Improvement of Deep Prediction Models via Mutual Information Regularization under Missing Data Distribution Shifts

Jihye Lee

Department of Statistics

The Graduate School of Sungshin Women's University

Improvement of Deep Prediction Models via Mutual Information Regularization under Missing Data Distribution Shifts

A Master's Thesis
Submitted to the
Graduate School of Sungshin Women's University

in partial fulfillment of the requirements
for the degree of
Master of Statistics

Jihye Lee

May, 2025

This is to certify that we have examined the
Master's Thesis of
Jihye Lee
Submitted to Department of Statistics

Approved as to style and content:

Thesis Advisor Dongha Kim

Committee Chairman Man Sik Park

Committee Member Seongoh Park

Committee Member Taehwa Choi

The Graduate School of Sungshin Women's University

Abstract

In real-world data analysis, missingness distributional shifts between training and test input datasets frequently occur, posing a significant challenge to achieving robust prediction performance. In this study, we propose a novel deep learning framework designed to address such shifts in missingness distributions. We begin by introducing a set of mutual information-based conditions, called *MI robustness conditions*, which guide a prediction model to extract label-relevant information while remaining invariant to diverse missingness patterns, thereby enhancing robustness to unseen missingness scenarios at test-time. To make these conditions practical, we propose simple yet effective techniques to derive loss terms corresponding to each and formulate a final objective function, termed *MIRRAMS (Mutual Information Regularization for Robustness Against Missingness Shifts)*. As a by-product, our analysis provides a theoretical interpretation of the principles underlying consistency regularization-based semi-supervised learning methods, such as FixMatch (Sohn et al., 2020). Extensive experiments across various benchmark datasets show that MIRRAMS consistently outperforms existing baselines and maintains stable performance across diverse missingness scenarios. Moreover, our approach achieves state-of-the-art performance even without missing data and can be naturally extended to address semi-supervised learning tasks, highlighting MIRRAMS as a powerful, off-the-shelf framework for general-purpose learning.

Keywords : Missing Data Distribution Shift, Deep Learning, Mutual Information

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In real-world scenarios, it is quite common for the missing data distribution to differ between the training and test datasets. This often arises due to changes in data collection environments, population shifts, or operational procedures. For example, hospitals may record patient variables differently (Gutiérrez-Sacristán-Isaac and Avillach; Zhou et al., 2023; Stokes et al., 2025) and financial regulations can affect how information is reported (Qian et al., 2022). In addition, survey respondent demographics may change over time (Meterko et al., 2015), all of which can lead to shifts in the missing data distribution.

Such shifts in missing data distributions degrade model performance, as patterns learned from the observed training data may no longer be effective in predicting the test data (Zhou et al., 2023; Gardner et al., 2023). This issue is particularly critical in tabular data, where each variable often carries distinct semantic meaning (Kim et al., 2023; Singh and Bedathur, 2023), making models more sensitive to missingness shifts than in unstructured domains such as image or text. Although this is a practically important and pressing challenge, and despite the recent development of numerous deep-learning-based architectures for tabular data (Bahri et al., 2021; Chen et al., 2023; Wu et al., 2024), to the best of the authors' knowledge, there has been limited research in the deep learning domain specifically aimed at addressing this problem.

In this paper, we propose a new deep learning framework for training prediction models that are robust to shifts in missing data distributions. To this end, we introduce a set of mutual information-based conditions, referred to as *MI robustness conditions*, designed to guide the prediction model in extracting label-relevant information from inputs, regardless

of missingness patterns in both the training and test data. However, since the missingness distribution at test-time is unseen and mutual information is generally intractable to compute, it is challenging to directly enforce the model to satisfy the MI robustness conditions.

To tackle these challenges, we introduce a couple of novel techniques that are simple yet highly effective. First, we first impose additional masking to training data to simulate diverse missingness scenarios and thereby better cover the potential support of the missingness distribution at test-time. And to encourage that the prediction model satisfies the MI robustness conditions, we propose alternative cross-entropy-based regularization terms. We theoretically show that minimizing each term guarantees the satisfaction of the corresponding component of the MI robustness conditions.

Interestingly, while developing the new regularization terms, we discover that one of our proposed terms closely resembles the consistency regularization term originally introduced by FixMatch (Sohn et al., 2020), one of the most widely used approaches for addressing semi-supervised learning (SSL) tasks. Therefore, as a by-product, our derivation provides a theoretical justification for the use of FixMatch and its subsequent variants. Moreover, this analysis offers a natural extension of our method to SSL.

By combining all the regularization terms, we formulate a new objective function and call our method *MIRRAMS (Mutual Information Regularization for Robustness Against Missingness Shifts)*. While MIRRAMS is primarily designed for tabular data domains, it can be readily extended to unstructured data types such as images and texts. We present Figure 1 to visualize our method.

We conduct extensive experiments by analyzing various benchmark datasets including performance comparison and ablation studies, and demonstrate that prediction models trained with our method exhibit robust performance, compared to existing state-of-the-art learning frameworks designed to handle missing values, across a variety of scenarios where the missing data distributions differ between training and test datasets. Furthermore, our

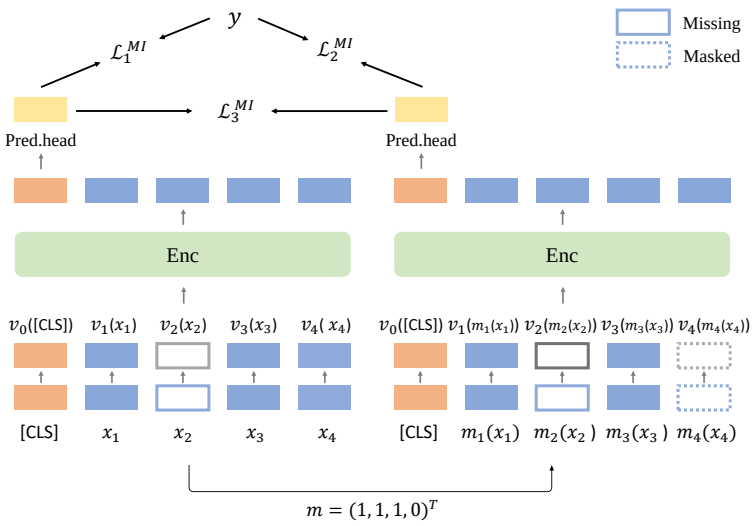

Figure 1: An illustration of the implementation of MIRRAMS on data with $p = 4$.

results indicate that MIRRAMS achieves superior performance even when no missing data are present, demonstrating its effectiveness in standard supervised learning settings. In addition, as mentioned above, our method also shows strong predictive performance in SSL settings.

The remainder of this paper is organized as follows. Section 2 provides a brief summary of recent deep learning studies related to our proposed method. Section 3 introduces our learning framework, MIRRAMS, along with a detailed explanation of its loss function. In Section 4, we present experimental analyses, including prediction performance comparisons across diverse missingness scenarios and ablation studies. Lastly, Section 5 concludes the paper with a summary of our findings and directions for future research. The key contributions of our work are:

- We propose mutual-information-based conditions, called *MI robustness conditions*, to guide prediction models to output robust predictions under missingness distribution shifts.

- We introduce a new, theoretically grounded learning framework, *MIRRAMS*, designed to encourage models to satisfy the robustness conditions.
- We empirically demonstrate the superiority of MIRRAMS across various scenarios with missingness distribution shifts, using widely used benchmark datasets.

Chapter 2

Background

Missing Data Handling To the best of the authors' knowledge, no existing deep learning study explicitly considers missingness distribution shifts; therefore, in this section, we focus on related research that addresses data with missing values. Roughly, existing deep learning-based approaches can be categorized into two groups: (1) imputing missing values and (2) treating missing values as masks.

A major number of studies adopt the conventional imputation strategies. For example, FT-Transformer(Gorishniy et al., 2021), SCARF (Bahri et al., 2021), ReConTab (Chen et al., 2023), XTab (Zhu et al., 2023) and SwitchTab (Wu et al., 2024), replace missing continuous features with the mean of observed values and categorical features with the most frequent category or encoded versions (e.g., one-hot or ordinal), and feed the fully imputed inputs into their networks. TabPFN (Hollmann et al., 2022) applies a simple rule by replacing missing entries with zero, although this may introduce semantic confusion when zero holds meaning. While widely used, such imputation techniques are considered naive, as they distort feature relationships and ignore the underlying mechanism of missingness, which may eventually lead to performance degradation.

In an alternative approach, a growing number of studies treat missing values as learnable mask tokens. Specifically, TabTransformer (Huang et al., 2020) defines a special embedding token for missing categorical values, used in place of missing entries during training and testing. However, this token is often under-fitted when missingness is sparse, and average embeddings are used instead. SAINT (Somepalli et al., 2021) extends this idea by

constructing a dedicated embedding table for missing values and learning their representations explicitly with binary missing indicators. This masking-based approach enables the model to condition its representations on missingness itself.

Additionally, VIME (Yoon et al., 2020) and UniTabE (Yang et al., 2023) introduce additional missingness during training by applying random masking and employing self-supervised learning to prevent overfitting and enhance robustness to missing values at test-time. PMAE (Kim et al., 2025) addresses missing data by imputing values using masked autoencoders (He et al., 2022) with a proportional masking strategy. However, they do not explicitly consider shifts in the missingness distribution between training and test data.

Mutual Information As a measure of the information shared by two random variables, mutual information (MI) quantifies the degree of dependence between them. It is defined as the Kullback–Leibler divergence between the joint distribution and the product of marginals: $I(X;Y) = D_{\mathrm{KL}}\big(P(X,Y) \,\|\, P(X)P(Y)\big)$. A large MI value implies that the joint distribution $P(X,Y)$ and the product of marginals $P(X)P(Y)$ differ substantially—i.e., X and Y are strongly dependent.

Many studies have leveraged MI to build high-performance prediction models. For example, the InfoMax principle (Hjelm et al., 2018) maximizes the MI between the target and the augmented input view to achieve strong performance. However, this approach can excessively capture unnecessary information, which may degrade predictive performance when the model encounters unseen data. To address this limitation, Tian et al. (2020) proposed the InfoMin principle, which minimizes the MI between augmented input views while ensuring that each view retains sufficient information about the target. This process removes noise and redundant shared information, thereby preventing performance degradation across diverse data scenarios.

Chapter 3

Proposed Method

3.1 Notations and Definitions

We introduce notations and definitions frequently used throughout this paper. Let $(\boldsymbol{X}, Y) \in \mathcal{X} \times [K]$ be a pair of fully observed input and corresponding class random variables where $\mathcal{X}$ represents the input space composed of p variables, which may include both continuous and categorical ones, and $[K] := \{1, \ldots, K\}$ denotes the set of class labels. We denote the true joint distribution of the fully observed input-output pairs by $\mathbb{P}$. The missingness distributions for the training and test input data are denoted by M^{tr} and M^{ts}, respectively. We assume that the true data distribution $\mathbb{P}$, as well as the missingness distributions M^{tr} and M^{ts}, are all unknown.

The observed training dataset is given by $\mathcal{D}^{\text{tr}} := \{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$, where each pair $(\boldsymbol{x}_i, y_i)$ is drawn independently from $M^{\text{tr}} \circ \mathbb{P}$. We also define the set of training inputs from $\mathcal{D}^{\text{tr}}$ as $\mathcal{D}_x^{\text{tr}}$, i.e., $\mathcal{D}_x^{\text{tr}} := \{\boldsymbol{x}_i\}_{i=1}^n$. Similarly, we assume that each input-output pair in the *unseen* test dataset is independently drawn from the distribution $M^{\text{ts}} \circ \mathbb{P}$. When the context is clear, we use the one-hot encoded representation of the label, that is, $y \in \{0, 1\}^K$, interchangeably with its scalar form. In addition, for simplicity, we use the dataset notation interchangeably with its corresponding empirical distribution whenever there is no confusion.

We define an additional masking distribution as $M_{\boldsymbol{r}}^{\text{m}}$, where $\boldsymbol{r} = (r_1, \ldots, r_p)^T \in (0, 1)^p$ denotes a masking ratio vector applied to each variable. Unless otherwise specified, we assume that M_r^{m} follows a product of p independent Bernoulli distributions with a com-

mon ratio r, that is, $M_r^{\mathrm{m}} = Ber(r) \otimes \cdots \otimes Ber(r)$, where a value of one indicates that the corresponding variable is observed, and zero indicates that it is masked. An instance of a masking vector is denoted by $\boldsymbol{m} \in \{0,1\}^p$.

