Expanding Not-MIWAE
Experiments on the Not-MIWAE Model

Laura Choquet, Pierre Pauchet

December 17, 2024

1 Introduction

1.1 Presentation and contributions of the article

The goal of this report is to expand on a paper written by Niels Bruun Ipsen, 2021. The original paper introduces a new autoencoder model specialized in reconstructing data with a missing pattern dependent on the value of the missing data itself. This peculiar missing pattern is known as Not Missing At Random (MNAR), as opposed to Missing At Random (MAR) which may depends on the observed data but not the missing one. This pattern poses a particularly difficult challenge in data reconstruction. The authors exposed an innovative lower-bound objective for the loss function that accounts for the conditional missing values. As a result, their model is capable of inferring missing data and reconstructing partial databases affected by MNAR masks, achieving a level of performance in reconstruction not previously demonstrated for datasets afflicted by MNAR. Their model is called *not-missing-at-random importance-weighted autoencoder* (*not*-MIWAE) which is able to impute missing data with the application of Deep latent variable models (DVLMs, Kingma and Welling, 2013), in the case of MNAR problems. The model is highly inspired by the missing data importance-weighted autoencoder(MIWAE, Mattei and Frellsen, 2019), itself based on the the importance-weighted autoencoder(IWAE,Burda et al., 2016).The general graphical model for the not-MIWAE is shown in the Figure 1.As for notations, we will refer to $z \sim p(z)$ as the classical latent variable. The observation model is parametrized by θ, $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. The second part is the missing model, a stochastic mapping from the data to the missing mask s parameterized by Φ such that $s \sim p(s|x)$.

We shall go over the specific of the not-MIWAE model, focusing on how it improves from its previous version, the MIWAE. We present the results from the paper as well as a partial reproduction of results observed in the paper.
In a second part, we present our original work, that aims to upgrade the capabilities of the not-MIWAE by going beyond what was presented in the article. As suggested by the closing words of the paper, we implement a latent variable pointing directly on the missing values mask to try to better the reconstruction and imputation losses. We shall also attempt to apply supervised learning models on the reconstructed data. Finally, we implement a joint reconstruction-regression model, and compare its results to the isolated models.

1.1.1 Background

For the remainder of the study, the author introduce the data matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \mathcal{X}_n$, containing n i.i.d samples of $\mathbf{x} \in \mathcal{X}$, with $\mathcal{X}$ being a p-dimensional feature space. $\mathbf{x}_{i,j}$ denotes the j'th feature of the i'th sample. The missing data are denoted $\mathbf{x}_i^m$ and the observations $\mathbf{x}_i^o$. The authors introduce a mask matrix $\mathcal{S} \in \{0,1\}^{n,p}$, where $s_{ij} = 1$ if x_{ij} is observed and $s_{ij} = 1$ otherwise.
The goal is to construct a parametric model $p_{\theta,\phi}(x,s) = p_\theta(x)p_\phi(s|x)$ for the joint distribution of a sample x and its mask s. In the context of MNAR, $p_\phi(s|x)$ depends both on the observed and missing data. To compute the likelihood, they marginalize the joint

distribution over the missing data $\mathbf{x}^m$:

$$p_{\theta,\phi}(\mathbf{x}^o, s) = \int p_\theta(\mathbf{x}^o, \mathbf{x}^m) p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) d\mathbf{x}^m$$

2 Not-MIWAE

In order to generate accurate data and capture the missing model mechanism, one needs to maximize the joint log-likelihood $l(\theta, \phi)$ with respect to the parameters. We introduce the latent variable z, and assume that the observation model is fully factorized.

$$l(\theta, \phi) = \sum_{i=1}^{n} \log \int p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) p_\theta(\mathbf{x}^o|z) p_\theta(\mathbf{x}^m|z) p(z) d\mathbf{x}^m dz$$

To make the log-likelihood tractable, we sample the latent variable from the variational distribution $q_\gamma(\mathbf{z}|\mathbf{x}^o)$, γ is a learnable parameter. We can now express : $\log p_{\theta,\psi}(x^o, s)$

$$\log p_{\theta,\phi}(x^o, s) = \log \int \frac{p_\theta(\mathbf{x}^o|z)p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) q_\gamma(\mathbf{z}|\mathbf{x}^o) p_\theta(\mathbf{x}^m|z) d\mathbf{x}^m dz \tag{1}$$

$$= \mathbb{E}_{z\sim q_\gamma(\mathbf{z}|\mathbf{x}^o), x^m \sim p_\theta(\mathbf{x}^m|z)} \left[\frac{p_\theta(\mathbf{x}^o|z)p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) \right] \tag{2}$$

