\{(i,j) : i \neq j,\ \lvert\widehat{\Omega}_{ij}\rvert \geq \tau\}$
τ	threshold on the entries of $\widehat{\Omega}$
$E^\star$	support of the off-diagonal entries of Ω
$\omega_{\min}$	smallest $\lvert\Omega_{ij}\rvert$ over the support $E^\star$
$\mathcal{F}_{\mathrm{shared}}$	propagation class with one shared map $A_{ij} = W$ on every edge
$\mathcal{F}_{\mathrm{adapt}}$	propagation class with column-specific maps $A_{ij} = T_i^\top S_j$
R_{ij}	relative rotation of the maps of columns i and j, $(Q_i^T)^\top Q_j^S$
$\mathcal{O}_i(M)$	the information available for reconstructing the target cell
$H_i^{(L)}$	final node representation of the target column, with the row index suppressed
$\mathcal{R}_i(f)$	reconstruction risk of the function f for column i
$\mathcal{R}_i^\star$	Bayes risk based on all information available after masking
$\eta_i(\mathcal{O}_i(M))$	Bayes decision $\mathbb{E}[X_i \mid \mathcal{O}_i(M)]$
$\mathcal{B}_i(M)$	mask-aware predictive neighborhood of the target column i
η_i^O	conditional mean $\mathbb{E}[X_i \mid \mathcal{O}_i(M)]$
η_i^H	conditional mean $\mathbb{E}[X_i \mid H_i^{(L)}, G, M]$
$\mathcal{F}$	class of reconstruction functions
$\mathcal{E}_{\mathrm{rep}}$	excess risk from information lost by diffusion
$\mathcal{E}_{\mathrm{dec}}$	excess risk from the reconstruction approximation
$\varepsilon_{\mathrm{rep}}, \varepsilon_{\mathrm{dec}}$	upper bounds on $\mathcal{E}_{\mathrm{rep}}$ and $\mathcal{E}_{\mathrm{dec}}$

Table 6: Symbols used for the column-adaptive graph diffusion and the masked reconstruction objective.

Symbol	Explanation
$H_{r,j}^{(\ell)}$	representation of node j in row r at layer ℓ
q	dimension of the column-specific coordinate space
$S_i^{(\ell)}$, $T_i^{(\ell)}$	source and target transformations of column i
$c_i^{S,(\ell)}$, $c_i^{T,(\ell)}$	positive learnable scalar factors of the conformal parameterization
$Q_i^{S,(\ell)}$, $Q_i^{T,(\ell)}$	orthogonal factors of the conformal parameterization, implemented with Householder reflections
$m_{j\to i}^{(\ell)}$	message sent from column j to column i
$A_{ij}^{(\ell)}$	induced source-to-target transformation, $(T_i^{(\ell)})^\top S_j^{(\ell)}$
$L_{\text{out}}^{(\ell)}$, $L_{\text{in}}^{(\ell)}$	source-oriented and target-oriented graph operators
D_{out}, D_{in}	the corresponding block-diagonal normalization operators
$\Delta_{\text{out}}^{(\ell)}$, $\Delta_{\text{in}}^{(\ell)}$	normalized versions of the two operators
$\mathcal{D}^{(\ell)}$	diffusion operator, $(\Delta_{\text{in}}^{(\ell)})^\top \Delta_{\text{out}}^{(\ell)}$
$\mathcal{W}^{(\ell)}$	learnable feature transformation
$W_1^{(\ell)}$, $W_2^{(\ell)}$	factors of $\mathcal{W}^{(\ell)}$ operating within the coordinate dimension and mixing feature channels
$\epsilon^{(\ell)}$	learnable node-wise scaling term
$\sigma(\cdot)$	nonlinear activation
M, M_{rj}	masking pattern, and its entry for the cell (r, j)
$\mathcal{M}_{\text{tr}}$	set of observed cells randomly masked during training
$H_{r,j}^{(L)}$	representation of node j in row r after the last diffusion layer
D_j	column-specific decoder
$\widehat{X}_{rj}$	reconstructed value of X_{rj}
$\mathcal{L}_{\text{imp}}$	masked reconstruction objective
ℓ_j	loss of column j within $\mathcal{L}_{\text{imp}}$

Table 7: Symbols used for the dataset, the column graph and the message-passing formulation.

Symbol	Explanation
N	number of samples (rows) in the dataset
D, d	number of columns; D is used in Section 2 and d in Sections 3 and 4
$\mathbf{X} \in \mathbb{R}^{N\times D}$	the tabular dataset
$\mathbf{x}_n$	the n-th row (sample) of $\mathbf{X}$
x_{nj}	the entry of row n in column j
$\mathcal{O}_n$	set of columns observed in row n
$\mathcal{M}_n$	set of columns missing in row n, $\{1,\dots,D\} \setminus \mathcal{O}_n$
$\mathbf{x}_{n,\mathcal{O}_n}$	the observed entries of row n
$p(\cdot \mid \cdot)$	conditional distribution; the imputation target is $p(\mathbf{x}_{n,\mathcal{M}_n} \mid \mathbf{x}_{n,\mathcal{O}_n})$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, G	the column graph, with node set $\mathcal{V}$ and edge set $\mathcal{E}$; G is used in Sections 4 and 6
$v_j \in \mathcal{V}$	the node of the graph corresponding to column j
$\widetilde{\mathbf{X}}$	the masked table
$\mathcal{S}$	set of artificially masked entries
$\ell(\cdot,\cdot)$	reconstruction loss
$\mathbf{h}_j^{(l)}$	representation of node v_j at layer l
$\mathcal{N}(j)$	neighbors of node v_j in the graph
$\phi^{(l)}$, $\text{AGG}^{(l)}$, $\psi^{(l)}$	message, aggregation and node-update functions of a message-passing layer

B PROOFS

B.1 PROOF OF PROPOSITION 1

Proof. For $i < j$, recursively expanding the autoregressive relation gives $Z_j = \rho^{j-i} Z_i + \sum_{k=i+1}^{j} \rho^{j-k} \varepsilon_k$. Since Z_i is independent of $\varepsilon_{i+1}, \ldots, \varepsilon_j$, $\mathrm{Cov}(Z_i, Z_j) = \rho^{j-i} \mathrm{Var}(Z_i)$. The chosen initialization and innovation variance imply $\mathrm{Var}(Z_t) = 1$ for all t. Therefore, $\mathrm{Corr}(Z_i, Z_j) = \rho^{j-i} = \rho^{|i-j|}$.

We next derive the precision structure. The joint density of the Markov chain can be written as $p(z) \propto \exp\left[-\frac{1}{2}\left(z_1^2 + \frac{1}{1-\rho^2}\sum_{t=2}^{d}(z_t - \rho z_{t-1})^2\right)\right]$. Expanding the quadratic form shows that the precision matrix is tridiagonal, with $\Omega_{11} = \Omega_{dd} = \frac{1}{1-\rho^2}$, $\Omega_{tt} = \frac{1+\rho^2}{1-\rho^2}, \quad 2 \le t \le d-1$, and $\Omega_{t,t+1} = \Omega_{t+1,t} = -\frac{\rho}{1-\rho^2}$. Thus, $\Omega_{ij} = 0, \quad |i-j| > 1$. For a non-degenerate Gaussian distribution, $\Omega_{ij} = 0 \iff Z_i \perp Z_j \mid Z_{-(i,j)}$. Therefore, non-adjacent variables can be marginally correlated while being conditionally independent. □

B.2 PROOF OF LEMMA 2

Proof. Consider first any non-edge $(i,j) \notin E^\star$. Then $\Omega_{ij} = 0$, and hence $|\widehat{\Omega}_{ij}| = |\widehat{\Omega}_{ij} - \Omega_{ij}| \le \epsilon < \tau$. Thus no non-edge is selected.

