Deep Generative Missingness Pattern-Set Mixture Models: Supplementary Materials

Sahra Ghalebikesabi
University of Oxford

Rob Cornish
University of Oxford

Chris Holmes[1]
University of Oxford

Luke J. Kelly
CEREMADE, CNRS
Université Paris-Dauphine

1 PROOFS

Proof of Proposition 1. Let μ denote the posterior distributional parameter of $(\mathbf{x_{mis}} \mid \mathbf{x_{obs}}, \mathcal{M}) \sim \mathcal{N}(\mu, c)$ for some fixed constant c. In the following, we assume that the distributional parameter is found by optimization of the expected likelihood given the augmented dataset

$$\mathbf{E_y}[P_\theta(\mathbf{x_{obs}}, \mathbf{x_{mis,1-y}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{y}))] = \boldsymbol{\pi} P_\theta(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) + (1-\boldsymbol{\pi}) P_\theta(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})). \tag{1}$$

Let θ denote the parameter set of the generative model. We write θ_1 for the parameter set where $\mu = \mu_1$ and θ_2 for the parameter set where $\mu = \mu_2$. For any two parameters μ_1 and μ_2 with

$$\begin{aligned}\boldsymbol{\pi} P_{\theta_1}(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) + (1-\boldsymbol{\pi}) P_{\theta_1}(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})) =& \boldsymbol{\pi} P_{\theta_2}(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) \\ &+ (1-\boldsymbol{\pi}) P_{\theta_2}(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})),\end{aligned}$$

we have to show that $\mu_1 = \mu_2$.

From the identifiability of the mean parameter of Gaussian distributions as sample mean in maximum likelihood estimation, it follows that

$$\begin{aligned}\boldsymbol{\pi} P_{\theta_1}(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) &= \boldsymbol{\pi} P_{\theta_2}(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) \text{ and thus} \\ (1-\boldsymbol{\pi}) P_{\theta_1}(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})) &= (1-\boldsymbol{\pi}) P_{\theta_2}(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})),\end{aligned} \tag{2}$$

where only the probabilities in Equation (2) depend on μ. The maximization of $P_{\theta_2}(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0}))$ equals a maximum likelihood estimation with a fully observed dataset of $\mathbf{x_{mis}}$. Again, from the identifiability of the mean parameter of Gaussian distributions, it follows that $\mu_1 = \mu_2$ from (2). □

Proof of ELBO in Table 1. The expected likelihood given the augmented dataset (1) is a weighted sum over the likelihood of the observed model $P_\theta(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0}))$ and the likelihood of the unobserved model $P_\theta(\mathbf{x_{obs}}, \mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1}))$.

The observed model can be learned by maximizing the ELBO which we can compute using Jensen's inequality:

$$\begin{aligned}\log P_\theta(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})) &= \log P_\theta(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}}, \mathbf{r}, \mathbf{z} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})) - \log P_\theta(\mathbf{r}, \mathbf{z} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{0})) \\ &\geq \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z} \mid \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}})}\Big[\log P_\theta(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}}, \mathbf{r}, \mathbf{z} \mid \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}}) - \log Q_\phi(\mathbf{r}, \mathbf{z} \mid \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}})\Big] \\ &= \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z} \mid \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}})}\Big[\log P_\theta(\mathbf{m} \mid \mathbf{r}, \mathbf{z}, \mathbf{x_{obs}}, \mathbf{x_{mis}}) + \log P_\theta(\mathbf{x_{obs}} \mid \mathbf{r}, \mathbf{z}) + \log P_\theta(\mathbf{x_{mis}} \mid \mathbf{r}, \mathbf{z}) \\ &\qquad + \log \frac{P_\theta(\mathbf{z} \mid \mathbf{r})}{Q_\phi(\mathbf{z} \mid \mathbf{r}, \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}})} + \log \frac{P_\theta(\mathbf{r})}{Q_\phi(\mathbf{r} \mid \mathbf{x_{obs}}, \mathbf{m}, \mathbf{x_{mis}})}\Big]\end{aligned}$$

