Appendix A. Details of NNGP

In this section, we provide the correspondence between infinitely wide fully connected neural networks and Gaussian processes which is proved in Lee et al. (2017). We remark that other types of neural networks, e.g. CNN, also works compatibly with the NNGP. Here we consider L-hidden-layer fully connected neural networks with input $\mathbf{x} \in \mathcal{R}^{d_{\text{in}}}$, layer width n^l (for l-th layer and $d_{\text{in}} := n^0$), parameter $\boldsymbol{\theta}$ consisting of weight $\mathbf{W}^l$ and bias $\mathbf{b}^l$ for each layer l in the network, pointwise nonlinearity ϕ, post-affine transformation (pre-activation) z_i^l and post-nonlinearity x_i^l for the i-th neuron in the l-th layer. We denote $x_i^0 = x_i$ for the input and use a Greek superscript $\mathbf{x}^\alpha$ to denote the α-th sample. Weight $\mathbf{W}^l$ and bias $\mathbf{b}^l$ have components W_{ij}^l and b_i^l independently drawn from normal distribution $\mathcal{N}\left(0, \frac{\sigma_w^2}{n^l}\right)$ and $\mathcal{N}\left(0, \sigma_b^2\right)$, respectively.

Then the i-th component of pre-activation z_i^0 is computed as:

$$z_i^0(\mathbf{x}) = \sum_{j=1}^{d_{\text{in}}} W_{ij}^0 x_j + b_i^0$$

where the pre-activation $z_i^0(\mathbf{x})$ emphasizes z_i^0 depends on the input $\mathbf{x}$. Since the weight $\mathbf{W}^0$ and bias $\mathbf{b}^0$ are independently drawn from normal distributions, $z_i^0(\mathbf{x})$ also follows a normal distribution. Likewise, any finite collection $\{z_i^0(\mathbf{x}^{\alpha=1}), \ldots, z_i^0(\mathbf{x}^{\alpha=k})\}$ which is composed of i-th pre-activation z_i^0 at k different inputs will have a joint multivariate normal distribution, which is exactly the definition of Gaussian process. Hence $z_i^0 \sim \mathcal{GP}(\mu^0, \mathcal{K}^0)$, where $\mu^0(\mathbf{x}) = \mathbb{E}[z_i^0(\mathbf{x})] = 0$ and

$$\mathcal{K}^0(\mathbf{x}, \mathbf{x}') = \mathbb{E}[z_i^0(\mathbf{x}) z_i^0(\mathbf{x}')] = \sigma_b^2 + \sigma_w^2 \left(\frac{\mathbf{x} \cdot \mathbf{x}'}{d_{\text{in}}}\right)$$

Notice that any two z_i^0, z_j^0 for $i \neq j$ are joint Gaussian, having zero covariance, and are guaranteed to be independent despite utilizing the same input.

Similarly, we could analyze i-th component of first layer pre-activation z_i^1:

$$z_i^1(\mathbf{x}) = \sum_{j=1}^{n^1} W_{ij}^1 x_j^1 + b_i^1 = \sum_{j=1}^{n^1} W_{ij}^1 \phi(z_j^0(\mathbf{x})) + b_i^1.$$

We obtain that $z_i^1 \sim \mathcal{GP}(0, \mathcal{K}^1)$, where

$$\begin{aligned}\mathcal{K}^1(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[z_i^1(\mathbf{x}) z_i^1(\mathbf{x}')] \\ &= \sigma_b^2 + \sigma_w^2 \left(\frac{\sum_{j=1}^{n^1} \phi(z_j^0(\mathbf{x}))\phi(z_j^0(\mathbf{x}'))}{n^1}\right)\end{aligned}$$