For any true edge $(i,j) \in E^\star$, the reverse triangle inequality gives $|\widehat{\Omega}_{ij}| \ge |\Omega_{ij}| - |\widehat{\Omega}_{ij} - \Omega_{ij}| \ge \omega_{\min} - \epsilon > \tau$. Thus every true edge is selected. Therefore, $\widehat{E}_\tau = E^\star$. □

B.3 PROOF OF THEOREM 3

Proof. By the global Markov property, node i is separated from $V \setminus (\mathcal{N}_i \cup \{i\})$ by $\mathcal{N}_i$. Hence, $Z_i \perp Z_{V\setminus(\mathcal{N}_i\cup\{i\})} \mid Z_{\mathcal{N}_i}$. By the definition of conditional independence, $p\left(Z_i \mid Z_{\mathcal{N}_i}, Z_{V\setminus(\mathcal{N}_i\cup\{i\})}\right) = p(Z_i \mid Z_{\mathcal{N}_i})$. The variables on the left-hand side together constitute Z_{-i}. Therefore, $p(Z_i \mid Z_{-i}) = p(Z_i \mid Z_{\mathcal{N}_i})$. □

B.4 PROOF OF THEOREM 5

Proof. We first establish inclusion. Let $W \in \mathbb{R}^{q\times q}$ be any shared transformation. Set $S_j = I_q$ for every source node and $T_i = W^\top$ for every target node. Then, for every edge $(i,j) \in E$, $T_i^\top S_j = (W^\top)^\top I_q = W$. Hence every shared propagation rule can be represented by the adaptive parameterization, proving $\mathcal{F}_{\text{shared}} \subseteq \mathcal{F}_{\text{adapt}}$.

To establish strictness, suppose that the graph contains $(j,i_1), \quad (j,i_2), \quad i_1 \ne i_2$. Choose $S_j = I_q, \quad T_{i_1} = I_q, \quad T_{i_2} = 2I_q$. Then $A_{i_1 j} = T_{i_1}^\top S_j = I_q$, whereas $A_{i_2 j} = T_{i_2}^\top S_j = 2I_q$. No single shared matrix W can satisfy $A_{i_1 j} = A_{i_2 j} = W$. Thus this adaptive propagation rule cannot be represented by a shared transformation, proving $\mathcal{F}_{\text{shared}} \subsetneq \mathcal{F}_{\text{adapt}}$.

Finally, in CAGI the transformations are dynamically generated from the current graph representations. Fix any realization of these generated transformations at a given layer. The above arguments then apply directly to that realization. Conversely, the shared case is recovered when the generation mechanism produces identical source and target transformations across columns. Therefore, dynamic generation preserves the inclusion relation pointwise. □

B.5 PROOF OF PROPOSITION 6

Proof. Substituting the parameterization gives

$$A_{ij} = (c_i^T Q_i^T)^\top (c_j^S Q_j^S) = c_i^T c_j^S (Q_i^T)^\top Q_j^S. \tag{10}$$

Define $R_{ij} = (Q_i^T)^\top Q_j^S$. Then

$$\begin{aligned} R_{ij}^\top R_{ij} &= (Q_j^S)^\top Q_i^T (Q_i^T)^\top Q_j^S \\ &= (Q_j^S)^\top Q_j^S \\ &= I_q. \end{aligned} \tag{11}$$

Thus R_{ij} is orthogonal. Therefore,

$$\begin{aligned} \|A_{ij}x\|_2 &= c_i^T c_j^S \|R_{ij}x\|_2 \\ &= c_i^T c_j^S \|x\|_2. \end{aligned} \tag{12}$$

□

B.6 Proof of Theorem 8

Proof. By predictive locality, $\eta_i(\mathcal{O}_i(M)) = \mathbb{E}\left[X_i \mid G, M, X_{\mathcal{B}_i(M)}\right]$. By information accessibility, $X_{\mathcal{B}_i(M)} = r_i(H_i^{(L)}, G, M)$. Therefore, the Bayes decision is measurable with respect to $(H_i^{(L)}, G, M)$. In particular, there exists a measurable function $\widetilde{g}_i$ such that $\eta_i(\mathcal{O}_i(M)) = \widetilde{g}_i(H_i^{(L)}, G, M)$. By decoder realizability, $g_i \in \mathcal{F}$ realizes this Bayes decision. Hence, $\mathcal{R}_i(g_i) = \mathcal{R}_i^\star$. Since $\mathcal{R}_i^\star$ is the minimum achievable risk when the full available information is accessible, $\inf_{f\in\mathcal{F}} \mathcal{R}_i(f) = \mathcal{R}_i^\star$.

For the approximation result, let $\eta_i^H = \eta_i(\mathcal{O}_i(M))$. If $\mathcal{F}$ is dense in L_2, then for every $\varepsilon > 0$ there exists $f \in \mathcal{F}$ such that $\mathbb{E}\left[\left(f(H_i^{(L)}, G, M) - \eta_i^H\right)^2\right] \le \varepsilon$. The standard conditional-expectation orthogonality identity then gives $\mathcal{R}_i(f) - \mathcal{R}_i^\star = \mathbb{E}\left[\left(f(H_i^{(L)}, G, M) - \eta_i^H\right)^2\right] \le \varepsilon$. Therefore, $\mathcal{R}_i(f) \le \mathcal{R}_i^\star + \varepsilon$. □

B.7 Proof of Theorem 9

Proof. For brevity, write $Y = X_i, \quad \eta = \eta_i^O, \quad \eta_H = \eta_i^H, \quad H = H_i^{(L)}$. Because H is a measurable function of $\mathcal{O}_i(M)$, both $\eta - \eta_H$ and $\eta_H - f_i(H)$ are measurable with respect to $\mathcal{O}_i(M)$.

Decompose the prediction error as

$$Y - f_i(H) = (Y - \eta) + (\eta - \eta_H) + (\eta_H - f_i(H)). \tag{13}$$

Since $\mathbb{E}[Y - \eta \mid \mathcal{O}_i(M)] = 0$, the residual $Y - \eta$ is orthogonal to both remaining terms. Moreover, $\eta_H = \mathbb{E}\left[\eta \mid H, G, M\right]$, and therefore

$$\mathbb{E}\left[\eta - \eta_H \mid H, G, M\right] = 0. \tag{14}$$

Thus,

$$\mathbb{E}\left[(\eta - \eta_H)(\eta_H - f_i(H))\right] = 0. \tag{15}$$

All cross terms therefore vanish, yielding

$$\begin{aligned} \mathcal{R}_i(f_i) = \mathbb{E}\left[(Y - \eta)^2\right] &+ \mathbb{E}\left[(\eta - \eta_H)^2\right] \\ &+ \mathbb{E}\left[(f_i(H) - \eta_H)^2\right]. \end{aligned} \tag{16}$$

