

# Generative Imputation and Stochastic Prediction

Mohammad Kachuee, Kimmo Karkkainen, Orpaz Goldstein, Sajad Darabi, and Majid Sarrafzadeh

**Abstract**—In many machine learning applications, we are faced with incomplete datasets. In the literature, missing data imputation techniques have been mostly concerned with filling missing values. However, the existence of missing values is synonymous with uncertainties not only over the distribution of missing values but also over target class assignments that require careful consideration. In this paper, we propose a simple and effective method for imputing missing features and estimating the distribution of target assignments given incomplete data. In order to make imputations, we train a simple and effective generator network to generate imputations that a discriminator network is tasked to distinguish. Following this, a predictor network is trained using the imputed samples from the generator network to capture the classification uncertainties and make predictions accordingly. The proposed method is evaluated on CIFAR-10 and MNIST image datasets as well as five real-world tabular classification datasets, under different missingness rates and structures. Our experimental results show the effectiveness of the proposed method in generating imputations as well as providing estimates for the class uncertainties in a classification task when faced with missing values.

**Index Terms**—Missing Data, Imputation, Incomplete Data, Generative Adversarial Networks, Classification Uncertainty.

## 1 INTRODUCTION

WHILE a large body of the machine learning literature is built upon the assumption of having access to complete datasets, in many real-world problems only incomplete datasets are available. The existence of missing values can be due to many different causes such as human subjects not adhering to certain questions or features not being collected frequently due to financial or experimental limitations, sensors failures, and so forth [1, 2, 3]. Data imputation techniques have been suggested as a solution to bridge this gap in the literature by replacing missing values with observed values.

Missing data imputation approaches can be categorized into single and multiple imputation methods. Single imputation methods try to replace each missing value with a plausible value that is the best fit given the value of other correlated features and knowledge extracted from the dataset [4, 5]. While these methods are easy to implement and use in practice, imputed values may induce bias by eliminating less likely but important values. Also, these methods do not suggest a way to measure to what extent the imputed values are representative of the missing values [6].

Multiple imputation (MI) techniques, as suggested by the name, try to use multiple imputed values to impute each missing value. The result would be having a set of imputed datasets that enables measuring how consistent and statistically significant are the results of the experiments [7]. While MI offers interesting statistical insights about the reliability of analysis on incomplete data, the insight is imprecise as it is mainly concerned about the population of data samples rather than individual instances. Specifically, MI methods reason about the statistical properties on a limited number of imputed datasets (less than 10 in most practical

implementations) on the population of samples within the dataset [8, 9].

The existence of missing values is synonymous with having uncertainty over these values that requires careful consideration. In many real-world applications, we are dealing with supervised problems that demand modeling and prediction based on incomplete data. Take for instance, prediction of class assignments given an image in which a large portion of the frame is missing. In such a scenario, based on the observed frame parts, there might be multiple probable class assignments each having a different likelihood. Here, we are not only interested in imputing missing values or measuring how robust our imputations are, but also it is highly desirable to measure the impact of missing values on the prediction outcome for each instance.

In this paper, we propose the idea of Generative Imputation and Stochastic Prediction (GI) as a novel approach to impute missing values and to measure class uncertainties arising from the distribution of missing values. The suggested approach is based on neural networks trained using an adversarial objective function. Additionally, a predictor is trained on the generated samples from the imputer network which is able to reflect the impact of uncertainties over missing values. This enables measuring different prediction outcomes and certainties for each specific instance. We evaluate the effectiveness of the proposed method on different incomplete image and tabular datasets under various missingness structures.<sup>1</sup>

## 2 RELATED WORKS

One of the simplest traditional methods for handling missing values includes imputing the occurrences of missing values with constant values such as zeros or using mean values. To enhance the accuracy of such imputations, alternatives such as k-nearest neighbors (KNN) [4] and maximum likelihood estimation (MLE) [5] have been suggested

• Authors are with the Department of Computer Science, University of California Los Angeles (UCLA), CA 90095.  
E-mail: mkachuee@cs.ucla.edu

Manuscript under review.

1. <https://github.com/mkachuee/GenerativeImputationStochasticPrediction>

to estimate values to be used given an observed context. While these methods are easy to implement and analyze, they often fail to capture the complex feature dependencies as well as structures present in many problems.

Rubin [7] suggested a categorization for missingness mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Under the assumption of MAR, the authors suggested multiple imputation (MI) as a stochastic imputation method. Here, instead of imputing missing values using a single value, several values are sampled to represent the distribution over the missing value. MI generates a few imputed complete datasets that are then used independently in statistical modeling [6, 8]. Recent work by Aleryani et al. [10] trains an ensemble of classifiers using bagging and stacking techniques based on multiple imputation of dataset samples, and studies the variance of the predictions made by each classifier. Usually, the final goal of MI is to measure the robustness of the final statistical analysis amongst the imputed datasets. In other words, it measures the quality of imputations and the statistical significance of analysis on the imputed data. It should be noted that the number of imputations used in MI is usually very limited. Also, often strong simplifying assumptions are made in modeling the data distribution (e.g., multi-variate Gaussian or Student's  $t$  distribution) which limit the applicability of this method [8, 9].

More recently, autoencoder architectures have been suggested as powerful density estimators capable of capturing complex distributions. Perhaps, denoising autoencoders (DAE) [11, 12] are one of the most intuitive approaches in which a neural network is trained to reconstruct and denoise its input. Following a more probabilistic perspective, variational autoencoders (VAE) [13] try to learn the data generating distribution via a latent representation. Specifically, conditional variational autoencoders (CVAE) [14] can be used to sample missing values conditioned on observed values. For instance, Mattei and Frellsen [15] suggested a method based on deep latent variable models and importance sampling that offers a tighter likelihood bound compared to the standard VAE bound. While these methods are powerful generative models applicable to missing data imputation, often samples generated using autoencoders are biased toward the mode of the distribution (e.g., resulting in blurry images, for vision tasks) [16, 17, 18].

Recently, due to the success of generative adversarial networks (GAN), there has been great attention toward applying them to impute missing values. For instance, Yoon et al. [19] suggested an imputation method based on adversarial and reconstruction loss terms. Li et al. [20] introduced the idea of using separate generator and discriminator networks to learn the missing data structure and data distribution. These methods have been quite successful and are able to present the state-of-the-art results. Though it should be noted that often the presence of additional loss terms may bias the generated samples toward the mode of the distribution being modeled. Also, these methods are often complicated to be applied in practical setups by practitioners. For instance, Yoon et al. [19] requires setting hyperparameters to adjust the influence of an MSE loss term as well as the rate of discriminator hint vectors. Also as

another example, Li et al. [20] uses three generators and three discriminators for the final imputer architecture.

From the perspective of supervised analysis, imputation and handling missing values are usually considered as a preprocessing step. A few exceptions exist such as Bayesian models and decision trees that permit direct analysis on incomplete data [21, 22]. Note that while certain Bayesian methods such as probabilistic Bayesian networks allow handling of missing values as unobserved variables. However, given an incomplete training dataset and without any known causal structure as a priori, learning such models is a very challenging problem with the complexity of at least NP-complete to learn the network architecture in addition to an iterative EM optimization to learn model parameters [23, 24].

Tran et al. [25] suggested a genetic programming method using multiple imputation to train a set of classifiers covering different combinations of observed features. While this method does not require any imputation at the prediction phase, it has significant limitations in the scale of the problems (i.e. the number of features/samples) that can be addressed due to the often combinatorial number of classifiers required.

We argue that the simplistic approach of imputing missing values as a preprocessing step discards uncertainties that exist in original incomplete data samples. Instead, there is a need for methods that reflect these uncertainties on the final predicted target distribution. This work suggests the idea of training a predictor on different imputed samples to capture the uncertainties over class assignments. Compared to MI, the suggested method interleaves imputation and training a downstream prediction model, enabling to estimate classification uncertainties for each instance.

### 3 PROPOSED METHOD

#### 3.1 Problem Definition

We make the general assumption of having access to an incomplete dataset  $\mathcal{D}$  consisting of a set of feature vector, mask vector, and target class pairs  $(\mathbf{x}_i, \mathbf{k}_i, y_i)$ . For each feature vector,  $\mathbf{x}_i \in \mathbb{R}^d$ , only a subset of the features is available. The mask vector  $\mathbf{k}_i \in \{0, 1\}^d$  is used to indicate available features and missing features by ones and zeros, respectively. Here, to represent features as fixed-width vectors, arbitrary (or *NaN*) values are used to fill missing values. Also, for convenience, we often use  $\mathbf{x}_i^{obs}$  and  $\mathbf{x}_i^{miss}$  to refer to the set of observed and missing features for the feature vector  $\mathbf{x}_i$ . Note that the  $(\mathbf{x}_i^{obs}, \mathbf{x}_i^{miss})$  notation does not use vectors for representation and instead is using sets for a more abstract representation rather than the fixed-length vector notation of  $(\mathbf{x}_i, \mathbf{k}_i)$ .

We define our objective in two steps: (i) Imputing missing values via sampling from the conditional distribution of missing features given observed features i.e.,  $P(\mathbf{x}_i^{miss}|\mathbf{x}_i^{obs})$ . (ii) Estimating the distribution of target classes given the observed features and the distribution of missing features i.e.,  $P(y|\mathbf{x}_i^{obs}, \mathbf{x}_i^{miss})$ . For the first part, we are interested in sampling from the conditional distribution rather than finding the mode of the distribution as the most probable imputation. Similarly, for the second part, we are interested in obtaining a distribution over the possible target

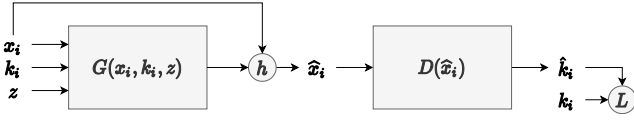


Fig. 1: Block diagram of the proposed adversarial imputation method.  $h$  represents the blending function of (1), and  $L$  is the adversarial loss function of (2).

assignments and the confidence of each class rather than maximum likelihood class assignments.

### 3.2 Generative Imputation

To generate samples from the distribution of missing features conditioned on the observed features, we follow the idea first suggested by Yoon et al. [19]. In this paradigm, a generator network is responsible for generating imputations while a discriminator is trying to distinguish imputed features from observed features (see Figure 1).

Specifically, the generator function  $G(x_i, k_i, z) \in \mathbb{R}^d$  generates an imputed feature vector, based on observed features, the corresponding mask, and a Gaussian noise vector ( $z$ ). Here,  $x_i$  is not revealing any information about missing values as they are represented by invalid values in  $x_i$  and are indicated by the mask vector  $k_i$ . In order to achieve the final imputed vector,  $\hat{x}_i$ , we blend (or, merge) the output of the generator with the input features to replace generated values with the exact values of observed features:

$$\hat{x}_{i,j} = \begin{cases} x_{i,j} & \text{if } k_{i,j} = 1 \\ G(x_i, k_i, z)_j & \text{if } k_{i,j} = 0 \end{cases}, \quad (1)$$

where  $x_{i,j}$  refers to  $j$ 'th feature of sample  $i$ . Also, note that by sampling  $z$  multiple times, we can obtain different imputation samples from the conditional distribution indicated by  $\hat{x}_i^l$  where  $l$  is the sample number.