Since $z_j^0 \sim \mathcal{GP}(0, \mathcal{K}^0)$, let $n^1 \to \infty$, the covariance is

$$\begin{aligned}
&\mathcal{K}^1(\mathbf{x}, \mathbf{x}') \\
=&\sigma_b^2 + \sigma_w^2 \iint \phi(z)\phi(z') \\
&\mathcal{N}\left(\begin{bmatrix} z \\ z' \end{bmatrix}; 0, \begin{bmatrix} \mathcal{K}^0(\mathbf{x}, \mathbf{x}) & \mathcal{K}^0(\mathbf{x}, \mathbf{x}') \\ \mathcal{K}^0(\mathbf{x}', \mathbf{x}) & \mathcal{K}^0(\mathbf{x}', \mathbf{x}') \end{bmatrix}\right) dz\, dz' \\
=&\sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_j^0 \sim \mathcal{GP}(0, \mathcal{K}^0)}\left[\phi(z_i^0(\mathbf{x}))\phi(z_i^0(\mathbf{x}'))\right]
\end{aligned}$$

This integral can be solved analytically for some activation functions, such as ReLU nonlinearity Cho and Saul (2009). If this integral cannot be solved analytically, it can be efficiently computed numerically Lee et al. (2017). Hence $\mathcal{K}^1$ is determined given $\mathcal{K}^0$.

We can extend previous arguments to general layers by induction. By taking each hidden layer width to infinity successively ($n^1 \to \infty, n^2 \to \infty, \dots$), we can conclude $z_i^l \sim \mathcal{GP}(0, \mathcal{K}^l)$, where $\mathcal{K}^l$ could be computed from the recursive relation

$$\begin{aligned}
\mathcal{K}^l(\mathbf{x}, \mathbf{x}') &= \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_j^{l-1} \sim \mathcal{GP}(0, \mathcal{K}^{l-1})}\left[\phi(z_i^0(\mathbf{x}))\phi(z_i^0(\mathbf{x}'))\right] \\
\mathcal{K}^0(\mathbf{x}, \mathbf{x}') &= \sigma_b^2 + \sigma_w^2 \left(\frac{\mathbf{x} \cdot \mathbf{x}'}{d_{\text{in}}}\right)
\end{aligned}$$

Hence, the covariance only depends on the neural network structure (including weight and bias variance, number of layers and activation function).

Appendix B. Implementation details

All the experiments run on Google Colab Pro with P100 GPU. For GAIN[1], Sinkhorn, Linear RR[2], and MIWAE[3], we use the open-access implementations provided by their authors, with the default or the recommended hyperparameters in their papers except MIWAE. For MIWAE, the default hyperparameters lead to running out RAM, hence we choose `h=128, d=10, K=20, L=1000`. For SoftImpute, the `lambda` hyperparameter is selected at each run through cross-validation and grid-point search, and we choose `maxit=500` and `thresh=1e-05`. For MICE, we use the `iterativeImputer`[4] method in the `scikit-learn` library with default hyperparameters Pedregosa et al. (2011). All NNGP-based methods uses a 3-layer fully connected neural network with ReLU activation function to impute missing values, where the initialization of weight and bias variances are set to 1 and 0 respectively. (We also tried other initialization of weight and bias variances and found that the result is very robust to these changes.) NNGP-based methods are implemented through Neural TangentsNovak et al. (2020). All the MI methods are used to multiply impute missing values for 10 times except GAIN, MIWAE and Linear RR, noting that the GAIN and MIWAE implementations from their authors conduct SI and Linear RR is

1. See https://github.com/jsyoon0823/GAIN
2. Same as Sinkhorn, see https://github.com/BorisMuzellec/MissingDataOT
3. See https://github.com/pamattei/miwae
4. See https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

computationally very expensive. We also include **not-MIWAE**[5] in the MNAR setting in the appendix. Similar to MIWAE, default hyperparameters of not-MIWAE lead to running out RAM, here we choose n_{hidden}`=128,` $n_{samples}$`=20, batch size=16, dl=p-1, L=1000, mprocess='selfmasking known'`. We observe that not-MIWAE is unstable and performs poorly. Probably because not-MIWAE is not scalable to high-dimensional data.

