Nazábal, A., Olmos, P. M., Ghahramani, Z. & Valera, I. (2020). Handling incomplete heterogeneous data using VAEs. *Pattern Recognition*, vol. 107, 107501.

DOI: 10.1016/j.patcog.2020.107501

Handling Incomplete Heterogeneous Data using VAEs

Alfredo Nazabal ANAZABAL@TURING.AC.UK
Alan Turing Institute
London, United Kingdom

Pablo M. Olmos OLMOS@TSC.UC3M.ES
University Carlos III in Madrid
Madrid, Spain

Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
University of Cambridge
Cambridge, United Kingdom
Uber AI Labs
San Francisco, US

Isabel Valera IVALERA@MPI-SWS.ORG
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Abstract

Variational autoencoders (VAEs), as well as other generative models, have been shown to be efficient and accurate to capture the latent structure of vast amounts of complex high-dimensional data. However, existing VAEs can still not directly handle data that are heterogenous (mixed continuous and discrete) or incomplete (with missing data at random), which is indeed common in real-world applications.

In this paper, we propose a general framework to design VAEs, suitable for fitting incomplete heterogenous data. The proposed HI-VAE includes likelihood models for real-valued, positive real valued, interval, categorical, ordinal and count data, and allows to estimate (and potentially impute) missing data accurately. Furthermore, HI-VAE presents competitive predictive performance in supervised tasks, outperforming supervised models when trained on incomplete data.

Data are usually organized and stored in databases, which are often large, heterogenous, noisy, and contain missing values. For example, an online shopping platform has access to heterogenous and incomplete information of its users, such as their age, gender, orders, wishing lists, etc. Similarly, Electronic Health Records of hospitals might contain different lab measurements, diagnoses and genomic information about their patients. Learning generative models that accurately capture the distribution, and therefore the underlying latent structure, of such incomplete and heterogeneous datasets may allow us to better understand the data, estimate missing or corrupted values, detect outliers, and make predictions (e.g., on patients' diagnose) on unseen data (Valera et al., 2017a).

Deep generative models have been recently proved to be highly flexible and expressive unsupervised methods, able to capture the latent structure of complex high-dimensional data. They efficiently emulate complex distributions from large high-dimensional datasets, generating new data points similar to the original real-world data, after training is completed (Kingma and Welling, 2014; Rezende and Mohamed, 2015). So far, the main focus in the literature is to enrich the prior or posterior of explicit generative models such as variational autoencoders (VAEs); or to propose alternative training objectives to the log-likelihood, leading to implicit generative models such as, e.g., generative adversarial networks (GANs) (Mescheder et al., 2017; Salimans et al., 2016). Indeed, we are witnessing a race between an ever-growing spectrum of VAE models, e.g. VAE with a VampPrior (Tomczak and Welling, 2018), Variational Lossy Autoencoder (Chen et al., 2016),

DVAE++ (Vahdat et al., 2018), Shape Variational Autoencoder (Nash and Williams, 2017) and GAN-style objective functions (f-GAN (Nowozin et al., 2016) , Wasserstein GANs (Arjovsky et al., 2017), MMD-GAN (Li et al., 2015), AdaGAN (Tolstikhin et al., 2017), feature-matching GAN (Mroueh et al., 2017), etc.). While all these approaches compete to generate the most realistic images or readable text, the deployment of such models to solve practically-relevant problems in arbitrary datasets, which are often incomplete and heterogenous, is being overlooked. In the following, we discuss these problems in more detail and why we believe our paper is relevant to data-scientists interested in exploiting the deep generative model pipeline in the data wrangling process. We provide with practical tools to handle both missing and heterogeneous data with little supervision from the user, who merely has to indicate the data type model of each attribute and the position of the missing data.

Currently deep generative models focus on highly-structured homogeneous data collections including, e.g., images (Gulrajani et al., 2017; Mescheder et al., 2017; Salimans et al., 2016), text (Yang et al., 2017) or speech (Bando et al., 2017), which are characterized by strong statistical dependencies between pixels or words. The dominant existing approach to account for heterogenous data follows a deep domain-alignment approach, designed to discover relations between two unpaired unlabelled datasets rather than modelling their joint distribution using a probabilistic generative model (Ganin et al., 2016; Kim et al., 2017; Liu et al., 2017; Zhu et al., 2017; Taigman et al., 2016; Castrejon et al., 2016). Surprisingly, not much attention has been paid to describe how deep generative models can be designed to effectively learn the distribution of, usually less structured, heterogeneous datasets. In these datasets there is no clear notion of correlation among the different attributes (or dimensions) to be exploited by weight sharing using convolutional or recurrent neural networks. As we show in this paper, preventing a few dimensions of the data dominating the training is a crucial aspect to effectively deploy deep generative models suitable for heterogeneous data.

Similarly, there is no clear discussion in the current literature on how to incorporate missing data during the training of deep generative models. Existing approaches consider either complete data during training (Kingma and Welling, 2014; Rezende and Mohamed, 2015; Mescheder et al., 2017; Salimans et al., 2016), or assume incomplete information only in one of the dimensions of the data, which corresponds to the one they aim to predict (e.g., the label in a classification task) (Sohn et al., 2015; Kingma et al., 2014). However, both approaches are not realistic enough, since it might be crucial for the performance of an unsupervised model to use all the available information during training. Recently, (Yoon et al., 2018) proposed a GAN approach to input missing data, where the generator completes the missing values given the observed ones, and the discriminator aims to distinguish between true and imputed values. However, this approach can only handle continuous or binary data, and it is not easily generalizable to heterogeneous data. As a consequence, strategies to effectively train deep generative models on incomplete and heterogeneous datasets are still required.

In this work, we present a general framework for VAEs that effectively incorporates incomplete data and heterogenous observations. Our design presents the following features:

i) a generative model that handles mixed numerical (continuous real-valued and positive real-valued, as well as discrete count data) and nominal (categorical and ordinal data) likelihood models, which we parametrize using deep neural networks (DNNs);

ii) a stable recognition model that handles Missing Data Completely at Random (MCAR) without increasing its complexity or promoting overfitting;

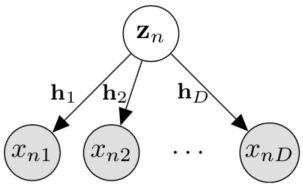
iii) a data-normalization input/output layer that prevents a few dimensions of the data dominating the training of the VAE, improving also the training convergence; and

iv) an ELBO (Evidence Lower Bound), used to optimize the parameters of both the generative and the recognition models, that is computed only on the observed data, regardless of the pattern of missing data.

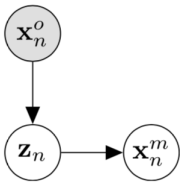
The resulting VAE is a fully unsupervised model which allows us not only to accurately solve unsupervised tasks, such as density estimation or missing data completion, but also supervised tasks (e.g., classification or regression) with incomplete input data. We provide the reader with specific guidelines to design VAEs for real-world data, which are compatible with modern efforts in the design of VAEs and implicit models (GANs), mainly oriented to prevent the mode-dropping effect (Arjovsky et al., 2017; Arora and Zhang, 2017). Our

yes	male	?	1.72
?	female	49	1.81
yes	female	38	1.60
?	male	?	1.81
no	?	25	1.67

(a) Database example



(b) Generative Model



(c) Recognition Model

Figure 1: a) Example of incomplete heterogenous data. Panel (b) shows our generative model, where every dimension in the observation vector $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$ corresponds to either a numerical or nominal variable, and therefore, the likelihood parameters of each dimension d are independently provided by an independent DNN $\mathbf{h}_d$. Additionally, panel (c) shows our recognition model to infer the missing data $\mathbf{x}_n^m$ from observed data $\mathbf{x}_n^o$.

empirical results show that our proposal outperforms competitors on a heterogenous data completion task, and provides comparable accuracy in classification tasks to deep supervised methods–which cannot handle missing values in the input data, therefore, requiring imputing missing inputs in the data.

1. Problem statement

We define a heterogeneous dataset as a collection of N objects, where each object is defined by D attributes and these attributes correspond to either numerical (continuous or discrete) or nominal variables. We denote each object in the dataset as a D-dimensional vector $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$, where each attribute d corresponds to one of the following data types:

- Numerical variables:
  1. Real-valued data, which takes values in the real line, i.e., $x_{nd} \in \mathbb{R}$.
  2. Positive real-valued data, which takes values in the positive real line, i.e., $x_{nd} \in \mathbb{R}^+$.
  3. (Discrete) count data, which takes values in the natural numbers, i.e., $x_{nd} \in \{1, \ldots, \infty\}$.
- Nominal variables:
  1. Categorical data, which takes values in a finite unordered set, e.g., $x_{nd} \in \{$'blue', 'red', 'black'$\}$.
  2. Ordinal data, which takes values in a finite ordered set, e.g., $x_{nd} \in \{$'never', 'sometimes', 'often', 'usually', 'always'$\}$.