For a given input $\boldsymbol{x} = (x_1, \ldots, x_p)^T$, $v(\boldsymbol{x}) = (v_0([\texttt{CLS}]), v_1(x_1), \ldots, v_p(x_p))^T \in \mathbb{R}^{(p+1)\cdot d}$ is defined as a concatenated embedding vector, where $[\texttt{CLS}]$ denotes the classification token and $v_0 \in \mathbb{R}^d$ is its corresponding embedding vector. Each $v_j(x_j) \in \mathbb{R}^d, j \in [p]$ represents the embedding function of the j-th input variable. Following (Somepalli et al., 2021), we implement each v_j using a multilayer perceptron (MLP) when x_j is a continuous variable, and a learnable token embedding vector when x_j is categorical. In addition, we introduce a separate token embedding vector for each variable to explicitly represent its missingness. All MLPs and token embeddings are learned during training.

Based on the concatenated embedding vector, we employ a transformer-based encoder denoted as $enc : \mathbb{R}^{(p+1)\cdot d} \to \mathbb{R}^{(p+1)\cdot d}$. We take the encoder output corresponding to the classification token $v_0([\texttt{CLS}])$ as the final representation of the input $\boldsymbol{x}$ for prediction. We employ a prediction head $g : \mathbb{R}^d \to \mathcal{S}^K$ on top of the representation vector, where $\mathcal{S}^K$ denotes the K-dimensional simplex. The overall prediction model for a given input $\boldsymbol{x}$ is then defined as $f(\boldsymbol{x}) := g \circ enc \circ v(\boldsymbol{x})$. We classify the predicted label of a given input as the index corresponding to the highest prediction score from f, i.e., $\hat{y}(\boldsymbol{x}) := \text{argmax}_{k\in[K]} f_k(\boldsymbol{x})$. The confidence of the prediction is defined as the highest predicted score: $\hat{c}(\boldsymbol{x}) := \max_{k\in[K]} f_k(\boldsymbol{x})$. If $\boldsymbol{x}$ contains missing values with a masking instance $\boldsymbol{m}$, its prediction is then represented as $f(m(\boldsymbol{x}))$.

3.2 Mutual-Information-Based Robustness Conditions

Our goal is to train a prediction model f that achieves high performance on unseen test data corrupted with an unknown missingness distribution. To this end, we employ the mutual information measure and introduce the following conditions–termed *MI robustness conditions*–formulated as:

$$\begin{array}{ll} I\left(M^{\text{tr}}(\boldsymbol{X}); Y\right) \underset{(i)}{\approx} I\left(f(M^{\text{tr}}(\boldsymbol{X})); Y\right) & \underset{(iii)}{\gtrless} \\ & I\left(f(M^{\text{tr}}(\boldsymbol{X})); f(M^{\text{ts}}(\boldsymbol{X}))\right) \\ I\left(M^{\text{ts}}(\boldsymbol{X}); Y\right) \underset{(ii)}{\approx} I\left(f(M^{\text{ts}}(\boldsymbol{X})); Y\right) & \underset{(iv)}{\gtrless} \end{array} \tag{3.1}$$

The detailed explanation of the MI robustness conditions is as follows. The function f should preserve information regarding each input's label distribution under the missingness distribution both in the training and test datasets ((i) and (ii) in (3.1)). And the prediction function should discard unnecessary information that is not relevant to predicting the label ((iii) and (iv) in (3.1)). Together, these conditions guide the model to extract only essential label-related information while remaining robust to various missingness patterns.

However, the above conditions in (3.1) are intractable to enforce directly, as the test-time missingness distribution, M^{ts}, is unknown–unlike M^{tr}, for which we have access to training data drawn from $M^{\text{tr}} \circ \mathbb{P}$. To tackle this issue, we construct an enlarged missingness distribution by applying an additional masking operation M_r^{m} to the training distribution M^{tr}, resulting in $M_r^{\text{m}} \circ M^{\text{tr}}$, and use it as a surrogate for the test-time missingness distribution. By applying this approximation to (3.1), we derive the following modified MI robustness conditions.

$$
\begin{array}{ccc}
I\left(M^{\mathrm{tr}}(\boldsymbol{X}); Y\right) \underset{(i)}{\approx} I\left(f(M^{\mathrm{tr}}(\boldsymbol{X})); Y\right) & & \\
& \overset{(iii)}{\wr} & I\left(f(M^{\mathrm{tr}}(\boldsymbol{X})); f(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X}))\right) \\
I\left(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X}); Y\right) \underset{(ii)}{\approx} I\left(f(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X})); Y\right) & \overset{(iv)}{\wr} &
\end{array}
\tag{3.2}
$$

Suppose that the support of the surrogate missingness distribution is sufficiently broad to cover that of the unknown test missingness distribution. Under this assumption, the prediction model f can ensure that satisfying conditions (ii) and (iv) in (3.2) implies the satisfaction of the corresponding conditions in (3.1). This, in turn, indicates that the resulting model is robust to distributional shifts in missingness. We also note that a careful choice of the masking ratio r is crucial, as excessively high masking ratios may cause a significant discrepancy between $I\left(M^{\mathrm{tr}}(\boldsymbol{X}); Y\right)$ and $I\left(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X}); Y\right)$, which leads to a violation of (3.2) and results in performance degradation. We empirically validate this observation in Section 4.

Remark 1. *The reference (Tian et al., 2020) first utilized mutual information to propose a criterion for designing encoders that yield optimal representations for downstream prediction tasks. They developed a criterion known as the InfoMin principle. Since the exact computation of each mutual information term is intractable, the authors replace it with the InfoNCE loss (Oord et al., 2018), which is widely used in contrastive learning tasks. This approach leverages the fact that the InfoNCE loss provides a lower bound on the mutual information between two random variables. In contrast, we propose an entirely different and effective approach to satisfy the mutual information-based conditions, thereby enabling efficient training of a high-performing prediction model tailored to our setting.*

3.3 Derivation of Loss Functions in MIRRAMS

In this section, we propose new loss functions to encourage the prediction model f to satisfy the *MI robustness conditions* in (3.2). Assuming that $I\left(M^{\text{tr}}(\boldsymbol{X}); Y\right) \approx I\left(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}); Y\right)$ with a carefully chosen masking distribution M_r^{m}, it is sufficient introduce three loss terms, each corresponding to conditions (i), (ii), and (iii) in (3.2).

Conditions (i)&(ii) For the condition (i), since even approximating $I\left(M^{\text{tr}}(\boldsymbol{X}); Y\right)$ is intractable, directly enforcing similarity between the two mutual information values is not feasible. To address this, we use the conditional entropy between Y and $f(M^{\text{tr}}(\boldsymbol{X}))$ as a surrogate loss term, given as:

$$H(Y|f(M^{\text{tr}}(\boldsymbol{X}))) := \mathbb{E}_{M^{\text{tr}}(\boldsymbol{X}),Y}\left[-\log P\left(Y|f(M^{\text{tr}}(\boldsymbol{X}))\right)\right], \tag{3.3}$$

The following proposition provides justification for using the conditional entropy to satisfy the condition (i). The proof is deferred to the Appendix A.

Proposition 1. *Consider a pair of two random variables (U, V) and a function ξ. Suppose there exists a positive constant $\epsilon > 0$ such that*

$$D_{KL}\left(P(V|U=u)||P(V|\xi(U)=\xi(u))\right) < \epsilon$$

for all $u \in$ supp(U), where $D_{KL}(\cdot||\cdot)$ denotes the KL divergence. Then, the following holds:

$$|I(U;V) - I(\xi(U);V)| < \epsilon.$$

We now present the implication of Proposition 1. To begin with, the conditional entropy

in (3.3) is given by:

$$\begin{aligned} H(Y|f(M^{\mathrm{tr}}(\boldsymbol{X}))) &= \mathbb{E}_{M^{\mathrm{tr},}(\boldsymbol{X}),Y}\left[-\log P(Y|f(M^{\mathrm{tr}}(\boldsymbol{X})))\right] \\ &= \mathbb{E}_{M^{\mathrm{tr}}(\boldsymbol{X}),Y}\left[\log \frac{P(Y|M^{\mathrm{tr}}(\boldsymbol{X}))}{P(Y|f(M^{\mathrm{tr}}(\boldsymbol{X})))} - \log(P(Y|M^{\mathrm{tr}}(\boldsymbol{X})))\right] \\ &= \mathbb{E}_{M^{\mathrm{tr}}(\boldsymbol{X})}\left[D_{KL}\left(P(Y|M^{\mathrm{tr}}(\boldsymbol{X}))||P(Y|f(M^{\mathrm{tr}}(\boldsymbol{X})))\right)\right] \\ &\quad + \mathbb{E}_{M^{\mathrm{tr}}(\boldsymbol{X}),Y}\left[-\log(P(Y|M^{\mathrm{tr}}(\boldsymbol{X})))\right] \end{aligned}$$

It is reformulated as:

$$H(Y|f(M^{\mathrm{tr}}(\boldsymbol{X}))) = \mathbb{E}_{M^{\mathrm{tr}}(\boldsymbol{X})}\left[D_{KL}\left(P(Y|M^{\mathrm{tr}}(\boldsymbol{X}))||P(Y|f(M^{\mathrm{tr}}(\boldsymbol{X})))\right)\right] + H(Y|M^{\mathrm{tr}}(\boldsymbol{X})).$$

Thus, minimizing (3.3) with respect to f corresponds to finding a function f such that the conditional distribution $P(Y|f(M^{\mathrm{tr}}(\boldsymbol{x})))$ closely approximates $P(Y|M^{\mathrm{tr}}(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathcal{X}$, in terms of KL divergence. Then, by Proposition 1, such a function f satisfies the condition (i) in (3.2). To implement (3.3) in practice, we approximate it using the empirical average of the cross-entropy over the training dataset, formulated as:

$$\mathcal{L}_1^{\mathrm{MI}} := \mathbb{E}_{(\tilde{\boldsymbol{X}},Y)\sim\mathcal{D}^{\mathrm{tr}}}\left[-\log P\left(Y|f(\tilde{\boldsymbol{X}})\right)\right]. \tag{3.4}$$

The fulfillment of the condition (ii) can be achieved in a manner similar to the condition (i), by minimizing the conditional entropy $H(Y|f(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X})))$, which can also be approximated by the empirical average of the cross-entropy loss, given as:

$$\mathcal{L}_2^{\mathrm{MI}} := \mathbb{E}_{(\tilde{\boldsymbol{X}},Y)\sim\mathcal{D}^{\mathrm{tr}}}\mathbb{E}_{M_r^{\mathrm{m}}}\left[-\log P\left(Y|f(M_r^{\mathrm{m}}(\tilde{\boldsymbol{X}}))\right)\right]. \tag{3.5}$$

Condition (iii) Lastly, we focus on the condition (iii) in (3.2). We note that the right-hand side of the condition (iii) can be decomposed as:

$$I\left(f(M^{\text{tr}}(\boldsymbol{X})); f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right) = H(f(M^{\text{tr}}(\boldsymbol{X}))) - H\left(f(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right). \tag{3.6}$$

Suppose that the training procedure has *sufficiently progressed* such that $f(M^{\text{tr}}(\boldsymbol{x}))$ captures nearly all the information relevant to its corresponding label, y. That implies $f(M^{\text{tr}}(\boldsymbol{x}))$ becomes nearly a one-hot encoded y, and therefore the information contained in $f(M^{\text{tr}}(\boldsymbol{X}))$ and $\hat{y}(M^{\text{tr}}(\boldsymbol{X}))$ becomes almost same. As a result, the mutual information in (3.6) can be approximated by

$$\begin{aligned} I\left(f(M^{\text{tr}}(\boldsymbol{X})); f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right) &\approx H(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))) - H\left(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right) \\ &\approx H(Y) - H\left(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right). \end{aligned}$$

Therefore, using the fact that $I\left(f(M^{\text{tr}}(\boldsymbol{X})); Y\right) = H(Y) - H(Y|f(M^{\text{tr}}(\boldsymbol{X})))$, the condition (iii) in (3.2) reduces to the following condition:

$$\begin{aligned} H(Y) - H(Y|f(M^{\text{tr}}(\boldsymbol{X}))) &\approx H(Y) - H\left(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right) \\ H(Y|f(M^{\text{tr}}(\boldsymbol{X}))) &\approx H\left(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right). \end{aligned} \tag{3.7}$$

Since encouraging condition (i) leads to minimization of $H(Y|f(M^{\text{tr}}(\boldsymbol{X})))$, the condition in (3.7) is fulfilled when $H\left(\hat{y}(M^{\text{tr}}(\boldsymbol{X}))|f(M_r^{\text{m}} \circ M^{\text{tr}}(\boldsymbol{X}))\right)$ is minimized as well. Consequently, we define the following loss term:

$$\mathcal{L}_3^{\text{MI}} := \mathbb{E}_{\tilde{\boldsymbol{X}} \sim \mathcal{D}_x^{\text{tr}}} \mathbb{E}_{M_r^{\text{m}}} \left[-\log P\left(\hat{y}(\tilde{\boldsymbol{X}})|f(M_r^{\text{m}}(\tilde{\boldsymbol{X}}))\right) \cdot \mathbb{I}(\hat{c}(\tilde{\boldsymbol{X}}) \geq \tau) \right], \tag{3.8}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and $\tau \in (0, 1)$ is a hyperparameter called confidence threshold.