According to the idea presented in Burda et al., 2016 with IWAE, they approximate the expectation with a Monte Carlo estimation, that ensure the objective is a lower-bound of the likelihood. Finally, they obtain :

$$\mathcal{L}_K(\theta, \phi, \gamma) = \sum_{i=1}^{n} \mathbb{E} \left[\log \frac{1}{K} \sum_{k=1}^{K} \omega_{ki} \right], \text{where } \omega_{ki} = \frac{p_\theta(\mathbf{x}_i^o|\mathbf{z}_{ki})p(\mathbf{z}_{ki})}{q_\gamma(\mathbf{z}_{ki}|\mathbf{x}_i^o)} p_\phi(s_i|\mathbf{x}_i^o, \mathbf{x}_{ki}^m) \tag{3}$$

$\mathbf{z}_{ki}$ and $\mathbf{x}_{ki}^m$ are respectively sampled over $q_\gamma(\mathbf{z}|\mathbf{x}_i^o)$ and $p_\theta(\mathbf{x}^m|z)$ The properties of the not-MIWAE objective are essentially the same as those of the MIWAE (Mattei and Frellsen, 2019). We could have discussed the reparametrization trick presented in Tucker et al., 2018 to obtain a lower-variance estimator of the gradient of the IWAE bound. Once the model has been trained, it can be used to impute missing data $\hat{x}^m$. The metric for this performance is the squared error and the optimal imputation is the conditional mean $\mathbb{E}_{x^m}[L(x^m, \hat{x}^m)|x^o, s$, that can also be computed with the self-normalized importance sampling. (See Appendix)

3 Experiments and results

3.1 Replication of the results of the paper : UCI database

We first aim to reproduce the results of the paper using the code found on an online *repository* linked in the article. The code makes use of `Tensorflow` 1.X, which is outdated on recent versions of Python. The hyperparameters described in the paper are globally consistent with those given in the paper, barring some rare exceptions. The architecture of the model also correlates to what the authors described.

Model	Paper Results		Reproduction	
-	Red Wine	White Wine	Red Wine	White Wine
RMSE MIWAE	1.62	1.55	1.63	1.54
RMSE not-MIWAE	1.07	1.04	1.06	1.03
low-rank joint model	1.42	1.39	-	-
RMSE MissForest	1.64	1.39	1.60	1.40
RMSE MICE	1.68	1.41	1.68	1.41
RMSE Mean	1.83	1.74	1.83	1.74

Table 1: *Reproduction studies for the single imputation RMSE for both red wine and white wine UCL dataset affected by MNAR.*

We ran single imputation RMSE studies after training the model on our machines for both the red wine and white wine dataset. Results are given in Table 1.
As shown in the table, our reproduction is very close to the results presented in the article. However, we were not able to replicate the low-rank model inspired from (Sportisse et al., 2020), described as useful in the paper but not found in the code. This is a dent in the reproducibility of an otherwise accessible and replicable article.

3.2 Latent variable on the mask

3.2.1 Introduction of additional latent variable on the mask

As discussed at the end of the article, we introduce an additional latent variable pointing directly influencing the mask $\mathbf{z}^{mask}$ as shown in the Figure 2, such that $\mathbf{s} \sim p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m, \mathbf{z}^{mask})$. In order to keep the likelihood tractable we introduce an additional variational distribution $q_\beta(\mathbf{z}|\mathbf{x}^o)$, parameterized by a learnable parameter β. Then, $\log p_{\theta,\phi}(x^o, s)$ from Equation (2) becomes:

$$= \log \int \frac{p_\theta(\mathbf{x}^o|z)p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} q_\gamma(\mathbf{z}|\mathbf{x}^o) \frac{p_\phi(s|\mathbf{x}^o, \mathbf{x}^m, \mathbf{z}^{mask})p(\mathbf{z}^{mask})}{q_\beta(\mathbf{z}^{mask}|\mathbf{x}^o)} q_\beta(\mathbf{z}^{mask}|\mathbf{x}^o) p_\theta(\mathbf{x}^m|z) d\mathbf{x}^m dz^{mask} dz$$