Additionally, we consider that a random set of entries in the data is incomplete, such that each object $\mathbf{x}_n$ can potentially contain any combination of observed and missing attributes. Let $\mathcal{O}_n$ ($\mathcal{M}_n$) be the index set of observed (missing) attributes for the n-th data point, where $\mathcal{O}_n \cap \mathcal{M}_n = \emptyset$. Also, let $\mathbf{x}_n^o$ ($\mathbf{x}_n^m$) represent the sliced $\mathbf{x}$ vector, stacking any dimension with index in $\mathcal{O}_n$ ($\mathcal{M}_n$). Figure 1(a) shows an example of an incomplete heterogenous dataset, where we observe that the different attributes (or dimensions) in the data correspond with different types of numerical and nominal variables, and missing values appear ubiquitously across the data.

Diverging from common trends in the deep generative community, we consider databases that do not contain highly-structured homogeneous data, but instead each observed object is a set of scalar mixed numerical and nominal, attributes and the underlying structure is in many cases mild. Since the dimensionality of these datasets can be relatively small (compared to images for instance), we need to carefully design the generative model to avoid overfitting on the observed data, while keeping the model flexible enough to incorporate both heterogeneous data types and random patterns of missing data.

2. Revisited VAE

In this section, we show how to extend the vanilla VAE introduced in (Kingma and Welling, 2014) to handle incomplete and heterogeneous data.

2.1 Handling incomplete data

In a standard VAE, missing data affect both the generative (decoder) and the recognition (encoder) models. The ELBO is defined over the complete data, and it is not straightforward to decouple the missing entries from rest of the data, particularly when these entries appear randomly in the dataset. To this end, we first propose to use the following factorization for the decoder (Figure 1(b)):

$$p(\mathbf{x}_n, \mathbf{z}_n) = p(\mathbf{z}_n) \prod_d p(x_{nd}|\mathbf{z}_n), \tag{1}$$

where $\mathbf{z}_n \in \mathbb{R}^K$ is the latent K-dimensional vector representation of the object $\mathbf{x}_n$, and $p(\mathbf{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$. This factorization allows to easily marginalize out from the variational ELBO the missing attributes for each object. We parametrize the likelihood $p(x_{nd}|\mathbf{z}_n)$ with the set of parameters $\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{z}_n)$—here, $\mathbf{h}_d(\mathbf{z}_n)$ is a DNN that transforms the latent variable $\mathbf{z}_n$ into the likelihood parameters $\boldsymbol{\gamma}_{nd}$.

Note that the above factorization of the likelihood allows us to separate the contributions of the observed data $\mathbf{x}_n^o$ from missing data $\mathbf{x}_n^m$ as

$$p(\mathbf{x}_n|\mathbf{z}_n) = \prod_{d \in \mathcal{O}_n} p(x_{nd}|\mathbf{z}_n) \prod_{d \in \mathcal{M}_n} p(x_{nd}|\mathbf{z}_n). \tag{2}$$

The recognition model also needs to account for incomplete data, such that the distribution of the latent variable $\mathbf{z}_n$ only depends on the observed attributes $\mathbf{x}_n^o$, i.e.,

$$q(\mathbf{z}_n, \mathbf{x}_n^m|\mathbf{x}_n^o) = q(\mathbf{z}_n|\mathbf{x}_n^o) \prod_{d \in \mathcal{M}_n} p(x_{nd}|\mathbf{z}_n). \tag{3}$$

The recognition model is graphically represented in Figure 1(c). Note that, we need a recognition model that is flexible enough to handle any combination of observed and missing attributes. To this end, we propose an *input drop-out* recognition distribution whose parameters are the output of a DNN with input $\tilde{\mathbf{x}}_n$, such that

$$q(\mathbf{z}_n|\mathbf{x}_n^o) = \mathcal{N}(\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)), \tag{4}$$

where the input $\tilde{\mathbf{x}}_n$ is a D-length vector that resembles the original observed vector $\mathbf{x}_n$ but the missing dimensions are replaced by zeros, and $\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n)$ and $\boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)$ are parametrized DNNs with input $\tilde{\mathbf{x}}_n$ whose output determine the mean and the diagonal covariance matrix of (4). By setting certain dimensions to zero in $\tilde{\mathbf{x}}_n$, the contribution of the missing attributes to $\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n)$ and $\boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)$ and their derivative with respect to the network parameters is zero.

An alternative approach, proposed in (Vedantam et al., 2017), consists of exploiting the properties of Gaussian distributions in the linear factor analysis case (Williams and Nash, 2018) and extending them to non-linear models, designing a factorized recognition model: $q(\mathbf{z}_n|\mathbf{x}_n^o) = p(\mathbf{z}_n) \prod_{d \in \mathcal{O}_n} q(\mathbf{z}_n|x_{nd})$, where $q(\mathbf{z}_n|x_{nd}) = \mathcal{N}\left(\boldsymbol{\mu}_d(x_{nd}), \boldsymbol{\Sigma}_d(x_{nd})\right)$, and therefore, $q(\mathbf{z}_n|\mathbf{x}_n^o) = \mathcal{N}\left(\boldsymbol{\mu}_q(\mathbf{x}_n^o), \boldsymbol{\Sigma}_q(\mathbf{x}_n^o)\right)$ with

$$\begin{aligned} \boldsymbol{\Sigma}_q^{-1}(\mathbf{x}_n^o) &= \mathbf{I}_K + \sum_{d \in \mathcal{O}_n} \boldsymbol{\Sigma}_d^{-1}(x_{nd}), \\ \boldsymbol{\mu}_q(\mathbf{x}_n^o) &= \boldsymbol{\Sigma}_q(\mathbf{x}_n^o) \left(\sum_{d \in \mathcal{O}_n} \boldsymbol{\mu}_d(x_{nd}) \boldsymbol{\Sigma}_d^{-1}(x_{nd}) \right). \end{aligned} \tag{5}$$

Note that, in contrast to our input drop-out recognition model, in this case we need to train an independent DNN per attribute d, which might not only result in a higher computational cost, as well as in overfitting, but

it also loses the ability of DNNs to amortize the inference of the parameters across attributes, and therefore, across different missing data patterns.

Given the above generative and recognition models, described respectively by (1) and (3), the ELBO of the marginal likelihood (computed only on the observed data $\mathbf{X}^o$) can be written as

$$\begin{aligned}\log p(\mathbf{X}^o) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n^o) \\ &= \sum_{n=1}^{N} \log \int p(\mathbf{x}_n^o, \mathbf{x}_n^m, \mathbf{z}_n) d\mathbf{z}_n \, d\mathbf{x}_n^m \\ &\geq \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{z}_n|\mathbf{x}_n^o)} \left[\sum_{d \in \mathcal{O}_n} \log p(x_{nd}|\mathbf{z}_n) \right] \\ &\quad - \sum_{n=1}^{N} \mathrm{KL}\left(q(\mathbf{z}_n|\mathbf{x}_n^o)||p(\mathbf{z}_n)\right), \end{aligned} \tag{6}$$

where the first term of the ELBO corresponds to the reconstruction term of the observed data $\mathbf{X}^o$, and the Kullback-Liebler (KL) divergence in the second term penalizes that the posterior $q(\mathbf{z}_n|\mathbf{x}_n^o)$ differs for the prior $p(\mathbf{z}_n)$. Note that the KL divergence can be computed in closed-form (Kingma and Welling, 2014).

Remark. This VAE for incomplete data can readily be used to estimate the missing values in the data as follows

$$p(\mathbf{x}_n^m|\mathbf{x}_n^o) \approx \int p(\mathbf{x}_n^m|\mathbf{z}_n) q(\mathbf{z}_n|\mathbf{x}_n^o) d\mathbf{z}_n \tag{7}$$

The KL term in (6), promotes a missing-data recognition model that does not rely on the observed attributes, i.e., $p(\mathbf{x}_n^m|\mathbf{x}_n^o) \approx \int p(\mathbf{x}_n^m|\mathbf{z}_n)\mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}) d\mathbf{z}_n$. When the data are highly structured (i.e, when the statistical dependencies among the attributes in the data are strong), the reconstruction term tends to dominate and therefore this situation is avoided. However, this might not be the case for non-structured heterogeneous data, for which the combination of a variety of likelihood reconstruction terms may result in overall reconstruction log-likelihoods that are comparable to the KL term during the optimization. In such cases, one could replace the standard Normal prior by a more structured prior to easily capture the (sometimes weak) statistical dependencies in the data. We discuss this approach in more detail in Section 3.