A discriminator network,  $D(\hat{x}_i)$ , is trained to distinguish real and imputed features by generating a predicted softmax mask output,  $\hat{k}_i$ . Here a binary cross-entropy loss per mask element is used as the adversarial objective function:

$$\begin{aligned} \max_G \min_D L(G, D) = \\ \mathbb{E}_{k \sim D, \hat{k} \sim D(G(x, k, z))} [k^T \log(\hat{k}) + (1 - k)^T \log(1 - \hat{k})]. \end{aligned} \quad (2)$$

The intuition behind this adversarial loss function is that given a generator function which captures the data distribution successfully, the discriminator would not be able to distinguish the parts of the feature vector that were originally missing.

Compared to Yoon et al. [19], the objective function of (2) does not have an MSE loss term. Instead, we use recent advances in GAN stabilization and training to improve the training process (see Section 3.4). While it is quite prevalent in the adversarial learning literature to use additional loss terms such as mean squared error (MSE) to enhance the quality of generated samples, we decided to keep our solution as simple as possible. Additionally, in our experiments, we provide supporting evidence that this simple loss function enables us to sample from the conditional distribution and prevents biased inclinations toward distribution modes.

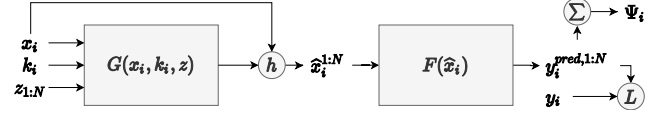


Fig. 2: Block diagram of the proposed stochastic prediction method.  $G$  represents a trained generative imputer (Section 3.2),  $L$  is the prediction loss function, and  $\Psi$  is the estimated classification certainty defined in (7).

### 3.3 Stochastic Prediction

To capture the distribution of target classes given incomplete data, we suggest the idea of stochastic prediction. As indicated in the previous section, the generator can be used to sample from the conditional distribution. Here, a predictor is trained based on the imputed samples to predict class assignments and to calculate the confidence of these assignments (see Figure 2). For instance, for a specific test sample at hand, if a certain missing feature is a strong indicator of the target class, we would like to observe the impact of different imputations for that feature on the final hypothesis.

Formally, we are interested in finding the certainty of class assignments given observed features:

$$\Psi = P(y|x_i^{obs}). \quad (3)$$

Here,  $\Psi$  is a vector where each element is representing a certain class. Rewriting (3) as a marginal we have:

$$\Psi = \int P(x_i^{miss}) P(y|x_i^{obs}, x_i^{miss}) dx_i^{miss}. \quad (4)$$

Approximating the integration using a summation, given enough samples,  $\Psi$  can be estimated by:

$$\Psi \approx \frac{1}{N} \sum P(y|x_i^{obs}, \hat{x}_i^{miss}), \quad (5)$$

where  $\hat{x}_i^{miss}$  are samples taken from the conditional distribution of missing features given observed ones. We use the suggested generative imputation method to generate samples required for this approximation. Rewriting (1) using Hadamard product and as function of the noise vector:

$$\hat{x}_i = k_i \odot x_i + (1 - k_i) \odot G(x_i, k_i, z) \quad (6)$$

Assuming that a predictor,  $F_\theta$ , is available which predicts class assignments for a complete feature vector,  $\Psi$  can be estimated as:

$$\Psi = \mathbb{E}_z[F_\theta(\hat{x}_i)] \approx \frac{1}{N} \sum_{l=1}^N F_\theta(\hat{x}_i^l). \quad (7)$$

Algorithm 1 presents the suggested algorithm for training the predictor. It consists of taking samples from the incomplete dataset, then imputing them using our generator network, and using the imputed samples to update the predictor. Note that, on each epoch and for each sample, the generator generates a new sample from the conditional distribution. Intuitively, it means that the predictor observes and learns to operate under different imputations for a given sample. This is different from approaches such as multiple imputation where several predictors are trained on different imputed versions of a dataset.

**Algorithm 1: Training the predictor.**


---

**Input:**  $G$  (trained imputer),  $\mathcal{D}$  (dataset)  
**Output:**  $F_\theta$  (trained predictor)  
**foreach** *Training Epoch* **do**  
  **foreach**  $(\mathbf{x}_i, \mathbf{k}_i, y_i)$  **in**  $\mathcal{D}$  **do**  
     $\mathbf{z} \sim N(0, I)$   
     $\hat{\mathbf{x}}_i \leftarrow \mathbf{k}_i \odot \mathbf{x}_i + (1 - \mathbf{k}_i) \odot G(\mathbf{x}_i, \mathbf{k}_i, \mathbf{z})$   
     $y_i^{pred} \leftarrow F_\theta(\hat{\mathbf{x}}_i)$   
     $loss \leftarrow L(y_i, y_i^{pred})$   
    Backpropagate  $loss$   
  Update  $F_\theta$

---

**Algorithm 2: Estimating target distributions.**


---

**Input:**  $G$  (trained imputer),  $F_\theta$  (trained predictor),  
 $(\mathbf{x}, \mathbf{k})$  (test sample),  $N$  (ensemble samples)  
**Output:**  $\Psi$  (distribution over target classes)  
 $\Psi \leftarrow \text{zeros} \in R^{\#classes}$   
**foreach** *Ensemble Sample 1 to N* **do**  
   $\mathbf{z} \sim N(0, I)$   
   $\hat{\mathbf{x}} \leftarrow \mathbf{k} \odot \mathbf{x} + (1 - \mathbf{k}) \odot G(\mathbf{x}, \mathbf{k}, \mathbf{z})$   
   $y^{pred} \leftarrow F_\theta(\hat{\mathbf{x}})$   
   $j \leftarrow \text{argmax}(y^{pred})$   
   $\Psi_j \leftarrow \Psi_j + \frac{1}{N}$

---

Algorithm 2 presents the suggested algorithm for making predictions and estimating target distributions given a trained predictor model. Here, a sample is imputed  $N$  times and inference on this set results in an ensemble of predictions over different imputations. The output of this algorithm can be interpreted as a distribution over the confidence of class assignments given a partially observed test sample. The following claims justify the validity of Algorithm 1 and Algorithm 2.

**Claim 1.** (Generalization of the predictor). *If we assume imputed  $\hat{\mathbf{x}}_i$ s are samples from the underlying feature distribution, then the assigned training set labels can be modeled as labels generated from a noisy labeling process.*

Claim 1 permits the analysis of the generalization and convergence for the predictor trained using Algorithm 1 based on current literature in training models with noisy labels [26, 27, 28]. From the analysis provided by Chen et al. [28], test accuracy in asymmetric label noise conditions is a quadratic function of the label noise:

$$P(y_i = \hat{y}_i) = (1 - \epsilon)^2 + \epsilon^2, \quad (8)$$

where  $\hat{y}_i$  is underlying true label for the imputed feature vector ( $\hat{\mathbf{x}}_i$ ), and  $y_i$  is the label provided by the incomplete dataset. In (8), label noise ratio,  $\epsilon$ , represents the probability of the label transition from a certain target class to another:

$$\epsilon = 1 - P(\hat{y}_i = j | y_i = j). \quad (9)$$

In practice,  $\epsilon$  is determined by the problem-specific underlying data distribution as well as the distribution of missing values.

Justification for claim 1 is straightforward, assume that  $\{\hat{y}_i^1 \dots \hat{y}_i^N\}$  are underlying true labels for each of  $\{\hat{\mathbf{x}}_i^1 \dots \hat{\mathbf{x}}_i^N\}$ . During training, for any imputed sample in

$\{\hat{\mathbf{x}}_i^1 \dots \hat{\mathbf{x}}_i^N\}$ , we use the dataset provided label,  $y_i$ , to calculate the loss and to update model parameters:

$$Loss_i = \sum_{l=1}^N L(y_i, F_\theta(\hat{\mathbf{x}}_i^l)). \quad (10)$$

In the case that any of  $\{\hat{y}_i^1 \dots \hat{y}_i^N\}$  is different from  $y_i$ , the loss term corresponding to that term would be calculated using a wrong label. Here, if we consider the average impact on gradients for batches of samples rather than individual cases, the overall impact on training would be very similar to the case of training using noisy labels:

$$Loss = \sum_{i=1}^{|\mathcal{D}|} \sum_{l=1}^N L(y_i, F_\theta(\hat{\mathbf{x}}_i^l)), \quad (11)$$

where  $|\mathcal{D}|$  is the number of dataset samples. Further, in this case, we can find the average label noise as:

$$\epsilon = \frac{\sum_{i=1}^{|\mathcal{D}|} \sum_{l=1}^N 1(y_i \neq \hat{y}_i^l)}{|\mathcal{D}| \cdot N} \quad (12)$$

**Claim 2.** (Approximation of the target distribution). *If we assume:*

- (i) imputed  $\hat{\mathbf{x}}_i$ s are valid samples from the underlying feature distribution:  $\hat{\mathbf{x}}_i \sim P(\mathbf{x} | \mathbf{x}_i^{obs})$ ,
  - (ii) a good predictor can be trained using the incomplete data (claim 1),
  - (iii) enough samples are used and the Monte Carlo estimator is unbiased:  $\frac{1}{N} \sum_{l=1}^N F_\theta(\hat{\mathbf{x}}_i^l) \rightarrow \mathbb{E}_z[F_\theta(\hat{\mathbf{x}}_i)]$  for  $N \rightarrow \infty$ ,
- then the target distribution,  $\Psi$ , can be estimated accurately.

This claim supports Algorithm 2 that is suggested to estimate the target distribution given a partially observed feature vector.

The first assumption is consistent with the theoretical analysis of generative adversarial networks that they can converge to the true underlying distribution [29, 30]. The second assumption is supported by Claim 1. Regarding the last assumption, each sample requires one forward computation of the generator and predictor networks which, based on the scalability of current network architectures, usually permits thousands of samples to be taken at a reasonable computational cost.

### 3.4 Implementation Details

As we conduct experiments on image and tabular datasets, we use different architectures for each. For image datasets, we used a generator and discriminator architectures similar to the ones suggested by Wang et al. [31]. However, we improved these architectures using self-attention layers [32]. It should be noted that, while Zhang et al. [32] suggests using a single self-attention layer in the middle of the network, we observed consistent improvements by inserting multiple self-attention layers before each residual block within the network. Furthermore, as input to the generator, we concatenate input image, mask, and a random  $\mathbf{z}$  frame along the channels dimension and use it as input. For tabular datasets, we use a simple 4 layer network consisting of fully-connected and batch-norm layers. Also, the input to the generator is the concatenation of a feature vector, mask vector, and a  $\mathbf{z}$  vector of size  $\frac{1}{8}$  of the input. For all experiments, we use an ensemble size ( $N$ ) equal to 128.