[1] denotes senior author

$$= \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z}|\mathbf{x_{obs}},\mathbf{m})}\Bigg[\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) + \log P_\theta(\mathbf{x_{obs}} \mid \mathbf{r},\mathbf{z}) + \log P_\theta(\mathbf{x_{mis}} \mid \mathbf{r},\mathbf{z}) + \log \frac{P_\theta(\mathbf{z} \mid \mathbf{r})}{Q_\phi(\mathbf{z} \mid \mathbf{r},\mathbf{x_{obs}},\mathbf{m})} + \log \frac{P_\theta(\mathbf{r})}{Q_\phi(\mathbf{r} \mid \mathbf{x_{obs}},\mathbf{m})}\Bigg],$$

where $\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) = \log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{x_{obs}},\mathbf{x_{mis}})$ for PSMVAE(a) and $\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) = \log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z})$ for PSMVAE(b). The last equation follows from the assumption that $Q_\phi(\cdot|\mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}}) = Q_\phi(\cdot|\mathbf{x_{obs}},\mathbf{m})$.

Similarly we obtain the lower bound of the unobserved model:

$$\begin{aligned}
\log P_\theta(\mathbf{x_{obs}},\mathbf{m} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) &= \log P_\theta(\mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}},\mathbf{r},\mathbf{z} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) - \log P_\theta(\mathbf{r},\mathbf{z},\mathbf{x_{mis}} \mid \mathcal{D}_{\boldsymbol{\pi}}(\mathbf{1})) \\
&\geq \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z},\mathbf{x_{mis}}|\mathbf{x_{obs}},\mathbf{m})}\Big[\log P_\theta(\mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}},\mathbf{r},\mathbf{z} \mid \mathbf{x_{obs}},\mathbf{m}) - \log Q_\phi(\mathbf{r},\mathbf{z},\mathbf{x_{mis}} \mid \mathbf{x_{obs}},\mathbf{m})\Big] \\
&= \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z},\mathbf{x_{mis}}|\mathbf{x_{obs}},\mathbf{m})}\Bigg[\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) + \log P_\theta(\mathbf{x_{obs}} \mid \mathbf{r},\mathbf{z}) + \log \frac{P_\theta(\mathbf{x_{mis}} \mid \mathbf{r},\mathbf{z})}{Q_\phi(\mathbf{x_{mis}} \mid \mathbf{x_{obs}},\mathbf{m},\mathbf{z},\mathbf{r})} \\
&\qquad + \log \frac{P_\theta(\mathbf{z} \mid \mathbf{r})}{Q_\phi(\mathbf{z} \mid \mathbf{r},\mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}})} + \log \frac{P_\theta(\mathbf{r})}{Q_\phi(\mathbf{r} \mid \mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}})}\Bigg],
\end{aligned}$$

where $\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}})$ is defined as before.

The weighted sum of the ELBOs (denoted by $\mathcal{L}(\mathbf{x_{obs}},\mathbf{m})$ in the following) can then be written as

$$\begin{aligned}
\mathcal{L}(\mathbf{x_{obs}},\mathbf{m}) = \mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z}|\mathbf{x_{obs}},\mathbf{m})}&\Bigg[\log P_\theta(\mathbf{x_{obs}} \mid \mathbf{r},\mathbf{z}) + \log \frac{P_\theta(\mathbf{z} \mid \mathbf{r})}{Q_\phi(\mathbf{z} \mid \mathbf{r},\mathbf{x_{obs}},\mathbf{m})} + \log \frac{P_\theta(\mathbf{r})}{Q_\phi(\mathbf{r} \mid \mathbf{x_{obs}},\mathbf{m})}\Bigg] \\
+ \pi\mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z},\mathbf{x_{mis}}|\mathbf{x_{obs}},\mathbf{m})}&\Bigg[\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) + \log \frac{P_\theta(\mathbf{x_{mis}} \mid \mathbf{r},\mathbf{z})}{Q_\phi(\mathbf{x_{mis}} \mid \mathbf{x_{obs}},\mathbf{m},\mathbf{z},\mathbf{r})}\Bigg] \\
+ (1-\pi)\mathbb{E}_{Q_\phi(\mathbf{r},\mathbf{z}|\mathbf{x_{obs}},\mathbf{m},\mathbf{x_{mis}})}&\Big[\log P_\theta(\mathbf{m} \mid \mathbf{r},\mathbf{z},\mathbf{x_{obs}},\mathbf{x_{mis}}) + \log P_\theta(\mathbf{x_{mis}} \mid \mathbf{r},\mathbf{z})\Big].
\end{aligned}$$