Minimizing (3.8) encourages the minimization of $H\left(\hat{y}(M^{\mathrm{tr}}(\boldsymbol{X}))|f(M_{\boldsymbol{r}}^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X}))\right)$, but this holds only when the overall predictions are made with high confidence. Such confidence typically emerges when the model f is *sufficiently trained*, which is consistent with the the assumptions under which condition in (3.7) is derived.

Final Loss function of MIRRAMS By combining the loss terms defined in (3.4), (3.5), and (3.8), the final objective function of our proposed method is formulated as:

$$\mathcal{L} := \mathcal{L}_1^{\mathrm{MI}} + \lambda_1 \cdot \mathcal{L}_2^{\mathrm{MI}} + \lambda_2 \cdot \mathcal{L}_3^{\mathrm{MI}}, \tag{3.9}$$

where $\lambda_1, \lambda_2 > 0$ are hyperparameters. We estimate the parameters of the prediction model f by minimizing the loss $\mathcal{L}$ using a gradient-descent-based optimizer, such as Adam (Kingma and Ba, 2014). In practice, the optimal combination of hyperparameters is selected using a validation dataset.

3.4 Rethinking Consistency Regularization-Based SSL Solvers

As a by-product, based on the derivation of loss functions in Section 3.3, we provide a theoretical perspective on SSL methods that employ consistency regularization. Consistency regularization is a core principle in SSL domain, which encourages a prediction model to produce similar predictions for the same input under different perturbations or views. One of the most prominent approaches built on this principle is FixMatch (Sohn et al., 2020), whose objective function is formulated as:

$$\mathcal{L}^{\mathrm{CE}} + \lambda \cdot \mathcal{L}^{\mathrm{FM}},$$

where

$$\mathcal{L}^{\mathrm{CE}} := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{L}^{\mathrm{tr}}}\mathbb{E}_{\alpha}\left[-\log P\left(Y|f(\alpha(\boldsymbol{X}))\right)\right],$$

$$\mathcal{L}^{\mathrm{FM}} := \mathbb{E}_{\boldsymbol{X}\sim\mathcal{U}^{\mathrm{tr}}}\mathbb{E}_{\alpha,\mathcal{A}}\left[-\log P\left(\hat{y}(\alpha(\boldsymbol{X}))|f(\mathcal{A}(\boldsymbol{X}))\right)\cdot\mathbb{I}(\hat{c}(\alpha(\boldsymbol{X}))\geq\tau)\right],$$

$\mathcal{L}^{\mathrm{tr}}$ and $\mathcal{U}^{\mathrm{tr}}$ denote the empirical distributions of the labeled and unlabeled training data, respectively, and $\lambda > 0, \tau \in (0,1)$ are two hyperparameters. Here, α and $\mathcal{A}$ represent weak and strong augmentations, respectively. In FixMatch (Sohn et al., 2020), the authors intuitively interpret minimizing this term as encouraging the model f to correctly classify labeled data while simultaneously producing consistent predictions under strong augmentations with those under weak ones, conditioned on high-confidence predictions from the weakly augmented inputs.

Based on our study, this loss function can be interpreted as guiding the training of the prediction model f to satisfy the following mutual information conditions:

$$I(Y;\alpha(\boldsymbol{X})) \underset{(i)}{\approx} I(Y;f(\alpha(\boldsymbol{X}))) \underset{(ii)}{\approx} I(f(\alpha(\boldsymbol{X}));f(\mathcal{A}(\boldsymbol{X}))). \tag{3.10}$$

Condition (i) in (3.10) encourages a prediction model f to extract label-relevant information, while condition (ii) discourages f from encoding redundant information that is not useful for label prediction, regardless of the type of augmentation applied. According to the theoretical analysis in Section 3.3, minimizing $\mathcal{L}^{\mathrm{CE}}$ and $\mathcal{L}^{\mathrm{FM}}$ respectively promotes the fulfillment of conditions (i) and (ii) in (3.10). Therefore, FixMatch trains a prediction model to extract label-relevant information while suppressing redundant features that do not contribute to label prediction, even under strong augmentations. This slightly different interpretation offers a more theoretically grounded explanation for the practical effectiveness of

FixMatch, as well as for subsequent SSL methods inspired by it, such as those proposed in (Xu et al., 2021; Zhang et al., 2021; Guo and Li, 2022; Wang et al., 2023).

Chapter 4

Experiments

We validate the effectiveness of our method under distributional shifts in missingness patterns between training and test datasets. Additionally, we empirically demonstrate that our method can be easily extended to SSL settings. For each experiment, we report the average performance over three trials with different random parameter initializations. Our implementation is based on the `PyTorch` framework and runs on two NVIDIA GeForce RTX 3090 GPUs.

Dataset Description We conduct experiments on ten widely used benchmark datasets, all of which are publicly available from either the UCI repository or AutoML benchmarks. Each dataset is split into training, validation, and test sets with a ratio pair of $(0.65, 0.15, 0.2)$. Detailed information about the datasets is provided in the Appendix B.

To simulate distributional shifts of missingness patterns between the training and test datasets, we use a $p \times p$ AR(1)-type covariance matrix with a correlation factor $\rho \in (0, 1)$ (In our experiments, we fix $\rho = 0.7$.). We then apply two different permutation operation to this base covariance matrix to construct two multivariate Gaussian distributions with zero mean and permuted covariance structures. These distributions are used to generate missingness patterns for the training and test data, respectively.

For each sample, we draw a noise vector from the corresponding Gaussian multivariate distribution and binarize its elements using a pre-specified α-quantile threshold of the standard Gaussian distribution to construct the missingness pattern, such that the marginal missing ratio for each variable is α. We vary α across both training and test datasets, specif-

ically setting $\alpha^{\text{tr}} \in \{0, 0.05, 0.1, 0.2\}$ for the training data and $\alpha^{\text{ts}} \in \{0, 0.1, 0.2, 0.3\}$ for the test data. We note that the case of $(\alpha^{\text{tr}}, \alpha^{\text{ts}}) = (0, 0)$ corresponds to the standard supervised learning setting, where neither the training nor the test data contain any missing values. Obviously, we apply the same missingness patterns to the validation dataset as in the training dataset.

Implementation Details We follow a similar architecture to that proposed by (Somepalli et al., 2021). Specifically, for the embedding layers, we use an MLP for each continuous variable and learnable tokenized vectors for each categorical variable. And we also adopt the transformer architecture (Vaswani et al., 2017) for the encoder and a separate MLP for the prediction head. Detailed descriptions of the architectures we employ are stated in Appendix B.

Our method involves four hyperparameters: $(r, \lambda_1, \lambda_2, \tau)$. We consider values $r \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$, $\lambda_1, \lambda_2 \in \{0, 1, 5, 10, 15, 20\}$, and $\tau \in \{0.8, 0.9, 0.95, 0.99\}$, and select the best combination based on performance on the validation dataset. For training, we use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-4}, a mini-batch size of 256, and up to 1,000 epochs.

4.1 Performance results

In this section, we demonstrate that our method trains prediction models that are robust to various cases of missingness distributional shifts. For comparison, we first consider five classical machine learning methods: (1) logistic regression with zero-value imputation (LR-0), (2) vanilla MLP with zero imputation (NN-0), (3) Random Forest (RF (Breiman, 2001)), (4) XGBoost (XGB (Chen and Guestrin, 2016)), and (5) CatBoost (CB (Prokhorenkova et al., 2018)). And we also consider three recent deep learning-based methods: (6) SAINT (Somepalli et al., 2021), (7) TabTransformer (TabTF, (Huang et al., 2020)),

and (8) SwitchTab (STab (Wu et al., 2024)). We implement the machine learning baselines using official `Python` packages, including `xgboost` and `scikit-learn`. For the deep learning-based methods, we implement them based on their publicly available `GitHub` repositories.

For each setting, we evaluate the averaged test AUC score of our proposed method across ten datasets and compare it against the baseline methods. The averaged results are summarized in Table 1, and the detailed per-dataset results, including standard deviations, are presented in Appendix B. We observe that our method consistently and significantly outperforms all baselines across all considered scenarios. Moreover, it yields stable AUC performance even as the missingness ratio in the test dataset increases, whereas other methods exhibit a sharp performance drop. These results highlight that our method enables prediction models to extract label-relevant information while suppressing the influence of missingness patterns, thereby enhancing robustness to distributional shifts in missingness.

Interestingly, our method also achieves improved performance in the standard supervised learning scenario where neither the training nor the test data contain missing values, i.e., $(\alpha^{\text{tr}}, \alpha^{\text{ts}}) = (0, 0)$. This observation can be attributed to the fact that imposing additional masking during training serves as an implicit form of regularization, preventing the prediction model from extracting excessive information that may lead to overfitting.

These results suggest that our approach is not only effective under distributional shifts in missingness, but also performs well in conventional settings where no missing data are present in either the training or test datasets. Furthermore, while other deep learning-based methods employ auxiliary decoder networks to boost performance, our method remains both effective and computationally efficient, as it requires only a prediction model.

Table 1: Comparison of averaged test AUC scores (%) across 10 tabular datasets under various combinations of $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$. All results were obtained using our own implementations. For each setting, the best result is highlighted in bold, and the second-best is underlined.

α^{tr}	0				0.05				0.1				0.2			
α^{ts}	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	72.03	68.43	66.65	64.86	71.99	69.04	67.55	66.40	70.99	68.49	67.37	66.30	71.39	69.38	69.16	68.00
NN-0	77.41	74.52	70.87	68.34	82.03	80.61	78.05	75.69	79.14	78.09	77.07	74.43	79.41	78.67	76.84	76.09
RF	88.19	86.43	85.15	83.12	87.77	87.40	86.71	85.28	87.77	87.22	86.61	85.52	87.90	87.24	84.31	84.47
XGB	91.17	86.41	82.7	78.97	91.55	89.58	88.15	85.65	91.21	89.90	88.03	85.79	89.76	88.81	86.77	85.20
CB	89.84	86.46	84.42	82.00	89.58	87.18	85.61	83.17	88.55	86.26	84.71	82.62	88.37	86.40	84.94	83.37
SAINT	90.03	86.15	82.84	80.01	89.83	87.09	84.33	81.77	89.68	87.72	85.65	83.47	89.91	89.38	87.83	86.25
TabTF	86.83	84.75	82.79	80.36	86.65	84.55	82.76	80.44	86.59	84.27	82.55	80.03	85.82	83.85	82.27	79.97
STab	85.40	81.83	79.64	77.93	84.67	81.61	80.19	78.88	86.16	82.98	81.66	79.17	84.78	81.83	80.20	78.79
Ours	**91.88**	**90.94**	**89.43**	**88.07**	**91.77**	**90.80**	**89.30**	**88.03**	**91.58**	**90.94**	**89.55**	**88.27**	**91.25**	**90.32**	**89.68**	**88.14**

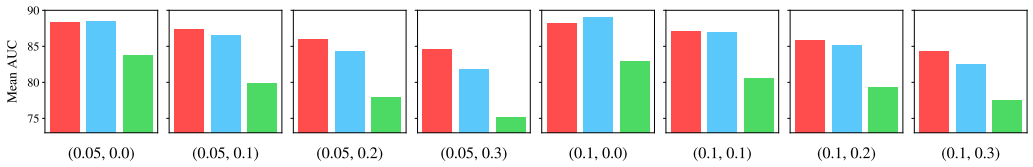

Figure 2: Comparison of averaged test AUC scores (%) across 9 tabular datasets under SSL settings. We compare three methods: (**red**) Ours, (**blue**) SAINT, and (**green**) SwitchTab. Only 10% of the training data is used as labeled data, while the remaining 90% is treated as unlabeled. In each panel, the x-axis denotes the combination of $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$.