We used Adam [33] for model optimization. Two time-scale update rule (TTUR) [34] was used to balance training the generator and discriminator networks. We explored best TTUR learning-rate settings from the set of  $\{0.001, 0.0005, 0.0001, 0.00005\}$ . Here, Adam parameters  $\beta_1$  and  $\beta_2$  are set to 0.5 and 0.999, respectively. Also, spectral normalization was used to stabilize both the generator and discriminator network in our experiments with image data [35]. For the predictor network, we used the default Adam settings as suggested by Kingma and Ba [33]. In all training procedures, we decay learning rate by a factor of 5 after reaching a plateau. For all experiments, we use a batch size of 64. Based on our experiments, we found that pretraining the discriminator while fixing the generator network for the first 5% of the training epochs helps the stability of training.

Further detail on exact architectures, experiments, etc. as well as additional results and ablation studies is provided in the appendices.

## 4 EXPERIMENTS

### 4.1 Datasets

To evaluate the proposed method we use CIFAR-10 [36] and MNIST benchmark [37] as image classification datasets as well as five non-image datasets: UCI Landsat [38]<sup>2</sup>, MIT-BIH arrhythmia [39], Diabetes, Cholesterol, and Hypertension classification [40]<sup>3</sup>. CIFAR-10 dataset consists of 60,000 32x32 images from 10 different classes. For this task, we use train and test sets as provided by the dataset. As a preprocessing step, we normalize pixel values to the range of [0,1] and subtract the mean image. The only data augmentation we use for this task is to randomly flip training images for each batch.

UCI Landsat consists of 6435 samples of 36 features from 6 different categories. We follow the same train and test split as provided by the dataset. MIT-BIH dataset consists of annotated heartbeat signals from which we used the preprocessed version available online<sup>4</sup> consisting of 92,062 samples of 5 different arrhythmia classes. Diabetes dataset is a real-world health dataset of 92,062 samples and 45 features from different categories such as questionnaire, demographics, medical examination, and lab results. The objective is to classify between three different diabetes conditions i.e., normal, pre-diabetes, and diabetes. Similarly, Cholesterol and Hypertension datasets have about 120 features and 50,000 samples each [40]. As MIT-BIH, Diabetes, Cholesterol, and Hypertension datasets do not provide explicit train and test sets, we randomly select 80% of samples as a training set and the rest as a test set. To preprocess our tabular datasets, statistical and unity based normalization are used to balance the variance of different features and center them around zero. Also, while different encoding and representation methods are suggested in the literature to handle categorical features [41, 42], in this paper, we take the simple approach of encoding categorical variables using one-hot representation and smoothing them by adding Gaussian noise with zero mean and variance equal to 5% of feature

variances. In our experiments, we observed a reasonable performance using the suggested simple smoothing; however, more advanced encoding methods are also applicable in this setup and can be applied to enhance the performances even further.

### 4.2 Missingness Mechanisms

In our experiments, we consider MCAR uniform and MCAR rectangular missingness structures. In MCAR uniform, each feature of each sample is missing based on a Bernoulli distribution with a certain missingness probability (i.e., missing rate) independent of other features. In addition to the case of uniform missingness, for image tasks, we use rectangular missingness/observation structure where rectangular regions of dataset images are missing/observed. To control the rate of missingness and decide on the regions that are missing for each case, we use a latent beta distribution that samples rectangular region's width and height such that the average missing rate is maintained. For missing rates less than 50% we make the assumption of having a random rectangular region to be missing, whereas for missing rates more than 50% we assume that only a random rectangular region is observed and the rest of the image is missing.

We would like to note that while the suggested solution in this paper is readily compatible with MAR structures, in our experiments, to simplify the presentation of results and to have a fair comparison with other work that does not support the MAR assumption, we limited the scope of our experiments to MCAR. Furthermore, to simulate incomplete datasets and to make sure the same features are missing without explicitly storing masks, we use hashed feature vectors to seed random number generators used to sample missing features. More detail is provided in Appendix B.

### 4.3 Evaluation Measures

Frchet inception distance (FID) [34] score is used to measure the quality of missing data imputation in experiments with images<sup>5</sup>. We also considered using root means squared error (RMSE); however, we decided to only include this result in the appendices as we observed an inconsistent behavior using RMSE in our comparisons as RMSE favors methods that show less variance rather than realistic and sharp samples from the distribution. Also, for each dataset and each missingness scenario, we report top-1 classification accuracy based on the majority vote estimated using Algorithm 2. Another measure that we use in this paper is the comparison between the estimated target certainties and average accuracies achieved for each confidence assignment. We run each experiment multiple times: 4 times for CIFAR-10 and 8 times for tabular datasets. We report the mean and standard deviation of results for each case.

We compare our results with MisGAN [20] and GAIN [19] as the state of the art imputation algorithms based on GANs as well as basic denoising autoencoder (DAE) [11] and multiple imputation by chained

2. [https://archive.ics.uci.edu/ml/datasets/Statlog+\(Landsat+Satellite\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite))

3. <https://github.com/mkachuee/Opportunistic>

4. <https://www.kaggle.com/shayanfazeli/heartbeat>

5. <https://github.com/mseitzer/pytorch-fid> is adapted to measure the FID scores.

equations (MICE) [43] as baselines. For experiments using MisGAN, we used the same architectures and hyper-parameters as suggested by the MisGAN authors<sup>6</sup>. The only modification was to adapt the last generator layer to generate images with resolutions as we use. Regarding GAIN, we used the same network architecture as our implementation of GI and hyper-parameters as used by the GAIN authors<sup>7</sup>. In the DAE implementation, due to the incomplete data assumption, only observed features appear in the loss function, ignoring reconstruction terms corresponding to missing features. Due to scalability issues, we were only able to use MICE for the smaller non-image datasets. For these methods, to train and evaluate classifiers, we use predictors trained on imputed datasets rather than the stochastic predictor suggested in Algorithm 1.

#### 4.4 Results

Figure 3 presents the comparison of FID scores on the CIFAR-10 dataset at different missing rates for uniform and rectangular missingness. As it can be inferred from these plots, GI outperforms other alternatives in all cases. Also, it can be seen that GAIN is able to provide more reasonable results for uniform missing data structure compared to MisGAN which is mainly effective in the rectangular missing data structure. One possible explanation for this behavior might be the fact that GAIN has an MSE loss term acting similar to an autoencoder loss smoothing noisy missing pixels. On the other hand, MisGAN tries to explicitly model missingness structure and is more successful in capturing a more structured missingness such as the case of a rectangular structure. Table 1 provides a comparison between the top-1 classification accuracy achieved using each method at different missing rates and structures. From this table, GI outperforms other work by achieving the best results in 5 out of 6 cases<sup>8</sup>.

Table 2 presents a comparison of classification accuracies for Landsat, MIT-BIH, Diabetes, Cholesterol, Hypertension, and MNIST datasets at different missing rates. In the Landsat, Cholesterol, Hypertension, and MNIST benchmarks, GI outperforms other work in all cases. Regarding the MIT-BIH experiments, GI outperforms other work for missing rates more than 30% while achieving similar accuracies to GAIN for lower missing rates. In the diabetes classification task, GI appears to be most effective imputing missing rates more than 20%.

Figure 4 shows a comparison of accuracy versus certainty plots for GI, MisGAN, and GAIN on Landsat dataset at different missing rates. To generate these figures we trained each imputation method and then used Algorithm 1 to train predictors on imputed samples. Finally, Algorithm 2 used to measure the average accuracy at different prediction confidence levels based on a sample of 128 imputations for each test example. As it can be seen from the plots, GI provides results closest to the ideal case of having

average confidence values equal to average accuracies. As suggested in (7) and supported by the experimental results, the proposed method is better calibrated compared to the traditional approach of imputing each sample as a preprocessing step prior to the prediction, ignoring the imputation uncertainties.

#### 4.5 Visualization using Synthesized Data

In order to provide further insight into the operation of GI and how imputations can potentially influence the outcomes of predictions, we conduct experiments on a synthesized dataset. The original underlying data distribution is generated by sampling 5000 samples from 4 Gaussians of standard deviation 0.1 centered on the vertices of a unit square. We assign two different classes to each cluster such that diagonal vertices are of the same class (see Figure 5a, classes are represented with colors). From this underlying distribution, we make an incomplete dataset with 50% of values missing.

The incomplete synthesized dataset is used to train GI and other imputation methods. We take a random test sample in which the second feature has a value of about 0.1 and the other feature is missing. Ideally, in the imputation phase, we would like to sample from the condition distribution i.e.  $P(x_1|x_2 = 0.1)$  (see Figure 5b). Here, in the prediction phase, an ideal method would decide on not making a confident classification and report the uncertainty. Note that solely observing the value of 0.1 for the second feature does not provide any useful evidence for the prediction. Figure 5c-f provide samples and classification results for GI, MisGAN, GAIN, and DAE. As it can be inferred from these figures, GI generates samples relatively similar to samples from the conditional distribution, and it also reflects this uncertainty over the prediction. On the other hand MisGAN, probably due to its complexity of using three different generators and discriminator pairs, is suffering from mode collapse to a higher degree and is unable to generate samples from the other class, resulting in over-confident assignments. Note that mode collapse is a well-known shortcoming of GANs, and although we observed a better behavior in our models, the results do not perfectly match the ground-truth distribution. GAIN, perhaps due to the MSE loss terms, is inclined towards the mean of the conditional distribution at the origin. DAE, as expected, due to its MSE loss term, only captures the expected value of the distribution mean hence reducing the MSE error and generates over-smoothed imputations.

### 5 ABLATION STUDY

Figure 6 presents a comparison between using (GI W/ Atten.) and not using (GI W/O Atten.) self-attention layers before each residual block in the proposed architecture. We report FID scores on CIFAR-10 with rectangular missingness. As it can be inferred from this comparison, using self-attention achieves a consistent improvement over the baseline. We also examined the case of uniform missingness; however, we did not observe any significant improvement for this case. One possible explanation could be the fact that

6. <https://github.com/steveli/misgan>

7. <https://github.com/jsyoon0823/GAIN>

8. An earlier version of this paper reported results that are different from the current manuscript. The current version is using the stochastic predictor exclusively on the suggested imputation method and trained using more precise hyper-parameter settings.