The first term equals the Bayes risk:

$$\mathbb{E}\left[(Y - \eta)^2\right] = \mathcal{R}_i^\star. \tag{17}$$

Hence,

$$\mathcal{R}_i(f_i) - \mathcal{R}_i^\star = \mathcal{E}_{\text{rep}} + \mathcal{E}_{\text{dec}}. \tag{18}$$

Finally, since $\eta_H = \mathbb{E}\left[\eta \mid H, G, M\right]$, the definition of conditional variance gives

$$\begin{aligned} \mathcal{E}_{\text{rep}} &= \mathbb{E}\left[(\eta - \eta_H)^2\right] \\ &= \mathbb{E}\left[\text{Var}\left(\eta \mid H, G, M\right)\right]. \end{aligned} \tag{19}$$

□

B.8 Proof of Corollary 10

Proof. By the exact excess-risk decomposition,

$$\mathcal{R}_i(f_i) - \mathcal{R}_i^\star = \mathcal{E}_{\text{rep}} + \mathcal{E}_{\text{dec}}(f_i). \tag{20}$$

Taking the infimum over $f_i \in \mathcal{F}_i$ gives

$$\begin{aligned} \inf_{f_i \in \mathcal{F}_i} \mathcal{R}_i(f_i) - \mathcal{R}_i^\star &= \mathcal{E}_{\text{rep}} + \inf_{f_i \in \mathcal{F}_i} \mathcal{E}_{\text{dec}}(f_i) \\ &\leq \varepsilon_{\text{rep}} + \varepsilon_{\text{dec}}. \end{aligned} \tag{21}$$

Therefore,

$$\inf_{f_i \in \mathcal{F}_i} \mathcal{R}_i(f_i) \leq \mathcal{R}_i^\star + \varepsilon_{\text{rep}} + \varepsilon_{\text{dec}}. \tag{22}$$

□

C Related Work

Statistical and Iterative Imputation. Statistical and iterative methods recover missing values from marginal statistics, local neighborhoods or column-wise conditional models: MICE alternates conditional models over columns to update each missing entry (van Buuren & Groothuis-Oudshoorn, 2011), MissForest combines this iterative idea with random forests to handle non-linear relations in mixed-type data (Stekhoven & Bühlmann, 2012), and HyperImpute brings the selection of the per-column model and its hyperparameters into the iterative procedure (Jarrett et al., 2022). Column-wise conditional prediction is therefore strong in modeling capacity, but it fits a separate predictive model per column.

Generative and Diffusion-Based Imputation. This family imputes by learning the data distribution and sampling the missing entries from it. Outside the diffusion route, GAIN generates missing entries with adversarial training (Yoon et al., 2018), MIWAE constructs an importance-weighted likelihood lower bound for incomplete observations (Mattei & Frellsen, 2019), and MCFlow alternately updates the density estimate of a normalizing flow and the missing values (Richardson et al., 2020). On the diffusion side, TabCSDI applies conditional score-based diffusion to mixed-type data (Zheng & Charoenphakdee, 2022), and DiffPuter places diffusion in an EM framework that alternates between distribution learning and missing-value updates (Zhang et al., 2025a). Both are among our baselines, and we contrast their generative form with the discriminative masked reconstruction used here. More recent work varies the training objective and the conditioning of the sampling step: MissDiff learns the distribution directly from tables with missing values using a masked denoising score-matching loss (Ouyang et al., 2023), and TabNAT combines conditional diffusion over continuous variables with categorical prediction over discrete ones (Zhang et al., 2025b). Deep generative models for tabular data are surveyed by Stoian et al. (2026).

Masked Reconstruction. Masked methods construct a self-supervised signal by masking and reconstructing observed entries (Yoon et al., 2020): ReMasker masks additional entries on top of the original missing ones and learns to reconstruct with a Transformer autoencoder (Du et al., 2024), and CACTI introduces a data-dependent inductive bias whose median truncated copy masking builds training masks from empirical missingness patterns, improving feature representations with the contextual semantics of column names (Gorla et al., 2025).

Graph-Based Imputation and Structured Message Passing. Graph methods organize the relations in a table explicitly by defining nodes, edges and a prediction target (Li et al., 2026). GRAPE treats samples and features as two types of nodes in a bipartite graph and observed cells as attributed edges, formulating feature imputation as edge-level prediction (You et al., 2020). MissGNN combines feature-imputation pretraining, node-label prediction and multi-view learning on the same kind of bipartite graph (Liu & Li, 2026). Beyond bipartite graphs, later work models direct relations among samples or features: IGRM repeatedly reconstructs a friend network among samples and uses it to guide information propagation (Zhong et al., 2023), EGG-GAE learns graph connections layer by layer within randomly sampled mini-batches (Telyatnikov & Scardapane, 2023), and BCGNN adds a complete directed feature graph on top of the bipartite graph, passing feature relations through element-wise attention and the sign of the estimated Spearman correlation

(Zhang et al., 2025c). CAGI builds its conditional-dependence graph from regularized latent partial correlations under a Gaussian copula representation, whose latent graph estimation under mixed continuous and binary data is studied by Fan et al. (2017), and separates sending and receiving maps, following CSNN (Ribeiro et al., 2026), which builds on earlier sheaf neural networks (Bodnar et al., 2022). Estimation of such a graph from mixed-type data is itself an established problem in statistics, where latent Gaussian copula models with polychoric and polyserial correlations are plugged into precision-matrix estimation routines (Göbler et al., 2022). Our work takes that estimation as given and asks what the resulting graph does inside an imputation model.

D Experimental Setup

Datasets. Experiments are conducted on 9 public tabular datasets covering regression (3), binary classification (4) and multi-class classification (2) tasks, with $2\,111$ to $41\,757$ rows and 8 to 47 modeled columns. The datasets and missing masks follow the benchmark suite released by DiffPuter (Zhang et al., 2025a) and use the same datasets, train/test partitions and missing masks as those of CACTI (Gorla et al., 2025). The column types and train/test partitions of each dataset are given in Table 8.

Missingness Settings. The missingness settings comprise three MCAR levels, 30%, 50% and 70%, and one level each of MAR and MNAR at 30% (Rubin, 1976). Masks are given separately for the training and the test set, and all methods share the same mask within a (dataset, setting) pair, so the comparison is position-wise paired.

Baselines. Apart from our method there are 9 baselines, taken from the comparison sets of DiffPuter and CACTI. The column-wise statistical and iterative methods are Mean (column mean), MissForest (Stekhoven & Bühlmann, 2012) and HyperImpute (Jarrett et al., 2022); the diffusion and masked-modeling methods are TabCSDI (Zheng & Charoenphakdee, 2022), ReMasker (Du et al., 2024), CACTI (Gorla et al., 2025) and DiffPuter (Zhang et al., 2025a); the graph method is GRAPE (You et al., 2020); and the optimal-transport method is MissingDataOT (Muzellec et al., 2020). Every baseline is run with the implementation and configuration of the benchmark suite it is taken from, which for most methods follows the official implementation and hyperparameters; Table 10 records the exact configuration, and where a configuration is a substitute for the original method, the table names the components used. The training budget follows the setting of the respective original paper rather than being forced to match in epochs, because convergence criteria differ.