Specifying $\pi = 1$ when $m_j = 0$ yields the ELBO, as shown in Table 1 of the main paper. □

2 DETAILS OF DATASETS AND EXPERIMENTS

2.1 Data

We split the datasets into a train, a validation and a test dataset respectively based on a 8:1:1 data split. The data is then normalized and mean imputed. The datasets can be found online in the GitHub repository. We compute the normalized RMSE which corresponds to the RMSE on the normalized data. We predict categorical and nominal variables by rounding the predictions of the models. Please refer to Table 1. Note that our results deviate from those of other papers because we do a train test split for the data which results in a smaller dataset in the training stage.

Dataset	Sample size	# continuous features	# discrete features
Adult	32.561	3	8
Breast	569	30	0
Credit	30.000	14	9
Letter	20.000	0	16
Spam	4.601	57	0
Wine	6.497	11	1

Table 1: Statistics of the datasets

2.2 Implementation Details

We use PyTorch to implement the VAE models. The tensorflow code for GAIN can be found online (`https://github.com/jsyoon0823/GAIN`) as well as the PyTorch implementation of MIWAE (`https://github.com/pamattei/miwae`). The code for not-MIWAE was provided by the main author of the corresponding paper, Niels Bruun Ipsen. We use the MissForest implementation of sklearn (IterativeImputer with ExtraTreesRegressor as estimator) and the MICE implementation of sklearn (IterativeImputer with BayesianRidgeRegressor).

We train all benchmark models with default parameters as specified in our code. We run all deep learning methods for 1,000 epochs and train them using the Adam optimizer (Kingma and Ba, 2014). All subgraphs of the VAE-based approaches have one hidden layer with 128 nodes. Unless otherwise stated, we choose the dimension of the latent Gaussian variable $\mathbf{z}$ equal to 20 and the number of categories k of the latent categorical variable $\mathbf{r}$ as 10. Only MIWAE has two hidden layers each with 128 hidden units as specified by Mattei and Frellsen (2019). We use a rectifier as activation function between two layers. We chose a batch size of 200 for the datasets used in the main paper and a batch size of 512 for MNIST. The number of trees used for MissForest is set to 10. We use the self-masking approach of not-MIWAE where $P(\mathbf{m} \mid \mathbf{x})$ is learned by a logistic regression which is independent for each feature.

The benchmark VAE architectures have been modeled as follows:

- *VAE*: a basic VAE with one Gaussian latent variable learned on the observed data (and not on the missingness mask).
- *GMVAE*: a VAE with a Gaussian Mixture prior. That is a VAE with one categorical latent variable $\mathbf{r}$ taking values from $\{1, ..., k\}$ and one conditionally normal distributed latent variable $\mathbf{z} \mid \mathbf{r}$. It learns the generative model $P(\mathbf{x_{obs}}|\mathbf{m}, \mathbf{z}, \mathbf{r}) = P(\mathbf{x_{obs}}, \mathbf{m} \mid \mathbf{z}, \mathbf{r})P(\mathbf{z} \mid \mathbf{r})P(\mathbf{r})$. This model framework is similar to the HIVAE model presented in (Nazabal et al., 2020). Instead of sampling $\mathbf{r}$ from a Gumbel-Softmax distribution (Jang et al., 2017), we instead learn $P(\cdot \mid \mathbf{r} = r)$ for each $r \in \{1, ..., k\}$ and weight the resulting loss for each category with its posterior probability.
- *DLGM*: a deep latent Gaussian model which extends the GMVAE by a second latent Gaussian variable $\mathbf{w}$ with the same dimensionality as $\mathbf{x}$. It learns the generative distribution $P(\mathbf{x_{obs}}, \mathbf{m}, \mathbf{z}, \mathbf{w}, \mathbf{r}) = P(\mathbf{x_{obs}}, \mathbf{m} \mid \mathbf{z}, \mathbf{r})P(\mathbf{w} \mid \mathbf{z}, \mathbf{r})P(\mathbf{z} \mid \mathbf{r})P(\mathbf{r})$. The inference model is structured in the same way as the inference model for the PSMVAE where $\mathbf{x_{mis}}$ is replaced by $\mathbf{w}$.