$$= \mathbb{E}_{z\sim q_\gamma(\mathbf{z}|\mathbf{x}^o), x^m \sim p_\theta(\mathbf{x}^m|z), z^{mask}\sim q_\beta(\mathbf{z}^{mask}|\mathbf{x}^o)} \left[\frac{p_\theta(\mathbf{x}^o|z)p(\mathbf{z})}{q_\gamma(\mathbf{z}^{mask}|\mathbf{x}^o)} \frac{p_\phi(s|\mathbf{x}^o, \mathbf{x}^m, \mathbf{z}^{mask})p(\mathbf{z}^{mask})}{q_\beta(\mathbf{z}^{mask}|\mathbf{x}^o)} \right]$$

And the Monte Carlo sampling (3), leads to :

$$\mathcal{L}_K(\theta, \phi, \gamma, \beta) = \sum_{i=1}^{n} \mathbb{E}\left[\log \frac{1}{K} \sum_{k=1}^{K} \omega_{ki} \right], \text{where } \omega_{ki} = \frac{p_\theta(\mathbf{x}_i^o|\mathbf{z}_{ki})p(\mathbf{z}_{ki})}{q_\gamma(\mathbf{z}_{ki}|\mathbf{x}_i^o)} \frac{p_\phi(s_i|\mathbf{x}_i^o, \mathbf{x}_{ki}^m, z_{ki}^{mask})p(z_{ki}^{mask})}{q_\beta(\mathbf{z}_{ki}^{mask}|\mathbf{x}_i^o)}$$

The imputation of the missing data is modified with the additionally latent variable $\mathbf{z}^{mask}$, the original imputation process is described in the appendix of the paper Niels Bruun Ipsen, 2021. The latent variable z_{mask} is introduce in the same way than above.

3.2.2 Results

Results are presented in Table 5 in the Appendix. We denote models with the suffix mask when the latent variable $z$mask is introduced into the model. Our model achieves better

RMSE performance compared to MIWAE, non-MIWAE, and other imputation models across various paradigms: Linear, Self-Masking, and Self-Masking with known masks. Moreover, we observe that our non-MIWAE models with masks often outperform their non-masked counterparts or, at worst, yield very similar results. These trends are consistent across both the White Wine and Red Wine UCI datasets (Dua, 2017). Such promising results highlight the potential of our approach and encourage further exploration on other datasets.

3.3 not-MIWAE used with supervised regression

3.3.1 Introduction of a supervised learning model

Following the recommandations at the end of the article, we now focus on the performance of the notMIWAE model when confronted with a different and complementary model : a MLP regressor with non-linearity. The goal of this part is to assess the following : we hypothesize that a joint regressor-notMIWAE model would perform better than a simple notMIWAE. Indeed, having access to the values of "labels" associated with samples would give the autoencoder more information when reconstructing, as well as more degrees of liberty when training. The chosen regressor model is kept simple on purpose (See Appendix)(**??**) The modified loss function was inspired by Ipsen et al., 2022. With y as the labels, x^o as the observed data, and x^m as the missing data, the conditional probability $p_\Psi(y|x^o)$ of finding the correct label through the regression model is defined as:

$$p_\Psi(y|x^o) = Z \int p_\Psi(y|x^o, x^m) p_\theta(x^m|x^o)\, dx^m \tag{4}$$

where Z is a normalization constant. As for the not-MIWAE, we assume that the observation model is factorized. We can then express the log-likelihood :

$$l(\theta, \phi, \psi) = \sum_{i=1}^{n} \log \int p_\Psi(y|\mathbf{x}^o, \mathbf{x}^m) p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) p_\theta(\mathbf{x}^o|z) p_\theta(\mathbf{x}^m|z) p(z) d\mathbf{x}^m dz$$

We still follow the Importance Sampling paradigm to make the log-likelihood tractable, by sampling the latent variable from the variational distribution. We can now express $\log p_{\theta,\phi,\psi}(x^o, s, y)$:

$$\log p_{\theta,\phi,\psi}(x^o, s, y) = \log \int \frac{p_\theta(\mathbf{x}^o|z) p_\Psi(y|\mathbf{x}^o, \mathbf{x}^m) p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) q_\gamma(\mathbf{z}|\mathbf{x}^o) p_\theta(\mathbf{x}^m|z)\, d\mathbf{x}^m\, dz \tag{5}$$

$$= \mathbb{E}_{z\sim q_\gamma(\mathbf{z}|\mathbf{x}^o),\, x^m \sim p_\theta(\mathbf{x}^m|z)} \left[\frac{p_\theta(\mathbf{x}^o|z) p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} p_\Psi(y|\mathbf{x}^o, \mathbf{x}^m) p_\phi(s|\mathbf{x}^o, \mathbf{x}^m) \right]. \tag{6}$$