Shared Input and Evaluation Scale. All methods run on the same input representation and the same evaluation scale. Numerical columns are standardized to $(x - \mu)/\sigma/2$, with the mean and standard deviation taken over observed training entries, and categorical columns keep integer labels. Evaluation counts only the originally missing positions, and reported errors are on the $(x - \mu)/\sigma$ scale, that is, in units of one observed training standard deviation of the column. Numerical performance is reported by MAE, RMSE and R^2 and categorical performance by accuracy. MAE and RMSE are pooled over the missing cells of all numerical columns, whereas R^2 is computed per column over the missing cells of that column and averaged with equal weight over the columns with more than 10 such cells. Accuracy is the pooled fraction of correctly recovered integer labels over the missing categorical cells. In-sample and out-of-sample refer to imputation on the training and the test set respectively, under the same masking protocol. The validation cells and the dynamically masked positions of each epoch are drawn only from originally observed positions, so model selection and early stopping never touch any missing ground truth. The validation cells are excluded from the pool from which the dynamic masking of each epoch is drawn and are never included in the training reconstruction loss; during training they are passed to the encoder as observed inputs. In the validation forward pass they are masked from the encoder and serve as the validation targets, so the validation loss is computed on held-out observed cells. Each table value is the mean over the runs available for that (dataset, setting), and the reported uncertainty is one standard deviation over those runs, which the per-dataset figures of Appendix E draw as error bars on their dataset bars. The runs of one pair differ only in the data partition they use: every configuration field is identical across them, including the random seed, so the spread of a cell measures the sensitivity to the partition rather than to initialization.

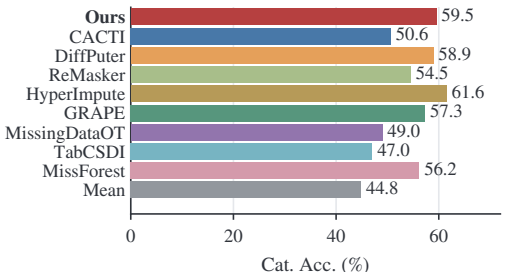


Figure 5: Accuracy on the categorical columns, pooled with equal weight over the 10 (setting, side) groups and over the 6 datasets that contain categorical columns.

E Experimental Details

Table 8: The 9 datasets used in the experiments. "Numerical/categorical columns" counts modeled columns and excludes the prediction target column. Datasets with 0 categorical columns take no part in the categorical metrics. All datasets use a $70\%/30\%$ training/test ratio, where the training set is the in-sample side and the test set the out-of-sample side of the main text.

Dataset	Rows	Numerical	Categorical	Train	Test	Task type
beijing	41 757	6	5	29 229	12 528	Regression
california	20 640	8	0	14 447	6 193	Regression
news	39 644	45	2	27 750	11 894	Regression
adult	32 561	6	8	22 792	9 769	Binary classification
default	30 000	14	9	21 000	9 000	Binary classification
magic	19 020	10	0	13 314	5 706	Binary classification
shoppers	12 330	10	7	8 631	3 699	Binary classification
bean	13 610	16	0	9 527	4 083	Multi-class classification
obesity	2 111	8	8	1 477	634	Multi-class classification

Table 9: Training and model configuration of our method, given by the argument names of our implementation. `num_layers` counts the message-passing layers and `gnn_hidden` is the width of the restriction-map learner; entries marked † are constants fixed in the implementation rather than arguments.

Training		*Model*	
`epochs`	1000	`num_layers`	4
`early_stop_patience`	50	`hidden_channels`	32
`batch_size`	128	`stalk_dim`	4
`lr`	2×10^{-3}	`dec_hidden`	128
`dropout`	0.0	`attr_emb_dim`	32
`grad_clip`	5.0	`val_emb_dim`	16
`lr_scheduler`	`plateau`	`l1_init`	0.1
`weight_decay`†	0.0	`gnn_hidden`†	32
`ema_decay`†	0.999	`edge_quantile`	0.5
`rand_mask_ratio`	0.5	`seed`†	0
`val_mask_ratio`	0.1	`num_initial_impute`	`mean`
`val_interval`	5	`cat_initial_impute`	`mode`

Table 10: Key hyperparameters of the baselines, as configured in the benchmark suite each is taken from; where a configuration substitutes components of the original method, those components are listed here.

Method	Hyperparameters
CACTI (Gorla et al., 2025)	`epochs`=600, `batch_size`=128, `lr`=1e-3, `mask_ratio`=0.9, `embed_dim`=64, `nencoder`=10, `ndecoder`=4, `nhead`=8, context embeddings = 1024-dim GTE-en-mlm-large text embeddings
ReMasker (Du et al., 2024)	`epochs`=600, `batch_size`=64, `blr`=1e-3, `mask_ratio`=0.5, `embed_dim`=32, `depth`=6, `decoder_depth`=4, `num_heads`=4
TabCSDI (Zheng & Charoenphakdee, 2022)	`epochs`=100, `batch_size`=256, `lr`=5e-4, `diff_steps`=100, `layers`=4, `channels`=128, `nheads`=4
GRAPE (You et al., 2020)	`epochs`=2×10^4, `lr`=1e-3, `node_dim`=64, `edge_dim`=64, `model_types`=EGSAGE×3, `impute_hiddens`=64, `known`=0.7
DiffPuter (Zhang et al., 2025a)	`max_iter`=4, `epochs`=10^4, `batch_size`=4096, `lr`=1e-4, `hid_dim`=1024, `num_steps`=50
MissingDataOT (Muzellec et al., 2020)	`niter`=500, `epsilon`=automatic, `batchsize`=128, `lr`=1e-2
Mean	`SimpleImputer(strategy=mean)`
MissForest (Stekhoven & Bühlmann, 2012)	`IterativeImputer(max_iter=100)` with base learner `RandomForestRegressor(n_estimators=10, max_depth=3)`
HyperImpute (Jarrett et al., 2022)	regression model pool = {random_forest_regressor, linear_regression, xgboost_regressor, catboost_regressor}, classification model pool = {random_forest, xgboost, catboost}, with `neural_nets` excluded

Table 11: Measured training cost of our method on the nine datasets under MCAR 30%. The experiment uses an RTX 3090 and conducts a validation set evaluation every 5 epochs. Training time is the wall-clock time of the training loop up to early stopping, inclusive of the periodic validation pass and exclusive of the one-off graph construction. Per-epoch time is that time divided by the number of epochs run. Training ends by early stopping, so the epoch count and hence the wall-clock vary across datasets.

Dataset	Epochs run	Training time (s)	Per epoch (s)	Parameters
beijing	256	3 586.4	14.01	108 666
california	201	1 164.5	5.79	97 824
news	411	17 844.5	43.42	107 538
adult	166	3 404.2	20.51	112 580
default	216	3 785.0	17.52	110 675
magic	191	1 805.8	9.45	98 242
shoppers	171	860.0	5.03	108 003
bean	471	2 640.2	5.61	99 496
obesity	336	414.1	1.23	101 543

F GRAPH CONSTRUCTION ALGORITHM

Algorithm 1 Construction of the conditional-dependence graph.