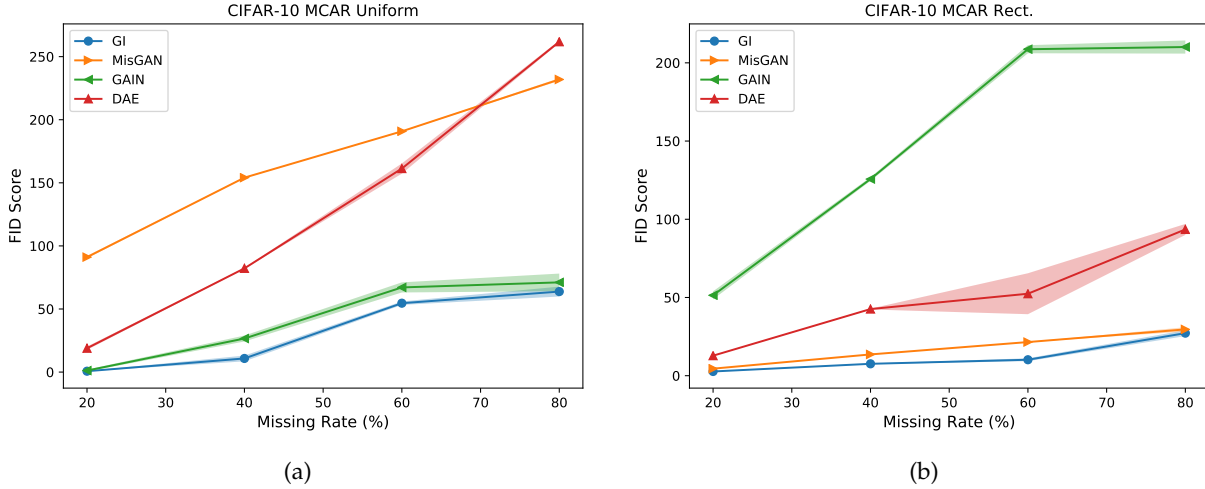


Fig. 3: Comparison of FID scores on CIFAR-10 dataset for (a) uniform and (b) rectangular missingness. Lower FID score is better. In many cases, variance values are very small and only observable by magnifying the figures.

TABLE 1: Top-1 CIFAR-10 classification accuracy for different missing rates and structures.

Method	Accuracy at Missing Rate (%)					
	MCAR Uniform			MCAR Rect.		
	20%	40%	60%	20%	40%	60%
GI	89.5 ( $\pm 0.45$ )	87.1 ( $\pm 0.54$ )	80.3 ( $\pm 0.26$ )	84.0 ( $\pm 0.03$ )	76.9 ( $\pm 0.03$ )	66.1 ( $\pm 0.16$ )
MisGAN	86.5 ( $\pm 0.31$ )	83.7 ( $\pm 0.40$ )	78.7 ( $\pm 0.26$ )	82.9 ( $\pm 0.44$ )	75.6 ( $\pm 0.20$ )	65.0 ( $\pm 0.31$ )
GAIN	88.7 ( $\pm 0.45$ )	86.0 ( $\pm 0.86$ )	81.8 ( $\pm 0.03$ )	81.7 ( $\pm 0.03$ )	73.6 ( $\pm 0.35$ )	58.4 ( $\pm 1.66$ )
DAE	88.0 ( $\pm 0.22$ )	84.0 ( $\pm 0.50$ )	79.8 ( $\pm 0.71$ )	83.3 ( $\pm 0.64$ )	75.5 ( $\pm 0.44$ )	63.8 ( $\pm 0.24$ )
Mean	85.7 ( $\pm 0.02$ )	83.4 ( $\pm 0.38$ )	79.2 ( $\pm 0.16$ )	82.7 ( $\pm 0.15$ )	75.3 ( $\pm 0.16$ )	64.0 ( $\pm 0.32$ )

imputing missing data with a uniform structure can be done by processing local regions and does not require attending to different distant regions across the image.

Figure 7 shows a comparison of classification accuracies for the Landsat dataset achieved using different ensemble sizes ( $N$ ). As it can be seen from this figure, higher values of  $N$  result in improved accuracies, especially for higher missing rates. Also, it can be observed that for  $N$  values more than 64 the difference is negligible.

To study the benefits of the suggested stochastic predictor, we conducted experiments comparing GI with its non-stochastic variation ( $N=1$ ). Here, the CIFAR-10 dataset with the rectangular missing structure and missing rates from 20% to as high as 90% is used. From Table 3 it can be inferred that as the rate of missingness increases, the benefits of the suggested predictor algorithm increase significantly. We hypothesize that at higher rates of missingness, the conditional distribution of missing features becomes multimodal. In such a scenario, the suggested method captures the uncertainties over the target distribution resulting in the predictor to make more reliable class assignments.

## 6 DISCUSSION

In the literature, despite the prevalence of missing values in many real-world scenarios, there has been less attention towards learning from incomplete datasets. Often handling missing values is being addressed as a preprocessing step

followed by typical predictor models. However, this simplistic approach to handle missing values may induce biases during the training due to the fact the subsequent model cannot distinguish imputed values and truly observed values fed as input [44]. Moreover, regardless of how good we impute missing features, the fact that a certain feature is missing bears an uncertainty about the values that feature may take. This uncertainty in the feature domain entails an uncertainty in the target assignments. In many real-world applications, it is of paramount importance to prevent biased predictions and estimate the prediction confidence. For instance, in health datasets which often contain many missing features per sample, it is critical to not only make predictions that are accurate on average but also reflect the confidence of certain diagnosis for a specific patient. It might be the case that a missing feature taking a critical value is less frequent but drastically impactful on the final outcome.

To address these issues, this paper suggests (i) a method consisting of a GAN-based imputer trained on incomplete data that is able to generate high-quality imputations from the conditional distribution of missing features given the observed ones. (ii) A predictor which is trained on samples generated by the imputer, which is capable of estimating the certainty of class assignments.

## 7 CONCLUSION

In this paper, we proposed a novel method to generate imputations and measure uncertainties over target class

TABLE 2: Comparison of classification accuracies at different missing rates.

Dataset	Method	Accuracy at Missing Rate (%) <sup>a</sup>			
		10%	20%	30%	40%
Landsat [38]	GI	<b>89.9</b> ( $\pm 0.36$ )	<b>89.6</b> ( $\pm 0.36$ )	<b>89.0</b> ( $\pm 0.03$ )	<b>88.0</b> ( $\pm 0.22$ )
	MisGAN	87.2 ( $\pm 0.01$ )	85.7 ( $\pm 0.19$ )	84.0 ( $\pm 0.61$ )	82.9 ( $\pm 0.75$ )
	GAIN	89.7 ( $\pm 0.42$ )	89.4 ( $\pm 0.56$ )	88.4 ( $\pm 0.71$ )	87.7 ( $\pm 0.10$ )
	DAE	89.4 ( $\pm 0.10$ )	88.6 ( $\pm 0.54$ )	87.5 ( $\pm 0.14$ )	86.6 ( $\pm 0.21$ )
	MICE	89.5 ( $\pm 0.16$ )	89.3 ( $\pm 0.10$ )	88.1 ( $\pm 0.49$ )	87.5 ( $\pm 0.03$ )
MIT-BIH [39]	GI	<b>98.5</b> ( $\pm 0.02$ )	<b>98.4</b> ( $\pm 0.03$ )	<b>98.2</b> ( $\pm 0.07$ )	<b>97.7</b> ( $\pm 0.03$ )
	MisGAN	97.8 ( $\pm 0.13$ )	97.4 ( $\pm 0.07$ )	96.7 ( $\pm 0.07$ )	96.2 ( $\pm 0.09$ )
	GAIN	<b>98.5</b> ( $\pm 0.02$ )	<b>98.4</b> ( $\pm 0.06$ )	98.0 ( $\pm 0.09$ )	97.5 ( $\pm 0.18$ )
	DAE	98.4 ( $\pm 0.02$ )	98.2 ( $\pm 0.11$ )	97.9 ( $\pm 0.09$ )	97.4 ( $\pm 0.02$ )
	MICE	98.4 ( $\pm 0.01$ )	98.3 ( $\pm 0.01$ )	98.1 ( $\pm 0.01$ )	97.5 ( $\pm 0.12$ )
Diabetes [40]	GI	89.6 ( $\pm 0.13$ )	<b>89.0</b> ( $\pm 0.03$ )	<b>88.2</b> ( $\pm 0.62$ )	<b>86.8</b> ( $\pm 0.38$ )
	MisGAN	89.7 ( $\pm 0.01$ )	88.9 ( $\pm 0.30$ )	87.6 ( $\pm 0.02$ )	86.4 ( $\pm 0.68$ )
	GAIN	89.2 ( $\pm 0.09$ )	88.3 ( $\pm 0.02$ )	86.9 ( $\pm 0.09$ )	83.8 ( $\pm 1.44$ )
	DAE	89.3 ( $\pm 0.05$ )	88.2 ( $\pm 0.19$ )	86.9 ( $\pm 0.09$ )	84.8 ( $\pm 0.03$ )
	MICE	<b>89.8</b> ( $\pm 0.08$ )	88.8 ( $\pm 0.01$ )	88.0 ( $\pm 0.08$ )	86.1 ( $\pm 0.02$ )
Cholesterol [40]	GI	<b>73.2</b> ( $\pm 0.12$ )	<b>72.2</b> ( $\pm 0.14$ )	<b>71.6</b> ( $\pm 0.30$ )	<b>70.4</b> ( $\pm 0.20$ )
	MisGAN	72.8 ( $\pm 0.31$ )	71.6 ( $\pm 0.13$ )	70.7 ( $\pm 0.15$ )	69.9 ( $\pm 0.13$ )
	GAIN	72.8 ( $\pm 0.22$ )	71.7 ( $\pm 0.27$ )	71.2 ( $\pm 0.05$ )	70.1 ( $\pm 0.08$ )
	DAE	73.0 ( $\pm 0.19$ )	71.6 ( $\pm 0.24$ )	70.8 ( $\pm 0.33$ )	70.2 ( $\pm 0.04$ )
	MICE	71.2 ( $\pm 0.10$ )	69.9 ( $\pm 0.13$ )	68.7 ( $\pm 0.02$ )	67.3 ( $\pm 0.21$ )
Hypertension [40]	GI	<b>77.8</b> ( $\pm 0.15$ )	<b>77.3</b> ( $\pm 0.32$ )	<b>77.2</b> ( $\pm 0.30$ )	<b>76.2</b> ( $\pm 0.07$ )
	MisGAN	77.0 ( $\pm 0.21$ )	76.9 ( $\pm 0.16$ )	76.1 ( $\pm 0.04$ )	75.4 ( $\pm 0.49$ )
	GAIN	77.5 ( $\pm 0.10$ )	76.8 ( $\pm 0.25$ )	76.6 ( $\pm 0.37$ )	75.8 ( $\pm 0.14$ )
	DAE	77.5 ( $\pm 0.30$ )	76.8 ( $\pm 0.11$ )	76.4 ( $\pm 0.58$ )	76.1 ( $\pm 0.05$ )
	MICE	76.7 ( $\pm 0.19$ )	75.9 ( $\pm 0.08$ )	74.7 ( $\pm 0.20$ )	73.0 ( $\pm 0.16$ )
MNIST [37]	GI	<b>99.0</b> ( $\pm 0.01$ )	<b>98.9</b> ( $\pm 0.07$ )	<b>98.7</b> ( $\pm 0.01$ )	<b>98.6</b> ( $\pm 0.01$ )
	MisGAN	98.6 ( $\pm 0.02$ )	98.3 ( $\pm 0.06$ )	98.1 ( $\pm 0.02$ )	97.5 ( $\pm 0.06$ )
	GAIN	98.8 ( $\pm 0.02$ )	98.7 ( $\pm 0.03$ )	98.6 ( $\pm 0.02$ )	98.5 ( $\pm 0.03$ )
	DAE	98.8 ( $\pm 0.08$ )	98.7 ( $\pm 0.03$ )	98.5 ( $\pm 0.08$ )	98.0 ( $\pm 0.06$ )
	MICE	98.7 ( $\pm 0.02$ )	98.6 ( $\pm 0.06$ )	98.4 ( $\pm 0.07$ )	98.3 ( $\pm 0.06$ )

<sup>a</sup>. Baseline accuracies for complete datasets (zero missing rate) are Landsat:90.9%, MIT-BIH:98.6%, Diabetes:90.7%, Cholesterol:73.6%, Hypertension:77.9%, MNIST:99.2%.