The Monte-Carlo estimate, replacing the expectation inside the logarithm Burda et al., 2016 lead to the following objective function :

$$\mathcal{L}_K(\theta, \phi, \gamma, \beta) = \sum_{i=1}^{n} \mathbb{E} \left[\log \left(\frac{1}{K} \sum_{k=1}^{K} \omega_{ki} \right) \right] \tag{7}$$

$$\text{with } \omega_{ki} \frac{p_\phi(s_i \mid \mathbf{x}_i^o, \mathbf{x}_{ki}^m) p_\theta(\mathbf{x}_i^o \mid \mathbf{z}_{ki}) p(\mathbf{z}_{ki}) p_\Psi(y_i \mid \mathbf{x}_i^o, \mathbf{x}_{ki}^m)}{q_\gamma(\mathbf{z}_{ki} \mid \mathbf{x}_i^o)}. \tag{8}$$

The contribution of a single fully observed point is

$$\log p_\Psi(y|x) + \mathbb{E}_{(\mathbf{z}_k)}\left[\log\left(\frac{1}{K}\sum_{i=1}^{K}\frac{p_\phi(s|\mathbf{x}^o,\mathbf{x}_k^m)p_\theta(\mathbf{x}^o|\mathbf{z}_k)p(\mathbf{z}_k)}{q_\gamma(\mathbf{z}_k|\mathbf{x}^o)}\right)\right] \tag{9}$$

which is exactly the sum of the likelihood objective of the regressor and the IWAE bound found in Niels Bruun Ipsen, 2021. Thus, we can train both models on a joint loss.

3.3.2 Results

First, we train a simple MLP regressor on the red wine dataset reconstructed by not-MIWAE. The structure of the regressor model is detailed on Figure **??**. We want to extablish the loss of the model when the weights of the not-MIWAE are frozen *ie* when the regressor learn on data reconstructed by simple imputation, and does not affect the not-MIWAE. We then want to compare this baseline loss with the loss of our *"Reg-not-MIWAE"*. This refers to the "joint model", a not-MIWAE combined with the MLP, trained end-to-end on a common objective. As shown in Table 2, the regressor part performs better when it is not joint with the not-MIWAE. This can be explained by the compromises that the joint loss shown in (9) entails : the weights are shared between the two models and thus the regressor is restrained in its ability to freely learn.

Test	Solo MLP	MLP in Reg-not-MIWAE
Test loss (MSE)	0.3059	0.3452
Train loss (MSE)	0.5905	0.6405

Table 2: *Train and test losses of solo MLP regressor vs the MLP part of Reg-not-MIWAE*

We now focus on the Auto-encoder part, and ask ourselves if it is capable of better reconstruction when exposed to labels of the data. As shown in 3, the two reconstruction models (not-MIWAE and reg-not-MIWAE) have similar performances in single imputation RMSE for unseen data subjected to the same MNAR pattern. However, our Reg-not-MIWAE performs significatively worse on the Train set. The model also seems to perform worse on training data, which is uncharacteristic.
For this comparison, we used the same hyperparameters in Reg-not-MIWAE as in not-MIWAE, when applicable. All parameters can be found in Appendix.

It appears that our Reg-not-MIWAE performs slightly worse than disjoint models both in reconstruction and regression. We can make the hypothesis that this is due to the simplistic architecture of the MLP we employed. Indeed, with a more flexible model, we believe that the regressor could fit the heavily non-linear nature of the target distribution better. This is corroborated by the shape of the regression outputs pictured in Figure 4 in Appendix.

Test	not-MIWAE	Reg-not-MIWAE
Single Imputation RMSE (Train set)	1.061	1.1606
Single Imputation RMSE (Test set)	0.8655	0.8884
RMSE - MICE	1.6805	1.6825
RMSE - Mean	1.8382	1.8414
RMSE - Random Forest	1.6095	1.6157

Table 3: *Performance of Reg-not-MIWAE vs not-MIWAE in RMSE*

4 Conclusion and discussion

In this article, we analyzed the not-MIWAE paper by Niels Bruun Ipsen, 2021, providing both an explanation and a successful reproduction of their results. The paper addresses a critical issue in machine learning—handling missing data under Missing Not At Random (MNAR) patterns. This problem is tackled effectively by the proposed model, using clever lower-bound of the objective function with importance sampling, as we shown in **2.** We also demonstrated the reproducibility of the original results, confirming the authors' method.