2.2 Handling heterogenous data

In contrast to homogeneous likelihood models, where the likelihood parameters can be directly captured by a joint DNN with shared weights across dimensions (for example, all the pixels in an image are often jointly modeled by a single convolutional DNN), parameter sharing in heterogenous likelihood models is not straightforward. Interestingly, the factorized decoder in (1) can be used to easily accommodate a variety of likelihood functions, one per input attribute, where an independent DNN, $\mathbf{h}_d(\cdot)$, models the likelihood parameters $\boldsymbol{\gamma}_{nd}$ of every likelihood model $p(x_{nd}|\mathbf{z}_n) = p(x_{nd}|\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{z}_n))$, as shown in Fig. 1(b).

Here we account for the numerical and nominal data types introduced in Section 1, for which we assume the following likelihood models:

1. Real-valued data. For real-valued data, we assume a Gaussian likelihood model, i.e.,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \mathcal{N}\left(\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\right), \tag{8}$$

with $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\}$, where the mean $\mu_d(\mathbf{z}_n)$ and the variance $\sigma_d^2(\mathbf{z}_n)$ are computed as the outputs of DNNs with input $\mathbf{z}_n$.

2. Positive real-valued data. For positive real-valued data, we assume a log-normal likelihood model, i.e.,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \log \mathcal{N}\left(\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\right), \tag{9}$$

with $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\}$, where the likelihood parameters $\mu_d(\mathbf{z}_n)$ and $\sigma_d^2(\mathbf{z}_n)$ (which corresponds to the mean and variance of the variable's natural logarithm) are the outputs of DNNs with input $\mathbf{z}_n$.

3. Count data. For count data, we assume a Poisson likelihood model, i.e,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \text{Poiss}\left(\lambda_d(\mathbf{z}_n)\right), \tag{10}$$

with $\boldsymbol{\gamma}_{nd} = \lambda_d(\mathbf{z}_n)$, where the mean parameter of the Poisson distribution corresponds to the output of a DNN.

4. Categorical data. For categorical data, codified using one-hot encoding, we assume a multinomial logit model such that the R-dimensional output of a DNN $\boldsymbol{\gamma}_{nd} = \{h_{d0}(\mathbf{z}_n), h_{d1}(\mathbf{z}_n), \ldots, h_{d(R-1)}(\mathbf{z}_n)\}$ represents the vector of unnormalized probabilities, such that the probability of every category is given by

$$p(x_{nd} = r|\boldsymbol{\gamma}_{nd}) = \frac{\exp^{-h_{dr}(\mathbf{z}_n)}}{\sum_{q=1}^{R} \exp^{-h_{dq}(\mathbf{z}_n)}}. \tag{11}$$

To ensure identifiability, we fix the value of $h_{d0}(\mathbf{z}_n)$ to zero.

5. Ordinal data. For ordinal data, codified using thermometer encoding,[1] we assume the ordinal logit model (McCullagh, 1980), where the probability of each (ordinal) category can be computed as

$$p(x_{nd} = r|\boldsymbol{\gamma}_{nd}) = p(x_{nd} \leq r|\boldsymbol{\gamma}_{nd}) - p(x_{nd} \leq r-1|\boldsymbol{\gamma}_{nd}), \tag{12}$$

with

$$p(x_{nd} \leq r|\mathbf{z}_n) = \frac{1}{1 + \exp^{-(\theta_r(\mathbf{z}_n) - h_d(\mathbf{z}_n))}}. \tag{13}$$

Here, the thresholds $\theta_r(\mathbf{z}_n)$ divide the real line into R regions and $\mathbf{h}_d(\mathbf{z}_n)$ indicates the region (category) in which x_{nd} falls. Therefore, the likelihood parameters are $\boldsymbol{\gamma}_{nd} = \{h_d(\mathbf{z}_n), \theta_1(\mathbf{z}_n) \ldots, \theta_{R-1}(\mathbf{z}_n)\}$, which we model as the output of a DNN. To guarantee that $\theta_1(\mathbf{z}_n) < \theta_2(\mathbf{z}_n) < \ldots < \theta_{R-1}(\mathbf{z}_n)$, we apply a cumulative sum function to the $R-1$ positive real-valued outputs of the network.

Moreover, for all the likelihood parameters which need to be positive, we use the softplus function $f(x) = \log(1 + \exp(x))$.

Remark. The caveat of the generative model in Figure 1 is that we are losing the ability of deep neural networks to capture correlations among data attributes by amortizing the parameters. An alternative would be to use the approach in (Suh and Choi, 2016), where categorical one-hot encoded variables are approximated by continuous variables using jitter noise (uniform on [0,1]). However, this approach does not allow to combine different likelihood models or distinguish categorical and ordinal data. In Section 3, we show how to solve this limitation by using a hierarchical model.

Handling heterogenous data ranges. Apart from different types of attributes, heterogeneous datasets commonly contain numerical attributes whose values correspond to completely different domains. For example, a dataset may contain the height of different individuals with values in the order of $1.5 - 2.0$ meters, and also their income, which might reach tens or even hundreds thousands of dollars per year. In order to learn the parameters of both the generative and the reconstruction models in Figure 1, one might rely on stochastic gradient descent using at every iteration a minibatch estimate of the ELBO in (6).[2] However, the heterogenous nature of the data and these differences of value ranges between continuous variables, result in broadly different likelihood parameters (e.g., the mean of the height is much lower than the mean of the income), leading in practice to heterogenous (and potentially unstable) gradient evaluations. To avoid that the gradient evaluations of the ELBO are dominated by a subset of attributes, we apply a batch normalization layer at the input of the reconstruction model for the numerical variables, and we apply the complementary batch denormalization at the output layer of the generative model to denormalize the likelihood parameters.

1. As an example, in an ordinal variable with three categories the lowest value is encoded as [100], the middle value as [110] and the highest value as [111].
2. Although here we use the standard ELBO for VAEs, tighter log-likelihood lower bound, such as the one proposed in the importance weight encoder (IWAE) in (Burda et al., 2015), could also be applied.

Table 1: HI-VAE probabilistic model

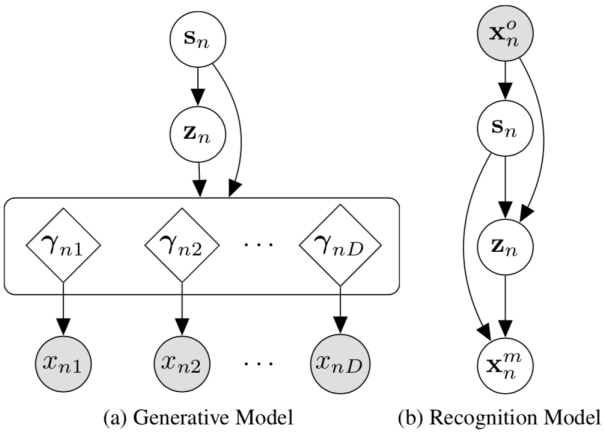
Generative	$p(\mathbf{x}_n, \mathbf{z}_n, \mathbf{s}_n) = p(\mathbf{s}_n)p(\mathbf{z}_n\|\mathbf{s}_n)\prod_d p(x_{nd}\|\boldsymbol{\gamma}_{nd})$
	$\boldsymbol{\gamma}_{nd} = h_d(\mathbf{y}_{nd}, \mathbf{s}_n)$, where $\mathbf{Y}_n = [\mathbf{y}_{n1}, \ldots, \mathbf{y}_{nD}] = \mathbf{g}(\mathbf{z}_n)$
Recognition	$q(\mathbf{s}_n, \mathbf{z}_n, \mathbf{x}_n^m\|\mathbf{x}_n^o) = q(\mathbf{s}_n\|\mathbf{x}_n^o)q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n)\prod_{d\in\mathcal{M}_n} p(x_{nd}\|\mathbf{z}_n, \mathbf{s}_n)$
	$q(\mathbf{s}_n\|\mathbf{x}_n^o) = \text{Categorical}(\boldsymbol{\pi}(\tilde{\mathbf{x}}_n))$
	$q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n) = \mathcal{N}(\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n, \mathbf{s}_n), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n, \mathbf{s}_n))$
ELBO	$\log p(\mathbf{X}^o) \geq \sum_{n=1}^N \left(\mathbb{E}_{q(\mathbf{s}_n,\mathbf{z}_n\|\mathbf{x}_n^o)}\left[\sum_{d\in\mathcal{O}_n}\log p(x_{nd}\|\mathbf{z}_n, \mathbf{s}_n)\right]\right)$ $- \sum_{n=1}^N \mathbb{E}_{q(\mathbf{s}_n\|\mathbf{x}_n^o)}\left[KL\left(q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n)\|\|p(\mathbf{z}_n\|\mathbf{s}_n)\right)\right] - \sum_{n=1}^N KL\left(q(\mathbf{s}_n\|\mathbf{x}_n^o)\|\|p(\mathbf{s}_n)\right)$
Likelihoods	Real-valued data: $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{y}_{nd}, \mathbf{s}_n), \sigma_d^2(\mathbf{s}_n)\}$ Positive real-valued data: $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{y}_{nd}, \mathbf{s}_n), \sigma_d^2(\mathbf{s}_n)\}$ Count data: $\boldsymbol{\gamma}_{nd} = \lambda_d(\mathbf{y}_{nd}, \mathbf{s}_n)$ Categorical: $\boldsymbol{\gamma}_{nd} = \{h_{d0}(\mathbf{y}_{nd}, \mathbf{s}_n), \ldots, h_{d(R-1)}(\mathbf{y}_{nd}, \mathbf{s}_n)\}$ Ordinal: $\boldsymbol{\gamma}_{nd} = \{h_d(\mathbf{y}_{nd}, \mathbf{s}_n), \theta_1(\mathbf{s}_n)\ldots, \theta_{R-1}(\mathbf{s}_n)\}$