Appendix C. Synthetic data experiments

C.1. Continuous data experiment

The simulation results are summarized over 100 Monte Carlo (MC) datasets. We also include not-MIWAE in MNAR. Note that the Each MC dataset has a sample size of $n = 200$ and includes $\mathbf{y}$, the fully observed outcome variable, and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$, the set of predictors and auxiliary variables. We consider the setting $p = 50$, $p = 250$, and $p = 1000$ (Here the use of p is a slight abuse of notation. In the main paper, p represents total number features which include predictors, auxiliary variables and the response.). $\mathbf{X}$ is obtained by rearranging the orders of $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_p)$ and $\mathbf{A}$ is generated from a first order autoregressive model with autocorrelation ρ and white noise ϵ. Here $\mathbf{a}_1$ is generated from standard normal distribution $\mathcal{N}(0, 1)$ if $\epsilon \sim \mathcal{N}(0, 0.1^2)$ or exponential distribution $\mathrm{Exp}(2)$ if $\epsilon \sim \mathrm{Exp}(0.4)$. To obtain $\mathbf{X}$, we firstly move the fourth variable in every five consecutive variables of $\mathbf{A}$ (e.g. $\mathbf{a}_4$, $\mathbf{a}_9$ and $\mathbf{a}_{14}$) to the right and then the fifth variable in every five consecutive variables of $\mathbf{A}$ (e.g. $\mathbf{a}_5$, $\mathbf{a}_{10}$ and $\mathbf{a}_{15}$) to the right. For a concrete example, if $p = 10$, $(\mathbf{a}_1, \ldots, \mathbf{a}_{10})$ becomes $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8, \mathbf{a}_4, \mathbf{a}_9, \mathbf{a}_5, \mathbf{a}_{10})$ after rearrangement. The response $\mathbf{y}$ depends on three variables of $\mathbf{X}$ indicated by a set q: given $\mathbf{X}$, $\mathbf{y}$ is generated from

$$\mathbf{y}_i = \beta_1 \cdot \mathbf{x}_{q[1]} + \beta_2 \cdot \mathbf{x}_{q[2]} + \beta_3 \cdot \mathbf{x}_{q[3]} + \mathcal{N}(0, \sigma_1^2) \tag{4}$$

where $\beta_i = 1$ for $i \in \{1, 2, 3\}$. For $p = 50$, $p = 250$ and $p = 1000$, the corresponding predictor set q is $\{40, 44, 48\}$ $\{210, 220, 230\}$ and $\{650, 700, 750\}$ respectively.

MAR or MNAR mechanism is considered in the simulation and the missing rate is around 40%. In particular, missing values are separately created in $\{\mathbf{x}_{\frac{3}{5}p+1}, \ldots, \mathbf{x}_{\frac{4}{5}p}\}$ and $\{\mathbf{x}_{\frac{4}{5}p+1}, \ldots, \mathbf{x}_p\}$ by using the following logit models for the corresponding missing indicators $\mathbf{R}_1$ and $\mathbf{R}_2$. If the missing mechanism is **MAR**:

$$\mathrm{logit}(\mathbb{P}(\mathbf{R}_1 = 1|\mathbf{X}, \mathbf{y})) = a_1 + a_2 \cdot \frac{5}{3p} \sum_{j=1}^{3p/5} \mathbf{x}_j + a_3 \cdot \mathbf{y} \tag{5}$$

$$\mathrm{logit}(\mathbb{P}(\mathbf{R}_2 = 1|\mathbf{X}, \mathbf{y})) = a_4 + a_5 \cdot \frac{5}{3p} \sum_{j=1}^{3p/5} \mathbf{x}_j + a_6 \cdot \mathbf{y} \tag{6}$$

5. See https://github.com/nbip/notMIWAE

If the missing mechanism is **MNAR**:

$$\text{logit}(\mathbb{P}(\mathbf{R}_1 = 1|\mathbf{X}, \mathbf{y})) = a_1 + a_2 * \frac{5}{p} \sum_{j=4p/5+1}^{p} \mathbf{x}_j + a_3 * \mathbf{y} \tag{7}$$

$$\text{logit}(\mathbb{P}(\mathbf{R}_2 = 1|\mathbf{X}, \mathbf{y})) = a_4 + a_5 \cdot \frac{5}{p} \sum_{j=3p/5+1}^{4p/5} \mathbf{x}_j + a_6 \cdot \mathbf{y} \tag{8}$$

