Introduction to Probabilistic Graphical Models and Deep Generative Models (2024), , 1–11

MATHÉMATIQUES VISION APPRENTISSAGE

REPORT

Deep Generative Modelling with Missing not at Random Data (not-MIWAE)

Simon Blotas and Charles Dezons

Project context

This report is the result of a project for the MVA course Introduction to Probabilistic Graphical Models and Deep Generative Models. The focus is on the main paper by Ipsen, Mattei, and Frellsen 2021, titled not-MIWAE: Deep Generative Modelling with Missing Not at Random Data. We present the core ideas of this method, implement the approach, and conduct experiments to evaluate its performance.

1. Introduction

There are many approaches to infer missing data, or to bypass missing data. The performance of those models depends on the assumption made on the mecanism behind the missing data. Usually we assume that the data is Missing At Random (MAR). This basically means that the missing pattern does not depend on the missing values. This allows us to compute an easier likelihood in inference based models, because we can simply marginalize over the missing values. In this paper, Niels Bruun Ipsen, Pierre-Alexandre Mattei and Jes Frellsen explore the cases where the data is Missing Not At Random (MNAR), which means that the missing data mechanism is dependent on the missing data themselves.

2. Contributions and Background

The whole approach is based on two important contributions, Deep Latent Variable Models and the not-missing-at-random importance-weighted autoencoder (not-MIWAE).

- **DLVMs**: Deep Latent Variable Models (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014) is a common framework used for inference and imputation in missing data problems, that shows impressive empirical results in the MAR and MCAR case, in particular for high-dimensional data. This approach is important to understand since the code that we will provide relies on this architecture.
- **not-MIWAE**: The Importance Weighted AutoEncoder (IWAE) was first introduced by Burda, Grosse, and Salakhutdinov 2016. Here, the not-MIWAE, allows for application of DLVMs to missing data problems where the missing mechanism is MNAR, and is inspired by the missing data importance-weighted autoencoder (Mattei and Frellsen 2019), a framework to train DLVMs in MAR scenarios.

3. Notations and Hypotheses

We consider a dataset $\mathbf{X} = (x_1, \dots, x_n) \in \mathcal{X}^n$, where each x_i is a sample with features x_i^o (observed) and x_i^m (missing). Missingness is described by a mask $s \in \{0, 1\}^p$, where $s_j = 1$ if feature j is observed, $s_j = 0$ if feature j is missing. The joint distribution $p_{\theta,\phi}(x, s)$ can be factorized as $p_{\theta,\phi}(x, s) = p_\theta(x)p_\phi(s|x)$, with three possible assumptions:

- **MCAR**: $p_\phi(s|x) = p_\phi(s)$; The data is missing completely at random and the mask doesn't depend on the missing nor observed data.
- **MAR**: $p_\phi(s|x) = p_\phi(s|x^o)$; The data is missing at random and the mask only depends on the observed data
- **MNAR**: $p_\phi(s|x)$ may depend on both x^o and x^m.

For θ and ϕ, likelihood estimation requires integrating out the missing data, except in the MNAR case, where both observed and missing data affect the likelihood.

4. Difference between the VAE and the IWAE and inference in the MNAR case

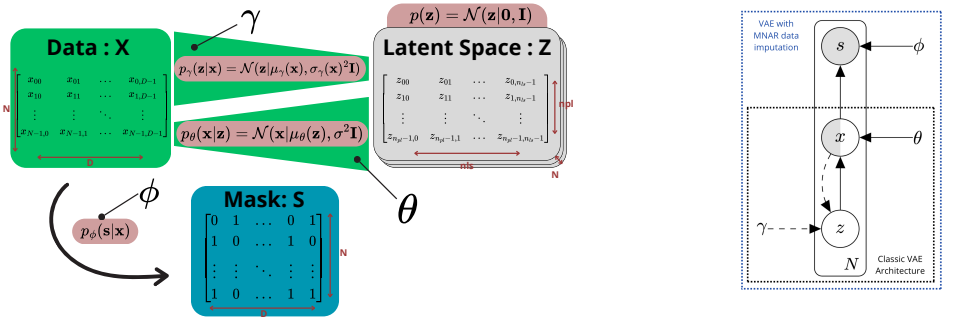


Figure 1. Schematic VAE with MNAR missing values on the left and Probabilistic Graphical Model of the not-MIWAE/VAE on the right