4.2 Extension to Semi-Supervised Learning

In the loss function in (3.9), the first and second term, $\mathcal{L}_1^{\text{CE}}$ and $\mathcal{L}_2^{\text{CE}}$, are computed using both inputs and labels, whereas the third term, $\mathcal{L}_3^{\text{CE}}$, is computed solely from inputs. Thus, our approach can be naturally extended to SSL tasks by replacing the expectation over labeled inputs in $\mathcal{L}_3^{\text{CE}}$ with that over unlabeled inputs. In this section, we claim that the modified MIRRAMS to SSL settings maintains strong performance even when only a small amount labeled dataset is available.

Figure 2 presents the averaged AUC scores of our method across nine datasets, compared to two recent deep learning-based approaches across several SSL settings, with varying $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. We exclude `arcene` dataset due to its small sample size (200

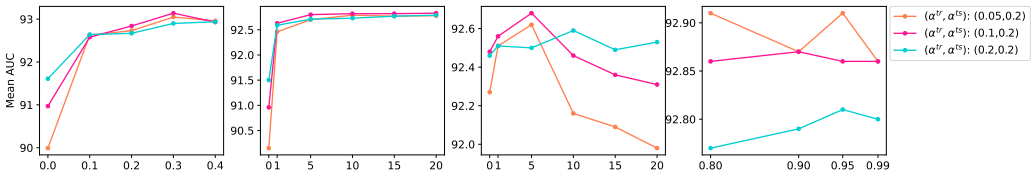

Figure 3: (**From left to right**) Test AUC scores for varying values of r, λ_1, λ_2, and τ, respectively.

samples in total), and the exact AUC values are provided in Appendix B. For each dataset, 10% of training data are used as labeled data, while remaining data treated as unlabeled data.

Our method consistently performs competitively or better than the baselines across all scenarios.

In addition, it maintains stable AUC scores even as the missingness ratio increases, while the performance of other methods degrades sharply. This further demonstrates the robustness of our approach under SSL settings. Together with the results in Section 4.1, these findings demonstrate our method's versatility across both supervised and semi-supervised learing tasks, positioning it as an off-the-shelf solution for a wide range of applications. Given that our extension to SSL is relatively naive and not fully explored, there remains substantial room for improving and adapting MIRRAMS specifically for SSL tasks–an intriguing direction for future research.

4.3 Ablation Studies

We conduct additional experiments to investigate the impact of hyperparameter choices on the performance of MIRRAMS. We consider three datasets–`adult`, `htru2`, and `qsar_bio`–and report the averaged test AUC scores for each setting. For each dataset, we vary a single hyperparameter while keeping the others fixed at their optimal values. A summary of the results is presented in Figure 3, and further details, including running time

analysis, are provided in Appendix B.

- Increasing the value of r generally enhances the performance of our method, but the marginal gains gradually decrease as r becomes larger.
- Regarding λ_1 and λ_2, our method performs well when both are set to positive values, indicating that the proposed loss terms $\mathcal{L}_2^{\text{MI}}$ and $\mathcal{L}_3^{\text{MI}}$ contribute meaningfully to model performance. Their effects, however, exhibit slightly different trends: increasing λ_1 up to 5 leads to consistent performance gains before saturating, while setting $\lambda_2 = 5$ generally yields the best results, with higher values resulting in performance degradation.
- Performance remains similar across all considered values of τ, indicating that our method is not sensitive to its choice as long as a reasonable value is used.

Chapter 5

Concluding remarks

In this paper, we proposed MIRRAMS, a deep learning framework for robust prediction under distributional shifts in missingness patterns. By formulating mutual-information-based robustness conditions and the corresponding loss terms, MIRRAMS guides models to extract label-relevant information robust to unseen missingness patterns. As a by-product, our method provides a theoretical justification for the effectiveness of consistency regularization-based SSL methods, such as FixMatch. Extensive experiments on ten benchmark datasets demonstrated the robustness of MIRRAMS, outperforming recent competitors designed to handle missing values across diverse missingness shift scenarios, and even when no missing data are present in either the training or test set. Additionally, we showed that MIRRAMS can be readily extended to address SSL tasks.

We note that our method does not employ any additional architectures, such as decoders, during training to improve prediction performance, whereas many recent approaches do. Therefore, combining our framework with auxiliary architectures used in other methods could further enhance performance, which we consider a promising direction for future research. Another meaningful avenue would be to design a version of MIRRAMS specifically tailored for SSL tasks. Moreover, our method is not limited to addressing distributional shifts in missingness within the input data. A natural next step is to adapt it to covariate shift scenarios (Huang et al., 2006; Gretton et al., 2009; Yu and Szepesvári, 2012) and domain adaptation settings (Ganin et al., 2015; Tzeng et al., 2017; Xie et al., 2018), where the input distributions differ between training and test data. Extending MIRRAMS to such scenarios would be a promising direction for future research.

Appendix

A. Proof of Proposition 1

It is sufficient to prove the result for the case where both U and V are discrete random variables. The extension to the continuous case follows straightforwardly. The mutual information can be reformulated as:

$$I(U;V) = H(V) - H(V|U) \quad \text{and} \quad I(c(U);V) = H(V) - H(V|c(U)),$$

where $H(\cdot|\cdot)$ denotes the conditional entropy. By the definition of the conditional entropy, we have

$$H(V|U) = \sum_{u,v} p(u,v) \cdot \log \frac{p(u)}{p(u,v)} = \sum_{u,v} p(u,v) \cdot [-\log p(v|u)]\,, \tag{A.1}$$

and similarly,

$$H(V|c(U)) = \sum_{u,v} p(u,v) \cdot \log \frac{p(c(u))}{p(c(u),v)} = \sum_{u,v} p(u,v) \cdot [-\log p(v|c(u))]\,. \tag{A.2}$$

By subtracting (A.1) from (A.2), we obtain

$$\begin{aligned} I(U;V) - I(c(U);V) &= H(V|c(U)) - H(V|U) \\ &= \sum_{u,v} p(u,v) \cdot \log \left[\frac{p(v|u)}{p(v|c(u))} \right] \\ &= \sum_{u} p(u) \sum_{v} p(v|u) \cdot \log \left[\frac{p(v|u)}{p(v|c(u))} \right] \\ &= \mathbb{E}_U \left[D_{KL} \left(P(V|U=u) || P(V|c(U)=c(u)) \right) \right] \\ &< \epsilon. \end{aligned}$$

Hence, the proof is complete. □

B. Algorithm for MIRRAMS

Algorithm 1 MIRRMAS algorithm.

Input : Training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^{\text{tr}}$, number of features p, batch size B, masking ratio r, model f, cross-entropy loss H, weights λ_1, λ_2, threshold τ, the maximum iteration T

$t \leftarrow 0$
while $t < T$ **do**
  Sample $\mathcal{B}_t = \{\boldsymbol{x}_i, y_i\}_{i=1}^B \sim \mathcal{D}^{\text{tr}}$
  Sample mask $m_1, m_2, \ldots, m_B \sim \text{Bernoulli}(r)^{\otimes p}$
  Compute the three losses:
    $\mathcal{L}_1^{\text{MI}} = \frac{1}{B} \sum_{i=1}^B H(y_i, f(\boldsymbol{x}_i))$
    $\mathcal{L}_2^{\text{MI}} = \frac{1}{B} \sum_{i=1}^B H(y_i, f(m_i(\boldsymbol{x}_i)))$
    $\mathcal{L}_3^{\text{MI}} = \frac{1}{B} \sum_{i=1}^B [\mathbb{I}(\hat{c}(\boldsymbol{x}_i) \geq \tau)\ H(\hat{y}(\boldsymbol{x}_i), f(m_i(\boldsymbol{x}_i)))]$
  Combine the losses:
    $\mathcal{L} = \mathcal{L}_1^{\text{MI}} + \lambda_1 \cdot \mathcal{L}_2^{\text{MI}} + \lambda_2 \cdot \mathcal{L}_3^{\text{MI}}$
  update f to minimize $\mathcal{L}$ through backpropagation.
  $t \leftarrow t + 1$
end while

C. Benchmark Dataset Information and Experiment Results

C.1 Benchmark Dataset Information

We evaluate a total of 10 tabular datasets. All datasets are publicly available and can be obtained from sources such as UCI, AutoML, or Kaggle competitions, as listed in Table C.1.1. And Table C.1.2 provides a summary of the basic information for all datasets we analyze.

Table C.1.1: Benchmark dataset links.

Dataset Name	URL
adult	https://www.kaggle.com/lodetomasi1995/income-classification
arcene	https://www.openml.org/search?type=data&sort=runs&id=1458
arrhythmia	https://archive.ics.uci.edu/dataset/5/arrhythmia
bank	https://archive.ics.uci.edu/ml/datasets/bank+marketing
blastChar	https://www.kaggle.com/blastchar/telco-customer-churn
churn	https://www.kaggle.com/shrutimechlearn/churn-modelling
htru2	https://archive.ics.uci.edu/ml/datasets/HTRU2
qsar_bio	https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation
shoppers	https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset
spambase	https://archive.ics.uci.edu/ml/datasets/Spambase

Table C.1.2: Description of benchmark datasets. All datasets used in our experiments are binary classification tasks. "Positive Class ratio (%)" indicates the proportion of samples labeled as belonging to the positive class.

Dataset	Dataset size	# Features	# Categorical	# Continuous	Positive class ratio (%)
adult	32561	14	8	6	24.08
arcene	200	783	0	783	44.00
arrhythmia	452	226	0	226	14.60
bank	45211	16	9	7	11.70
blastChar	7043	20	17	3	26.54
churn	10000	11	3	8	20.37
htru2	17898	8	0	8	9.16
qsar_bio	1055	41	0	41	33.74
shoppers	12330	17	2	15	15.47
Spambase	4601	57	0	57	39.40

C.2 Architecture Description

Regarding the embedding mechanism, we employ a single-hidden-layer MLP with 100 hidden units for each continuous variable. For categorical variables, each distinct value is treated as a token and mapped to a learnable embedding vector. Additionally, for each variable, we introduce a separate learnable embedding vector to represent missing values. Notably, a distinct embedding is assigned for the missing case of each variable, allowing the model to capture variable-specific missingness.

We employed a Transformer-based architecture to model the tabular data. By default, the Transformer encoder depth was set to 6, the number of attention heads to 8, and the embedding dimension to 32. For high-dimensional datasets with more than 100 input features, the embedding dimension was reduced to 4 and the batch size was restricted to 64. Furthermore, for `Arrhythmia` dataset, the encoder depth was reduced to 1 to lower the risk of overfitting, and for `Arcene` dataset, the depth and number of heads were respectively adjusted to 4 and 1 in order to tailor the model capacity to the limited sample size.

C.3 Detailed Results for Supervised Learning Tasks

Table C.3.1-C.3.20 present the detailed results of the average AUC scores and their standard deviations, computed from three independent runs of each method across all benchmark datasets in the supervised learning tasks.