If $\mathbf{R}_1 = 1$ or 0, then $\{\mathbf{x}_{\frac{3}{5}p+1}, \ldots, \mathbf{x}_{\frac{4}{5}p}\}$ is missing or observed, respectively; similarly, if $\mathbf{R}_2 = 1$ or 0, then $\{\mathbf{x}_{\frac{4}{5}p+1}, \ldots, \mathbf{x}_p\}$ is missing or observed, respectively.

C.2. Discrete data experiment

In the discrete data analysis, we append one binary variable $\mathbf{x}_{p+1}$ on the last column of $\mathbf{X}$ in the above section. We consider the setting $p = 1000$. The binary variable is generated through:

$$\mathbf{x}_{p+1} = \begin{cases} 1 & \text{if } \mathbf{x}_{10} + \mathbf{x}_{50} + \mathbf{x}_{100} > 0 \\ 0 & \text{otherwise} \end{cases}.$$

The fully observed response $\mathbf{y}$ is also generated from eq. (4) and the corresponding predictor set q is $\{1001, 701, 751\}$. Hence β_1 is the coefficient of the binary variable in the regression model. Here missing values are separately created in $\{\mathbf{x}_{\frac{3}{5}p+1}, \ldots, \mathbf{x}_{\frac{4}{5}p}\}$ and $\{\mathbf{x}_{\frac{4}{5}p+1}, \ldots, \mathbf{x}_{p+1}\}$ with the corresponding missing indicators $\mathbf{R}_1$ and $\mathbf{R}_2$, which are also generated form (5), (6) or (7), (8) depending on the specific missing mechanism.

Before MI-NNGPs impute, the binary variable is encoded into an one-hot, zero-mean vector (i.e. entries of -0.5 for the incorrect class and 0.5 for the correct class). After imputing this one-hot vector in the incomplete cases, the class with higher value is regarded as the imputation class.

C.3. Experiment setting

- Table 2: Continuous data experiment, MAR, $n = 200$, $p = 250$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 3: Continuous data experiment, MAR, $n = 200$, $p = 1000$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 11: Discrete data experiment, MAR, $n = 200$, $p = 1000$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = -1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 1$, $a_5 = 2$, $a_6 = -2$
- Table 5: Continuous data experiment, MAR, $n = 200$, $p = 50$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 6: Continuous data experiment, MAR, $n = 200$, $p = 1000$, $\rho = 0.75$, $\epsilon \sim \text{Exp}(0.4)$, $\sigma_1 = 1$, $a_1 = -3$, $a_2 = -1$, $a_3 = 1.5$, $a_4 = 1$, $a_5 = 1.5$, $a_6 = -1$

- Table 7: Continuous data experiment, MNAR, $n = 200$, $p = 50$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 8: Continuous data experiment, MNAR, $n = 200$, $p = 250$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 9: Continuous data experiment, MNAR, $n = 200$, $p = 1000$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = 1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 0$, $a_5 = 2$, $a_6 = -2$
- Table 10: Continuous data experiment, MNAR, $n = 200$, $p = 1000$, $\rho = 0.75$, $\epsilon \sim \text{Exp}(0.4)$, $\sigma_1 = 1$, $a_1 = -3$, $a_2 = -1$, $a_3 = 1.5$, $a_4 = 1$, $a_5 = 1.5$, $a_6 = -1$
- Table 12: Discrete data experiment, MNAR, $n = 200$, $p = 1000$, $\rho = 0.95$, $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\sigma_1 = 0.5$, $a_1 = -1$, $a_2 = -2$, $a_3 = 3$, $a_4 = 1$, $a_5 = 2$, $a_6 = -2$