TABLE 3: Comparison of CIFAR-10 accuracies for the stochastic (N=128) and the deterministic (N=1) predictor under rectangular missingness.

Method	Accuracy at Missing Rate (%)					
	20%	40%	60%	70%	80%	90%
GI (N=128)	84.0	76.9	66.1	59.1	46.0	32.1
GI (N=1)	83.6	75.7	65.1	56.7	42.8	29.4
% difference (normalized)	0.5	1.6	1.5	4.1	6.9	8.4

assignments based on incomplete feature vectors. We evaluated the effectiveness of the suggested approach on image and tabular data via using different measures such as FID distance, classification accuracy, and confidence versus accuracy plots. According to the experiments, the proposed method not only can generate accurate imputations but also is able to model prediction uncertainties arising from missing values. The proposed method is applicable to many real-world applications where only an incomplete dataset is available, and modeling classification uncertainties is a necessity.

## APPENDIX A NETWORK ARCHITECTURES

Table 4 shows the exact architectures used in this paper. To show each layer or block we used the following notation.  $C \times S y P z - t$  represents a 2-d convolution layer of kernel size  $x$ , stride  $y$ , padding  $z$ , and number of output channels  $t$  followed by ReLU activation.  $Attn$  represents a self-attention layer similar to Zhang et al. [32].  $R - x$  represents a residual block consisting of two 2-d convolutions with kernel size 3 (padding size 1), batch normalization, and ReLU activation.  $CT \times S y P z - t$  is the convolution transpose corresponding to  $C \times S y P z - t$ .  $FC - x$  is representing a linear fully-connected layer of  $x$  output neurons with biases. We use spectral normalization as suggested by [35] for all convolutional layers in both generator and discriminator networks.

## APPENDIX B MISSING DATA MECHANISMS

In this paper, we conduct experiments on two mechanisms for missing values: MCAR uniform and MCAR rectangular.



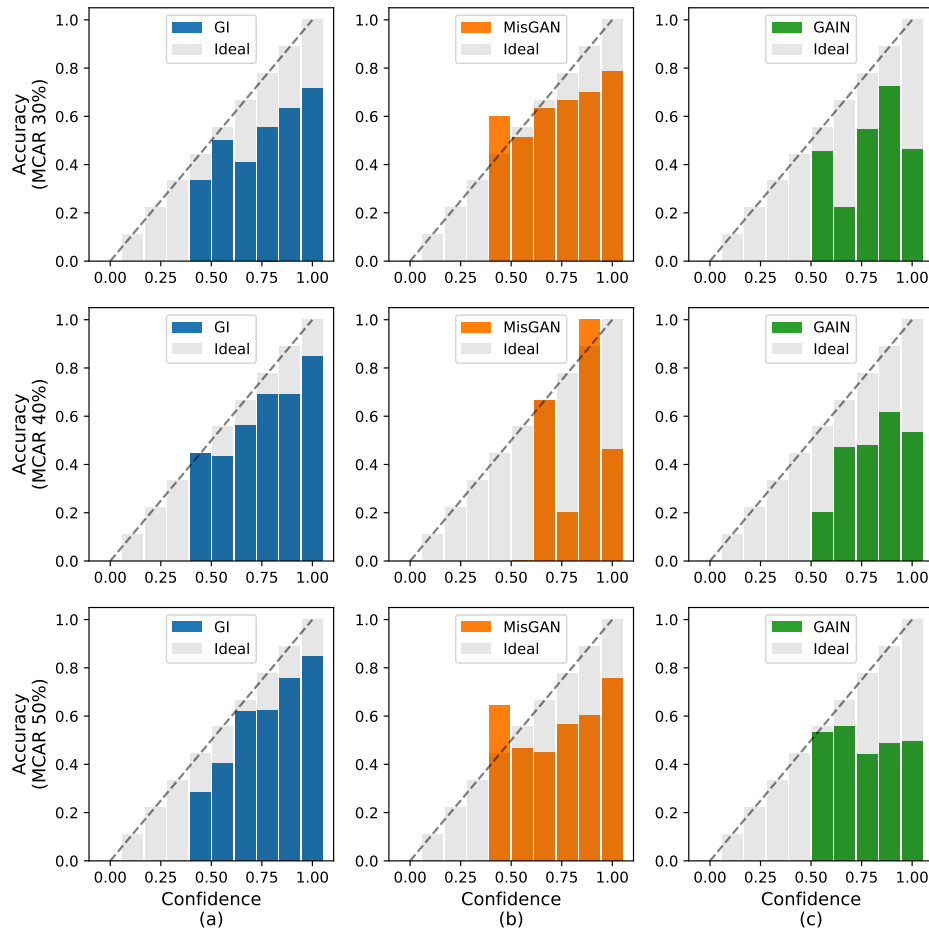


Fig. 4: Accuracy versus certainty plots for (a) GI, (b) MisGAN, and (c) GAIN on Landsat dataset at the missing rate of 30%, 40%, and 50%.

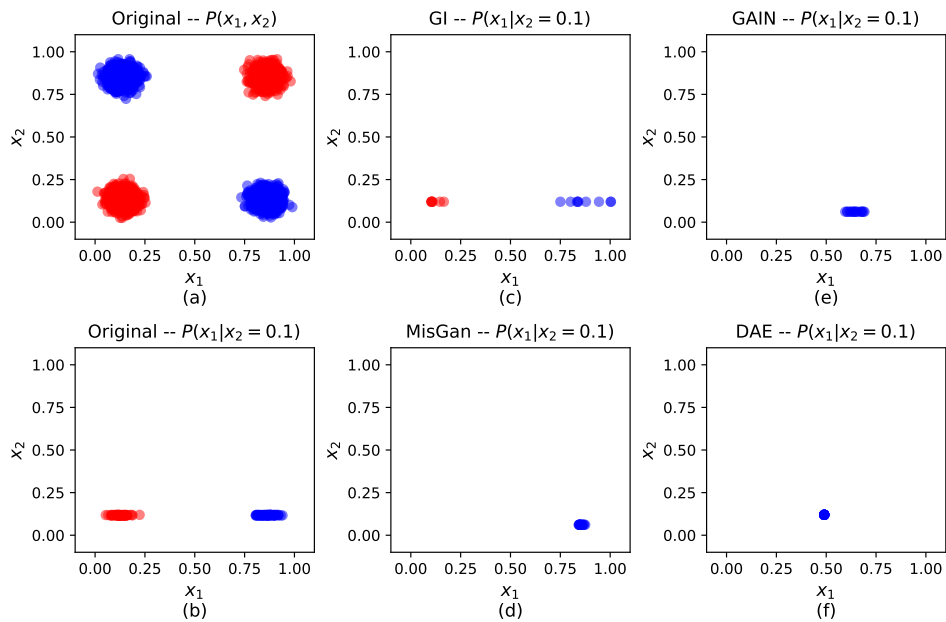


Fig. 5: Evaluation using synthesized data: (a) samples from the underlying distribution, (b) samples from the conditional underlying distribution, (c-f) samples from the conditional distribution generate by GI, MisGAN, GAIN, and DAE.

TABLE 4: Network architectures used in our experiments.

Dataset	Generator/Discriminator Architecture	Predictor Architecture
CIFAR-10	C7S1P3-64, C3S2P1-128, Attn, R-128, Attn, R-128, Attn, R-128, Attn, R-128, CT3S2P1-128, CT7S1P3-3, Tanh/Sigmoid	ResNet-18 [45] <sup>a</sup>
Landsat	FC-64, Sigmoid, BNorm, FC-64, Sigmoid, BNorm, FC-64, Sigmoid, BNorm, FC-36, Tanh/Sigmoid	FC-64, ReLU, BNorm, FC-64, ReLU, BNorm, FC-6, Softmax
MIT-BIH	FC-1860, ReLU, BNorm, FC-1860, ReLU, BNorm, FC-1860, ReLU, BNorm, FC-186, Tanh/Sigmoid	FC-1860, ReLU, BNorm, FC-1860, ReLU, BNorm, FC-5, Softmax
Diabetes	FC-45, ReLU, BNorm, FC-45, ReLU, BNorm, FC-45, ReLU, BNorm, FC-45, Tanh/Sigmoid	FC-22, ReLU, BNorm, FC-22, ReLU, BNorm, FC-3, Softmax
Cholesterol	FC-242, ReLU, BNorm, FC-242, ReLU, BNorm, FC-242, ReLU, BNorm, FC-121, Tanh/Sigmoid	FC-242, ReLU, BNorm, FC-242, ReLU, BNorm, FC-2, Softmax
Hypertension	FC-240, ReLU, BNorm, FC-240, ReLU, BNorm, FC-240, ReLU, BNorm, FC-120, Tanh/Sigmoid	FC-240, ReLU, BNorm, FC-240, ReLU, BNorm, FC-2, Softmax
MNIST	FC-1568, ReLU, BNorm, FC-1568, ReLU, BNorm, FC-1568, ReLU, BNorm, FC-784, Tanh/Sigmoid	FC-1568, ReLU, BNorm, FC-1568, ReLU, BNorm, FC-10, Softmax

a. <https://github.com/kuangliu/pytorch-cifar>

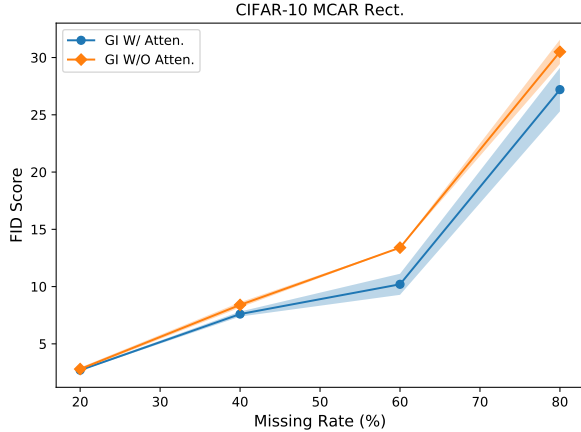


Fig. 6: Comparison of FID scores achieved with (GI W/ Atten.) and without (GI W/O Atten.) self-attention layers on CIFAR-10 dataset and rectangular missingness. Lower FID score is better.

As in our experiments and comparisons, we consider the case where only an incomplete dataset is available for training. It is crucial to guarantee that each method has only access to a unique incomplete version of each sample. However, it is relatively expensive to load and store feature masks for each sample in the dataset. Instead, we generate missing values during the data load for each batch. A hashing mechanism is used to ensure that the same parts are missing for each sample throughout the training. Note that we set system, python, and external library hash seeds to fixed values to ensure the consistency between different runs.