Let $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ denote a dataset of N i.i.d. samples where each observed datapoint $\mathbf{x}^{(i)}$ is obtained by first sampling a latent vector $\mathbf{z}$ from the prior $p_\theta(\mathbf{z})$ and then sampling $\mathbf{x}^{(i)}$ itself from the scene model $p_\theta(\mathbf{x}|\mathbf{z})$. Now we introduce an auxiliary distribution $q_\gamma(\mathbf{z}|\mathbf{x})$ (with its own parameters) as an approximation to the true, but unknown posterior $p_\theta(\mathbf{z}|\mathbf{x})$. We can show (please refer to Appendix 1.1 as those equations are important to understand why the code works) that the estimator to maximize in the IWAE framework is a lower-bound on the plain data log-likelihood:

$$\mathcal{L}_k^{\text{IWAE}}\left(\theta, \gamma; \mathbf{x}^{(i)}\right) = \mathbb{E}\left[\log \frac{1}{k}\sum_{l=1}^{k} w^{(i,l)}\right] \leq \log \mathbb{E}\left[\frac{1}{k}\sum_{l=1}^{k} w^{(i,l)}\right] = \log p_\theta(\mathbf{x}^{(i)}). \tag{1}$$

Where:

$$\text{(unnormalized) importance weights:} \quad w^{(i,l)} = \frac{p_\theta(\mathbf{x}^{(i)}, \mathbf{z}^{(l)})}{q_\gamma(\mathbf{z}^{(l)}|\mathbf{x}^{(i)})}. \tag{2}$$

MNAR case

In the MNAR case, we optimize both data-generation and missingness mechanisms. The objective is the joint log-likelihood:

$$\ell(\theta, \phi) = \sum_{i=1}^{n} \log p_{\theta,\phi}(x_i, s_i), \tag{3}$$

but direct maximum likelihood estimation is intractable due to integrals over missing and latent variables. To address this, it is possible to do exactly the same thing as for VAE and IWAE: we

introduce a variational distribution $q_\gamma(z|x^o)$ to estimate a tractable lower bound using importance sampling. Let's rewrite the contribution of data points

$$\log p_{\theta,\phi}(x^o, s) = \log \int p_\phi(s|x^o, x^m)\, p_\theta(x^o|z)\, p_\theta(x^m|z)\, p(z)\, dz\, dx^m \tag{4}$$

The contribution of a single observation becomes:

$$\log p_{\theta,\phi}(x^o, s) = \log \mathbb{E}_{z\sim q_\gamma(z|x^o), x^m\sim p_\theta(x^m|z)} \left[\frac{p_\phi(s|x^o, x^m) p_\theta(x^o|z) p(z)}{q_\gamma(z|x^o)} \right] \tag{5}$$

The objective function is then estimated by Monte Carlo sampling:

$$\mathcal{L}_K(\theta, \phi, \gamma) = \sum_{i=1}^{n} \mathbb{E} \left[\log \frac{1}{K} \sum_{k=1}^{K} w_{ki} \right], \quad w_{ki} = \frac{p_\phi(s_i|x_i^o, x_{ki}^m) p_\theta(x_i^o|z_{ki}) p(z_{ki})}{q_\gamma(z_{ki}|x_i^o)}. \tag{6}$$

where w_{ik} represents the importance weights for the samples. We show in Appendix 1.2 how to backpropagate through samples from $q_\gamma(z|x_i^\circ)$, $p_\theta(x^m|z)$ using the reparameterization trick.

4.1 Imputation

Once the model has been trained, it is possible to use it to impute the missing values. We are going to use the squared error as a loss function $L(x^m, \hat{x}^m)$. Then optimal imputations $\hat{x}^m$ minimize

$$\mathbb{E}_{x^m}\left[L(x^m, \hat{x}^m) \middle| x^o, s \right].$$

As discussed in Appendix 1.3, we can show that in this case,

$$\hat{x}^m = \mathbb{E}[x^m|x^o, s] \approx \sum_{k=1}^{K} \alpha_k \mathbb{E}[x^m|x^o, s], \quad \text{with} \quad \alpha_k = \frac{w_k}{w_1 + \cdots + w_K} \tag{7}$$

Where by luck, the weights $w_1, \ldots, w_K$ are identical to the ones used for training (see equation (19))

$$\forall k \leq K, \quad w_k = \frac{p_\phi(s|x^o, x_k^m) p_\theta(x^o|z_k) p(z_k)}{q_\gamma(z_k|x^o)} \tag{8}$$

5. Using Prior Information via the Missing Data Model

The missing data mechanism may be known or learned. More knowledge about the missing process helps improve model design, as shown by Molenberghs et al. 2008. For MNAR, incorporating prior information is critical to correctly model missingness.