In particular, for real-valued variables, we shift and scale the input data to the recognition model to ensure that the normalized minibatch has zero mean and variance equal to one. These shift and scale normalization parameters, μ' and σ', are afterwards used to denormalize the likelihood parameters of the Gaussian distribution, i.e., $x_{nd} \sim \mathcal{N}\left(\sigma'\boldsymbol{\mu}_d(\mathbf{z}_n) + \mu', \sigma'^2\boldsymbol{\sigma}_d^2(\mathbf{z}_n)\right)$. For positive real-valued variables, for which a log-Normal model is used, we apply the same batch normalization at the encoder and denormalization at the decoder used for real-valued variables, but to the natural logarithm of the data, instead of directly to the data. We note that count variables are not batch denormalized at the decoder, but a normalized $\log(\cdot)$ transformation is used to feed the recognition network. With this batch normalization and denormalization layers at respectively the recognition and the generative models, we do not only enforce more stable evaluations (free of numerical errors) of the gradients, but we also speed-up the convergence of the optimization.

3. The Heterogeneous-Incomplete VAE (HI-VAE)

In the previous section, we have introduced a simple VAE architecture that handles incomplete and heterogeneous data. However, this approach might be too restrictive to capture complex and high-dimensional data. More specifically, we have assumed a standard Gaussian prior on the latent variables $\mathbf{z}_n$, which might be too narrow based on the literature (Tomczak and Welling, 2018) and particularly problematic when the final goal is to estimate missing values in unstructured datasets (refer to the discussion under (7)). Similarly, we have assumed a generative model that fully factorizes for every (heterogenous) dimension in the data, losing the properties of an amortized generative model where the different dimensions share the weights of a common DNN capturing the relationships between attributes (as CNNs capture correlations between pixels in an image). In this section, we overcome these limitations of the model discussed in the previous section and remark that the models proposed in this paper are, in fact, compatible with the current developments in VAE literature.

In order to prevent the KL term in (6) from dominating the ELBO, thus penalizing rich posterior distributions for $\mathbf{z}_n$, we can impose structure in the latent variable representation $\mathbf{z}_n$ through its prior distribution.



(a) Generative Model (b) Recognition Model

Figure 2: Graphical models for the generative and recognition models of the HI-VAE.

We propose a Gaussian mixture prior $p(\mathbf{z}_n)$ (Dilokthanakul et al., 2016), such that

$$p(\mathbf{s}_n) = \text{Categorical}(\boldsymbol{\pi}) \tag{14}$$

$$p(\mathbf{z}_n|\mathbf{s}_n) = \mathcal{N}(\boldsymbol{\mu}_p(\mathbf{s}_n), \mathbf{I}_K), \tag{15}$$

where $\mathbf{s}_n$ is a one-hot encoding vector representing the component in the mixture, i.e., the mean and the variance of the Gaussian that generates $\mathbf{z}_n$. For simplicity, we assume a uniform Gaussian mixture with $\pi_\ell = 1/L$ for all ℓ.

Moreover, to ease that the model accurately captures the statistical dependencies among heterogeneous attributes, we propose a hierarchical structure that allows different attributes to share network parameters (i.e., to amortize the generative model). More specifically, we introduce an intermediate homogenous representation of the data $\mathbf{Y} = [\mathbf{y}_{n1}, \dots, \mathbf{y}_{nD}]$, which is jointly generated by a single DNN with input $\mathbf{z}_n$, $\mathbf{g}(\mathbf{z}_n)$. Then, the likelihood parameters of each attribute d are the output of an independent DNN with inputs $\mathbf{y}_{nd}$ and $\mathbf{s}_n$, such that $p(x_{nd}|\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{y}_{nd}, \mathbf{s}_n))$. Note that, in this hierarchical structure, the top level (from $\mathbf{z}_n$ to $\mathbf{Y}_n$) captures statistical dependencies among the attributes through the shared DNN $\mathbf{g}(\mathbf{z}_n)$, while the bottom level in the hierarchy (from $\mathbf{Y}_n$ and $\mathbf{s}_n$ to $\mathbf{x}_n$) accounts for heterogeneous likelihood models using d independent DNNs $\mathbf{h}_d(\mathbf{y}_{nd}, \mathbf{s}_n)$.

The resulting generative model, that is hereafter referred to as Heterogeneous-Incomplete VAE (HI-VAE), is shown in Figure 2 and formulates as indicated in Table 1, which also shows how we parametrize in the HI-VAE the different likelihood models provided in Section 3.2.[3] Note that $\boldsymbol{\pi}(\tilde{\mathbf{x}}_n)$ is the soft-max output of a DNN with input $\tilde{\mathbf{x}}_n$. The Gumbel-softmax reparameterization trick (Jang et al., 2016) is used to draw differentiable samples from $q(\mathbf{s}_n, \mathbf{z}_n|\mathbf{x}_n^o)$.

4. Experiments

In this section, we first compare the performance of the HI-VAE to other methods in the literature for data completion tasks in heterogeneous data. Then, we focus on a classification task, where we evaluate the classification degradation due to performing mean imputation for the missing data in supervised models compared to using the fully generative HI-VAE, which does not require data imputation. An additional empirical comparison between the HI-VAE with an input drop-out encoder, a factorized encoder as in (5)

3. Other likelihood functions (e.g., a Gamma distribution) and data types (e.g., interval data using e.g. a Beta distribution) can be readily be incorporated.

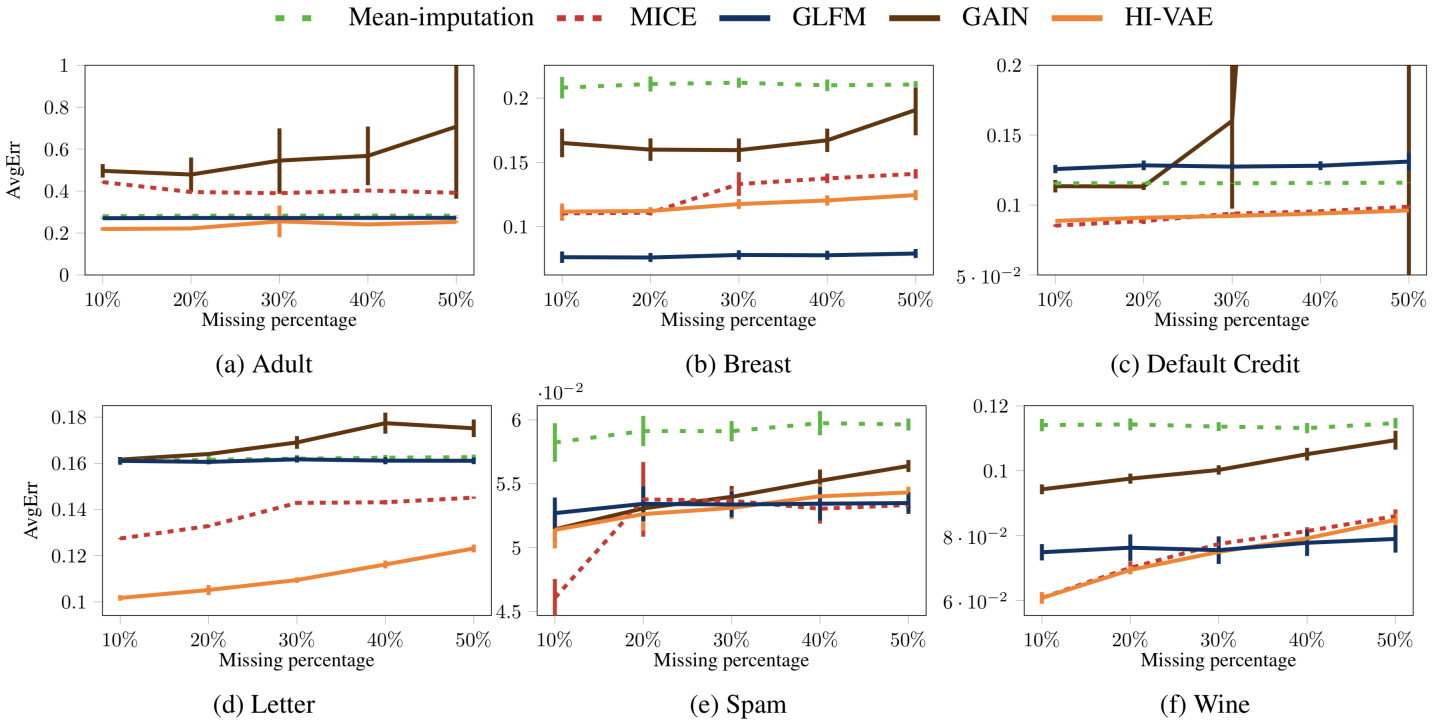


Figure 3: *Missing Data.* Average imputation error for different percentages of missing data (completely at random).