Require: data matrix X; the observed rows $\mathcal{J}_j$ of each column j; the partition of the columns into the numeric–numeric, numeric–categorical and categorical–categorical blocks

Ensure: edge set E and partial-correlation scores C_{ij}

$\widehat{Z}_{rj} \leftarrow \Phi^{-1}\big(\widehat{\mathrm{rank}}(X_{rj}) - 1/2\big)/n_j\big)$ for $r \in \mathcal{J}_j$

$\widehat{R}_{jk} \leftarrow$ Pearson, polyserial or polychoric over $\mathcal{J}_j \cap \mathcal{J}_k$, according to the pair of column types; $\widehat{R}_{jk} \leftarrow 0$ if $|\mathcal{J}_j \cap \mathcal{J}_k| < 10$; $\widehat{R}_{jj} \leftarrow 1$

$\Xi \leftarrow U \max(\Lambda, 10^{-8})U^\top$ for the eigen-decomposition $U\Lambda U^\top$ of the symmetrized $\widehat{R}$; $\widehat{R}_{\mathrm{psd}} \leftarrow \mathrm{diag}(\Xi)^{-1/2}\Xi\,\mathrm{diag}(\Xi)^{-1/2}$

$\widehat{\Omega} \leftarrow (\widehat{R}_{\mathrm{psd}} + 10^{-8}I)^{-1}$; $C_{ij} \leftarrow -\widehat{\Omega}_{ij}/\sqrt{\widehat{\Omega}_{ii}\widehat{\Omega}_{jj}}$

for each block b **do**

  $E \leftarrow E \cup \{(i,j) \in b : |C_{ij}| > \tau_b\}$, with τ_b the 0.5-quantile of $\{|C_{ij}|\}_{(i,j)\in b}$

end for

connect every column left isolated to its strongest partner $\arg\max_{k\neq i} |C_{ik}|$

join the remaining components with the edges of a maximum spanning forest over $\{|C_{ij}|\}$

return E

G GRAPH MODELS

G.1 GRAPH MESSAGE PASSING

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathbf{h}_j^{(l)}$ denote the representation of node v_j at layer l, with $\mathbf{h}_j^{(0)}$ initialized from its corresponding node features. A standard message-passing layer updates each node by aggregating information from its neighbors:

$$\mathbf{m}_j^{(l)} = \mathrm{AGG}^{(l)}\left(\left\{\phi^{(l)}\left(\mathbf{h}_j^{(l)}, \mathbf{h}_k^{(l)}\right) : k \in \mathcal{N}(j)\right\}\right), \tag{23}$$

followed by

$$\mathbf{h}_j^{(l+1)} = \psi^{(l)}\left(\mathbf{h}_j^{(l)}, \mathbf{m}_j^{(l)}\right), \tag{24}$$

where $\mathcal{N}(j)$ denotes the neighbors of node v_j, $\phi^{(l)}$ is a message function, $\mathrm{AGG}^{(l)}$ is a permutation-invariant aggregation function, and $\psi^{(l)}$ is a node-update function. In conventional message passing, the same message and update functions are shared across nodes and edges. This shared parameterization provides an efficient and general graph learning framework.

G.2 BIDIRECTIONAL GRAPH DIFFUSION

To integrate the source and target transformations into the graph propagation process, we construct two complementary graph operators (Ribeiro et al., 2026). Consider the column graph as a directed graph by replacing each undirected edge with two directed edges. For a row-level representation $H_r^{(\ell)}$, define the source-oriented diffusion operator as

$$\left(L_{\mathrm{out}}^{(\ell)} H_r^{(\ell)}\right)_i = \sum_{j\in\mathcal{N}(i)} \left[\left(S_i^{(\ell)}\right)^\top S_i^{(\ell)} H_{r,i}^{(\ell)} - \left(T_i^{(\ell)}\right)^\top S_j^{(\ell)} H_{r,j}^{(\ell)}\right]. \tag{25}$$

Similarly, define the target-oriented operator by

$$\left(\left(L_{\mathrm{in}}^{(\ell)}\right)^\top H_r^{(\ell)}\right)_i = \sum_{j\in\mathcal{N}(i)} \left[\left(T_i^{(\ell)}\right)^\top T_i^{(\ell)} H_{r,i}^{(\ell)} - \left(T_i^{(\ell)}\right)^\top S_j^{(\ell)} H_{r,j}^{(\ell)}\right]. \tag{26}$$

The off-diagonal interaction blocks of both operators are therefore determined by $-\big(T_i^{(\ell)}\big)^\top S_j^{(\ell)}$, which directly implements the column-specific source-to-target transformation introduced above.

The edge strength a_{ij} multiplies this off-diagonal term, while the self term carries no edge weight, so that the off-diagonal blocks reproduce the weighted message of Section 3.2. For numerical stability, we use normalized versions of the two operators, $\Delta_{\text{out}}^{(\ell)} = D_{\text{out}}^{-\frac{1}{2}} L_{\text{out}}^{(\ell)} D_{\text{out}}^{-\frac{1}{2}}$, $\left(\Delta_{\text{in}}^{(\ell)}\right)^\top = D_{\text{in}}^{-\frac{1}{2}} \left(L_{\text{in}}^{(\ell)}\right)^\top D_{\text{in}}^{-\frac{1}{2}}$, where D_{out} and D_{in} are the corresponding block-diagonal normalization operators. We then compose the two operators to form the proposed diffusion operator, $\mathcal{D}^{(\ell)} = \left(\Delta_{\text{in}}^{(\ell)}\right)^\top \Delta_{\text{out}}^{(\ell)}$. Given a learnable feature transformation $\mathcal{W}^{(\ell)}(H) = (I_d \otimes W_1^{(\ell)}) H W_2^{(\ell)}$, where $W_1^{(\ell)}$ operates within the latent coordinate dimension and $W_2^{(\ell)}$ mixes feature channels, the hidden representation is updated according to

$$H_r^{(\ell+1)} = \left(\mathbf{1} + \epsilon^{(\ell)}\right) \odot H_r^{(\ell)} - \sigma\left(\mathcal{D}^{(\ell)} \mathcal{W}^{(\ell)}\left(H_r^{(\ell)}\right)\right), \tag{27}$$

where $\epsilon^{(\ell)}$ is a learnable node-wise scaling term and $\sigma(\cdot)$ denotes a nonlinear activation.

The composition of the two operators is important: the first operator propagates source information through the graph, while the second aligns and aggregates the resulting information at the receiving columns. This structure allows the model to distinguish the roles of the columns as information sources and information receivers, while retaining a compact column-level parameterization.

Remark: A standard shared-transformation graph network is recovered as a special case when all columns use the same source and target transformations. In contrast, allowing $S_i^{(\ell)}$ and $T_i^{(\ell)}$ to vary across columns gives the model the flexibility to adapt the propagation rule to the heterogeneous representation spaces of tabular variables.

H Masked Cell Reconstruction

The proposed graph diffusion module is trained using a masked reconstruction objective. Let $\mathcal{M}_{\text{tr}} \subseteq \{(r, j) : M_{rj} = 1\}$ denote a set of observed cells randomly selected for masking during training. For each $(r, j) \in \mathcal{M}_{\text{tr}}$, the original value X_{rj} is replaced with the missing-value representation before graph propagation. The model then produces a contextual representation $H_{r,j}^{(L)}$ after L diffusion layers. A column-specific decoder $D_j : \mathbb{R}^q \to \mathcal{X}_j$ maps the final representation to the reconstructed value, $\widehat{X}_{rj} = D_j\left(H_{r,j}^{(L)}\right)$. The training objective is the reconstruction loss over the artificially masked cells: $\mathcal{L}_{\text{imp}} = \frac{1}{|\mathcal{M}_{\text{tr}}|} \sum_{(r,j)\in\mathcal{M}_{\text{tr}}} \ell_j\left(\widehat{X}_{rj}, X_{rj}\right)$, where ℓ_j is chosen according to the data type of column j. For continuous variables, ℓ_j can be a regression loss such as mean squared error, whereas categorical variables can be trained using an appropriate classification loss.