Algorithm 3 presents the procedure used for generating missing values with uniform structure. This algorithm is sampling independent Bernoulli distributions with probabilities equal to the missing rate. Algorithm 4 shows the

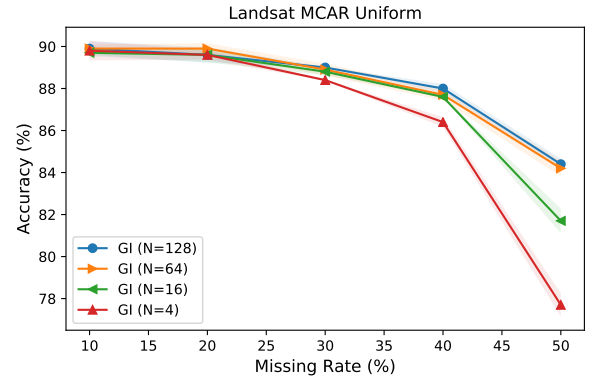


Fig. 7: Comparison of classification accuracies achieved with different ensemble size ( $N$ ).

outline for the rectangular missing structure used in image experiments. It consists of selecting a random point as the center of the rectangle and then deciding on parameters to be used for the beta distribution based on the missing rate. Finally, the width and height of the rectangular region are sampled from the latent beta distribution. In other words, we generate rectangular regions centered at random locations within the image which have width and height values determined by samples from a latent beta distribution. Here, distribution parameters,  $\alpha$  and  $\beta$ , are used to control the average missing rate. The outcome would be rectangular regions of different shape at different locations within the frame with the expected portion of missing area equal to the missing rate.

In order to decide on the beta distribution parameters i.e.  $\alpha$  and  $\beta$  we use numerical simulations. Specifically, we fix one of the parameters to 1 and change the other parameter in the range of [1,10], while measuring the average missing rate caused by each case. Figure 8 shows the missing rates caused by different beta distribution parameters. The

**Algorithm 3: MCAR uniform generation.**


---

**Input:**  $x$  (complete feature),  $r$  (missing rate)  
**Output:**  $x_m$  (incomplete feature)  
 $seed_x \leftarrow hash(x)$   
 $k \leftarrow 1 - Bernoulli(seed_x, shape(x), prob = r)$   
 $x_m \leftarrow k \odot x + (1 - k) \odot NaN$

---

**Algorithm 4: MCAR rect. generation.**


---

**Input:**  $x$  (complete feature),  $r$  (missing rate)  
**Output:**  $x_m$  (incomplete feature)  
 $seed_x \leftarrow hash(x)$   
 $n_x, n_y \leftarrow shape(x)$   
 $(p_x, p_y) \sim (uniform(0, n_x), uniform(0, n_y))$   
 $\alpha, \beta \leftarrow beta\_params(r) // beta\_params$  gives  
 $\alpha, \beta$  for each missing rate based on  
numerical simulations  
 $(w, h) \sim (Beta(\alpha, \beta) \times n_x, Beta(\alpha, \beta) \times n_y)$   
 $k \leftarrow rect\_mask(p_x, p_y, w, h)$   
 $x_m \leftarrow k \odot x + (1 - k) \odot NaN$

---

first half of Figure 8 (missing rates less than about 0.18) corresponds to setting  $\beta$  to 1 and changing  $\alpha$  values; and the other half fixing  $\alpha$  to 1 and changing  $\beta$  values. To generate missing rates more than 50% we invert our masks and limit the observation to the rectangular region while the rest of the image is missing. Note that missing rates indicate the ratio of features that are missing on the average case. As we are using a latent model for sampling width and height for the rectangles, the actual missing ratios for each specific sample differs between samples. See Table 5 for visual examples of different missing rates and missing structures.

## APPENDIX C

### ANALYSIS OF THE RMSE MEASURE

Table 6 presents the comparison of different imputation methods using the RMSE measure on CIFAR-10 for different missing structures and rates. Generally, RMSE values for the uniform missing structure are lower than their rectangular counterparts. It is consistent with our intuition that imputing uniform missingness is most similar to denoising problems where the RMSE measure is frequently used. Additionally, comparing the performance of different imputation methods using the FID measure (Section 4.4) does not demonstrate a clear correlation to results shown in Table 6. Nonetheless, it is well-known that the FID measure is more suited to measuring the performance of generated images from the underlying distribution [34].

Similarly, in Table 7, we provide RMSE values corresponding to experiments on the tabular datasets. Here, GAIN and DAE provide very similar results that are generally better than GI or MisGAN. This signifies our hypothesis that the MSE loss term may skew generated samples toward the mean of the distribution, resulting in better RMSE values but not necessarily higher final classification accuracies (see Table 2). Table 8 presents R-squared ( $R^2$ ) values for the Landsat, MIT-BIH, and Diabetes datasets. While  $R^2$  is a

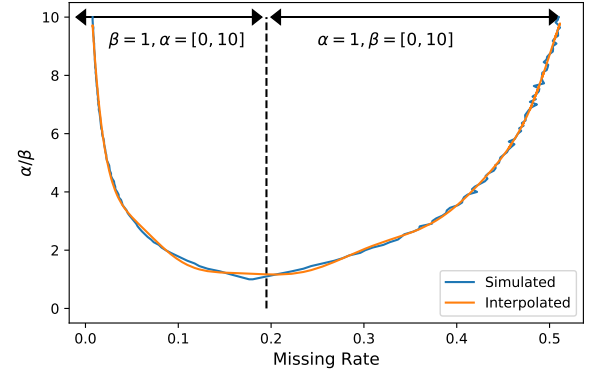


Fig. 8: Simulation results for measuring average missing rate given different beta distribution parameters.

good performance measure for regression problems (often better than RMSE), it may not be the best metric for imputation problems. The main reason is that, for imputation problems, there may be multiple valid solutions based on the observed features.

## APPENDIX D

### IMPACT OF TRAINING NOISE

Addition of noise to input vectors often serves as an input augmentation and results in improved generalization accuracies. In order to verify that the improved GI performance is not merely due to the introduction of noise in the suggested architecture, we conducted an experiment by adding different amounts of Gaussian noise during the training process for GAIN and GI. Specifically, we compared how the CIFAR-10 test accuracies change at different degrees of training noise for uniform and rectangular missingness structures at the average missing rate of 40%.

According to Table 9, adding small amounts of Gaussian noise (e.g., std=0.0125) improves the generalization under uniform missingness for both GI and GAIN. Even in this case, GI is still outperforming GAIN in terms of final classification performance. It is also interesting to point out that for the case of rectangular missingness adding Gaussian noise results in a consistent reduction in the classification accuracy for both methods.

## APPENDIX E

### IMPACT OF THE MSE LOSS TERM

In our earlier discussions, we stated that the MSE loss term used in GAIN would bias the distribution of generated samples toward the mean of the distribution. Here, a synthesized dataset is used to illustrate the impact of MSE loss term on the distribution of generated samples. A hyperparameter,  $\lambda$ , controls the weight of the MSE term in the final objective function. As it can be observed from Figure 9, the higher the  $\lambda$  parameter, the lower the variance of the generated samples (i.e., more bias toward the mean of the distribution).

TABLE 5: Examples of uniform and rectangular missing structures at different missing rates.

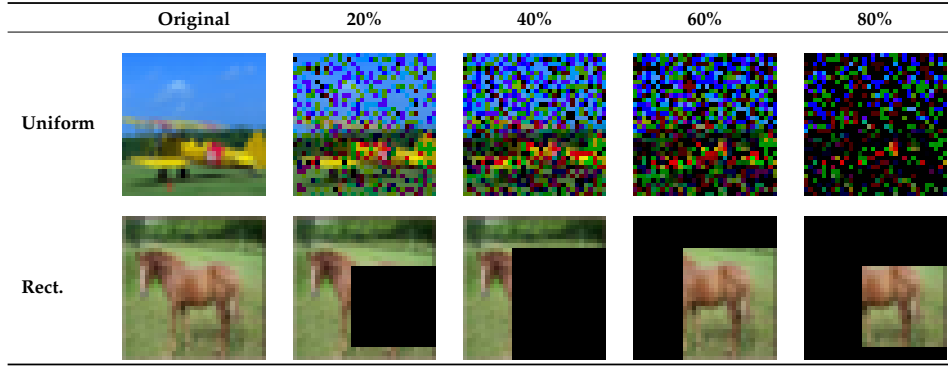


TABLE 6: Comparison of imputation RMSE values for CIFAR-10 at different missing structures and rates.

Method	RMSE at Missing Rate (%)					
	MCAR Uniform			MCAR Rect.		
	20%	40%	60%	20%	40%	60%
GI	0.026 ( $\pm 0.003$ )	0.057 ( $\pm 0.008$ )	0.090 ( $\pm 0.006$ )	0.097 ( $\pm 0.02$ )	0.148 ( $\pm 0.001$ )	0.660 ( $\pm 0.010$ )
MisGAN	0.079 ( $\pm 0.001$ )	0.161 ( $\pm 0.001$ )	0.257 ( $\pm 0.002$ )	0.106 ( $\pm 0.005$ )	0.158 ( $\pm 0.004$ )	0.250 ( $\pm 0.001$ )
GAIN	0.027 ( $\pm 0.003$ )	0.045 ( $\pm 0.001$ )	0.072 ( $\pm 0.005$ )	0.340 ( $\pm 0.047$ )	0.511 ( $\pm 0.001$ )	0.660 ( $\pm 0.010$ )
DAE	0.036 ( $\pm 0.001$ )	0.075 ( $\pm 0.002$ )	0.121 ( $\pm 0.005$ )	0.116 ( $\pm 0.007$ )	0.160 ( $\pm 0.001$ )	0.233 ( $\pm 0.029$ )

TABLE 7: Comparison of imputation RMSE values for Landsat, MIT-BIH, and Diabetes datasets at different missing rates.