Table 2: *Imputation error.* Average and standard deviation of the imputation error for a 20% of missing data, evaluated exclusively over **numeric variables**.

Model	Adult	Breast	DefaultCredit	Letter	Spam	Wine
Mean imputation	0.111 ± 0.002	–	0.056 ± 0.001	–	0.053 ± 0.001	0.103 ± 0.002
MICE	0.108 ± 0.002	–	$\mathbf{0.035 \pm 0.002}$	–	0.052 ± 0.003	0.074 ± 0.002
GLFM	$\mathbf{0.083 \pm 0.001}$	–	0.051 ± 0.005	–	0.052 ± 0.001	0.082 ± 0.004
GAIN	0.225 ± 0.192	–	0.044 ± 0.002	–	$\mathbf{0.049 \pm 0.001}$	0.086 ± 0.002
HI-VAE	0.106 ± 0.002	–	0.043 ± 0.001	–	0.052 ± 0.001	$\mathbf{0.074 \pm 0.001}$

to handle missing data, and a non-structured VAE with a simple Gaussian prior instead of a mixture model distribution at the latent space is provided in the Appendix. The code to reproduce all our experiments can be found in the following public repository `https://github.com/probabilistic-learning/HI-VAE`.

4.1 Missing data imputation

In our first experiment, we evaluate the performance of the proposed HI-VAE at imputing missing data. We use six different heterogenous datasets from the UCI repository (Lichman, 2013), which vary both in the number of instances and attributes, as well as in the statistical data types of the attributes. The details of all these datasets are provided in the Appendix. For each dataset we generate 10 different incomplete datasets, removing completely at random a percentage of the data, ranging from a 10% deletion to a 50%.

Comparison. We compare the performance of the following methods for missing data imputation:

- **Mean Imputation**: We use as baseline an algorithm that imputes the mean of each continuous attribute and the mode of each discrete attribute.
- **MICE:** Multiple Imputation by Chained Equations (Azur et al., 2011), which is an iterative method that performs a series of supervised regression models, in which missing data is modeled conditional upon the other variables in the data, including those imputed in previous rounds of the algorithm. We use MICE implementation within the *fancyimpute* package `https://github.com/iskandr/fancyimpute`, which in its current implementation only allows to pick for a homogeneous regression model for all attributes, independently of whether they are numerical or nominal.
- **GLFM:** General latent feature model for heterogeneous data (Valera et al., 2017b), which was initially introduced for table completion in heterogeneous datasets in (Valera and Ghahramani, 2014). This method handles all the numerical and nominal data types described in Section 1 and perform MCMC inference. We run 5000 iterations of the sampler using the available implementation in `https://github.com/ivaleraM/GLFM`.
- **GAIN:** Generative adversarial network for missing data imputation (Yoon et al., 2018), which uses MSE as a loss function for numerical variables, and cross-entropy for binary variables. We train GAIN for 2000 epochs using the network specifications and hyperparameters reported in (Yoon et al., 2018).
- **HI-VAE:** Model introduced in Section 3, which we implement in TensorFlow using only one dense layer for all the parameters of the encoder and decoder of the HI-VAE). We set the dimensionality of $\mathbf{z}, \mathbf{y}$ and $\mathbf{s}$ to 10, 5 and 10, respectively. The parameter τ of the Gumbel-Softmax is annealed using a linear decreasing function on the number of epochs, from 1 to 10^{-3}. We train our algorithms for 2000 epochs using minibatches of 1000 samples.

Imputation strategy. Once the HI-VAE model is trained, the imputation of missing data is performed in a three step process: First, we perform the MAP estimate of $q(\mathbf{z}_n, \mathbf{s}_n|\mathbf{x}_n^o)$ to obtain $\hat{\mathbf{z}}_n$ and $\hat{\mathbf{s}}_n$. With these MAP estimates, we evaluate the generative model, obtaining $\hat{\mathbf{Y}}_n = \mathbf{g}(\hat{\mathbf{z}}_n)$ and $\hat{\gamma}_{nd} = \mathbf{h}_d(\hat{\mathbf{y}}_{nd}, \hat{\mathbf{s}}_n)$ for every attribute. Finally, the imputed values $\hat{\mathbf{x}}_n$ are obtained computing the mode of each distribution $p(x_{nd}|\hat{\gamma}_{nd})$, where the computation of the mode depends on the likelihood model of the attribute. A further discussion on imputation methods is available in the Appendix.

Imputation error. We compare the above models in terms of average imputation error computed as AvgErr $= 1/D \sum_d \text{err}(d)$, where we use the normalized root mean square error (NRMSE) for numerical variables, the accuracy error for categorical variables, and the displacement error for ordinal variables. See the Appendix for a precise definition of each error metric.

Results. Figure 3 summarizes the average imputation error (AvgErr) for each database as we vary the fraction of missing data. Observe that the proposed HI-VAE (with input drop-out encoder) presents the more robust results across all datasets. The second more robust model is the GLFM, which performs best in small datasets (Breast and Wine). This might be explained by the fact that while it accounts mixed nominal and discrete data, it relies on Gibbs-sampling for inference, scaling and mixing poorly for larger datasets. In contrast, the

Table 3: *Imputation error.* Average and standard deviation of the imputation error for a 20% of missing data, evaluated exclusively over **nominal variables**.

Model	Adult	Breast	DefaultCredit	Letter	Spam	Wine
Mean imputation	0.405 ± 0.002	0.211 ± 0.006	0.2 ± 0.001	0.162 ± 0.002	0.393 ± 0.014	0.248 ± 0.014
MICE	0.601 ± 0.002	0.111 ± 0.002	0.163 ± 0.003	0.133 ± 0.0	0.168 ± 0.012	0.02 ± 0.004
GLFM	0.407 ± 0.003	$\mathbf{0.076 \pm 0.003}$	0.236 ± 0.012	0.161 ± 0.001	0.154 ± 0.02	$\mathbf{0.006 \pm 0.001}$
GAIN	0.66 ± 0.025	0.16 ± 0.009	0.211 ± 0.005	0.164 ± 0.001	0.276 ± 0.017	0.236 ± 0.014
HI-VAE	$\mathbf{0.304 \pm 0.006}$	0.112 ± 0.003	$\mathbf{0.158 \pm 0.001}$	$\mathbf{0.105 \pm 0.002}$	$\mathbf{0.111 \pm 0.009}$	0.016 ± 0.003

Table 4: *Prediction Accuracy.* Average and standard deviation of the classification error when we use 50% of the labels for training and assume complete input data and 10% and 50% of missing values in input data (right-hand table).

% Missing	Model	Breast	DefaultCredit	Letter	Spam	Wine
0%	DLR	0.041 ± 0.01	$\mathbf{0.179 \pm 0.002}$	0.142 ± 0.003	$\mathbf{0.081 \pm 0.005}$	0.018 ± 0.003
	CVAE	0.04 ± 0.012	$\mathbf{0.179 \pm 0.001}$	$\mathbf{0.14 \pm 0.004}$	0.081 ± 0.006	0.016 ± 0.002
	HIVAE	$\mathbf{0.026 \pm 0.005}$	0.2 ± 0.004	0.372 ± 0.012	0.096 ± 0.007	$\mathbf{0.014 \pm 0.002}$
10%	DLR	0.04 ± 0.009	$\mathbf{0.184 \pm 0.001}$	0.229 ± 0.002	0.09 ± 0.005	0.027 ± 0.003
	CVAE	0.048 ± 0.009	0.184 ± 0.002	$\mathbf{0.227 \pm 0.003}$	$\mathbf{0.088 \pm 0.006}$	0.025 ± 0.003
	HIVAE	$\mathbf{0.031 \pm 0.007}$	0.201 ± 0.002	0.498 ± 0.006	0.103 ± 0.008	$\mathbf{0.022 \pm 0.006}$
50%	DLR	0.08 ± 0.014	$\mathbf{0.196 \pm 0.003}$	$\mathbf{0.496 \pm 0.005}$	$\mathbf{0.134 \pm 0.008}$	0.078 ± 0.006
	CVAE	0.101 ± 0.038	0.197 ± 0.003	$\mathbf{0.496 \pm 0.005}$	0.138 ± 0.009	0.078 ± 0.005
	HIVAE	$\mathbf{0.052 \pm 0.012}$	0.205 ± 0.003	0.749 ± 0.012	0.138 ± 0.005	$\mathbf{0.042 \pm 0.005}$

MICE and GAIN[4] are outperformed by the Mean-imputation baseline in several datasets, most likely, due to the fact that they do not account for different types of mixed nominal and numerical attributes.