We then impute $\mathbf{x_{mis}}$ by sampling from the conditional distribution of $\mathbf{x_{obs}}$.

In contrast to Nazabal et al. (2020), we do not use the Gumbel softmax distribution for sampling the categorical variable $\mathbf{r}$ (Jang et al., 2017), but instead integrate out $\mathbf{r}$ in the loss function. See Dilokthanakul et al. (2016) for a similar approach. In the implementation, we weight the log-likelihood of the observed data with the inverse of 1-*missingness rate*. Otherwise increasing the missingness will lead to a higher weight of the log-likelihood of the missingness mask compared to the log-likelihood of the observed data.

3 ADDITIONAL EXPERIMENTS

3.1 Multiple Imputation

We follow Mattei and Frellsen (2019) and assess the performance of our models in the multiple imputation setting by computing the test accuracy of the predicted target variable when a one layer classification network is trained on the dataset where each observation with missing entries was imputed 20 times. The test set is again made up of 20 multiple imputations.

3.2 Preliminary Results On Image Inpainting

We also show that our method can be used for image inpainting using the MNIST dataset (LeCun et al., 2010). We induced missingness completely random as before. We then trained our method using 500 epochs, a batch size of 512, $\pi = 0$ and only one importance sample. Please refer to Figure 1 for the results. Missing pixels were highlighted in red in the first row of each subfigure. The second row shows inpainted images using

Algorithm	20% MCAR	80% MCAR	20% MNAR	80% MNAR
PSMVAE(a)	.9482 ± .0211	.8466 ± .0627	.9489 ± .0116	.9501 ± .0101
PSMVAE(b)	.9429 ± .0180	.8791 ± .0511	.9577 ± .0063	.9496 ± .0157
MICE	.9545 ± .0141	.8105 ± .0337	.9578 ± .0096	.9578 ± .0096

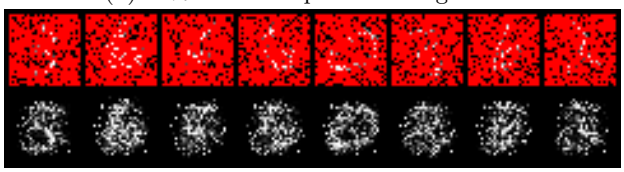
Table 2: Test accuracy of a one layer neural network for the prediction of breast cancer using the breast dataset which was imputed multiple times

the corresponding imputation algorithm. As we see, the PSMVAE(b) yields sharper images than GAIN does. Especially when the missingness is high, the imputations of our model suffer from considerably less noise than those of GAIN.