Table C.3.1: Comparison of averaged test AUC scores (%) on `adult` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	60.90	59.03	57.99	57.03	60.98	59.09	58.05	57.07	60.92	59.04	58.00	57.02	60.78	58.94	57.92	56.93
NN-0	51.19	51.28	51.27	51.20	47.16	50.45	52.88	55.60	50.12	52.59	54.51	56.59	49.78	52.26	54.29	56.47
RF	90.84	89.22	86.95	85.08	90.71	89.68	88.19	87.15	90.67	89.68	87.96	86.74	90.58	89.65	88.06	86.89
XGB	92.79	84.83	78.22	72.92	92.77	91.80	90.40	88.78	92.70	91.78	90.47	89.01	92.67	91.83	90.63	89.26
CB	91.14	88.14	84.79	81.77	91.02	89.49	87.55	85.68	90.89	89.57	87.84	86.10	90.55	89.41	87.90	86.25
SAINT	91.29	88.17	84.90	82.15	91.20	90.13	88.74	86.94	91.24	90.42	89.00	87.65	90.84	89.98	89.06	87.86
TabTF	87.81	86.36	84.71	82.66	87.74	86.59	85.16	83.26	87.52	86.50	85.34	83.84	87.75	86.77	85.68	84.20
STab	89.56	88.43	86.80	84.82	89.26	88.28	86.82	84.87	89.44	88.46	87.22	85.25	89.49	88.59	87.34	85.47
Ours	92.19	91.42	90.47	88.99	92.16	91.43	90.43	88.97	92.11	91.38	90.43	89.01	92.08	91.25	90.33	88.90

Table C.3.2: Standard deviation comparison of test AUC scores (%) on `adult` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.009	0.008	0.006	0.008	0.013	0.010	0.008	0.009	0.013	0.010	0.008	0.010	0.015	0.012	0.010	0.010
NN-0	0.023	0.017	0.014	0.014	0.056	0.057	0.058	0.055	0.009	0.025	0.034	0.044	0.005	0.023	0.032	0.038
RF	0.002	0.003	0.008	0.007	0.003	0.002	0.003	0.001	0.003	0.002	0.005	0.002	0.002	0.001	0.004	0.000
XGB	0.003	0.009	0.010	0.011	0.002	0.001	0.003	0.005	0.003	0.002	0.002	0.004	0.003	0.002	0.002	0.005
CB	0.018	0.013	0.007	0.009	0.018	0.021	0.023	0.024	0.019	0.022	0.024	0.027	0.024	0.026	0.028	0.032
SAINT	0.007	0.020	0.028	0.035	0.007	0.009	0.008	0.006	0.007	0.010	0.012	0.014	0.010	0.013	0.012	0.014
TabTF	0.003	0.005	0.005	0.004	0.003	0.004	0.005	0.011	0.005	0.006	0.007	0.005	0.004	0.005	0.004	0.004
STab	0.003	0.004	0.001	0.005	0.003	0.001	0.003	0.007	0.003	0.003	0.003	0.005	0.003	0.005	0.003	0.002
Ours	0.002	0.002	0.001	0.004	0.002	0.001	0.001	0.004	0.003	0.002	0.001	0.004	0.003	0.002	0.001	0.005

Table C.3.3: Comparison of averaged test AUC scores (%) on `arcene` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	63.89	64.43	64.35	62.19	61.81	61.57	61.50	60.26	60.65	62.11	61.50	60.73	59.57	59.49	60.80	59.57
NN-0	49.31	51.70	52.43	55.02	67.94	67.21	62.31	59.45	61.73	61.27	55.13	55.71	74.69	72.18	65.32	65.28
RF	50.66	51.00	50.31	52.43	48.61	50.12	50.62	49.69	47.61	49.61	47.76	47.84	49.46	49.42	48.96	46.53
XGB	85.34	85.03	82.48	84.57	81.79	78.16	75.93	72.61	78.74	76.43	74.73	68.63	77.93	75.69	72.80	71.72
CB	84.99	81.40	81.52	78.67	84.26	82.64	81.79	81.33	82.41	83.95	85.26	84.72	78.16	77.82	78.32	79.32
SAINT	83.68	83.91	83.68	85.38	81.56	82.18	80.71	81.79	89.58	88.19	86.27	86.88	87.96	89.43	87.89	89.89
TabTF	91.51	90.28	88.43	88.43	90.66	89.12	86.81	86.34	90.66	89.97	87.58	84.64	88.27	87.65	84.80	82.64
STab	85.26	78.78	81.17	79.32	84.18	78.63	78.24	75.46	80.79	79.48	80.25	76.23	85.19	79.47	76.85	75.23
Ours	89.58	88.43	88.19	87.58	87.65	87.11	86.11	85.88	75.54	76.08	74.69	72.99	74.85	75.23	74.61	72.84

Table C.3.4: Standard deviation comparison of test AUC scores (%) on `arcene` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.142	0.165	0.146	0.143	0.199	0.190	0.191	0.192	0.202	0.195	0.206	0.203	0.212	0.198	0.211	0.204
NN-0	0.012	0.022	0.090	0.116	0.204	0.168	0.204	0.188	0.113	0.126	0.106	0.085	0.039	0.068	0.066	0.113
RF	0.176	0.180	0.138	0.154	0.192	0.174	0.155	0.195	0.175	0.168	0.158	0.123	0.163	0.154	0.182	0.134
XGB	0.079	0.026	0.092	0.098	0.103	0.088	0.092	0.094	0.136	0.112	0.077	0.088	0.026	0.038	0.048	0.096
CB	0.061	0.031	0.013	0.040	0.083	0.076	0.083	0.045	0.073	0.051	0.026	0.039	0.113	0.082	0.053	0.066
SAINT	0.102	0.106	0.090	0.076	0.065	0.064	0.043	0.013	0.008	0.012	0.010	0.032	0.008	0.049	0.039	0.021
TabTF	0.058	0.058	0.058	0.059	0.046	0.045	0.050	0.058	0.044	0.045	0.053	0.069	0.063	0.046	0.062	0.070
STab	0.094	0.095	0.092	0.042	0.100	0.087	0.111	0.063	0.096	0.066	0.103	0.121	0.111	0.088	0.08	0.052
Ours	0.068	0.030	0.043	0.049	0.057	0.041	0.051	0.045	0.197	0.184	0.184	0.203	0.188	0.181	0.183	0.202

Table C.3.5: Comparison of averaged test AUC scores (%) on `arrhythmia` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	64.50	59.35	59.55	55.67	64.19	58.87	55.84	55.84	63.85	59.21	58.42	58.14	64.16	61.24	61.10	62.39
NN-0	76.99	75.32	72.22	67.51	78.97	77.58	75.22	74.28	80.48	76.24	74.15	71.66	81.65	77.62	74.50	71.81
RF	73.14	70.44	69.11	73.53	75.70	73.90	69.68	73.70	71.65	72.71	72.41	72.21	73.00	71.84	71.95	65.26
XGB	86.55	81.87	79.02	75.84	83.01	79.97	81.03	77.81	82.59	81.24	80.42	77.30	81.42	82.33	79.55	78.62
CB	82.64	85.37	82.59	77.30	83.21	78.78	77.20	75.05	79.30	76.64	76.60	72.49	83.46	82.11	81.80	79.10
SAINT	83.91	77.13	71.59	69.40	85.82	79.66	73.08	70.89	84.73	80.53	78.54	74.68	87.31	84.44	82.93	79.92
TabTF	81.04	70.75	58.41	46.71	81.63	69.54	57.93	46.71	82.48	68.17	55.12	44.06	80.20	63.83	52.93	39.76
STab	79.07	70.44	66.36	64.23	75.19	63.04	62.24	61.20	80.23	70.73	71.55	68.00	79.93	70.68	65.63	60.25
Ours	83.52	78.57	76.15	77.02	84.67	81.04	80.39	80.34	84.53	84.53	81.86	80.98	82.59	81.72	82.05	80.96

Table C.3.6: Standard deviation comparison of test AUC scores (%) on `arrhythmia` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.152	0.132	0.099	0.043	0.087	0.061	0.062	0.052	0.121	0.096	0.046	0.050	0.142	0.116	0.109	0.092
NN-0	0.127	0.118	0.115	0.137	0.105	0.082	0.060	0.068	0.073	0.065	0.041	0.089	0.081	0.059	0.066	0.058
RF	0.150	0.166	0.165	0.144	0.127	0.141	0.168	0.155	0.178	0.140	0.153	0.144	0.079	0.144	0.150	0.081
XGB	0.068	0.106	0.126	0.158	0.066	0.059	0.062	0.056	0.043	0.038	0.033	0.047	0.126	0.110	0.113	0.103
CB	0.143	0.088	0.070	0.053	0.044	0.024	0.032	0.034	0.037	0.045	0.044	0.040	0.062	0.059	0.025	0.033
SAINT	0.093	0.189	0.225	0.247	0.054	0.136	0.187	0.222	0.082	0.091	0.125	0.127	0.056	0.055	0.068	0.075
TabTF	0.069	0.056	0.060	0.105	0.068	0.060	0.069	0.122	0.073	0.068	0.059	0.122	0.069	0.032	0.038	0.086
STab	0.040	0.063	0.071	0.054	0.015	0.043	0.048	0.012	0.104	0.071	0.060	0.052	0.103	0.114	0.142	0.081
Ours	0.062	0.100	0.102	0.076	0.066	0.092	0.063	0.069	0.076	0.085	0.111	0.133	0.101	0.120	0.125	0.129

Table C.3.7: Comparison of averaged test AUC scores (%) on `bank` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	69.12	65.35	61.86	59.33	66.64	63.32	60.17	57.91	63.55	60.89	58.23	56.37	55.63	54.81	53.43	52.58
NN-0	83.20	78.25	73.58	69.89	82.77	78.56	74.46	70.94	82.65	78.87	75.06	71.72	83.52	80.75	77.43	74.41
RF	90.45	87.06	84.21	82.35	90.18	88.25	86.04	83.89	90.00	88.24	85.96	83.92	89.82	88.11	85.80	83.68
XGB	93.16	84.45	78.15	72.81	93.05	91.36	89.37	87.23	92.80	91.27	89.30	87.19	92.82	91.33	89.54	87.57
CB	91.35	87.16	83.40	79.38	90.92	88.71	86.35	83.90	90.76	88.58	86.30	83.93	90.42	88.41	86.27	83.95
SAINT	92.19	88.92	85.29	81.64	91.97	89.96	87.52	84.74	92.00	90.25	88.03	85.57	91.68	89.92	87.86	85.59
TabTF	88.33	86.36	84.34	83.13	88.04	86.75	85.38	84.21	87.71	86.56	85.52	84.55	86.87	85.71	84.70	84.03
STab	90.76	88.05	84.82	81.82	90.10	87.88	85.20	82.52	89.74	87.47	84.99	82.80	88.58	86.67	84.27	82.39
Ours	93.73	92.45	90.73	88.77	93.73	92.42	90.70	88.73	93.63	92.25	90.44	88.45	93.26	91.95	90.13	88.18

Table C.3.8: Standard deviation comparison of test AUC scores (%) on `bank` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.010	0.014	0.014	0.014	0.014	0.015	0.013	0.013	0.019	0.019	0.016	0.015	0.033	0.026	0.019	0.016
NN-0	0.005	0.012	0.013	0.020	0.006	0.015	0.016	0.023	0.009	0.019	0.020	0.028	0.018	0.028	0.028	0.035
RF	0.001	0.011	0.017	0.011	0.001	0.004	0.006	0.005	0.003	0.006	0.006	0.004	0.004	0.007	0.009	0.011
XGB	0.003	0.005	0.006	0.006	0.003	0.006	0.003	0.003	0.002	0.004	0.004	0.004	0.002	0.003	0.002	0.003
CB	0.024	0.028	0.021	0.022	0.024	0.034	0.033	0.038	0.023	0.033	0.033	0.037	0.024	0.034	0.034	0.039
SAINT	0.018	0.018	0.011	0.006	0.019	0.022	0.022	0.021	0.019	0.021	0.018	0.016	0.018	0.020	0.019	0.021
TabTF	0.003	0.004	0.004	0.003	0.003	0.003	0.005	0.009	0.003	0.003	0.001	0.004	0.001	0.004	0.003	0.009
STab	0.001	0.005	0.002	0.003	0.003	0.004	0.002	0.004	0.001	0.004	0.002	0.005	0.009	0.004	0.006	0.003
Ours	0.004	0.004	0.005	0.004	0.004	0.004	0.005	0.003	0.005	0.005	0.006	0.005	0.006	0.005	0.007	0.005