The missing data model can be framed as a classification problem, with a Bernoulli distribution for mask prediction:

$$p_\phi(s|x^o, x^m) = \prod_{j=1}^{p} \pi_{\phi,j}(x)^{s_j} (1 - \pi_{\phi,j}(x))^{1-s_j}, \tag{9}$$

where $\pi_{\phi,j}(x)$ is the probability of observing feature j. The mapping $\pi_{\phi,j}(x)$ may depend on the value of x_j, with a possible sigmoid form.

6. Experiments with synthetic data

In this section, we evaluate the performance of different imputation methods on synthetic datasets.

6.1 Gaussian distribution

Data generation and masking mechanism

We generate a 2D Gaussian distribution centered at $(0, 0)$, with covariance matrix $\mathbf{Cov}$:

$$\mathcal{N}(\mathbf{0}, \mathbf{Cov}) \text{ where } \mathbf{Cov} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

The missing values mask is defined as follows: for each sample, the x_1 component is masked if it exceeds the mean of all x_1 values in the dataset, which in this case is 0.

6.2 Ring-shaped distribution

Data generation and masking mechanism

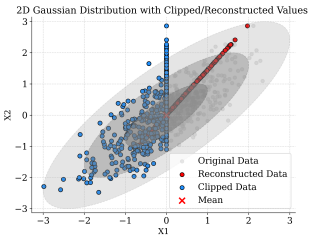
We generate a 2D ring distribution by sampling angles uniformly in $[0, 2\pi]$ and radius from a Gaussian distribution with mean $\mu = 4.5$ and variance σ^2, $\sigma = 0.5$. The Cartesian coordinates are derived as: $x_1 = r\cos(\theta), \quad x_2 = r\sin(\theta)$.

The missing values mask is defined as follows: for each sample, the x_1 component is masked if the sample lies within a specific angular sector.

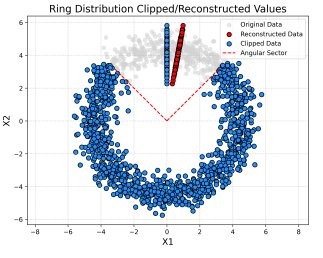
6.3 Comparison of results

Reconstruction quality

Figure 2 shows side-by-side examples of the original data, clipped data, and reconstructed data using notMIWAE for both the Gaussian and ring datasets.



(a) Reconstructed data for Gaussian example using notMIWAE.



(b) Reconstructed data for ring example using notMIWAE.

Figure 2. Comparison of reconstructed data for Gaussian (left) and ring (right) examples using notMIWAE.

For the Gaussian reconstruction shown in Figure 2a, we observe that the model reconstructs the data well. When the value of x_1 is missing, the model uses the known value of x_2 to project the missing point onto a line that approximately aligns with the mean of the Gaussian distribution corresponding to x_2. This behavior demonstrates that the model effectively captures the underlying distribution and utilizes the available information for accurate imputation.

In contrast, the reconstruction of the ring dataset in Figure 2b is less successful. When x_1 values are missing, the model shifts the reconstruction of these points onto a line that is offset to the right, rather than following the expected ring shape. Ideally, the model should infer that the missing values adhere to the circular structure of the data. While the reconstructed values are close to the real values, they do not align with the expected geometric pattern. This highlights a limitation of the model when applied to non-linear and more complex data distributions.

RMSE performance

Table 1 summarizes the RMSE computed for the six imputation methods across both datasets.

Table 1. RMSE comparison for Gaussian and ring datasets across different imputation methods.