(a) 20% MCAR inpainted using PSMVAE(b)

(b) 20% MCAR inpainted using GAIN

(c) 80% MCAR inpainted using PSMVAE(b)

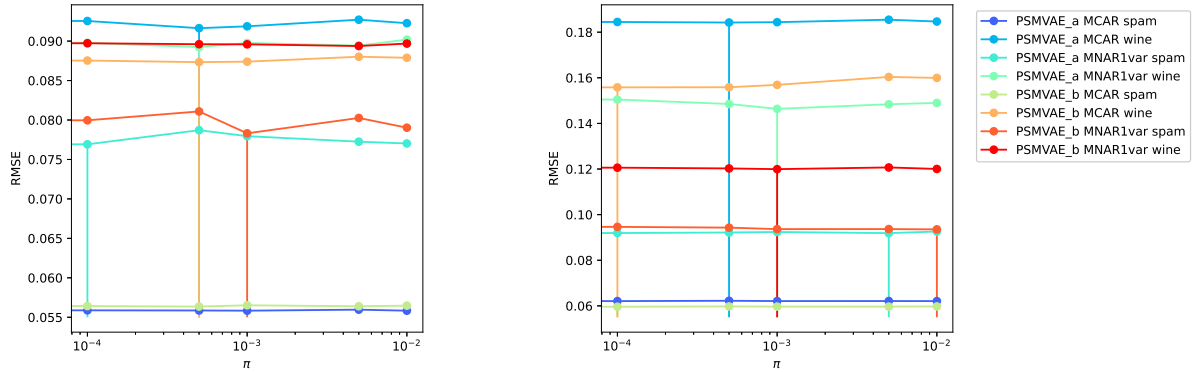


(d) 80% MCAR inpainted using GAIN

Figure 1: MNIST images with pixels MCAR at different rates inpainted with different imputation algorithms

3.3 Robustness Study

We now assess the robustness of our method. Please refer to Figure 2 for an illustration of the relationship between the RMSE and the hyperparameter π. The vertical lines highlight the minimum of the loss curves. We notice that the optimal value of π is usually on a similar scale as the optimal value of the weight decay parameter. Only on the MCAR spam dataset it is optimal to have a $\pi = 0$. Please note that the decay of the curves seems infinitesimal because we compare different methods and datasets within the same figure.



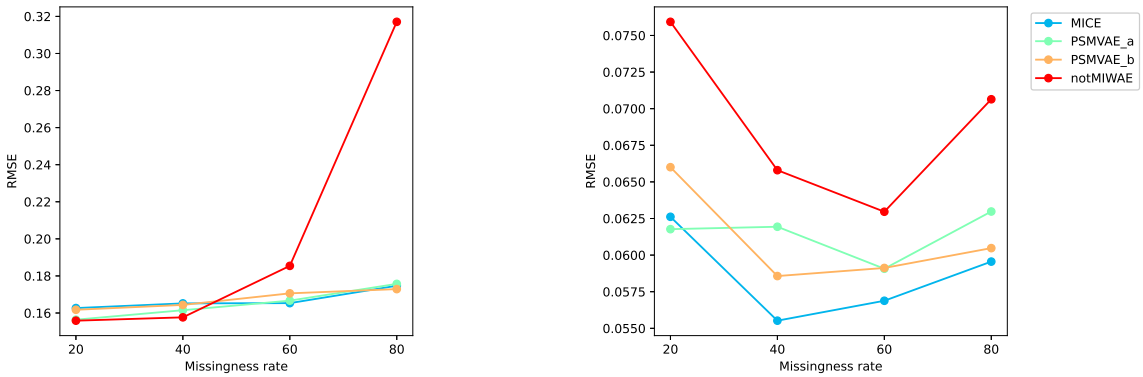
(a) 20% missingness rate

(b) 80% missingness rate

Figure 2: RMSE of imputation for different values of π for a missingness rate of 20% (left) and a missingness rate of 80% (right)