Table C.3.9: Comparison of averaged test AUC scores (%) on `blastChar` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	79.52	74.14	69.41	65.23	77.88	74.02	70.33	67.04	76.52	74.09	71.40	68.79	74.79	72.70	70.22	67.86
NN-0	80.74	75.86	70.68	67.08	83.57	81.22	78.66	75.54	83.45	81.39	79.44	76.63	82.67	81.11	79.42	77.23
RF	83.68	82.20	80.19	78.59	83.35	82.13	80.88	79.40	83.10	82.04	81.07	80.07	83.27	82.02	80.90	80.07
XGB	83.90	79.32	75.15	71.97	84.39	83.29	81.91	80.18	84.66	83.87	82.53	81.23	83.85	83.07	81.99	80.93
CB	84.31	82.29	80.16	77.52	84.60	83.22	81.62	79.64	84.32	83.22	81.94	79.98	84.21	83.10	81.79	80.07
SAINT	84.54	82.98	81.10	78.51	84.46	82.84	81.68	79.68	84.43	82.96	81.73	79.61	84.51	83.76	82.67	81.20
TabTF	82.70	81.80	81.12	79.77	82.64	81.79	81.26	80.11	82.50	81.66	81.24	80.20	82.39	81.67	81.34	80.31
STab	78.43	76.06	73.55	70.55	78.12	76.54	75.16	72.07	78.32	76.91	75.85	73.74	78.56	77.37	76.31	74.63
Ours	85.23	84.54	83.66	82.71	85.25	84.47	83.58	82.52	85.14	84.48	83.54	82.55	85.14	84.48	83.68	82.89

Table C.3.10: Standard deviation comparison of test AUC scores (%) on `blastChar` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.014	0.009	0.012	0.008	0.011	0.013	0.017	0.013	0.015	0.017	0.016	0.012	0.024	0.026	0.023	0.019
NN-0	0.028	0.014	0.020	0.034	0.008	0.012	0.016	0.023	0.009	0.012	0.015	0.020	0.018	0.021	0.022	0.030
RF	0.005	0.008	0.014	0.011	0.006	0.009	0.010	0.008	0.009	0.009	0.008	0.005	0.007	0.008	0.009	0.002
XGB	0.011	0.010	0.013	0.019	0.007	0.011	0.008	0.011	0.009	0.013	0.008	0.006	0.012	0.018	0.015	0.012
CB	0.017	0.021	0.023	0.023	0.013	0.016	0.016	0.016	0.016	0.016	0.015	0.014	0.015	0.017	0.020	0.020
SAINT	0.003	0.002	0.001	0.000	0.005	0.007	0.007	0.012	0.004	0.010	0.015	0.022	0.003	0.005	0.010	0.010
TabTF	0.014	0.012	0.014	0.013	0.014	0.013	0.015	0.015	0.013	0.013	0.013	0.015	0.013	0.014	0.014	0.017
STab	0.014	0.016	0.025	0.027	0.026	0.026	0.026	0.020	0.025	0.024	0.022	0.019	0.021	0.025	0.022	0.025
Ours	0.009	0.007	0.009	0.013	0.009	0.007	0.011	0.016	0.009	0.008	0.011	0.016	0.007	0.008	0.011	0.019

Table C.3.11: Comparison of averaged test AUC scores (%) on `churn` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	53.92	52.57	51.15	51.49	51.89	50.90	49.93	50.74	50.49	49.86	49.05	50.05	48.59	48.37	47.98	49.22
NN-0	58.11	56.43	55.46	53.77	57.09	56.64	55.57	54.63	54.56	53.67	53.37	53.22	51.48	51.98	51.51	52.26
RF	81.98	78.86	75.65	72.31	82.29	80.16	78.08	75.18	82.20	79.71	77.59	74.20	81.55	79.67	77.29	74.40
XGB	86.14	80.34	75.25	69.66	86.03	84.14	82.09	79.42	85.85	84.09	81.96	79.52	85.82	84.35	82.24	80.04
CB	85.87	83.03	80.71	77.10	85.27	83.11	81.11	78.44	85.08	83.04	81.12	78.56	84.64	82.90	81.13	78.82
SAINT	86.23	83.49	80.93	78.83	86.04	84.28	82.31	79.90	85.88	84.25	82.37	80.24	85.96	84.82	82.87	81.45
TabTF	66.69	66.01	66.12	65.34	66.35	65.64	65.43	64.39	66.68	66.19	65.83	65.19	66.02	65.60	65.56	65.57
STab	63.39	62.62	62.41	61.33	63.33	62.61	61.67	60.62	64.89	63.94	62.81	62.04	63.35	62.84	62.24	61.45
Ours	85.83	84.09	82.40	80.64	85.68	84.01	82.29	80.47	85.41	83.79	82.08	80.19	84.98	83.42	82.10	79.99

Table C.3.12: Standard deviation comparison of test AUC scores (%) on `churn` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.020	0.024	0.024	0.025	0.038	0.042	0.037	0.034	0.033	0.039	0.034	0.031	0.030	0.036	0.032	0.031
NN-0	0.002	0.012	0.024	0.017	0.012	0.006	0.008	0.013	0.023	0.026	0.017	0.010	0.046	0.037	0.034	0.024
RF	0.021	0.029	0.033	0.038	0.019	0.020	0.028	0.027	0.020	0.030	0.027	0.042	0.020	0.024	0.032	0.037
XGB	0.004	0.013	0.004	0.012	0.003	0.003	0.005	0.004	0.006	0.006	0.004	0.006	0.005	0.005	0.004	0.009
CB	0.011	0.013	0.012	0.013	0.014	0.020	0.019	0.024	0.011	0.019	0.018	0.021	0.015	0.018	0.017	0.022
SAINT	0.000	0.001	0.001	0.001	0.000	0.001	0.001	0.001	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.001
TabTF	0.017	0.014	0.012	0.012	0.019	0.018	0.017	0.019	0.014	0.013	0.015	0.017	0.009	0.011	0.014	0.018
STab	0.034	0.028	0.026	0.016	0.074	0.056	0.049	0.042	0.069	0.054	0.048	0.039	0.043	0.042	0.033	0.028
Ours	0.010	0.011	0.008	0.005	0.011	0.012	0.008	0.005	0.012	0.013	0.010	0.008	0.012	0.013	0.010	0.007

Table C.3.13: Comparison of averaged test AUC scores (%) on `htru2` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	96.65	92.79	88.63	84.47	95.68	94.36	92.56	90.53	95.56	94.38	92.74	90.86	95.46	94.39	92.87	91.03
NN-0	96.89	93.51	89.85	86.35	96.89	95.99	94.51	92.39	96.83	96.21	95.00	93.43	96.70	96.17	95.14	93.86
RF	96.42	96.05	95.48	94.18	96.05	95.77	95.78	94.55	96.18	96.07	95.30	94.49	95.72	96.09	95.48	94.89
XGB	97.31	95.32	92.67	89.14	97.36	96.90	96.05	95.18	97.43	97.16	96.51	95.78	97.40	97.27	96.74	96.17
CB	97.08	95.69	94.21	92.45	97.03	96.28	95.43	93.80	97.26	96.61	95.93	94.90	97.03	96.47	95.89	95.13
SAINT	98.18	95.47	91.62	87.52	97.95	97.61	96.97	95.89	97.92	97.76	97.36	96.72	97.94	97.82	97.64	97.08
TabTF	97.29	97.63	97.91	98.52	97.22	97.65	98.09	98.69	97.16	97.63	98.19	98.71	96.83	97.39	97.98	98.59
STab	97.39	92.11	87.86	83.58	97.32	96.37	94.22	91.94	97.29	96.70	95.43	94.19	97.11	96.57	95.58	94.96
Ours	97.57	97.42	96.89	96.31	97.58	97.36	96.74	96.19	97.57	97.44	96.90	96.29	97.52	97.33	96.87	96.21

Table C.3.14: Standard deviation comparison of test AUC scores (%) on `htru2` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.007	0.010	0.013	0.013	0.007	0.006	0.001	0.005	0.006	0.005	0.001	0.006	0.007	0.005	0.003	0.008
NN-0	0.005	0.003	0.006	0.011	0.006	0.006	0.005	0.004	0.006	0.005	0.008	0.008	0.007	0.005	0.006	0.005
RF	0.008	0.006	0.002	0.008	0.002	0.004	0.002	0.001	0.007	0.008	0.005	0.004	0.005	0.005	0.001	0.001
XGB	0.004	0.006	0.015	0.024	0.004	0.003	0.005	0.006	0.004	0.003	0.005	0.004	0.004	0.004	0.006	0.006
CB	0.007	0.009	0.011	0.010	0.006	0.008	0.008	0.016	0.006	0.007	0.007	0.005	0.005	0.008	0.008	0.008
SAINT	0.000	0.011	0.021	0.028	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TabTF	0.005	0.002	0.005	0.002	0.007	0.002	0.003	0.001	0.007	0.002	0.002	0.001	0.007	0.003	0.005	0.002
STab	0.005	0.007	0.005	0.011	0.006	0.006	0.007	0.004	0.009	0.010	0.009	0.009	0.004	0.005	0.002	0.003
Ours	0.006	0.006	0.006	0.007	0.005	0.005	0.007	0.008	0.005	0.006	0.008	0.011	0.005	0.006	0.006	0.006

Table C.3.15: Comparison of averaged test AUC scores (%) on `qsar_bio` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	89.42	81.66	76.47	73.38	86.90	81.72	77.52	75.08	88.31	83.63	80.01	77.38	85.84	81.48	78.07	75.12
NN-0	92.27	85.11	81.37	76.03	91.53	87.87	85.82	83.25	91.84	88.60	86.92	84.28	90.86	88.27	86.05	83.08
RF	86.10	84.49	83.92	83.07	84.18	83.01	81.99	79.92	83.22	82.41	81.95	80.69	82.74	82.10	80.63	78.80
XGB	90.44	86.11	81.07	79.20	92.27	90.27	88.19	86.90	90.21	88.21	86.12	84.74	90.25	89.10	87.67	86.53
CB	91.07	88.78	86.41	84.34	90.47	88.81	86.66	84.13	90.29	89.00	87.04	84.40	89.46	88.26	86.28	84.81
SAINT	90.93	89.04	88.75	86.67	90.13	87.70	87.87	85.64	91.36	86.97	87.52	84.66	90.69	88.63	88.94	88.19
TabTF	91.97	90.73	89.67	88.75	92.04	90.81	89.84	88.84	91.75	90.84	89.77	88.78	91.38	90.42	89.61	88.72
STab	88.97	88.00	86.24	84.94	89.42	88.18	86.29	85.36	89.10	88.72	87.15	85.20	88.90	87.11	86.10	84.82
Ours	91.88	91.22	90.67	89.82	91.70	90.63	90.14	89.24	91.70	90.85	90.66	87.84	91.47	90.67	90.63	89.93

Table C.3.16: Standard deviation comparison of test AUC scores (%) on `qsar_bio` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.022	0.022	0.009	0.032	0.015	0.015	0.022	0.026	0.011	0.032	0.035	0.037	0.029	0.026	0.028	0.019
NN-0	0.012	0.017	0.012	0.017	0.007	0.017	0.023	0.018	0.004	0.010	0.023	0.016	0.010	0.006	0.011	0.008
RF	0.001	0.011	0.003	0.002	0.001	0.008	0.012	0.019	0.006	0.014	0.015	0.022	0.003	0.007	0.030	0.017
XGB	0.013	0.025	0.034	0.043	0.009	0.013	0.020	0.025	0.022	0.030	0.044	0.047	0.015	0.013	0.008	0.006
CB	0.019	0.025	0.021	0.020	0.016	0.026	0.011	0.024	0.021	0.035	0.028	0.032	0.024	0.032	0.035	0.037
SAINT	0.001	0.000	0.001	0.004	0.002	0.001	0.005	0.012	0.058	0.008	0.011	0.005	0.001	0.001	0.001	0.001
TabTF	0.009	0.016	0.008	0.016	0.011	0.017	0.009	0.019	0.009	0.017	0.011	0.020	0.012	0.014	0.011	0.021
STab	0.030	0.018	0.016	0.010	0.026	0.010	0.021	0.015	0.036	0.022	0.024	0.019	0.036	0.020	0.027	0.036
Ours	0.003	0.006	0.003	0.005	0.006	0.005	0.009	0.005	0.002	0.010	0.005	0.025	0.005	0.010	0.002	0.007