Building on this foundation, we extended the model by implementing a latent value on the mask operator. With this modified masked not-MIWAE, we obtain results that outperform the baseline not-MIWAE, highlighting the potential for further improvements in the treatment of missing data. This modification, while simple, yields promising results that suggest the potential for better handling of incomplete datasets.
In a second part, we attempted to combine not-MIWAE and supervised learning models, namely an MLP regressor. By employing a joint loss, we hoped that the autoencoder could learn from the predictions of the label to improve its grasp on the missing data. It seems, however, that the joint model performs worse than two independent model, both in regression and in imputation. We can understand this difference by reckoning that the objective of the two models contradict each other.

Several areas remain to be explored for further improvement. For the not-MIWAE-mask approach, testing the model on different datasets, such as those mentioned in Niels Bruun Ipsen, 2021, including MNIST, with varying patterns of missing data could provide deeper insights into its robustness and adaptability. Additionally, for the reg-MIWAE approach, incorporating more complex regressors to better capture non-linearities in the data could potentially improve performance, allowing the joint model to outperform its two-part counterpart.

As indicated by recent works such as Ghalebikesabi et al. (2021), there are many openings for future research in this domain. The continuous advancement of models like not-MIWAE could lead to significant improvements in the handling of missing data across a wide range of applications.

References

Burda, Y., Grosse, R., & Salakhutdinov, R. (2016). Importance weighted autoencoders. https://arxiv.org/abs/1509.00519

Dua, D. (2017). *Home - UCI machine learning repository*. Retrieved December 17, 2024, from https://archive.ics.uci.edu/

Ipsen, N. B., Mattei, P.-A., & Frellsen, J. (2022). How to deal with missing data in deep supervised learning ?

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *CoRR*, *abs/1312.6114*. https://api.semanticscholar.org/CorpusID:216078090

Mattei, P.-A., & Frellsen, J. (2019). MIWAE: Deep generative modelling and imputation of incomplete data sets (K. Chaudhuri & R. Salakhutdinov, Eds.). *?*, *97*, 4413–4423. https://proceedings.mlr.press/v97/mattei19a.html

Niels Bruun Ipsen, J. F., Pierre-Alexandre Mattei. (2021). Not-miwae: Deep generative modelling with missing not at random data. *arXiv*. https://doi.org/10.48550/arXiv.2006.12871

Sportisse, A., Boyer, C., & Josse, J. (2020, January 29). Imputation and low-rank estimation with missing not at random data. https://doi.org/10.48550/arXiv.1812.11409

Tucker, G., Lawson, D., Gu, S., & Maddison, C. J. (2018). Doubly reparameterized gradient estimators for monte carlo objectives. https://arxiv.org/abs/1810.04152

Appendix

4.1 Background and notations

4.2 Code

All code for this article was turned in as a .zip file to the professoral team. It is also available online at the following online repository`https://github.com/pierre-pauchet/MVA/PGM`.

4.3 Parameters

The parameters can be found in Table **??**. They can also be found in the code.

Parameter	**MIVAE**	**Regressor**
`N, D`	Data shape	Data Shape
`n_latent`	$D-1$	-
`n_hidden`	128	128
`n_samples`	20	-
`max_iter`	100,000	300
`batch_size`	16	-
`n_classes`	5	-
`n_outputs`	-	1
`learning_rate`	-	2×10^{-4}