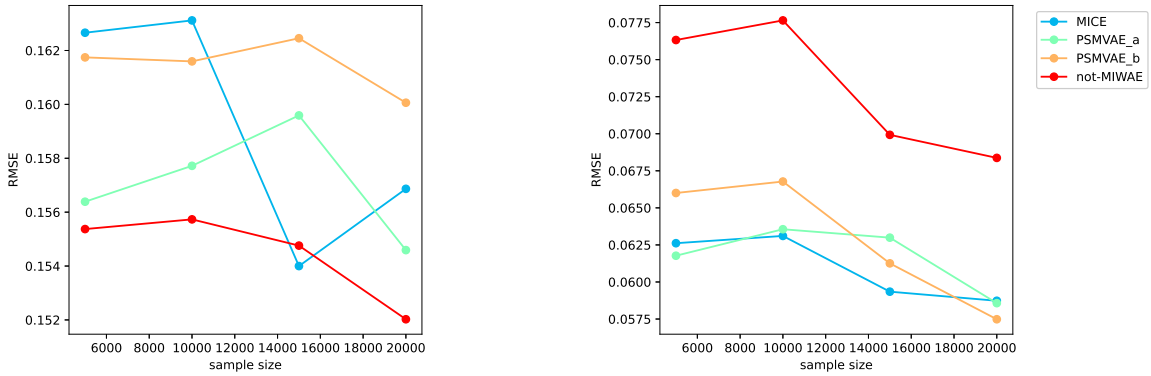
In the following we compare the robustness of our models and two benchmarks on the credit dataset. When assessing the relationship of the RMSE and the missingness rate (see Figure 3), we note that not-MIWAE is

not robust to increasing the missingness rate. This could stem from the fact that increasing the missingness increases the fraction of the optimization loss that comes from the likelihood of the missingness mask compared to the likelihood of the observed data. The larger the missingness, the better do our models compare relative to MICE.



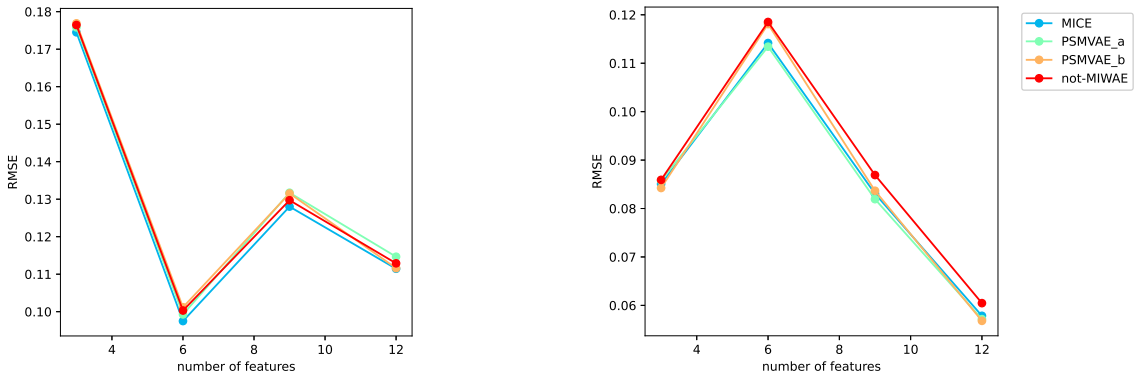
(a) MCAR (b) MNAR

Figure 3: RMSE of imputation for different missingness rates (in %) when data of the credit dataset are missing



(a) MCAR (b) MNAR

Figure 4: RMSE of imputation for different sample sizes when 20% of the data of the credit dataset are missing



(a) MCAR (b) MNAR

Figure 5: RMSE of imputation for different number of features when 20% of the data of the credit dataset are missing

When we change the sample size of the training dataset (see Figure 4), we note that the performance of the deep learning methods improves strictly and that the PSMVAE(b) performs better than MICE for large sample sizes.

Eventually, we compare the robustness of the RMSE when the number of features is changed (Figure 5). We see that the performance of all methods in general decreases when the number of features increases (except for one increase for all methods). The relative performance of our models hereby improves, the more features there are.

References

Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., and Shanahan, M. (2016). Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. *arXiv preprint arXiv:1611.02648*.

Jang, E., Gu, S., and Poole, B. (2017). Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database.

Mattei, P.-A. and Frellsen, J. (2019). MIWAE: deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, pages 4413–4423.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. (2020). Handling Incomplete Heterogeneous Data using VAEs. *Pattern Recognition*, page 107501.