Dataset	Method	RMSE at Missing Rate (%)			
		10%	20%	30%	40%
Landsat [38]	GI	0.040 ( $\pm 0.005$ )	0.067 ( $\pm 0.007$ )	0.076 ( $\pm 0.020$ )	0.136 ( $\pm 0.002$ )
	MisGAN	0.068 ( $\pm 0.001$ )	0.096 ( $\pm 0.001$ )	0.118 ( $\pm 0.001$ )	0.136 ( $\pm 0.001$ )
	GAIN	0.018 ( $\pm 0.001$ )	0.024 ( $\pm 0.001$ )	0.030 ( $\pm 0.001$ )	0.037 ( $\pm 0.001$ )
	DAE	0.020 ( $\pm 0.001$ )	0.031 ( $\pm 0.001$ )	0.041 ( $\pm 0.001$ )	0.052 ( $\pm 0.001$ )
MIT-BIH [39]	GI	0.038 ( $\pm 0.001$ )	0.060 ( $\pm 0.004$ )	0.071 ( $\pm 0.002$ )	0.095 ( $\pm 0.002$ )
	MisGAN	0.073 ( $\pm 0.007$ )	0.092 ( $\pm 0.002$ )	0.115 ( $\pm 0.003$ )	0.111 ( $\pm 0.001$ )
	GAIN	0.032 ( $\pm 0.008$ )	0.046 ( $\pm 0.001$ )	0.055 ( $\pm 0.004$ )	0.067 ( $\pm 0.007$ )
	DAE	0.029 ( $\pm 0.001$ )	0.048 ( $\pm 0.008$ )	0.061 ( $\pm 0.009$ )	0.068 ( $\pm 0.003$ )
Diabetes [40]	GI	0.080 ( $\pm 0.002$ )	0.118 ( $\pm 0.008$ )	0.149 ( $\pm 0.020$ )	0.189 ( $\pm 0.009$ )
	MisGAN	0.082 ( $\pm 0.004$ )	0.111 ( $\pm 0.002$ )	0.133 ( $\pm 0.001$ )	0.151 ( $\pm 0.001$ )
	GAIN	0.064 ( $\pm 0.001$ )	0.092 ( $\pm 0.001$ )	0.119 ( $\pm 0.001$ )	0.140 ( $\pm 0.001$ )
	DAE	0.065 ( $\pm 0.001$ )	0.093 ( $\pm 0.001$ )	0.118 ( $\pm 0.001$ )	0.143 ( $\pm 0.001$ )

## APPENDIX F IMPACT OF THE DISCRIMINATOR HINT VECTOR

Yoon et al. [19] suggested the idea of guiding the discriminator network using a hint mechanism. A hint vector reveals a subset of features that are missing to the discriminator. In Figure 10 and 11 we provide a comparison of learning curves for GI implemented using different hint rates. From Figure 10, using the hint mechanism does not result in any noticeable improvement in the final imputation quality justifying the added complexity. For the case of the rectangular missing structure in Figure 11; however, using the hint vector causes instabilities in the training process. One possible explanation is: providing even a small portion of the mask as a hint, due to the deterministic nature of the rectangular shape it is equivalent to providing region boundaries to the discriminator making it obvious for the discriminator. In GAN training we generally want to have

equal competition between the generator and discriminator.

## APPENDIX G IMPACT OF OTHER MISSINGNESS MECHANISMS

Throughout this paper, we conducted experiments based on the MCAR assumptions. In this section, we provide additional experiments on the MNIST dataset using two sample-dependent missingness mechanisms: (i) foreground pixels missing at different rates, (ii) background pixels missing at different rates. Note that in these experiments, to prevent the trivial case of imputers learning to always impute constant values, we let all image pixels have at least 10% chance of being missing. For instance, to examine the foreground missingness, we let all background pixels have 10% chance of missingness while we set the missing rate for foreground pixels at different rates in the range of 10% to 40%. Otherwise, as MNIST pixel values are mostly

TABLE 8: Comparison of imputation  $R^2$  values for Landsat, MIT-BIH, and Diabetes datasets at different missing rates.

Dataset	Method	$R^2$ at Missing Rate (%)			
		10%	20%	30%	40%
Landsat [38]	GI	0.951 ( $\pm 0.001$ )	0.898 ( $\pm 0.001$ )	0.840 ( $\pm 0.001$ )	0.783 ( $\pm 0.001$ )
	MisGAN	0.946 ( $\pm 0.001$ )	0.887 ( $\pm 0.001$ )	0.851 ( $\pm 0.001$ )	0.825 ( $\pm 0.040$ )
	GAIN	0.996 ( $\pm 0.001$ )	0.994 ( $\pm 0.001$ )	0.992 ( $\pm 0.001$ )	0.987 ( $\pm 0.001$ )
	DAE	0.990 ( $\pm 0.001$ )	0.980 ( $\pm 0.002$ )	0.961 ( $\pm 0.005$ )	0.942 ( $\pm 0.006$ )
MIT-BIH [39]	GI	0.986 ( $\pm 0.001$ )	0.965 ( $\pm 0.002$ )	0.947 ( $\pm 0.001$ )	0.913 ( $\pm 0.006$ )
	MisGAN	0.949 ( $\pm 0.009$ )	0.923 ( $\pm 0.008$ )	0.889 ( $\pm 0.010$ )	0.876 ( $\pm 0.006$ )
	GAIN	0.988 ( $\pm 0.007$ )	0.980 ( $\pm 0.003$ )	0.964 ( $\pm 0.010$ )	0.957 ( $\pm 0.002$ )
	DAE	0.992 ( $\pm 0.001$ )	0.98 ( $\pm 0.010$ )	0.969 ( $\pm 0.003$ )	0.937 ( $\pm 0.010$ )
Diabetes [40]	GI	0.953 ( $\pm 0.007$ )	0.900 ( $\pm 0.008$ )	0.851 ( $\pm 0.008$ )	0.823 ( $\pm 0.001$ )
	MisGAN	0.964 ( $\pm 0.002$ )	0.934 ( $\pm 0.004$ )	0.909 ( $\pm 0.001$ )	0.876 ( $\pm 0.001$ )
	GAIN	0.979 ( $\pm 0.001$ )	0.955 ( $\pm 0.001$ )	0.929 ( $\pm 0.001$ )	0.896 ( $\pm 0.001$ )
	DAE	0.979 ( $\pm 0.001$ )	0.955 ( $\pm 0.001$ )	0.928 ( $\pm 0.002$ )	0.895 ( $\pm 0.002$ )

TABLE 9: Top-1 CIFAR-10 classification accuracy at 40% missing rate using added training noise.

Noise STD	Accuracy (%)			
	MCAR Uniform (40%)		MCAR Rect. (40%)	
	GI	GAIN	GI	GAIN
0.0	87.1	86.0	<b>76.9</b>	73.6
0.0125	<b>87.3</b>	86.3	76.8	73.3
0.025	86.5	86.6	76.7	73.2
0.05	85.6	84.7	73.7	72.4
0.1	82.0	80.6	68.7	67.0

distributed around 0 or 1, it is quite easy for our imputers to learn constant and near-perfect imputations, making the task too easy.

Table 10 shows a comparison of the uniform, foreground, and background missingness mechanisms for the MNIST dataset. According to the results, GI is able to outperform other work in all of the cases. This is consistent with the general formulation presented in this work which does not impose any domain-specific prior over the structure of missing values and hence is robust to the missingness mechanism.

TABLE 10: Comparison of classification accuracies for MNIST at different missing rates and missing types.

	Method	Accuracy at Missing Rate (%)			
		10%	20%	30%	40%
Uniform	GI	<b>99.0</b> ( $\pm 0.01$ )	<b>98.9</b> ( $\pm 0.07$ )	<b>98.7</b> ( $\pm 0.01$ )	<b>98.6</b> ( $\pm 0.01$ )
	MisGAN	98.6 ( $\pm 0.02$ )	98.3 ( $\pm 0.06$ )	98.1 ( $\pm 0.02$ )	97.5 ( $\pm 0.06$ )
	GAIN	98.8 ( $\pm 0.02$ )	98.7 ( $\pm 0.03$ )	98.6 ( $\pm 0.02$ )	98.5 ( $\pm 0.03$ )
	DAE	98.8 ( $\pm 0.08$ )	98.7 ( $\pm 0.03$ )	98.5 ( $\pm 0.08$ )	98.0 ( $\pm 0.06$ )
Foreground	GI	<b>99.0</b> ( $\pm 0.01$ )	<b>98.9</b> ( $\pm 0.01$ )	<b>98.9</b> ( $\pm 0.02$ )	<b>98.8</b> ( $\pm 0.03$ )
	MisGAN	98.6 ( $\pm 0.01$ )	98.5 ( $\pm 0.06$ )	98.3 ( $\pm 0.06$ )	98.3 ( $\pm 0.07$ )
	GAIN	98.9 ( $\pm 0.08$ )	98.8 ( $\pm 0.06$ )	98.8 ( $\pm 0.02$ )	98.7 ( $\pm 0.09$ )
	DAE	98.8 ( $\pm 0.01$ )	98.7 ( $\pm 0.06$ )	98.7 ( $\pm 0.02$ )	98.5 ( $\pm 0.07$ )
Background	GI	<b>99.0</b> ( $\pm 0.02$ )	<b>99.0</b> ( $\pm 0.01$ )	<b>99.0</b> ( $\pm 0.02$ )	<b>98.8</b> ( $\pm 0.01$ )
	MisGAN	98.5 ( $\pm 0.09$ )	98.3 ( $\pm 0.06$ )	98.2 ( $\pm 0.02$ )	97.8 ( $\pm 0.04$ )
	GAIN	98.8 ( $\pm 0.04$ )	98.8 ( $\pm 0.06$ )	98.8 ( $\pm 0.04$ )	98.7 ( $\pm 0.04$ )
	DAE	98.8 ( $\pm 0.04$ )	98.7 ( $\pm 0.04$ )	98.6 ( $\pm 0.02$ )	98.6 ( $\pm 0.04$ )

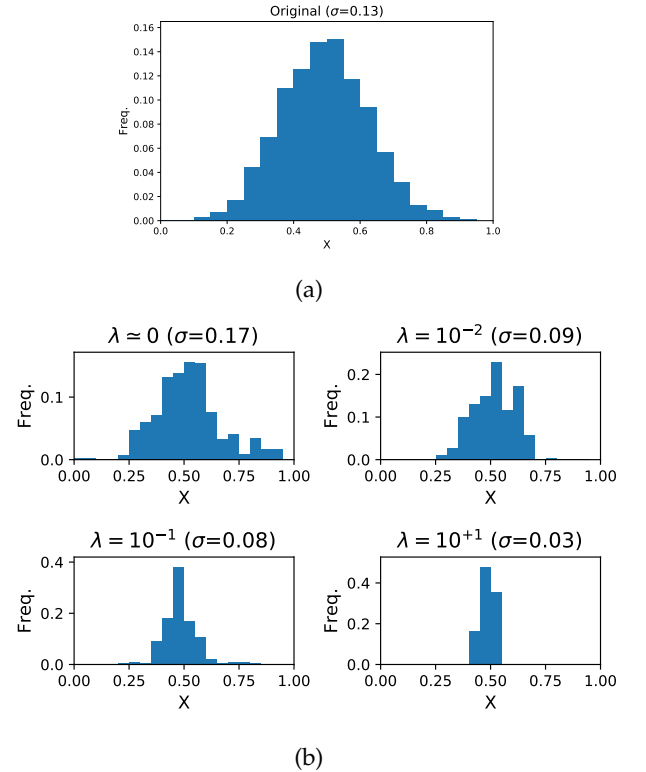


Fig. 9: Comparison of generating samples from a Gaussian distribution (a) samples from the original distribution, (b) samples generated using GAIN imputers with different significance of the MSE term (controlled by  $\lambda$ ).

## APPENDIX H VISUAL RESULTS

Figure 12 provides examples of masked CIFAR-10 images that are imputed using the proposed method. The variance among imputed samples is representing different possibilities for completing the missing parts. For each input sample, we also show the class assignment certainties estimated from an ensemble of 128 imputations, of which three randomly selected samples are shown here. In certain examples, the missing part is not causing a noticeable uncertainty

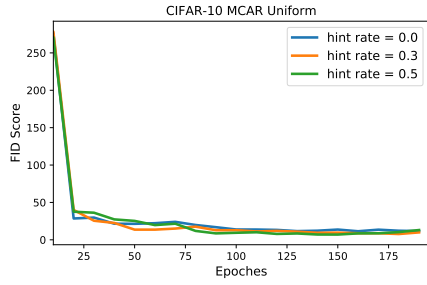


Fig. 10: Learning curves for CIFAR-10 with uniform missing structure at different discriminator hint rates.