Table 4: *Model Parameters for MIWAE and Regressor*

4.4 Graphical models for modified MIWAE with latent variable on the mask

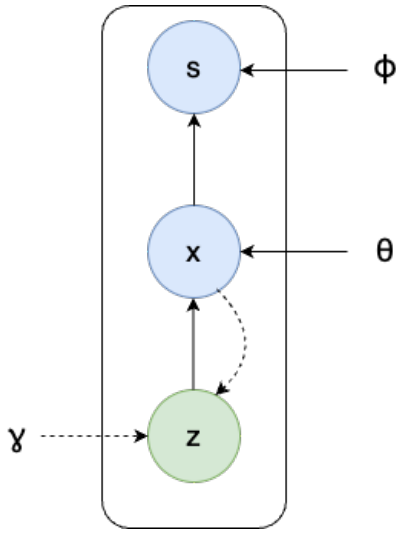

Figure 1: *Graphical model of the not-MIWAE.*

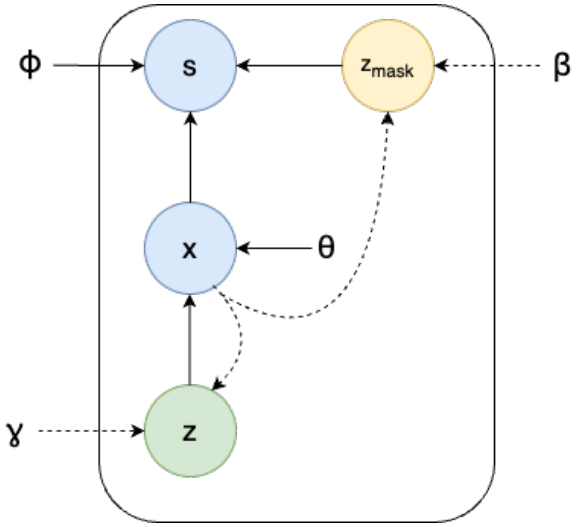

Figure 2: *Graphical model of the modified not-MIWAE.*

4.5 Regressor Architechture

As explained previously, the regressor architecture was kept simple on purpose. To go further, one could try to implement a more robust and adaptable structure such as a CNN.

4.6 Detailed results for for modified MIWAE with latent variable on the mask

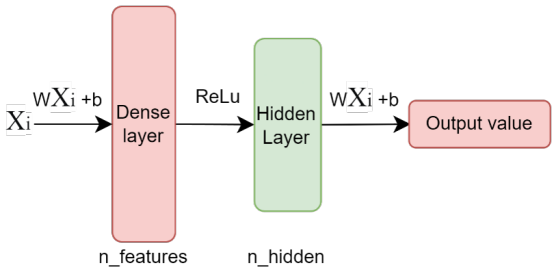

Figure 3: *Structure of the regressor model. We set* $n_{hidden} = 128$*, while* $n_{features}$ *depends on the dataset used.*

Table 5: *RMSE comparison of models on white wine and red wine datasets. The best results, as well as the model with the latent variable z_{mask}, are highlighted in* ***bold****. The second-best performances are shown in blue.*

Model	White Wine	Red Wine
Linear		
MIWAE	1.54273	1.66189
not_MIWAE	1.37157	1.32128
not_MIWAE_mask	**1.35925**	**1.24951**
mean	1.73928	1.83821
MICE	1.41151	1.68054
Self-masking		
not_MIWAE_PPCA	1.002	1.143
not_MIWAE_PPCA_Mask	**0.996**	1.1226
MIWAE	1.54181	1.63876
not_MIWAE	1.04110	1.13257
not_MIWAE_mask	1.05727	**1.08386**
mean	1.73928	1.83821
MICE	1.41151	1.68054
Self-masking known		
not_MIWAE_PPCA	0.99827	1.125
not_MIWAE_PPCA_mask	**0.97321**	1.124
MIWAE	1.55240	1.63987
not_MIWAE	1.04805	1.07733
not_MIWAE_mask	1.07587	**1.07332**
mean	1.73928	1.83821
MICE	1.41151	1.68054

4.7 Shape of regression outputs

The non-linearity of the target grade is quite apparent in this figure. It is possible that a MLP with more non-linearity would be able to capture the discontinous nature of the objective better.

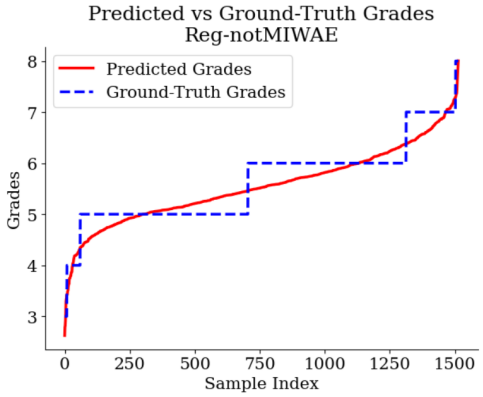

Figure 4: *Performance of Reg-not-MIWAE on the red wine training dataset affected by MNAR pattern.*