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	2.7	**0.0132**	**-0.0017**	**0.92**	0.0623	0.0642
GAIN	SI	35.9	1.356	0.3213	0.38	0.1142	0.4262
MIWAE	SI	46.5	0.0361	-0.0238	0.90	0.0632	0.0738
Linear RR	SI	628.4	0.1712	**0.0358**	0.91	0.1287	0.1568
MICE	MI	2.1	0.0200	**0.0031**	**0.97**	0.0644	0.0567
Sinkhorn	MI	42.1	0.1081	-0.1225	0.60	0.0978	0.1269
MI-NNGP1	MI	3.4	**0.0129**	**0.0048**	**0.95**	0.0621	0.0647
MI-NNGP1-BS	MI	4.8	**0.0177**	**0.0052**	**0.97**	0.0794	0.0624
MI-NNGP2	MI	5.5	**0.0092**	**0.0083**	**0.96**	0.0639	0.0563
MI-NNGP2-BS	MI	13.5	**0.0105**	**0.0083**	**0.98**	0.0705	0.0574
Complete data	-	-	-	0.0025	0.98	0.0605	0.0524
Complete case	-	-	-	0.1869	0.79	0.2298	0.2419
ColMean Imp	SI	-	0.4716	0.5312	0.28	0.2242	0.1729

Table 5: Gaussian data with $n = 200$ and $p = 51$ under MAR. Approximately **40%** features and **92%** cases contain missing values.

C.4. Varying missing rates experiment

Here we state clearly the varying missing rates experiment. Similar to the data generation process in the continuous data experiment, each MC dataset has sample size of $n = 200$ and each sample includes a response y and $p = 1000$ features. When generating variable set $\mathbf{A}$, $\mathbf{a}_1$ is drawn from $\mathcal{N}(0, 1)$ and the remaining variables are generated through first order autoregressive model with autocorrelation $\rho = 0.95$ and white noise $\mathcal{N}(0, 0.1^2)$. $\mathbf{X}$ is obtained by firstly moving the seventh variable and ninth variable in every ten consecutive variables of $\mathbf{A}$ (e.g., $\mathbf{a}_7$, $\mathbf{a}_9$, $\mathbf{a}_{17}$ and $\mathbf{a}_{19}$) to the right and then the eighth variable and tenth variable in every ten consecutive variables of $\mathbf{A}$ (e.g., $\mathbf{a}_8$, $\mathbf{a}_{10}$, $\mathbf{a}_{18}$ and $\mathbf{a}_{20}$) to the right. Given $\mathbf{X}$, $\mathbf{y}$ is generated from (4) with corresponding predictor set $q = \{910, 950, 990\}$. Missing values are separately created in two groups of variables under MAR by using the

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	37.8	0.6284	-0.5896	0.92	0.6348	0.4858
GAIN	SI	130.9	1.691	-0.7217	0.40	0.4611	2.229
MIWAE	SI	58.4	1.530	0.5626	0.05	0.1522	0.1456
Sinkhorn	MI	42.1	0.3845	-0.2077	1.00	0.4566	0.2832
MI-NNGP1	MI	3.1	0.3296	**-0.0421**	0.75	0.1757	0.3220
MI-NNGP1-BS	MI	3.4	0.4543	-0.0570	1.00	0.3366	0.2466
MI-NNGP2	MI	3.9	**0.2358**	0.1098	0.80	0.2312	0.3383
MI-NNGP2-BS	MI	10.3	**0.2516**	**-0.0242**	**0.95**	0.3203	0.3037
Complete data	-	-	-	0.0156	0.95	0.0978	0.0938
Complete case	-	-	-	0.1726	0.89	0.3984	0.4534
ColMean Imp	SI	-	0.3794	-0.1506	1.00	0.4556	0.3123

Table 6: Exponential data with $n = 200$ and $p = 1001$ under MAR. Approximately **40%** features and **92%** cases contain missing values. Here Linear RR and MICE are not included due to running out of RAM.