A deeper understanding of the results in Figure 3 can be obtained by separately analyzing the error in numeric variables (real, positive and count variables) in Table 2, and nominal variables (categorical/ordinal variables) in Table 3. In both cases, we use 20% of missing data. While for numeric variables HI-VAE achieves a comparable error w.r.t. the rest of methods, it is in the imputation of nominal variables where HI-VAE achieves a remarkable gain, being the best performing method in four out of six cases. These results demonstrate the superior ability of HI-VAE to exploit underlying correlations among the set of heterogeneous attributes. For a further discussion on the imputation for each type of nominal and numerical variable, refer to the Appendix. We note that we use the same HI-VAE configuration (i.e., DNN structure and number of latent variables) in all experiments and, therefore, further improvements could be achieved by cross-validating the structure of the HI-VAE for each database.

4.2 Predictive Task

Finally, although the HI-VAE is a fully unsupervised generative model, we evaluate its performance at solving a classification task, a multi-class classification problem for the Letter dataset (with 26 classes corresponding to the different letters) and a binary classification problem for the rest. We use 50% of the data for training, which for HI-VAE means that we remove 50% of the labels to train the generative model. Regarding the training data, we consider three different scenarios: the first assumes complete input attributes in the training set (no missing data), the second assumes 10% of missing values in the input training data, and the third assumes 50% of missing values. Since these supervised methods cannot handle missing data, we impute the mean of each attribute to the missing input values during training. Here, we compare our HI-VAE with two supervised methods: deep logistic regression (DLR) and the conditional VAE (CVAE) in (Sohn et al., 2015).

Results. Table 4 summarizes the results, where we observe that our HI-VAE method provides competitive results in all cases except for the Letter database. This may be due to the fact that we are using the same HI-VAE configuration for all datasets, independently of their complexity.Furthermore, note HI-VAE provides the best results for both Wine and Breast, while showing less degradation with increasing fraction of missing input data in the DefaultCredit and Spam. These results show that a fully generative model might be preferred over a supervised model with imputed data.

4. We would like to clarify that the reported results do not quite match those provided in (Yoon et al., 2018), despite using the code and the hyperparameters provided by the authors. For the sake of reproducibility, we will incorporate the GAIN implementation to our public repository.

5. Acknowledgments

The authors wish to thank Christopher K. I. Williams, for fruitful discussions and helpful comments to the manuscript. Alfredo Nazabal would like to acknowledge the funding provided by the UK Government's Defence & Security Programme in support of the Alan Turing Institute. The work of Pablo M. Olmos is supported by Spanish government MEC under grant TEC2016-78434-C3-3-R, by Comunidad de Madrid under grant IND2017/TIC-7618, and by the European Union (FEDER). Zoubin Ghahramani acknowledges support from the Alan Turing Institute (EPSRC Grant EP/N510129/1) and EPSRC Grant EP/N014162/1, and donations from Google and Microsoft Research. Isabel Valera is supported by the MPG Minerva Fast Track program. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017. URL http://arxiv.org/abs/1701.07875.

Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *CoRR*, abs/1706.08224, 2017. URL http://arxiv.org/abs/1706.08224.

Melissa J. Azur, Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1):40–49, 3 2011. ISSN 1049-8931. doi: 10.1002/mpr.329.

Yoshiaki Bando, Masato Mimura, Katsutoshi Itoyama, Kazuyoshi Yoshii, and Tatsuya Kawahara. Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization. *CoRR*, abs/1710.11439, 2017. URL http://arxiv.org/abs/1710.11439.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *CoRR*, abs/1611.02731, 2016. URL http://arxiv.org/abs/1611.02731.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, 2016.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2946645.2946704.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5769–5779, 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *CoRR*, abs/1611.01144, 2016. URL http://arxiv.org/abs/1611.01144.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1857–1865, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/kim17a.html.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*. 2014.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1718–1727, 2015. URL http://jmlr.org/proceedings/papers/v37/li15.html.

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 700–708. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6672-unsupervised-image-to-image-translation-networks.pdf.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142, 1980. ISSN 00359246. URL http://www.jstor.org/stable/2984952.

Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2391–2400, 2017. URL http://proceedings.mlr.press/v70/mescheder17a.html.

Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching GAN. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2527–2535, 2017. URL http://proceedings.mlr.press/v70/mroueh17a.html.

Charlie Nash and Chris Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. *Computer Graphics Forum*, 36(5):1–12, 2017. doi: 10.1111/cgf.13240. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13240.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. pages 271–279, 2016.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/rezende15.html.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015.

Suwon Suh and Seungjin Choi. Gaussian copula variational autoencoders for mixed data. *arXiv preprint arXiv:1604.04960*, 2016.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *CoRR*, abs/1611.02200, 2016. URL http://arxiv.org/abs/1611.02200.

Ilya O. Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. *CoRR*, abs/1701.02386, 2017. URL http://arxiv.org/abs/1701.02386.

Jakub M. Tomczak and Max Welling. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1214–1223, 2018.

Arash Vahdat, William G. Macready, Zhengbing Bian, and Amir Khoshaman. Dvae++: Discrete variational autoencoders with overlapping transformations, 2018. URL http://arxiv.org/abs/1802.04920. cite arxiv:1802.04920.

I. Valera, M. F. Pradier, M. Lomeli, and Z. Ghahramani. General latent feature models for heterogeneous datasets. *arXiv preprint arXiv:1706.03779*, 2017a.

Isabel Valera and Zoubin Ghahramani. General table completion using a bayesian nonparametric model. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 981–989. Curran Associates, Inc., 2014.

Isabel Valera, Melanie F. Pradier, and Zoubin Ghahramani. General latent feature modeling for data exploration tasks. *CoRR*, abs/1707.08352, 2017b. URL http://arxiv.org/abs/1707.08352.

Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. *arXiv preprint arXiv:1705.10762*, 2017.

Christopher KI Williams and Charlie Nash. Autoencoders and probabilistic inference with missing data: An exact solution for the factor analysis case. *arXiv preprint arXiv:1801.03851*, 2018.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/yang17d.html.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18), Stockholm (Sweden), July 2018.*, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017. doi: 10.1109/ICCV.2017.244. URL https://doi.org/10.1109/ICCV.2017.244.

Table 5: *Datasets*

Database	Objects	Attributes	# Real	# Positive	# Categorical	# Ordinal	# Count
Adult	32561	12	0	3	6	1	2
Breast	699	10	0	0	1	9	0
Default Credit	30000	24	6	7	4	6	1
Letter	20000	17	0	0	1	16	0
Spam	4601	58	0	57	1	0	0
Wine	6497	13	0	11	1	0	1

Appendix

Error metrics.

We compare the above models in terms of average imputation error computed as AvgErr $= 1/D \sum_d \text{err}(d)$, where we use the following error metrics for each attribute, since the computation of the errors depends on the type of variable we are considering:

- Normalized Root Mean Square Error (NRMSE) for numerical variables, i.e.,

$$\text{err}(d) = \frac{\sqrt{1/n \sum_n (x_{nd} - \hat{x}_{nd})^2}}{\max(\mathbf{x}_d) - \min(\mathbf{x}_d)}. \tag{16}$$

- Accuracy error for categorical variables, i.e.,

$$\text{err}(d) = \frac{1}{n} \sum_n I(x_{nd} \neq \hat{x}_{nd}). \tag{17}$$

- Displacement error for ordinal variables, i.e.,

$$\text{err}(d) = \frac{1}{n} \sum_n |\frac{x_{nd} - \hat{x}_{nd}}{R}|. \tag{18}$$

Databases characteristics.

We use six databases borrowed from the UCI repository `https://archive.ics.uci.edu/ml/index.php`. We summarize their main characteristics in Table 5.

Imputation error per attribute.

We augment the experimental evaluation in Section 5.1 of the main document by illustrating the imputation error per attribute when we have a 20% fraction of missing data. It can be seen that HI-VAE is in general superior for imputing nominal variables (ordinal or categorical ones).

Variations on the HI-VAE construction.

