We thank the reviewers for their time and relevant comments.

Why we use our architecture to approximate the self-masking Bayes predictor - to rev. #3 and #4: Reviewer #4 mentioned a lack of clarity in the paragraph entitled "Differences between MNAR and M(C)AR predictors" (l.128-133). This paragraph actually touches upon a point raised by Reviewer #3: why our method can be used for self-masking missingness (also discussed l.190-192). These questions suggest that this paragraph should be more detailed. Hereafter, we explain it in more details and give an answer to reviewer #3.

The expression of the M(C)AR Bayes predictors is given by (eq. 4 in the paper):

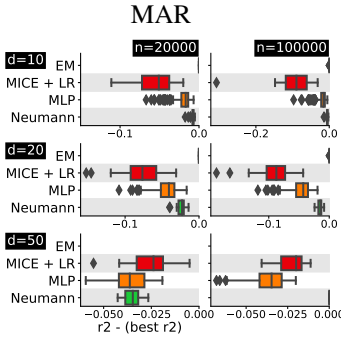
$$f^\star(X_{obs}, M) = \beta_0^\star + \langle \beta_{obs}^\star, X_{obs} \rangle + \langle \beta_{mis}^\star, \mu_{mis} + \Sigma_{mis,obs}(\Sigma_{obs})^{-1}(X_{obs} - \mu_{obs}) \rangle \quad (1)$$

The expression of the (MNAR) self-masking Bayes predictor is more complicated (eq. 5 in the paper). To study this expression, we approximate $D_{mis}\Sigma_{mis|obs}^{-1}$ by Id. Then, the self-masking Bayes predictor becomes:

$$f^\star(X_{obs}, M) \approx \beta_0^\star + \left\langle \beta_{obs}^\star, X_{obs} \right\rangle + \left\langle \beta_{mis}^\star, \frac{1}{2}(\tilde{\mu}_{mis} + \mu_{mis}) + \frac{1}{2}\Sigma_{mis,obs}\left(\Sigma_{obs}\right)^{-1}\left(X_{obs} - \mu_{obs}\right) \right\rangle \quad (2)$$

Thus, under this approximation, the self-masking Bayes predictor can be modeled by our proposed architecture (just as the M(C)AR Bayes predictor), the only difference being the targeted value for parameter μ of the network (in blue in the two models above) and a scaling factor of $1/2$ for W_{mix} (in orange). A less coarse approximation also works: $D_{mis}\Sigma_{mis|obs}^{-1} \approx \hat{D}_{mis}$ where $\hat{D}$ is a diagonal matrix. In this case, the proposed architecture can perfectly model the self-masking Bayes predictor: the parameter μ of the network should target $(Id + \hat{D})^{-1}(\tilde{\mu} + \hat{D}\mu)$ and W_{mix} should target $(Id + \hat{D})^{-1}\hat{D}\Sigma$ instead of simply Σ in the M(C)AR case. Consequently, **our architecture can well approximate the self-masking Bayes predictor by adjusting the values learned for the parameters μ and W_{mix} if $D_{mis}\Sigma_{mis|obs}^{-1}$ are close to diagonal matrices**. We will add this discussion to the Appendix.

Experimental results under the MAR scenario - to rev.#3 The results are presented in the figure below:



The MAR data was generated as follows: first, a subset of variables with *no* missing values is randomly selected (10%). The remaining variables have missing values according to a logistic model with random weights, re-scaled so as to attain the desired proportion of missing values on those variables (50%). Note that we cannot compute the Bayes rate, so instead of showing (R2 - Bayes rate) we show (R2 - best R2).

As can be seen from the figure, the trends oberved for MAR are the same as those for MCAR. We will add this figure to the appendix. We have not tested the scenario where MAR and self-masking missingness happen at the same time.

Modeling non linear functions - to rev. #1 and #2 The reviewers pointed out that the proposed architecture is limited to linear models. Indeed, our theoretical foundations derive from the study of linear models. However, as a differentiable architecture, it can be readily included as a building block in more complex networks. For example, the layer W_β can be replaced by a MLP.

Robustness to non Gaussian data - to rev. #2 and #3 The crux of the method is to capture the covariance of the data, which relates the different slopes of the models on incomplete data. This covariance will be relevant even on non-Gaussian data, though we can so far only develop formal arguments under Gaussian assumptions.

What is ν in eq.(8) - to rev. #1 It is the smallest eigenvalue of the covariance matrix Σ. We apologize, the definition was inadequately moved to the appendix; we will move it to the main text.

What happens with few samples - to rev. #2 Reviewer #2 made a comment related to the number of samples. The difficulty of a problem is linked to the ratio # dimensions/# samples, the higher the ratio, the more difficult the problem. The highest ratio for which we presented experiments is for dimension 50 and 20000 samples. For such ratio and higher, using a depth of 0 for our architecture is enough, more depth triggers overfitting. We showed theoretically in section 3.4 that the MLP can model a depth-0 Neumann network. Thus the MLP, whether in MCAR or selfmaking, is on par or slightly better than the Neumann network for high ratios. As for MICE+LR, it has an advantage over MLP and Neumann for high ratios in MCAR, but not in selfmasking, because Mice assumes the data to be M(C)AR.