Remark: Importantly, the masking procedure does not alter the learned graph topology. The conditional dependency graph is estimated from the available training data and shared across row-level graph instances, while the masking process provides self-supervised reconstruction targets for learning the graph diffusion parameters.

Remark: At inference time, the observed entries remain fixed and only the missing cells are reconstructed. The model therefore performs imputation by propagating information through the learned conditional dependency structure while adapting the representation of messages to the source and target columns.

I Complete Results of the Mechanism Analysis

I.1 Edge-Weight Structure Before and After Inversion

The scores below are computed on the complete data with the estimator and the latent correlation matrix restated in Appendix F.

BILL_AMT1–BILL_AMT6 of default. These six columns are the bill amounts of the same account over six consecutive months. On the marginal correlations after rank transformation, the values of the 15 pairs all lie in 0.71–0.89, a ratio of about 1.25 between the largest and the smallest. Used as edge weights, these six columns form an almost weight-uniform complete subgraph, and "direct coupling between adjacent months" and "a detour across a month" are multiplied by nearly the same

Table 12: The four graph-construction variants on `default` under MCAR 30%, both sides; `partial` is ours. They run on one split and one mask, so every comparison is paired; bold is the column best.

	In-sample				Out-of-sample			
	MAE	RMSE	R^2	acc	MAE	RMSE	R^2	acc
complete	0.2305	0.6286	0.6107	0.7687	0.2252	0.6165	0.5114	0.7674
random	0.2339	0.6389	0.5960	0.7615	0.2266	0.6019	0.6226	0.7591
marginal	0.2252	0.6159	0.6307	0.7605	0.2199	0.5863	0.6412	0.7586
partial (ours)	**0.2105**	**0.5894**	**0.6510**	**0.7805**	**0.2057**	**0.5763**	**0.6609**	**0.7804**

scalar during propagation. After switching to the partial-correlation scores given by the inverse of the same matrix, the absolute values of the same 15 pairs drop to 0.07–0.27 and split into three non-overlapping groups by month distance: lag 1 0.17–0.27, lag 2 0.11–0.16, and lag $\geq$ 3 0.07–0.10. This agrees with our domain reading that bill balances are carried over month by month and adjacent months share more settlement information. We read it only as relative structure on this column family and not as recovery of the conditional-independence structure.

Effect on graph construction and propagation. In this example the edge set selected by the within-block retention ratio is identical under the two weightings, so the difference lies in the magnitude each edge carries rather than in which edges are present: partial correlation stratifies the weights by the coupling the estimator targets, whereas marginal correlation leaves the direct and the detour pairs at nearly the same level.

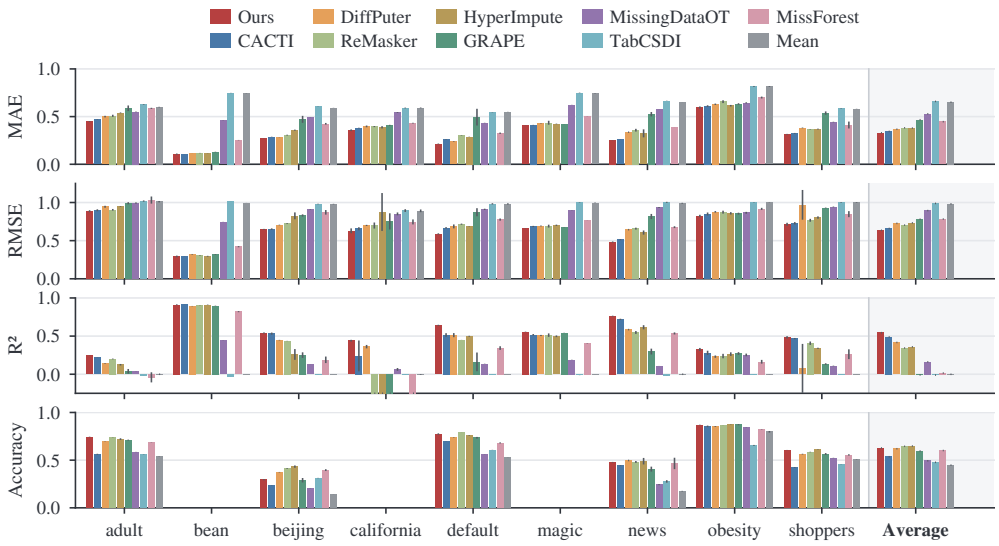


Figure 6: In-sample per-dataset results under MCAR (missing rate 30%). The four panels from top to bottom are MAE, RMSE, R^2 and Accuracy. Each panel contains 9 datasets plus Average, and lower is better for the first two. bean, california and magic have no categorical column, so those three groups are blank in the Accuracy panel, and the Average bar of that panel averages the 6 datasets that have categorical columns. The remaining settings are in the figures that follow.

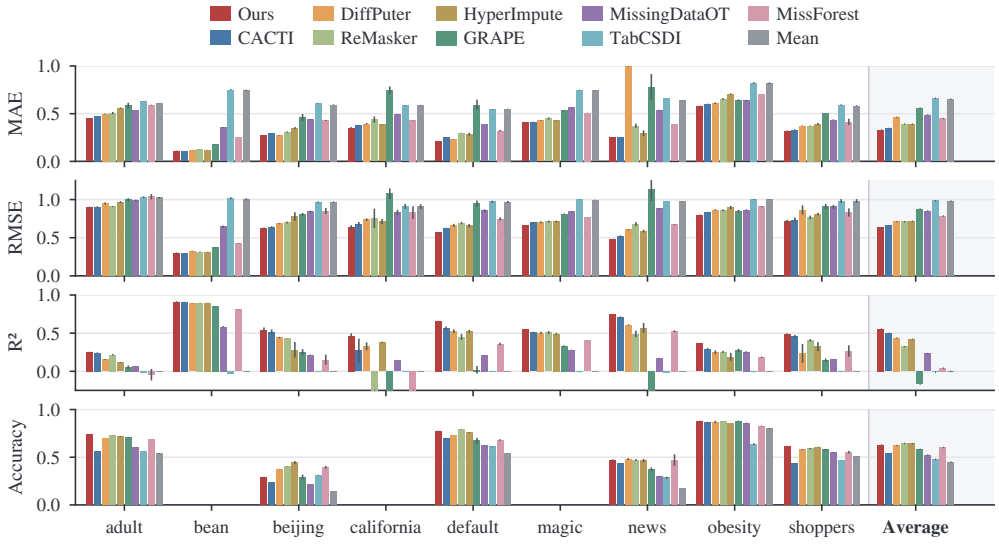


Figure 7: Per-dataset results for MCAR (30%) out-of-sample.