Table C.3.17: Comparison of averaged test AUC scores (%) on `shoppers` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	57.00	55.42	54.81	53.37	56.10	54.78	54.10	52.90	56.61	55.17	54.43	53.17	55.64	54.31	53.60	52.47
NN-0	91.47	87.82	84.17	80.25	91.28	88.87	85.94	82.43	91.50	89.07	86.19	83.03	90.38	88.11	85.58	82.60
RF	91.59	87.74	85.25	81.62	91.27	89.04	87.68	85.83	90.79	89.21	87.53	86.26	91.08	89.32	87.98	86.37
XGB	93.17	88.51	84.22	80.57	93.03	90.89	89.16	86.33	93.07	91.25	89.66	87.25	92.66	90.96	89.40	87.43
CB	92.08	88.67	85.72	82.08	92.30	89.93	87.79	84.65	92.32	90.13	87.96	85.05	92.07	89.90	87.75	84.99
SAINT	93.39	90.46	87.16	83.02	93.39	91.92	89.57	87.07	93.45	92.13	89.82	86.81	93.08	91.97	90.05	87.17
TabTF	82.64	84.39	86.27	87.50	82.83	84.35	85.45	86.59	82.69	84.06	85.03	86.43	82.08	83.26	84.32	85.67
STab	87.58	85.12	82.35	79.75	87.25	85.47	83.33	80.11	86.15	84.56	82.28	79.63	86.43	84.7	81.89	79.57
Ours	93.17	91.98	89.93	88.04	92.85	91.88	89.89	88.03	92.86	91.92	90.02	88.11	92.61	91.65	89.76	87.74

Table C.3.18: Standard deviation comparison of test AUC scores (%) on `shoppers` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.022	0.023	0.018	0.013	0.025	0.029	0.025	0.021	0.012	0.015	0.010	0.009	0.005	0.006	0.012	0.012
NN-0	0.004	0.013	0.009	0.010	0.003	0.005	0.010	0.012	0.007	0.009	0.012	0.014	0.010	0.006	0.009	0.010
RF	0.004	0.010	0.006	0.007	0.002	0.001	0.004	0.010	0.002	0.001	0.003	0.012	0.002	0.001	0.002	0.005
XGB	0.005	0.007	0.010	0.006	0.004	0.003	0.006	0.010	0.002	0.002	0.003	0.011	0.005	0.004	0.007	0.009
CB	0.016	0.012	0.016	0.020	0.007	0.014	0.024	0.029	0.009	0.014	0.023	0.029	0.008	0.014	0.021	0.027
SAINT	0.000	0.004	0.007	0.006	0.001	0.003	0.008	0.025	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
TabTF	0.009	0.006	0.008	0.009	0.011	0.006	0.010	0.004	0.012	0.005	0.006	0.002	0.012	0.003	0.007	0.009
STab	0.005	0.003	0.009	0.008	0.004	0.007	0.006	0.009	0.005	0.004	0.004	0.013	0.003	0.002	0.003	0.009
Ours	0.003	0.003	0.002	0.006	0.003	0.007	0.005	0.002	0.002	0.006	0.005	0.002	0.001	0.005	0.004	0.003

Table C.3.19: Comparison of averaged test AUC scores (%) on `Spambase` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	84.44	78.54	74.72	71.23	84.35	78.66	74.93	71.57	84.58	80.22	77.42	74.78	87.06	84.04	82.15	79.78
NN-0	97.75	93.76	89.75	86.62	97.75	96.96	96.15	95.37	97.73	97.03	96.38	95.68	97.54	96.90	96.38	95.79
RF	96.47	95.16	94.10	92.61	95.88	95.30	94.51	93.34	95.63	95.01	94.25	93.60	95.58	95.27	94.45	93.99
XGB	98.31	93.38	87.73	81.22	98.56	98.14	97.35	96.54	98.42	98.09	97.44	96.76	98.48	98.22	97.71	97.11
CB	97.61	97.24	96.55	95.59	97.58	97.21	96.53	95.67	97.49	97.21	96.55	95.85	97.36	97.08	96.48	95.84
SAINT	98.16	95.95	92.85	90.81	98.17	97.28	95.90	94.11	97.94	97.63	96.54	95.20	97.75	97.40	96.52	95.51
TabTF	98.19	98.55	98.69	98.76	98.01	98.45	98.67	98.76	97.93	98.48	98.72	98.81	97.65	98.39	98.70	98.79
STab	97.22	96.61	95.84	94.82	97.20	96.66	96.12	95.13	97.40	96.80	96.25	95.38	97.42	96.98	96.39	95.79
Ours	97.85	97.45	97.01	96.44	97.66	97.33	96.89	96.40	97.71	97.37	96.95	96.49	97.58	97.25	96.71	96.31

Table C.3.20: Standard deviation comparison of test AUC scores (%) on `Spambase` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations.

$\alpha^{\mathbf{tr}}$	0				0.05				0.1				0.2			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3	0	0.1	0.2	0.3
LR-0	0.039	0.048	0.051	0.057	0.043	0.055	0.059	0.067	0.046	0.061	0.066	0.075	0.041	0.056	0.059	0.072
NN-0	0.005	0.014	0.024	0.036	0.004	0.004	0.003	0.003	0.005	0.004	0.005	0.006	0.005	0.004	0.004	0.006
RF	0.008	0.000	0.005	0.006	0.005	0.003	0.005	0.006	0.003	0.002	0.007	0.008	0.006	0.004	0.006	0.007
XGB	0.003	0.005	0.002	0.009	0.001	0.000	0.002	0.004	0.001	0.001	0.002	0.004	0.000	0.001	0.002	0.005
CB	0.016	0.018	0.019	0.019	0.016	0.018	0.020	0.020	0.017	0.019	0.021	0.021	0.018	0.019	0.022	0.022
SAINT	0.000	0.005	0.009	0.012	0.006	0.014	0.006	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TabTF	0.001	0.001	0.003	0.002	0.001	0.001	0.003	0.003	0.001	0.001	0.003	0.003	0.002	0.001	0.002	0.003
STab	0.004	0.005	0.008	0.007	0.004	0.008	0.007	0.008	0.004	0.004	0.006	0.006	0.002	0.004	0.005	0.005
Ours	0.004	0.002	0.002	0.001	0.004	0.003	0.003	0.003	0.004	0.003	0.002	0.003	0.004	0.003	0.002	0.002

C.4 Semi Supervised Learning Results

Table C.4.1-C.4.9 report the average AUC scores and corresponding standard deviations, obtained from three independent runs of each method on all benchmark datasets under the semi-supervised learning setting. The `arcene` dataset was excluded from this evaluation due to its limited sample size (Please refer to Table C.1.2).

Table C.4.1: Comparison of averaged test AUC scores (%) on `adult` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	90.45 (0.000)	86.93 (0.000)	83.26 (0.002)	80.36 (0.002)	90.33 (0.001)	89.15 (0.001)	87.82 (0.004)	86.51 (0.004)
STab	88.61 (0.005)	87.38 (0.007)	85.95 (0.007)	84.05 (0.003)	88.66 (0.003)	87.63 (0.005)	86.30 (0.003)	84.50 (0.002)
Ours	91.04 (0.003)	90.24 (0.002)	89.04 (0.001)	87.59 (0.003)	91.03 (0.003)	90.21 (0.001)	88.95 (0.003)	87.34 (0.008)

Table C.4.2: Comparison of averaged test AUC scores (%) on `arrhythmia` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	75.11 (0.038)	76.03 (0.015)	73.95 (0.005)	69.06 (0.011)	80.03 (0.031)	75.19 (0.037)	73.90 (0.030)	65.88 (0.043)
STab	76.76 (0.046)	65.85 (0.099)	64.98 (0.127)	60.80 (0.124)	71.90 (0.072)	66.03 (0.050)	61.67 (0.095)	58.58 (0.072)
Ours	74.54 (0.132)	72.97 (0.122)	73.39 (0.137)	74.18 (0.131)	73.92 (0.141)	72.83 (0.113)	72.60 (0.138)	73.25 (0.132)

Table C.4.3: Comparison of averaged test AUC scores (%) on `bank` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	89.84 (0.000)	86.59 (0.002)	83.52 (0.003)	80.26 (0.005)	90.20 (0.002)	87.87 (0.003)	85.52 (0.004)	83.04 (0.006)
STab	89.52 (0.001)	86.94 (0.008)	83.80 (0.003)	81.05 (0.006)	88.34 (0.002)	86.09 (0.005)	83.31 (0.003)	80.77 (0.008)
Ours	90.77 (0.003)	89.43 (0.003)	87.60 (0.004)	85.51 (0.001)	90.51 (0.002)	89.14 (0.003)	87.33 (0.002)	85.12 (0.003)

Table C.4.4: Comparison of averaged test AUC scores (%) on `blastChar` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	83.17 (0.004)	81.43 (0.007)	79.83 (0.007)	77.16 (0.008)	83.11 (0.004)	81.47 (0.005)	80.10 (0.005)	77.90 (0.003)
STab	79.10 (0.029)	76.58 (0.027)	74.08 (0.023)	71.60 (0.031)	78.79 (0.035)	76.81 (0.026)	74.97 (0.027)	71.95 (0.014)
Ours	83.28 (0.013)	82.40 (0.009)	81.65 (0.012)	80.44 (0.018)	82.96 (0.012)	82.00 (0.009)	81.31 (0.013)	80.19 (0.016)

Table C.4.5: Comparison of averaged test AUC scores (%) on `churn` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	84.21 (0.002)	81.45 (0.003)	79.64 (0.004)	77.34 (0.005)	83.47 (0.007)	81.11 (0.003)	79.22 (0.000)	76.97 (0.006)
STab	61.94 (0.032)	61.10 (0.028)	60.35 (0.015)	59.58 (0.020)	63.09 (0.095)	62.10 (0.085)	61.25 (0.076)	60.09 (0.069)
Ours	83.38 (0.015)	81.49 (0.015)	79.82 (0.010)	77.70 (0.007)	83.28 (0.017)	81.20 (0.017)	79.47 (0.014)	77.27 (0.011)

Table C.4.6: Comparison of averaged test AUC scores (%) on `HTRU2` for various $(\alpha^{\text{tr}}, \alpha^{\text{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	96.91 (0.000)	96.36 (0.000)	95.31 (0.000)	94.45 (0.001)	97.23 (0.000)	97.10 (0.001)	96.49 (0.001)	95.64 (0.004)
STab	96.97 (0.008)	92.34 (0.032)	88.86 (0.043)	86.00 (0.054)	96.86 (0.008)	95.81 (0.012)	93.69 (0.016)	92.59 (0.016)
Ours	97.03 (0.003)	96.61 (0.004)	96.03 (0.002)	95.51 (0.003)	97.11 (0.005)	96.72 (0.004)	96.21 (0.002)	95.59 (0.002)

Table C.4.7: Comparison of averaged test AUC scores (%) on `shoppers` for various $(\alpha^{\mathrm{tr}}, \alpha^{\mathrm{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	89.97 (0.017)	88.21 (0.014)	87.84 (0.016)	85.27 (0.024)	89.40 (0.012)	87.31 (0.007)	87.26 (0.011)	85.26 (0.008)
STab	90.51 (0.011)	88.31 (0.005)	86.50 (0.010)	85.13 (0.010)	89.41 (0.010)	88.15 (0.013)	86.99 (0.024)	85.08 (0.019)
Ours	89.34 (0.003)	88.48 (0.002)	87.49 (0.010)	86.69 (0.003)	89.07 (0.004)	88.20 (0.003)	87.75 (0.015)	86.67 (0.008)

Table C.4.8: Comparison of averaged test AUC scores (%) on `qsar_bio` for various $(\alpha^{\mathbf{tr}}, \alpha^{\mathbf{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	90.06 (0.017)	87.45 (0.016)	84.89 (0.019)	81.96 (0.017)	89.77 (0.011)	85.64 (0.010)	82.73 (0.019)	79.63 (0.024)
STab	85.92 (0.007)	83.39 (0.002)	81.03 (0.008)	78.26 (0.015)	84.59 (0.011)	83.04 (0.009)	80.63 (0.015)	78.04 (0.024)
Ours	90.82 (0.005)	89.65(0.005)	87.87 (0.006)	86.24 (0.009)	90.41 (0.007)	89.22 (0.009)	87.58 (0.007)	85.78 (0.010)

Table C.4.9: Comparison of averaged test AUC scores (%) on `Spambase` for various $(\alpha^{\mathrm{tr}}, \alpha^{\mathrm{ts}})$ combinations. Values in parentheses indicate the corresponding standard deviations.