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	2.1	**0.0119**	**-0.0053**	**0.93**	0.0624	0.0611
GAIN	SI	35.9	1.4822	0.4448	0.24	0.1187	0.4641
MIWAE	SI	46.5	0.0361	-0.0238	0.90	0.0632	0.0738
not-MIWAE	SI	40.8	0.7566	-0.0436	0.91	0.0518	0.0969
Linear RR	SI	407	0.1760	**0.0412**	0.91	0.1314	0.1567
Sinkhorn	MI	27.9	0.1103	-0.1340	0.63	0.1006	0.1278
MICE	MI	2.1	0.0198	**0.0036**	**0.98**	0.0636	0.0559
MI-NNGP1	MI	4.7	**0.0130**	**0.0027**	**0.95**	0.0621	0.0651
MI-NNGP1-BS	MI	3.9	**0.0177**	**0.0026**	**0.97**	0.0799	0.0631
MI-NNGP2	MI	10.4	**0.0088**	**0.0085**	**0.96**	0.0614	0.0536
MI-NNGP2-BS	MI	10.1	**0.0106**	**0.0093**	**0.97**	0.0711	0.0564
Complete data	-	-	-	0.0025	0.98	0.0605	0.0524
Complete case	-	-	-	0.2143	0.78	0.2340	0.4201
ColMean Imp	SI	-	0.4772	0.5597	0.24	0.2246	0.1720

Table 7: Gaussian data with $n = 200$ and $p = 51$ under MNAR. Approximately **40%** features and **92%** cases contain missing values.

following logit models for the corresponding missing indicators $\mathbf{R}_1$ and $\mathbf{R}_2$:

$$\text{logit}(\mathbb{P}(\mathbf{R}_1 = 1|\mathbf{X}, \mathbf{y})) = 1 - \frac{1}{50}\sum_{j=1}^{100}\mathbf{x}_j + 3\mathbf{y}$$

$$\text{logit}(\mathbb{P}(\mathbf{R}_2 = 1|\mathbf{X}, \mathbf{y})) = \frac{1}{50}\sum_{j=1}^{100}\mathbf{x}_j - 2\mathbf{y}$$

If the missing rate is 20%, the first group is $\{\mathbf{x}_{801}, \dots, \mathbf{x}_{900}\}$ and the second group is $\{\mathbf{x}_{901}, \dots, \mathbf{x}_{1000}\}$. If the missing rate is 40%, the first group is $\{\mathbf{x}_{601}, \dots, \mathbf{x}_{800}\}$ and the second group is $\{\mathbf{x}_{801}, \dots, \mathbf{x}_{1000}\}$. If the missing rate is 60%, the first group is $\{\mathbf{x}_{401}, \dots, \mathbf{x}_{700}\}$ and the second group is $\{\mathbf{x}_{701}, \dots, \mathbf{x}_{1000}\}$. If the missing rate is 80%, the first group is $\{\mathbf{x}_{201}, \dots, \mathbf{x}_{600}\}$ and the second group is $\{\mathbf{x}_{601}, \dots, \mathbf{x}_{1000}\}$.

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	15.3	0.0194	-0.0997	0.84	0.1182	0.1358
GAIN	SI	53.2	0.8618	0.6212	0.18	0.1502	0.5088
MIWAE	SI	47.6	0.0502	0.0695	0.90	0.1356	0.1410
not-MIWAE	SI	41.7	1.4701	0.1040	0.65	0.1084	0.1624
Linear RR	SI	3009.6	0.0658	0.1823	0.90	0.1782	0.0935
MICE	MI	48.6	0.0233	**-0.0049**	**0.93**	0.1160	0.1244
Sinkhorn	MI	29.9	0.0757	**0.0117**	**0.97**	0.1839	0.1523
MI-NNGP1	MI	3.4	**0.0116**	**0.0069**	**0.93**	0.1147	0.1215
MI-NNGP1-BS	MI	3.4	**0.0149**	**0.0140**	**0.96**	0.1285	0.1179
MI-NNGP2	MI	10.4	**0.0085**	**-0.0024**	**0.95**	0.1123	0.1148
MI-NNGP2-BS	MI	10.3	**0.0094**	**-0.0018**	**0.96**	0.1177	0.1147
Complete data	-	-	-	-0.0027	0.90	0.1098	0.1141
Complete case	-	-	-	0.2518	0.89	0.3385	0.3319
ColMean Imp	SI	-	0.1414	0.3539	0.72	0.2210	0.1712

Table 8: Gaussian data with $n = 200$ and $p = 251$ under MNAR. Approximately **40%** features and **90%** cases contain missing values.