In Figures 5 and 6 we compare three different approaches to implement the HI-VAE generative model. We compare the HI-VAE with mixture model prior distribution at the latent space with input dropout (HI-VAE), which is the model we use in the main document, with a HI-VAE that uses the factorized model (5) in the main document to handle missing data (HI-VAE factorized), and a HI-VAE in which the latent variables $\mathbf{z}$ in the generative model are Gaussian distributed, e.g. we do not use a mixture model at the latent space (HI-VAE Gaussian prior). We compare the results in terms of imputation errors, Figure 5, and in terms of test log-likelihood, Figures 6. The standard HI-VAE provides slightly better error performance for both Default Credit and Wine datasets, and provides the best test log-likelihood in the Breast dataset.

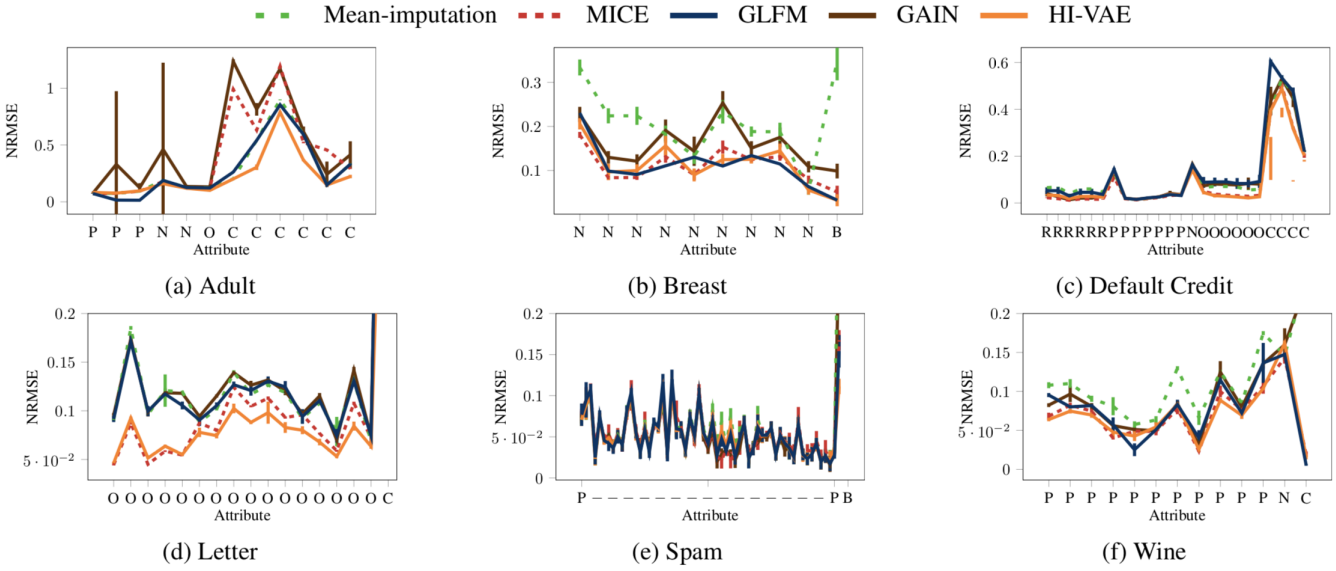


Figure 4: *Missing Data.* Imputation error per attribute in the data for a 20% of missing values. Here, 'R' stands for real-valued variables, 'P' for positive real-valued , 'C' for categorical, 'O' for ordinal and 'N' for count.

Beyond a slight imputation improvement in some cases, HI-VAE has much less parameters than HI-VAE factorized (which has a different NN per missing dimension in the inference model) and, compared to HI-VAE Gaussian prior, the structure provided by the mixture model in the latent space naturally yields data clustering at the latent space, hence providing more discriminative data embeddings and emphasises more interpretable generative models. For instance, in Figure 7, we show the latent space induced by the HI-VAE Gaussian prior and the HI-VAE for the Breast dataset, when both use a latent space dimension of 2 and there is a 50% of missing data. It can be observed that HI-VAE induces more separated and disentangled clusters.

HI-VAE imputation: sampling vs. mode.

Once we have trained the generative model, to impute missing data we can either sample from the generative model or use the inferred parameters of the output distribution, e.g. impute the mode of the inferred distribution (this is what we did in the main document). To illustrate the differences, we show in Figures 8 and 9 the goodness of fit provided by the HI-VAE and the GLFM in a positive real-valued variable and a categorical variable with 6 categories, both belonging to the Adult dataset. Specifically, we show (top row) the true distribution of the data together with the HI-VAE output distribution for the observed data and the HI-VAE output distribution for the missing values. We show results for HI-VAE using the mode of the distribution and HI-VAE using one sample for imputation. We also show results for the GLFM. Further, in the bottom row we show the Q-Q plot for the positive-real variable and the confusion matrix for the categorical one. See the figure caption for more details. Note that, while both the HI-VAE and the GLFM result in a good fitting of the positive variable (although the HI-VAE provides a smoother, and thus, more realistic distribution for the data); the GLFM fails at capturing the categorical variable–it assigns all the probability to a single category. These results are consistent with Table 3 in the paper, which demonstrate the superior ability of the HI-VAE to perform missing data imputation in nominal variables.

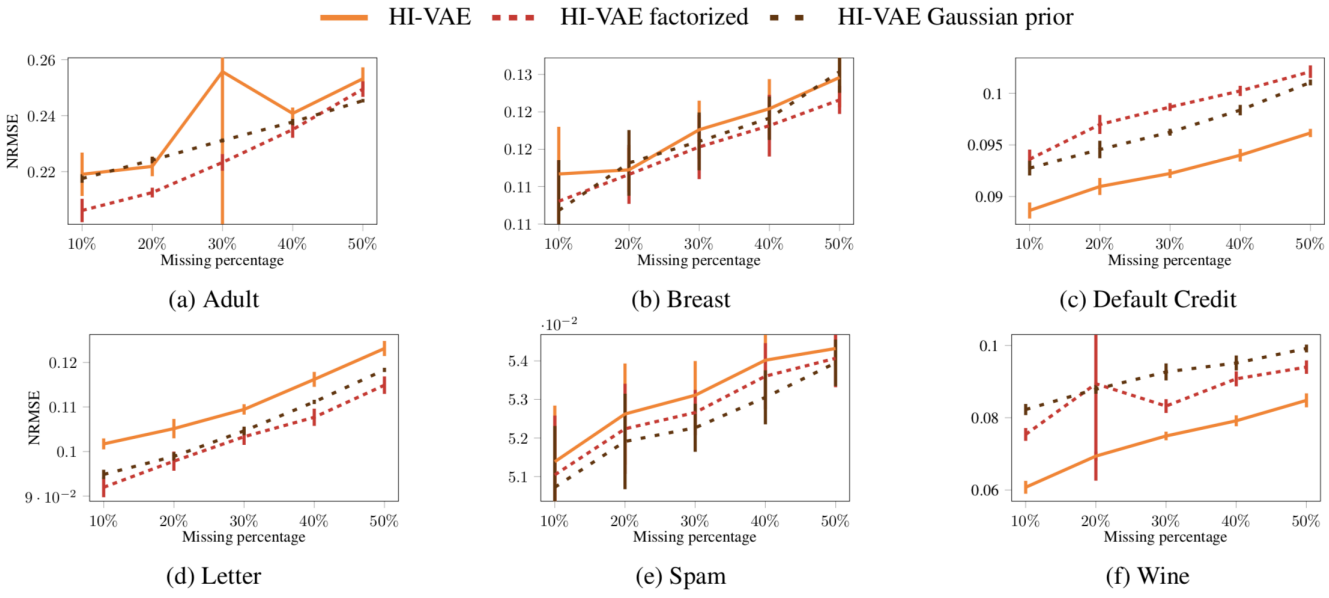


Figure 5: *Missing Data.* Imputation error for different percentages of missing data (completely at random).