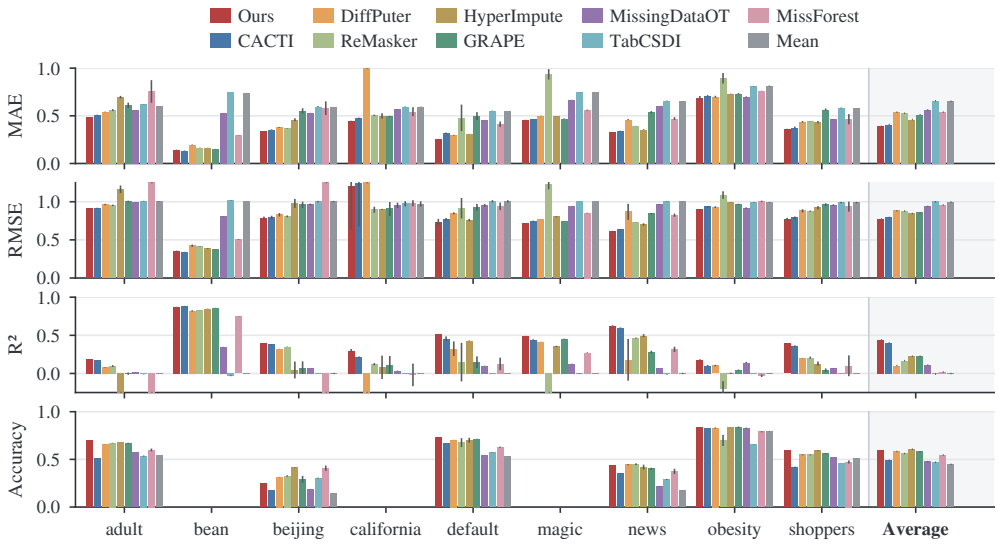


Figure 8: Per-dataset results for MCAR (50%) in-sample.

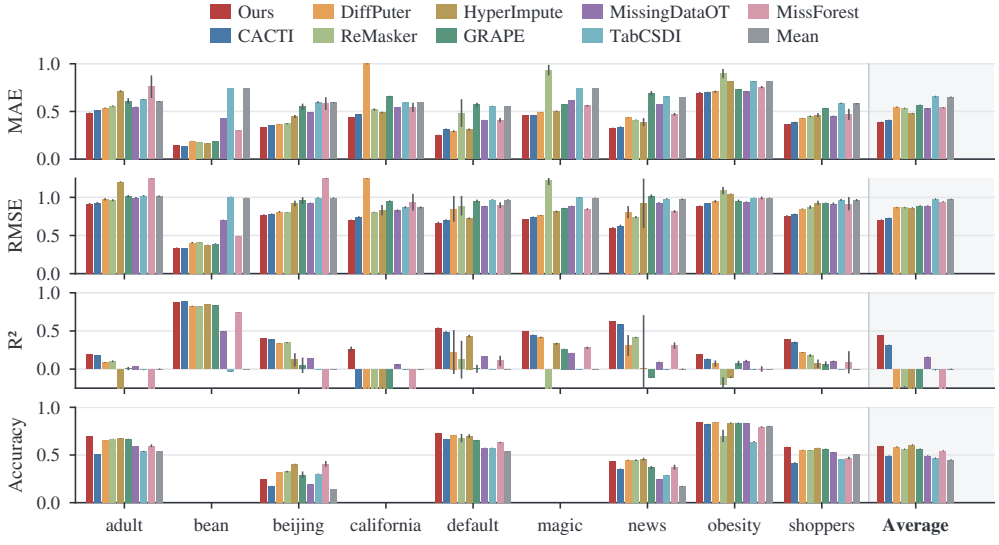


Figure 9: Per-dataset results for MCAR (50%) out-of-sample.

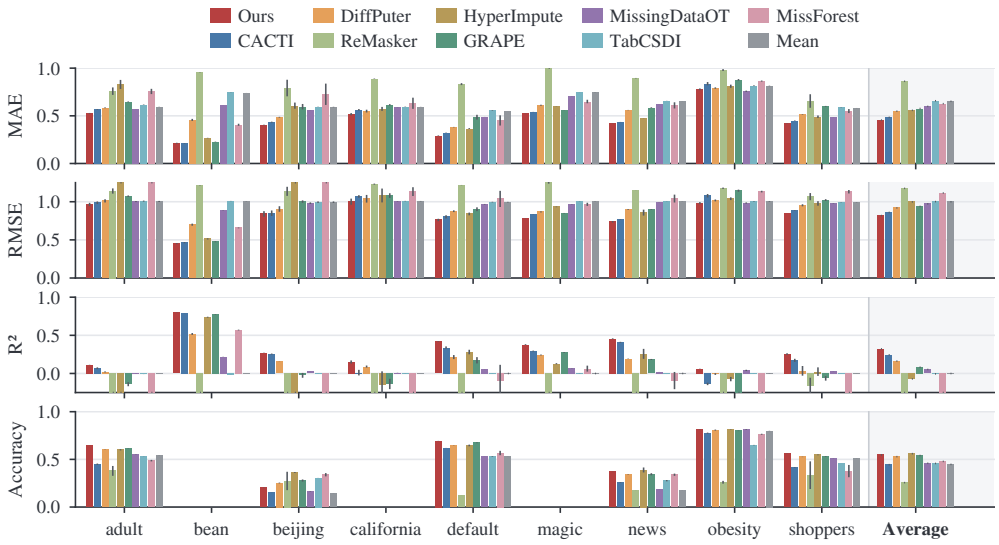


Figure 10: Per-dataset results for MCAR (70%) in-sample.

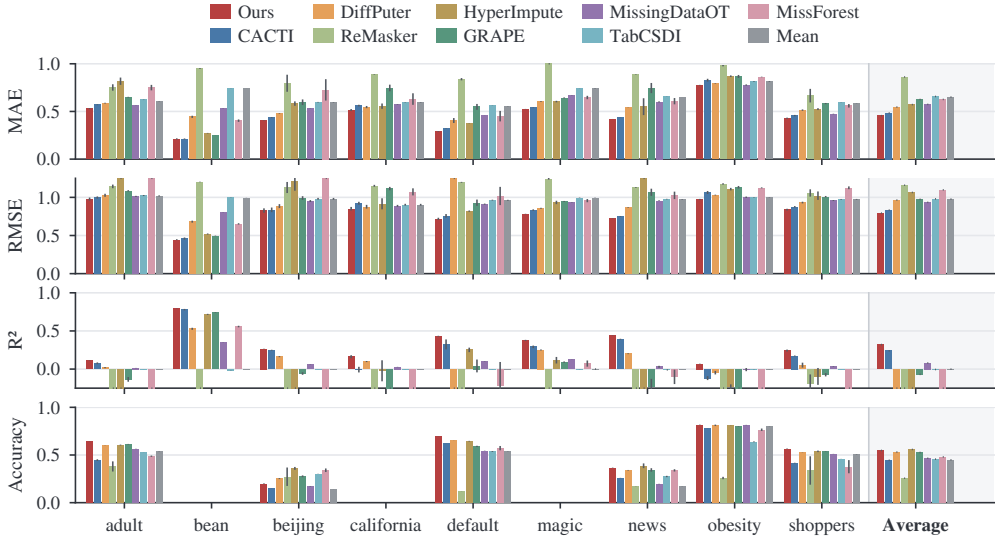


Figure 11: Per-dataset results for MCAR (70%) out-of-sample.