Dataset	Method	notMIWAE	MIWAE	KNN	MICE	Random Forest
Gaussian	RMSE	**1.152**	1.153	1.320	1.155	1.328
Ring	RMSE	**1.971**	1.977	3.046	1.975	4.034

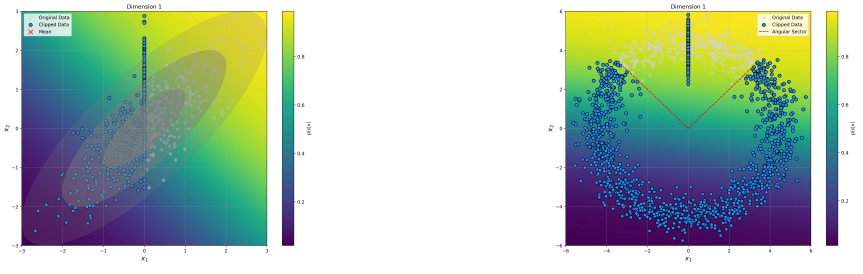
Table 1 shows that **notMIWAE** consistently achieves the lowest RMSE across both datasets, slightly outperforming **MIWAE** and **MICE**, which also perform well and remain competitive. These methods demonstrate robustness to different data structures, with minimal differences in RMSE values.

In contrast, heuristic methods like **KNN** and **Random Forest** exhibit significantly higher RMSEs, particularly on the more complex ring dataset. This highlights their limitations in capturing intricate patterns and interdependencies, making them less suitable for such tasks.

Overall, probabilistic approaches like notMIWAE and MIWAE emerge as the most effective methods, with MICE providing a strong non-probabilistic alternative. Further analysis on larger datasets or varying missingness patterns could deepen insights into their performance.

Comparison of inferred distributions

Figure 3 illustrates the inferred conditional distributions $p(x_1 \text{ missing} \mid x)$ for both datasets, projected in 2D.



(a) Inferred $p(s|x)$ for Gaussian dataset. **(b)** Inferred $p(s|x)$ for ring dataset.

Figure 3. Comparison of inferred $p(s|x)$ for Gaussian (left) and ring (right) datasets.

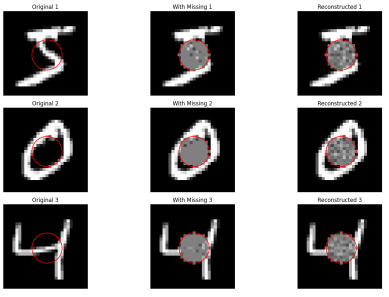
For the Gaussian dataset, shown in Figure 3a, the model captures the dynamics of the missing data process effectively. It infers that as x_1 approaches the mean, the probability of it being missing increases. Similarly, as x_2 increases, the likelihood of x_1 being missing also rises, which aligns with the expected behavior. The inferred $p(s|x)$ profile corresponds to the application of a sigmoid function on the learned weights and biases of a linear layer. In 2D, this results in a distribution that separates the plane into two regions using a line, here approximated by $f(x) = -x$. This behavior reflects a reasonable approximation of the missing data dynamics for the Gaussian case.

In contrast, for the ring dataset shown in Figure 3b, the separating line is slightly bellow the angular sector where missing x_1 values begin to appear. This placement is sensible, as a linear boundary is the best approximation available for partitioning the plane in this scenario. However, the model does not fully capture the specific angular sector of the ring where values are missing. This limitation is likely due to the simplicity of using a single linear layer to estimate $p(s|x)$.

We explored replacing the linear layer with a small neural network (adding one additional layer with a ReLU activation) to enhance the model's expressiveness. However, this modification did

not improve the prediction of $p(s|x)$. Moreover, it led to a decrease in the model's reconstruction performance, suggesting that increased complexity in this layer may compromise other aspects of the model's learning.

7. Experimentation with a Real Dataset



(a) Reconstructed MNIST examples with imputed values.

Method	RMSE
notMIWAE	0.9301
MIWAE	0.9285
KNN	0.9353

(b) RMSE comparison across imputation methods for MNIST dataset.

Figure 4. Visual and quantitative results of imputation methods. (a) Example of reconstructed MNIST data with missing values imputed. (b) RMSE comparison across different imputation methods.

This experiment is based on the MNIST dataset, a widely used benchmark in image processing tasks.

To simulate missing data, a circular mask is applied to each image. The mask is characterized by a fixed radius and center, with pixels inside the circle treated as missing if their absolute values exceed a specified threshold. This process mimics real-world scenarios where certain regions of data are corrupted or incomplete, such as in cases of sensor defects.

8. Discussion

Upon visual inspection on Figure 4a, the reconstructed images from the notMIWAE model do not effectively recover the digits within the occluded regions. Instead of reconstructing meaningful structures, the model tends to produce a blur of pixels within the masked areas, failing to restore the digit's original shape.