$\alpha^{\mathbf{tr}}$	0.05				0.1			
$\alpha^{\mathbf{ts}}$	0	0.1	0.2	0.3	0	0.1	0.2	0.3
SAINT	95.81 (0.000)	94.41 (0.002)	92.33 (0.003)	90.81 (0.004)	95.63 (0.002)	94.44 (0.006)	92.57 (0.007)	91.02 (0.008)
STab	96.62 (0.011)	96.09 (0.010)	95.35 (0.013)	94.19 (0.011)	96.26 (0.011)	96.01 (0.009)	95.22 (0.012)	94.40 (0.009)
Ours	95.68 (0.012)	95.17 (0.009)	94.35 (0.009)	93.56 (0.006)	95.57 (0.014)	94.91 (0.011)	94.11 (0.012)	93.18 (0.011)

C.5 Ablation studies

Optimal Masking Ratio Selection Positive values of r generally lead to improved performance of our method. That indicates that our method is not sensitive to the value of r in practice, as long as it is chosen reasonably. This result is somewhat in contrast to our initial claim that a careful selection of r is critical to ensure the condition $I\left(M^{\mathrm{tr}}(\boldsymbol{X}); Y\right) \approx I\left(M_r^{\mathrm{m}} \circ M^{\mathrm{tr}}(\boldsymbol{X}); Y\right)$ is satisfied, thereby allowing us to focus solely on conditions (i) through (iii) of our MI robustness conditions. We conjecture that this outcome may be attributed to the fact that our experiments were conducted under the MCAR (Missing Completely At Random) assumption. Exploring more complex missingness settings–such as MAR (missing at random) or MNAR (missing not at random)–would be a valuable extension of our study.

Table C.5.1: Averaged results of test AUC scores with various values of r.

$(\alpha^{\mathbf{tr}}, \alpha^{\mathbf{ts}})$	(0.05, 0.2)					(0.1, 0.2)					(0.2, 0.2)				
r	0	0.1	0.2	0.3	0.4	0	0.1	0.2	0.3	0.4	0	0.1	0.2	0.3	0.4
adult	88.91	90.13	90.38	90.54	90.62	89.46	90.07	90.36	90.53	90.60	89.68	90.03	90.30	90.34	90.46
htru2	95.90	96.82	97.32	97.52	97.38	96.92	96.91	97.52	97.59	97.42	97.01	97.12	97.23	97.57	97.70
qsar_bio	85.17	90.93	90.50	91.10	90.88	86.53	90.77	90.63	91.30	90.76	88.14	90.76	90.49	90.80	90.66

Loss Term Analysis With respect to the hyperparameters λ_1 and λ_2, which control the weights of the proposed loss components $\mathcal{L}_1^{\mathrm{MI}}$ and $\mathcal{L}_2^{\mathrm{MI}}$, we observe that the method performs best when both values are set to be positive. This indicates that these MI-based regularization terms contribute meaningfully to model learning. We note that $\mathcal{L}_1^{\mathrm{MI}}$ and $\mathcal{L}_2^{\mathrm{MI}}$ appear to play a similar role at first glance, as both encourage the prediction model to maintain correct predictions even under additional masking operations. However, the third term, $\mathcal{L}_2^{\mathrm{MI}}$, functions as a loss only for inputs on which the model exhibits high confidence. As a result, $\mathcal{L}_2^{\mathrm{MI}}$ places greater loss weight on high-confidence examples, effectively propagating reliable information to more challenging instances and thereby enhancing the overall

learning process.

Table C.5.2: Averaged results of test AUC scores with various values of λ_1.

$(\alpha^{\mathbf{tr}}, \alpha^{\mathbf{ts}})$	(0.05, 0.2)						(0.1, 0.2)						(0.2, 0.2)					
λ_1	0	1	5	10	15	20	0	1	5	10	15	20	0	1	5	10	15	20
adult	89.71	90.26	90.34	90.38	90.37	90.39	89.72	90.23	90.35	90.36	90.39	90.37	89.82	90.16	90.25	90.30	90.29	90.29
htru2	96.05	96.95	97.31	97.32	97.32	97.32	96.07	97.36	97.49	97.52	97.49	97.52	96.83	97.21	97.24	97.23	97.23	97.22
qsar_bio	89.34	90.07	90.52	90.50	90.51	90.59	88.43	90.43	90.56	90.63	90.58	90.59	88.85	90.34	90.45	90.49	90.51	90.51

Table C.5.3: Averaged results of test AUC scores with various values of λ_2.

$(\alpha^{\mathbf{tr}}, \alpha^{\mathbf{ts}})$	(0.05, 0.2)						(0.1, 0.2)						(0.2, 0.2)					
λ_2	0	1	5	10	15	20	0	1	5	10	15	20	0	1	5	10	15	20
adult	90.37	90.38	90.41	90.44	90.44	90.44	90.38	90.36	90.42	90.41	90.42	90.41	90.29	90.30	90.31	90.33	90.32	90.31
htru2	97.30	97.32	97.40	97.42	97.39	97.40	97.50	97.52	97.50	97.53	97.53	97.51	97.23	97.23	97.23	97.23	97.22	97.20
qsar_bio	90.47	90.50	90.64	91.12	90.40	90.32	90.62	90.63	90.57	90.54	90.52	90.31	90.45	90.49	90.59	90.63	90.86	90.99

Optimal Confidence Threshold Selection Lastly, we find that the method is relatively insensitive to the choice of the temperature parameter τ across the range of values we considered. This robustness indicates that the effect of $\mathcal{L}_2^{\mathrm{MI}}$–namely, propagating reliable information to more challenging instances–remains effective as long as τ is set within a reasonable range. In addition, our method does not require fine-tuning of τ to achieve strong performance, which improves usability in practical settings.

Table C.5.4: Averaged results of test AUC scores with various values of τ.

$(\alpha^{\mathbf{tr}}, \alpha^{\mathbf{ts}})$	(0.05, 0.2)				(0.1, 0.2)				(0.2, 0.2)			
τ	0.8	0.9	0.95	0.99	0.8	0.9	0.95	0.99	0.8	0.9	0.95	0.99
adult	90.40	90.38	90.40	90.37	90.37	90.39	90.37	90.36	90.30	90.29	90.30	90.28
htru2	97.31	97.30	97.30	97.17	97.51	97.49	97.49	97.50	97.23	97.23	97.23	97.23
qsar_bio	91.03	90.93	91.03	91.05	90.71	90.71	90.71	90.73	90.79	90.85	90.89	90.89

Running Time Analysis Although our method requires two forward passes per update–one for the original training data and another for the additionally masked version–it maintains faster running time compared to other deep learning–based methods, in a case exceeding a three times speedup. This efficiency is largely due to the fact that our approach does

not employ any auxiliary architectures such as decoders, while other competitors do. Therefore, this result highlights our method as an efficient learning framework in terms of both memory usage and computational cost.

Table C.5.5: Comparison of running times. All results are reported as the average running time per epoch across the benchmark datasets, rescaled such that the running time of our method is normalized to one.

Method	MIRRAMS	SwitchTab	SAINT	TabTransformer
Time(ratio)	1	1.3365	2.5837	3.7881

References

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Suiyao Chen, Jing Wu, Naira Hovakimyan, and Handong Yao. Recontab: Regularized contrastive representation learning for tabular data. *arXiv preprint arXiv:2310.18541*, 2023.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *CoRR*, abs/1505.07818, 2015. URL `http://arxiv.org/abs/1505.07818`.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36: 53385–53432, 2023.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems 34:*, pages 18932–18943, 2021.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Lan-Zhe Guo and Yufeng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8082–8094. PMLR, 2022. URL https://proceedings.mlr.press/v162/guo22e.html.

Carlos Sáez-Alba Gutiérrez-Sacristán-Isaac and Kohane-Juan García-Gómez-Paul Avillach. Ehrtemporalvariability: delineating temporal data-set shifts in electronic health records. *GigaScience–2020*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.org/10.1109/CVPR52688.2022.01553.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Pro-*

cessing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006, pages 601–608. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

Jiwoo Kim, Sunghoon Kwon, and Dongha Kim. A review and suggestions for synthetic data generation strategies using deep generative models. *The Korean Data & Information Science Society*, 34(5):791–810, 2023.

Jungkyu Kim, Kibok Lee, and Taeyoung Park. To predict or not to predict? proportionally masked autoencoders for tabular data imputation. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 17886–17894. AAAI Press, 2025. doi: 10.1609/AAAI.V39I17.33967. URL https://doi.org/10.1609/aaai.v39i17.33967.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mark Meterko, Joseph D Restuccia, Kelly Stolzmann, David Mohr, Caitlin Brennan, Justin Glasgow, and Peter Kaboli. Response rates, nonresponse bias, and data quality: results from a national survey of senior healthcare leaders. *Public Opinion Quarterly*, 79(1): 130–144, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical fea-

tures. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6639–6649, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

Hongyi Qian, Baohui Wang, Ping Ma, Lei Peng, Songfeng Gao, and You Song. Managing dataset shift by adversarial validation for credit scoring. In *Pacific Rim International Conference on Artificial Intelligence*, pages 477–488. Springer, 2022.

Rajat Singh and Srikanta Bedathur. Embeddings for tabular data: A survey. *arXiv preprint arXiv:2302.11777*, 2023.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

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.

Tyrel Stokes, Hyungrok Do, Saul Blecker, Rumi Chunara, and Samrachana Adhikari. Domain adaptation under mnar missingness. *arXiv preprint arXiv:2504.00322*, 2025.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2962–2971. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.316. URL https://doi.org/10.1109/CVPR.2017.316.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PDrUPTXJI_A.

Jing Wu, Suiyao Chen, Qi Zhao, Renat Sergazinov, Chen Li, Shengjie Liu, Chongchao Zhao, Tianpei Xie, Hanqing Guo, Cheng Ji, et al. Switchtab: Switched autoencoders are effective tabular learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15924–15933, 2024.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceed-*

ings of Machine Learning Research, pages 5419–5428. PMLR, 2018. URL http://proceedings.mlr.press/v80/xie18c.html.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yufeng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11525–11536. PMLR, 2021. URL http://proceedings.mlr.press/v139/xu21e.html.

Yazheng Yang, Yuqi Wang, Guang Liu, Ledell Wu, and Qi Liu. Unitabe: A universal pretraining protocol for tabular foundation model in data science. *arXiv preprint arXiv:2307.09249*, 2023.

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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yaoliang Yu and Csaba Szepesvári. Analysis of kernel mean matching under covariate shift. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL http://icml.cc/2012/papers/330.pdf.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Pro-*

cessing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 18408–18419, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/995693c15f439e3d189b06e89d145dd5-Abstract.html.

Helen Zhou, Sivaraman Balakrishnan, and Zachary Lipton. Domain adaptation under missingness shift. In *International Conference on Artificial Intelligence and Statistics*, pages 9577–9606. PMLR, 2023.

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. Xtab: Cross-table pretraining for tabular transformers. *arXiv preprint arXiv:2305.06090*, 2023.

국문 초록

결측치 분포 변화에 대한 강건성을 위한 상호정보량 정규화 기반 딥러닝 예측 모델 학습 방법론에 대한 연구

성신여자대학교
대학원
통계학과
이지혜

현실 세계의 데이터 분석에서는 학습과 테스트 시 입력 데이터의 결측 분포가 달라지는 현상이 자주 발생하며, 이는 예측 성능의 일관성을 크게 저해하는 요인으로 작용합니다. 본 연구에서는 이러한 결측 분포 변화를 해결하기 위해 설계된 새로운 심층 학습 프레임워크를 제안합니다. 상호정보량을 기반으로 한 조건을 도입하여, 예측 모델이 다양한 결측 패턴에 불변하면서도 레이블 관련 정보를 최대한 보존하도록 유도함으로써 테스트 시 관측되지 못한 결측 시나리오에 대해서도 강건성을 향상시킵니다. 이러한 조건을 실제로 적용하기 위해, 새로운 정규화 항을 도출하기 위한 간단하면서도 효과적인 기법을 제안하고, 이에 대응하는 목적 함수를 개발하여 이를 MIRRAMS라 명명합니다. 부수적으로, 본 연구는 FixMatch와 같은 일관성 정규화 기반 준지도학습 방법론에 대한 이론적 해석을 제공합니다. 다양한 벤치마크 데이터셋을 사용한 실험을 통해 MIRRAMS가 기존 기법 대비 모든 결측 시나리오에서 안정적이며 우수한 성능을 보임을 확인합니다. 아울러 결측 데이터가 전혀 없는 일반적인 지도학습에서도 최첨단 성능을 달성하고, 준지도학습 과제로 자연스럽게 확장 가능하다는 점을 강조하여 MIRRAMS를 범용 학습을 위한 강력한 기성 프레임워크로 제안합니다.

핵심용어 : 결측치 분포 변화, 딥러닝, 상호정보량