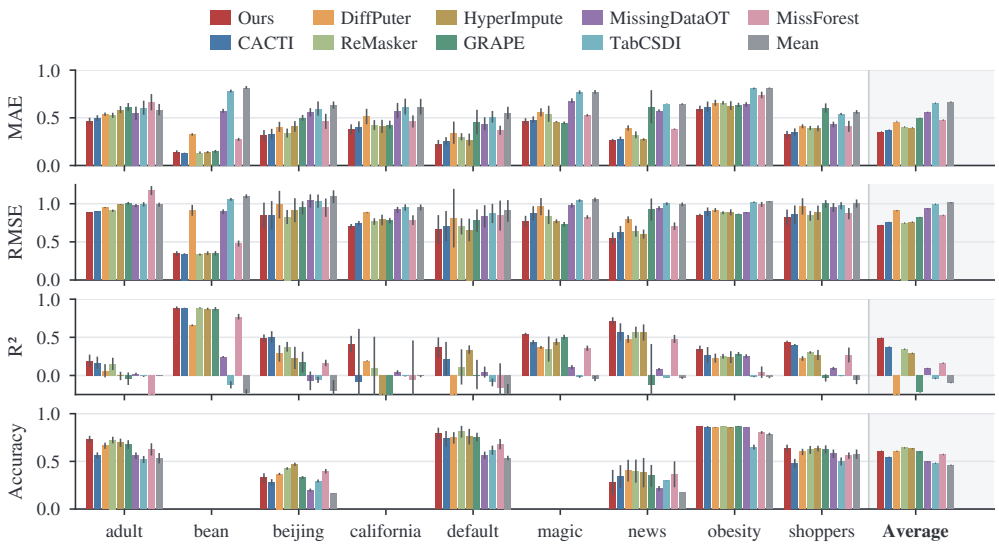


Figure 12: Per-dataset results for MAR (30%) in-sample.

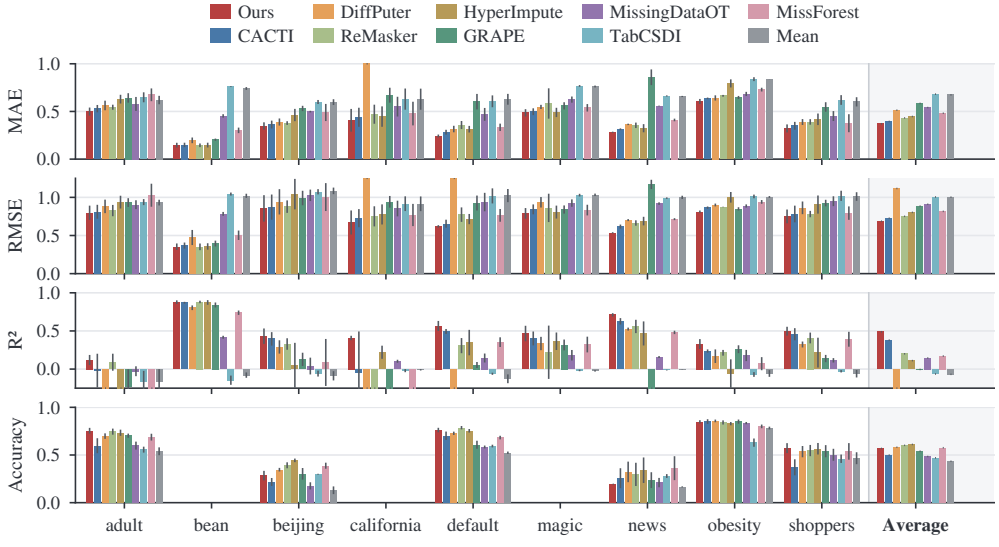


Figure 13: Per-dataset results for MAR (30%) out-of-sample.

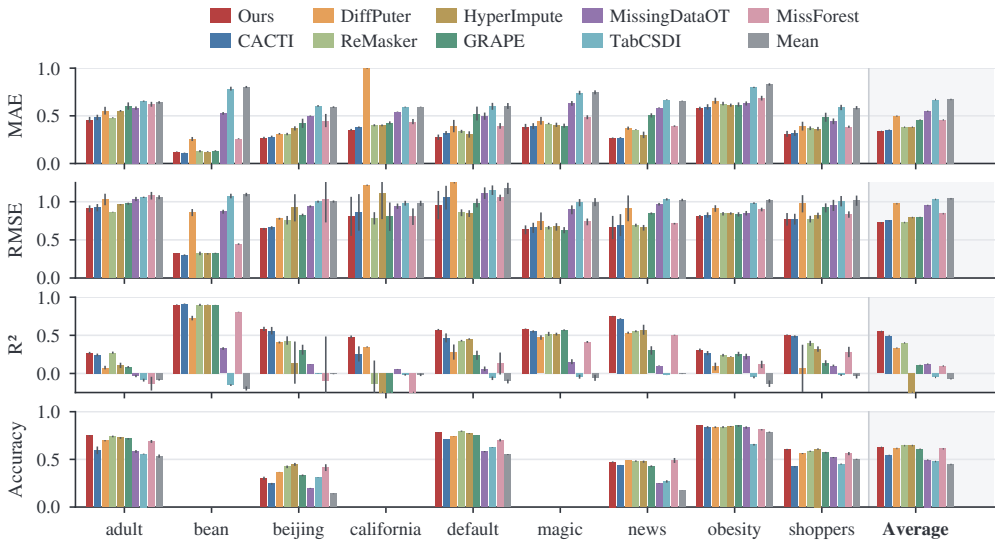


Figure 14: Per-dataset results for MNAR (30%) in-sample.

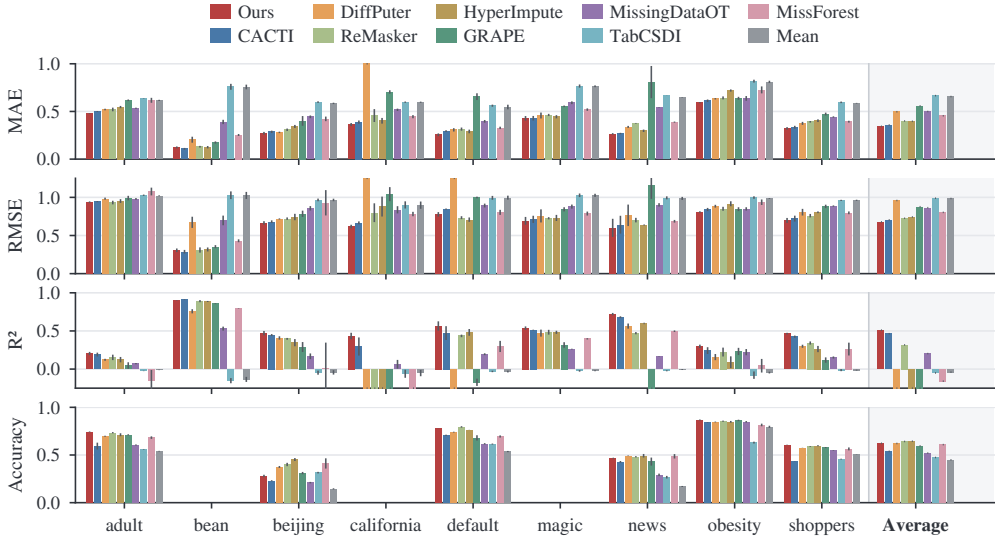


Figure 15: Per-dataset results for MNAR (30%) out-of-sample.