From the RMSE analysis Table 4, the performance of notMIWAE and MIWAE are comparable to simpler methods like KNN, which merely imputes missing values based on the mean of neighboring pixels. This suggests that notMIWAE and MIWAE are not effectively exploiting its probabilistic framework to provide superior reconstructions.

Several factors could explain these limitations:

1. **Data Representation:** The current implementation feeds the model with flattened vectors of the MNIST images, rather than leveraging convolutional architectures. Flattening the data removes the spatial structure, making it harder for the model to learn meaningful patterns and relationships inherent to images. Using an encoder-decoder architecture with convolutional layers could significantly improve the reconstruction quality by preserving spatial relationships.
2. **Model Complexity:** The lack of an advanced decoder or specialized image processing techniques limits the ability of the model to capture the complexity of the digit structures. Implementing inpainting methods (Qin et al. 2021)or convolutional approaches could provide more efficient and accurate results for image data.
3. **Computational Constraints:** Due to computational limitations, certain methods like MICE or Random Forest were excluded from comparison. These methods, although computationally intensive, could potentially offer better performance if optimized for this specific task.

References

Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *International conference on learning representations (iclr).*

Dezons, Simon Blotas Charles. 2024. *Notebook implementation of the notmiwae method.* https://colab.research.google.com/drive/1-PI2CMnVC_fVyHvny4msXO5mrAJQBFAD?usp=sharing. Accessible online.

Figurnov, Michael, Shakir Mohamed, and Andriy Mnih. 2018. Implicit reparameterization gradients. Submitted on 22 May 2018, last revised 30 Jan 2019, *arXiv preprint arXiv:1805.08498.*

Ipsen, Niels Bo, Pierre-Alexandre Mattei, and Jes Frellsen. 2021. not-MIWAE: deep generative modelling with missing not at random data. In *Proceedings of the 9th international conference on learning representations.*

Kingma, Diederik P, and Max Welling. 2013. Auto-encoding variational bayes. Submitted on 20 Dec 2013, last revised 10 Dec 2022, *arXiv preprint arXiv:1312.6114.*

Mattei, Pierre-Alexandre, and Jes Frellsen. 2019. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. In *Proceedings of the 36th international conference on machine learning,* 97:4413–4423. Proceedings of Machine Learning Research. PMLR.

Molenberghs, Geert, Caroline Beunckens, Cristina Sotto, and Michael G. Kenward. 2008. Every missingness not at random model has a missingness at random counterpart with equal fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70 (2): 371–388. https://doi.org/10.1111/j.1467-9868.2007.00634.x.

Qin, Zhen, Qingliang Zeng, Yixin Zong, and Fan Xu. 2021. Image inpainting based on deep learning: a review. *Displays* 69:102028. ISSN: 0141-9382. https://doi.org/https://doi.org/10.1016/j.displa.2021.102028. https://www.sciencedirect.com/science/article/pii/S0141938221000391.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning (icml),* 1278–1286.

Appendix 1. Proofs

Appendix 1.1 VAE formulation

Let's first go back to the VAE architecture described by Kingma and Welling (Kingma and Welling 2013). Let $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ denote a dataset of N i.i.d. samples where each observed datapoint $\mathbf{x}^{(i)}$ is obtained by first sampling a latent vector $\mathbf{z}$ from the prior $p_\theta(\mathbf{z})$ and then sampling $\mathbf{x}^{(i)}$ itself from the scene model $p_\theta(\mathbf{x}|\mathbf{z})$. Now we introduce an auxiliary distribution $q_\gamma(\mathbf{z}|\mathbf{x})$ (with its own parameters) as an approximation to the true, but unknown posterior $p_\theta(\mathbf{z}|\mathbf{x})$ (Note that in the paper from Kingma and Welling, the posterior approximation is denoted by $q_\phi(\mathbf{z}|\mathbf{x})$, instead of $q_\gamma(\mathbf{z}|\mathbf{x})$, but to we chose to keep the notation from Mattei and Frelsen where ϕ is associated with the mask). Accordingly, the data likelihood of a one sample $\mathbf{x}^{(i)}$ can be stated as follows

$$p_\theta\left(\mathbf{x}^{(i)}\right) = \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right] = \int p_\theta(\mathbf{z}) p_\theta\left(\mathbf{x}^{(i)}|\mathbf{z}\right) d\mathbf{z} = \int p_\theta\left(\mathbf{x}^{(i)}, \mathbf{z}\right) d\mathbf{z} \tag{10}$$