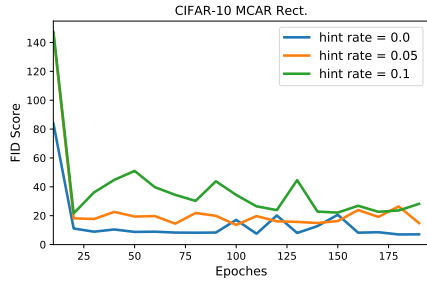
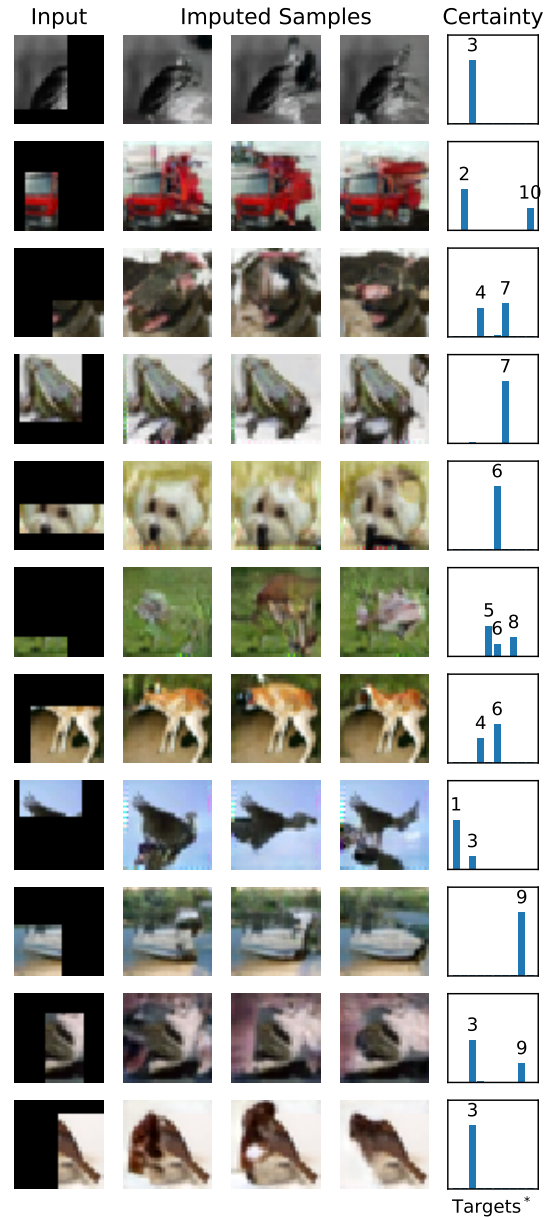


Fig. 11: Learning curves for CIFAR-10 with rectangular missing structure at different discriminator hint rates.

over target assignments, while in others it leads to some confusion over target assignments based on the different viable imputations.

## REFERENCES

- [1] G. Eisele, H. Vachon, G. Lafit, P. Kuppens, M. Houben, I. Myin-Germeys, and W. Viechtbauer, "The effects of sampling frequency and questionnaire length on perceived burden, compliance, and careless responding in experience sampling data in a student population," *PsyArXiv preprint PsyArXiv:10.31234*, 2020.
- [2] S. Lin, X. Wu, G. Martinez, and N. V. Chawla, "Filling missing values on wearable-sensory time series data," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 46–54.
- [3] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, "Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model," *Knowledge-Based Systems*, p. 105592, 2020.
- [4] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein, "Imputing missing data for gene expression arrays," 1999.
- [5] T. W. Anderson, "Maximum likelihood estimates for a multivariate normal distribution when some observations are missing," *Journal of the American Statistical Association*, vol. 52, no. 278, pp. 200–203, 1957.
- [6] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. Wiley, 2019, vol. 793.
- [7] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.



\* 1: plane, 2: car, 3: bird, 4: cat, 5: deer, 6: dog, 7: frog, 8: horse, 9: ship, 10: truck,

Fig. 12: Visual samples of imputed CIFAR-10 images and estimated classification certainties from each incomplete input. The estimated certainty for each target class assignment is represented by a bar for each class, where the height shows the relative confidence.

- [8] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art." *Psychological methods*, vol. 7, no. 2, p. 147, 2002.
- [9] J. S. Murray *et al.*, "Multiple imputation: a review of practical and theoretical findings," *Statistical Science*, vol. 33, no. 2, pp. 142–159, 2018.
- [10] A. Aleryani, W. Wang, and B. De La Iglesia, "Multiple imputation ensembles (mie) for dealing with missing data," *SN Computer Science*, vol. 1, pp. 1–20, 2020.
- [11] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th in-*



- ternational conference on Machine learning.* ACM, 2008, pp. 1096–1103.
- [12] S. Ryu, M. Kim, and H. Kim, “Denoising autoencoder-based missing value imputation for smart meters,” *IEEE Access*, vol. 8, pp. 40 656–40 666, 2020.
- [13] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [14] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *Advances in neural information processing systems*, 2015, pp. 3483–3491.
- [15] P.-A. Mattei and J. Frellsen, “missiwae: Deep generative modelling and imputation of incomplete data,” *arXiv preprint arXiv:1812.02633*, 2018.
- [16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [17] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially learned inference,” *arXiv preprint arXiv:1606.00704*, 2016.
- [18] M. Śmieja, M. Kołomycki, Ł. Struski, M. Juda, and M. A. T. Figueiredo, “Can auto-encoders help with filling missing data?” in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.
- [19] J. Yoon, J. Jordon, and M. Van Der Schaar, “Gain: Missing data imputation using generative adversarial nets,” *arXiv preprint arXiv:1806.02920*, 2018.
- [20] S. C.-X. Li, B. Jiang, and B. Marlin, “Learning from incomplete data with generative adversarial networks,” in *International Conference on Learning Representations*, 2019.
- [21] T. D. Nielsen and F. V. Jensen, *Bayesian networks and decision graphs.* Springer Science & Business Media, 2009.
- [22] S. Zhang, Z. Qin, C. X. Ling, and S. Sheng, ““missing is useful”: missing values in cost-sensitive decision trees,” *IEEE transactions on knowledge and data engineering*, vol. 17, no. 12, pp. 1689–1693, 2005.
- [23] A. Darwiche, *Modeling and reasoning with Bayesian networks.* Cambridge university press, 2009.
- [24] R. E. Neapolitan *et al.*, *Learning bayesian networks.* Pearson Prentice Hall Upper Saddle River, NJ, 2004, vol. 38.
- [25] C. T. Tran, M. Zhang, P. Andreae, and B. Xue, “Multiple imputation and genetic programming for classification with incomplete data,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 521–528.
- [26] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, “Learning with noisy labels,” in *Advances in neural information processing systems*, 2013, pp. 1196–1204.
- [27] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, “Training deep neural networks on noisy labels with bootstrapping,” *arXiv preprint arXiv:1412.6596*, 2014.
- [28] P. Chen, B. Liao, G. Chen, and S. Zhang, “Understanding and utilizing deep neural networks trained with noisy labels,” *arXiv preprint arXiv:1905.05040*, 2019.
- [29] S. Arora, A. Risteski, and Y. Zhang, “Do gans learn the distribution? some theory and empirics,” 2018.
- [30] S. Liu, O. Bousquet, and K. Chaudhuri, “Approximation and convergence properties of generative adversarial learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5545–5553.
- [31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.
- [32] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” *arXiv preprint arXiv:1805.08318*, 2018.
- [33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [34] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.
- [35] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
- [36] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” *Citeseer, Tech. Rep.*, 2009.
- [37] Y. LeCun, C. Cortes, and C. J. Burges, “The mnist database of handwritten digits, 1998,” *URL <http://yann.lecun.com/exdb/mnist>*, vol. 10, p. 34, 1998.
- [38] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [39] G. B. Moody and R. G. Mark, “The impact of the mit-bih arrhythmia database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [40] M. Kachuee, K. Karkkainen, O. Goldstein, D. Zamanzadeh, and M. Sarrafzadeh, “Nutrition and health data for cost-sensitive learning,” *arXiv preprint arXiv:1902.07102*, 2019.
- [41] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” *arXiv preprint arXiv:1611.01144*, 2016.
- [42] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, “Handling incomplete heterogeneous data using vaes,” *arXiv preprint arXiv:1807.03653*, 2018.
- [43] S. v. Buuren and K. Groothuis-Oudshoorn, “mice: Multivariate imputation by chained equations in r,” *Journal of statistical software*, pp. 1–68, 2010.
- [44] J. Yi, J. Lee, K. J. Kim, S. J. Hwang, and E. Yang, “Why not to use zero imputation? correcting sparsity bias in training neural networks,” in *International Conference on Learning Representations*, 2020.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.



**Mohammad Kachuee** (S14) received the M.S. and B.S. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran. He is currently pursuing the Ph.D. degree in computer science with the University of California at Los Angeles, Los Angeles, CA, USA. His current research interests include representation learning, health analytics, and designing machine learning algorithms for healthcare applications.



**Kimmo Karkkainen** received his M.Sc. and B.Sc degrees in computer science and engineering from Aalto University, Helsinki, Finland. He is currently pursuing a Ph.D. degree in computer science with the University of California, Los Angeles, CA, USA. His current research interests include computer vision, mobile health, and designing machine learning algorithms for health applications.



**Orpaz Goldstein** received his B.Sc degree in software engineering from Shenkar college of engineering and design in Tel-Aviv Israel. The M.Sc degree from University of California Los Angeles, USA where he is currently pursuing a Ph.D in computer science as part of the eHealth analytics lab. His current research interests include machine learning applied to live data streams on edge networks, and AI healthcare applications.



**Sajad Darabi** (S14) received the B.Eng. degree (Hons.) in electrical engineering from McGill University, Montreal, QC, Canada. He is currently pursuing the Ph.D. degree in computer science with the eHealth Analytics Lab, University of California at Los Angeles, Los Angeles, CA, USA. His current research interests include sports analytics and designing machine learning algorithms for mobile and health applications.



**Majid Sarrafzadeh** (F96) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at UrbanaChampaign, Urbana, IL, USA, in 1987. He joined Northwestern University, Evanston, IL, USA, as an Assistant Professor in 1987. In 2000, he joined the Computer Science Department, University of California at Los Angeles, Los Angeles, CA, USA, where he is currently a Distinguished Professor. He was in collaborative with many industries. In 2000, he was a Co-Founder of two

companies, which were both acquired around 2004. He is currently a Co-Founder of three companies in healthcare technology. He has authored or co-authored over 600 papers and co-authored 5 books. He is a named inventor on many U.S. patents. His current research interests include embedded computing with emphasis on health analytics.