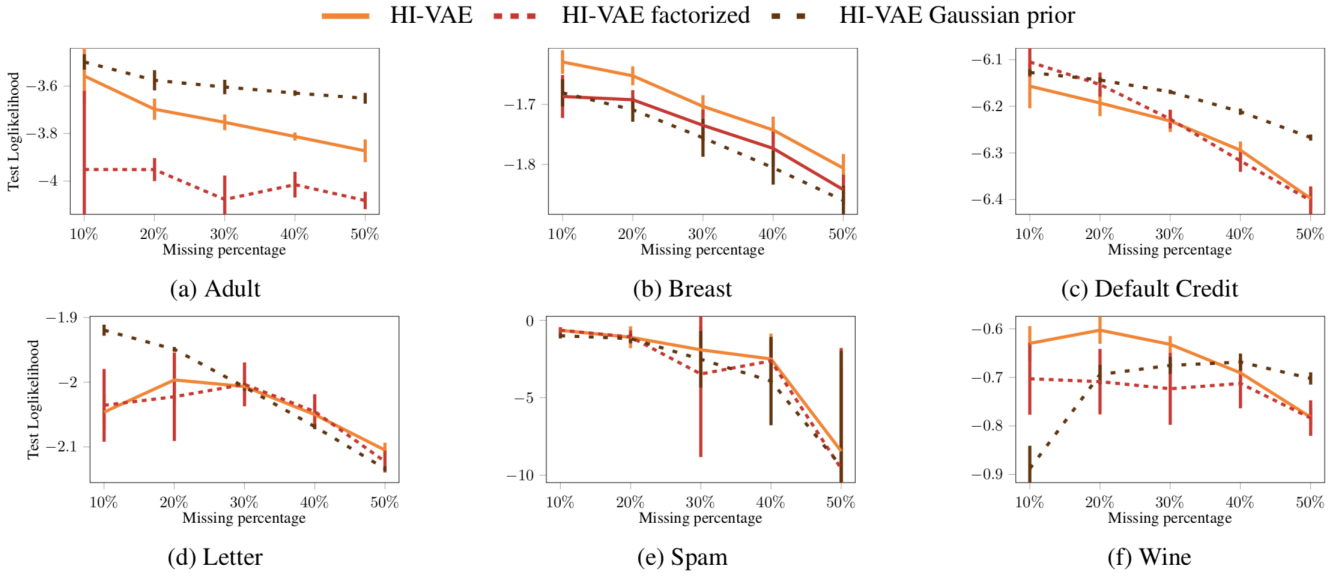


Figure 6: *Missing Data.* Test log-likelihood for different percentages of missing data (completely at random).

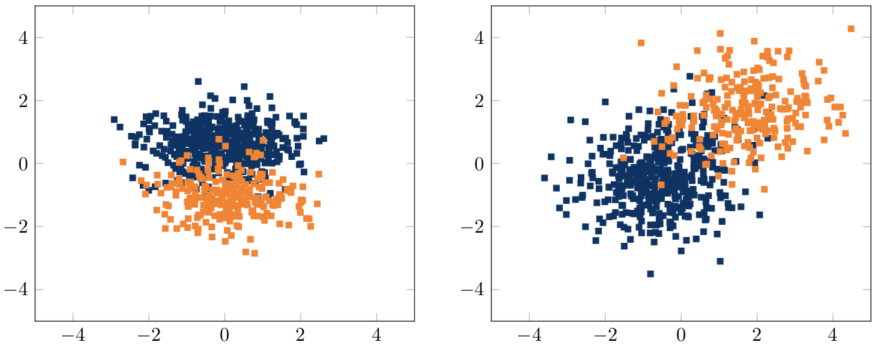


Figure 7: Latent space using the Breast dataset and a 50% of missing data for the HI-VAE Gaussian prior (a) and the HI-VAE (b).

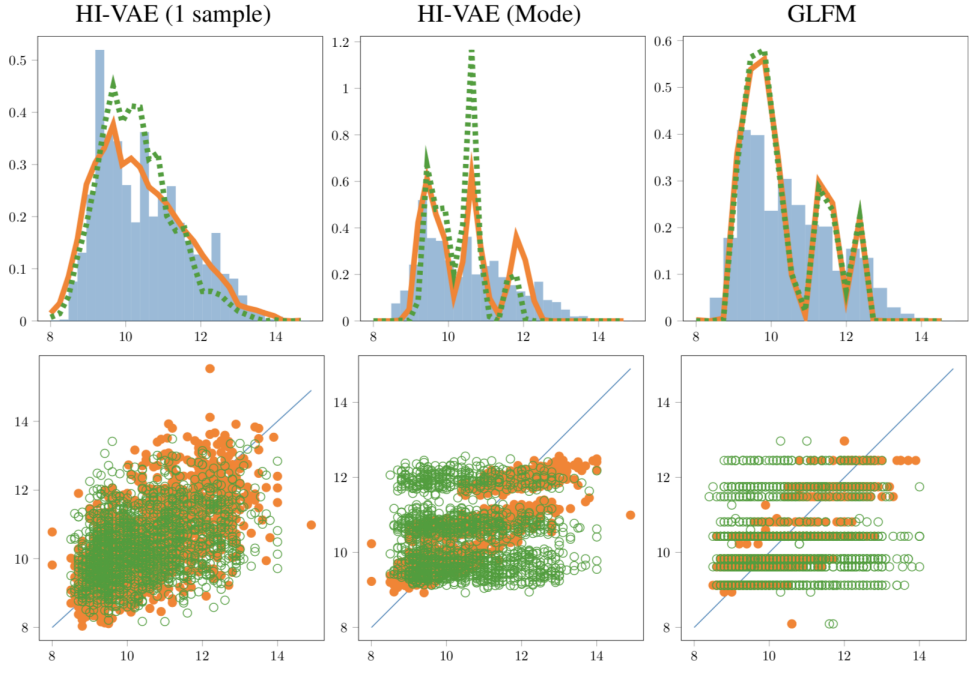


Figure 8: We demonstrate the fit provided by the HI-VAE and the GLFM in a positive real-valued variable of the Adult dataset. Top row depicts the true empirical data distribution (shadowed histogram) and the inferred data distribution for the observed attributes in dashed line and for the missing data in solid line. The bottom row shows the Q-Q plot (observed in orange (•) marker and missing in green (○) marker). The left-most column shows the results for the HI-VAE when we sample from the model posterior distribution (given the observed data) to impute, while for the center column we use the mode of the posterior.

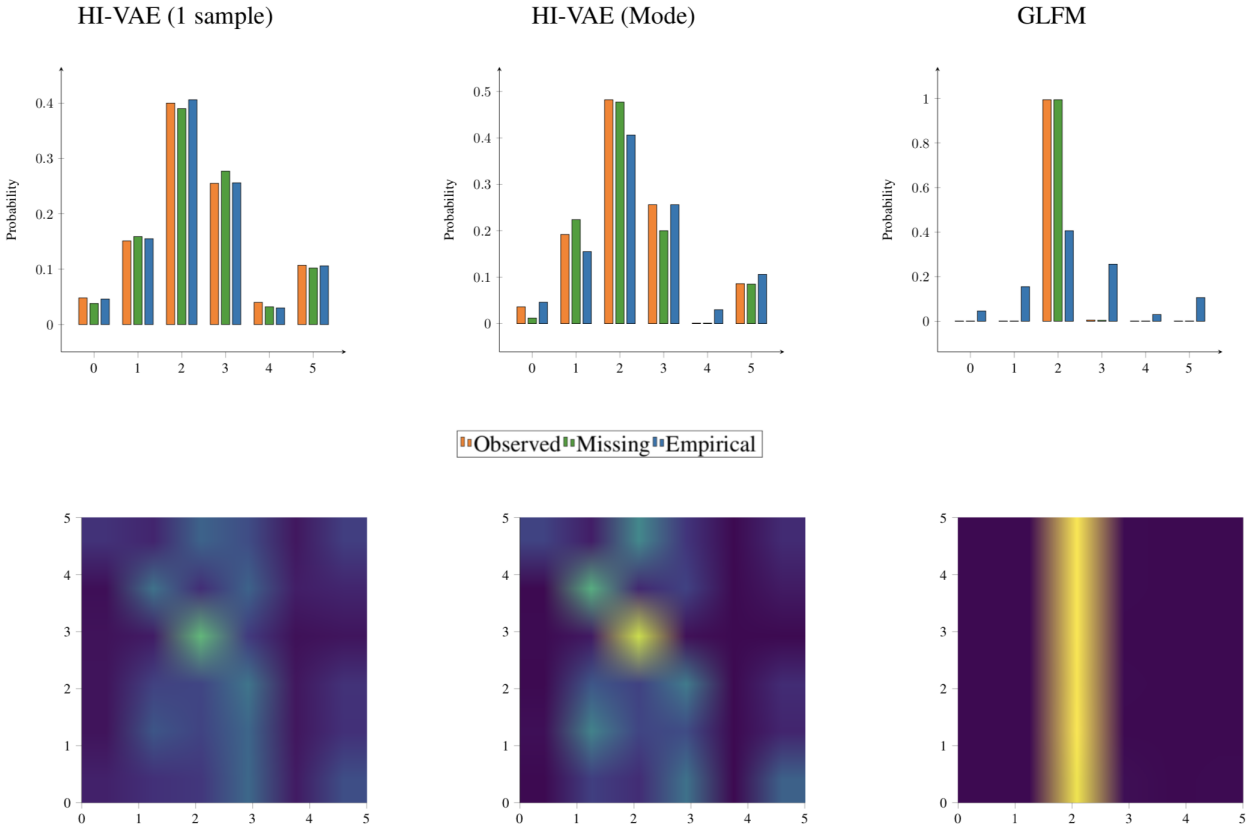


Figure 9: We demonstrate the fit provided by the HI-VAE and the GLFM in a categorical variable with 6 categories of the Adult dataset. Top row depicts the true empirical data distribution and the inferred data distribution for the observed attributes and for the missing data. The bottom row shows the missing data confusion matrix.