Now, we use the simple trick of *importance sampling* to change the sampling distribution into the approximated posterior, i.e.,

$$p_\theta\left(\mathbf{x}^{(i)}\right) = \int \frac{p_\theta\left(\mathbf{x}^{(i)}, \mathbf{z}\right)}{q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)} q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right) d\mathbf{z} = \mathbb{E}_{\mathbf{z}\sim q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\frac{p_\theta\left(\mathbf{x}^{(i)}, \mathbf{z}\right)}{q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\right] \tag{11}$$

VAE Formulation

In the standard VAE approach, we use the *evidence lower bound (ELBO)* on $\log p_\theta(\mathbf{x})$ as the objective function. This can be derived by applying Jensen's Inequality on the data log-likelihood:

$$\log p_\theta\left(\mathbf{x}^{(i)}\right) = \log \mathbb{E}_{\mathbf{z}\sim q_\gamma(\mathbf{z}|\mathbf{x}^{(i)})}\left[\frac{p_\theta\left(\mathbf{x}^{(i)},\mathbf{z}\right)}{q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\right] \geq \mathbb{E}_{\mathbf{z}\sim q_\gamma(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log \frac{p_\theta\left(\mathbf{x}^{(i)},\mathbf{z}\right)}{q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\right] = \mathcal{L}_{\text{ELBO}} \quad (12)$$

Using simple algebra, this can be rearranged into

$$\mathcal{L}_{\text{ELBO}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\gamma(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_\theta\left(\mathbf{x}^{(i)}|\mathbf{z}\right)\right]}_{\text{Reconstruction Accuracy}} - \underbrace{D_{KL}\left(q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)\|p_\theta(\mathbf{z})\right)}_{\text{Regularization}} \quad (13)$$

While the regularization term can usually be solved analytically, the reconstruction accuracy in its current formulation poses a problem for backpropagation: Gradients cannot backpropagate through a sampling operation. To circumvent this problem, the standard VAE formulation includes the reparametrization trick: Substitute sampling $\mathbf{z}\sim q_\gamma$ by using a deterministic mapping $\mathbf{z} = g_\gamma(\boldsymbol{\epsilon},\mathbf{x})$ with the differential transformation g_γ of an auxiliary noise variable $\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})$.

As a result, we can rewrite the ELBO as follows

$$\mathcal{L}_{\text{ELBO}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) = \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}\left[\log p_\theta\left(\mathbf{x}^{(i)}|g_\gamma\left(\boldsymbol{\epsilon},\mathbf{x}^{(i)}\right)\right)\right] - D_{KL}\left(q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right)|0 p_\theta(\mathbf{z})\right) \quad (14)$$

Lastly, the expectation is approximated using Monte-Carlo integration, leading to the standard VAE objective

$$\tilde{\mathcal{L}}_k^{\text{VAE}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) = \frac{1}{k}\sum_{l=1}^{k}\log p_\theta\left(\mathbf{x}^{(i)}|g_\gamma\left(\boldsymbol{\epsilon}^{(l)},\mathbf{x}^{(i)}\right)\right) - D_{KL}\left(q_\gamma\left(\mathbf{z}|\mathbf{x}^{(i)}\right),p_\theta(\mathbf{z})\right) \quad (15)$$

with $\boldsymbol{\epsilon}^{(l)}\sim p(\boldsymbol{\epsilon})$

IWAE Formulation

Before we introduce the IWAE estimator, remind that the Monte-Carlo estimator of the data likelihood (when the sampling distribution is changed via importance sampling, see VAE Formulation) is given by

$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\gamma(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x})}\right] \approx \frac{1}{k}\sum_{l=1}^{k}\frac{p_\theta(\mathbf{x},\mathbf{z}^{(l)})}{q_\gamma(\mathbf{z}^{(l)}|\mathbf{x})} \quad \text{with} \quad \mathbf{z}^{(l)}\sim q_\gamma(\mathbf{z}|\mathbf{x}). \quad (16)$$

As a result, the data log-likelihood estimator for one sample $\mathbf{x}^{(i)}$ can be stated as follows

$$\log p_\theta(\mathbf{x}^{(i)}) \approx \log\left[\frac{1}{k}\sum_{l=1}^{k}\frac{p_\theta(\mathbf{x}^{(i)},\mathbf{z}^{(i,l)})}{q_\gamma(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)})}\right] = \tilde{\mathcal{L}}_k^{\text{IWAE}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) \quad (17)$$

with $\mathbf{z}^{(i,l)}\sim q_\gamma(\mathbf{z}|\mathbf{x}^{(i)})$

which leads to an empirical estimate of the IWAE objective. However, Burda et al. (Burda, Grosse, and Salakhutdinov 2016) do not use the data log-likelihood in its plain form as the true IWAE objective. Instead, they introduce the IWAE objective as follows

$$\mathcal{L}_k^{\text{IWAE}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) = \mathbb{E}_{\mathbf{z}^{(i,1)},\ldots,\mathbf{z}^{(i,k)}\sim q_\gamma(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log\frac{1}{k}\sum_{l=1}^{k}\frac{p_\theta(\mathbf{x}^{(i)},\mathbf{z}^{(l)})}{q_\gamma(\mathbf{z}^{(l)}|\mathbf{x}^{(i)})}\right]. \tag{18}$$

For notation purposes, they denote

$$\text{(unnormalized) importance weights:}\quad w^{(i,l)} = \frac{p_\theta(\mathbf{x}^{(i)},\mathbf{z}^{(l)})}{q_\gamma(\mathbf{z}^{(l)}|\mathbf{x}^{(i)})}. \tag{19}$$

By applying Jensen's Inequality, we can see that in fact the (true) IWAE estimator is merely a lower-bound on the plain data log-likelihood

$$\mathcal{L}_k^{\text{IWAE}}\left(\theta,\gamma;\mathbf{x}^{(i)}\right) = \mathbb{E}\left[\log\frac{1}{k}\sum_{l=1}^{k}w^{(i,l)}\right] \leq \log\mathbb{E}\left[\frac{1}{k}\sum_{l=1}^{k}w^{(i,l)}\right] = \log p_\theta(\mathbf{x}^{(i)}). \tag{20}$$

They could prove that with increasing k the lower bound gets strictly tighter and approaches the true data log-likelihood in the limit of $k \to \infty$. Note that since the empirical IWAE estimator $\tilde{\mathcal{L}}_k^{\text{IWAE}}$ can be understood as a Monte-Carlo estimator on the true data log-likelihood, in the empirical case this property can simply be deduced from the properties of Monte-Carlo integration.

Appendix 1.2 Reparameterization for the not-MIWAE Objective, and ways of adjusting the model to concrete cases

As seen in equation (6), to obtain unbiased estimates of gradients of the bound $L_K(\theta,\phi,\gamma)$, we will need to backpropagate through samples from $q_\gamma(z|x_i^o)$, $p_\theta(x^m|z)$. To make this tractable, we can use the reparametrization trick from the VAE formulation in (14) and pick for both $q_\gamma(z|x_i^o)$ and $p_\theta(x^m|z)$, a simple distribution from a class of distributions well known (gaussian in our code). But the distribution can be anything as soon as it is reparameterizable. For instance, Figurnov et al.(Figurnov, Mohamed, and Mnih 2018) gives a list of such distributions. It can be a good way of improving the code, if we know already what kind of distribution will fit best the encoder $q_\gamma(z|x_i^o)$ and the decoder $p_\theta(x^m|z)$.

Appendix 1.3 Imputation

If one wants to be more precise, it is possible to have a larger choice of loss functions rather than only using the squared error. The goal is to minimize the loss function thanks to moments of the conditional distribution of the missing values given the observed. For more details about this process, please refer to Mattei and Frellsen 2019, equations (10-11). These moments can be estimated via self-normalized importance sampling. For any function of the missing data $h(x^m)$,

$$\mathbb{E}[h(x^m)|x^o,s] = \int h(x^m)p(x^m|x^o,s)\,dx^m. \tag{21}$$

Using Bayes's theorem, we get

$$\mathbb{E}[h(x^m)|x^o,s] = \int h(x^m)\frac{p(s|x^o,x^m)p(x^m,x^o)}{p(s,x^o)}\,dx^m, \tag{22}$$

Once again, we introduce the latent variable:

$$\mathbb{E}[h(x^m)|x^o, s] = \iint h(x^m)\frac{p(s|x^o, x^m)p(x^m|z)p(x^o|z)p(z)}{p(s, x^o)}\, dz\, dx^m. \tag{23}$$

Using self-normalized importance sampling on this last integral with proposal $q_\gamma(z|x^o)p_\theta(x^m|z)$ leads to the estimate

$$\hat{x}^m = \mathbb{E}[h(x^m)|x^o, s] \approx \sum_{k=1}^{K} \alpha_k h(x_k^m), \quad \text{with} \quad \alpha_k = \frac{w_k}{w_1 + \cdots + w_K}. \tag{24}$$

where the weights $w_1, \ldots, w_K$ are incidentally identical to the ones used for training:

$$\forall k \leq K, \quad w_k = \frac{p_\phi(s|x^o, x_k^m)p_\theta(x^o|z_k)p(z_k)}{q_\gamma(z_k|x^o)}. \tag{25}$$

and $(z_1, x_1^m), \ldots, (z_K, x_K^m)$ are K i.i.d. samples from $(q_\gamma(z|x^o), p_\theta(x^m|z))$. If $\mathbb{E}[h(x^m)|z]$ is easy to compute, then it is preferable to choose a Rao-Blackwellized version of equation (24):

$$\hat{x}^m = \mathbb{E}[h(x^m)|x^o, s] \approx \sum_{k=1}^{K} \alpha_k \mathbb{E}[h(x^m)|z_k]. \tag{26}$$

Squared error

When L corresponds to the squared error, we can simply use the above calculation where h is the identity function :

$$\hat{x}^m = \mathbb{E}[x^m|x^o, s] \approx \sum_{k=1}^{K} \alpha_k \mathbb{E}[x^m|x^o, s], \quad \text{with} \quad \alpha_k = \frac{w_k}{w_1 + \cdots + w_K} \tag{27}$$

Multiple imputation

It is also possible to perform multiple imputation with the same computations. It can be handy when, instead of imputing the missing values directly, we want to obtain samples from $p(x^m|x^o)$. Those samples are approximate and are imputed using sampling importance resampling with the same set of weights described before.

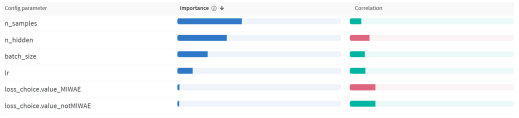
Appendix 2. Hyperparameter Optimization using W&B Sweep

To optimize the performance of our model (the detailed architecture is provided in the code notebook Dezons 2024), we conducted a hyperparameter search using the Weights and Biases (W&B) Sweep framework. The optimization targeted the minimization of the RMSE metric on the reconstructed data for 2D Gaussian example. The sweep employed a Bayesian search strategy to explore the hyperparameter space efficiently. The hyperparameters included:

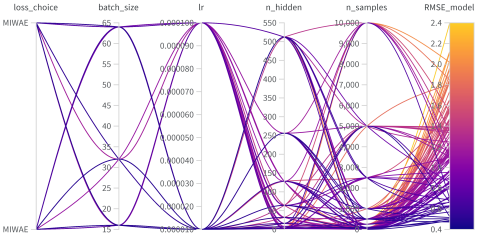
- **Learning rate (lr):** Explored values were 10^{-3}, 10^{-4}, and 10^{-5}.
- **Batch size:** Values included 16, 32, and 64.
- **Number of samples (n_samples):** Tested values ranged from 1 to 1000, including intermediate values like 2.5 and 250.
- **Number of hidden units (n_hidden):** Explored sizes included 16, 32, 64, 128, and 256.
- **Loss choice:** Two loss functions, "MIWAE" and "notMIWAE," were evaluated.

- **Maximum iterations (max_iter):** This was fixed at 10,000 for all runs.

For each experiment, a new combination of hyperparameters was used to train an encoder-decoder architecture. After training, the model reconstructed the data, and its performance was evaluated using the RMSE metric. The results, including the RMSE and corresponding hyperparameter values, were logged into W&B.



(a) Correlation between hyperparameters and RMSE.



(b) Visualization of RMSE scores for different hyperparameter values.

Figure 5. Results from the W&B hyperparameter sweep.

The sweep allowed us to identify the optimal combination of hyperparameters, enabling the model to achieve a minimal RMSE while balancing computational efficiency.