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	25.1	0.0443	-0.2550	0.52	0.1570	0.2164
GAIN	SI	111.1	0.7395	0.6488	0.18	0.1719	0.5830
MIWAE	SI	53.2	0.1116	0.5249	0.35	0.1902	0.2523
not-MIWAE	SI	45.4	3.981	0.9897	0.0	0.1080	0.1451
Sinkhorn	MI	116.9	0.0889	0.5445	0.38	0.2406	0.2237
MI-NNGP1	MI	4.9	**0.0119**	**0.0351**	0.89	0.1194	0.1422
MI-NNGP1-BS	MI	4.9	**0.0166**	**0.0383**	**0.94**	0.1424	0.1416
MI-NNGP2	MI	10.5	**0.0085**	**0.0356**	0.91	0.1160	0.1310
MI-NNGP2-BS	MI	9.9	**0.0092**	**0.0343**	**0.93**	0.1263	0.1301
Complete data	-	-	-	0.0350	0.94	0.1122	0.1173
Complete case	-	-	-	0.2824	0.76	0.3447	0.4201
ColMean Imp	SI	-	0.1130	0.7022	0.11	0.2572	0.1941

Table 9: Gaussian data with $n = 200$ and $p = 1001$ under MNAR. Approximately **40%** features and **90%** cases contain missing values. Here Linear RR and MICE are not included due to running out of RAM.

Appendix D. ADNI data experiments

D.1. Data Availability

The de-identified ADNI dataset is publicly available at http://adni.loni.usc.edu/.

D.2. Experiment details

This section details the ADNI data experiment. Here we use a large-scale dataset from ADNI study. The original dataset includes 19822 features and one continuous response variable ($\mathbf{y}$), the VBM right hippocampal volume, for 649 patients. Here we preprocess features and the response by removing their means. Among these 19822 features, we only select 10000 features which have maximal correlation with the response to analyze and rank them in the

Models	Style	Time(s)	Imp MSE	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	39.1	0.6682	-0.6784	0.90	0.6805	0.4632
GAIN	SI	91.0	1.6974	0.0331	0.33	0.4187	2.2123
MIWAE	SI	57.4	1.4937	0.3981	0.30	0.1437	0.1659
not-MIWAE	SI	43.6	26.7277	0.7388	0.00	0.0928	0.1524
Sinkhorn	MI	53.2	0.3837	-0.2698	1.0	0.4362	0.2886
MI-NNGP1	MI	3.5	0.3296	**-0.0354**	0.76	0.1764	0.3166
MI-NNGP1-BS	MI	3.6	0.4545	-0.0546	0.99	0.3324	0.2466
MI-NNGP2	MI	4.1	**0.2360**	0.0835	0.82	0.2301	0.3327
MI-NNGP2-BS	MI	10.3	**0.2501**	**-0.0449**	**0.94**	0.3183	0.3043
Complete data	-	-	-	0.0156	0.95	0.0978	0.0938
Complete case	-	-	-	0.1663	0.89	0.4038	0.4639
ColMean Imp	SI	-	0.3793	-0.1574	1.0	0.4559	0.3087

Table 10: Exponential data with $n = 200$ and $p = 1001$ under MNAR. Approximately **40%** features and **92%** cases contain missing values. Here Linear RR and MICE are not included due to running out of RAM.

Models	Style	Time(s)	Imp MSE	Imp accu	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	21.8	0.0431	0.3331	-0.2793	0.64	0.2156	0.2263
GAIN	SI	98.9	0.8942	0.3331	0.8610	0.24	0.1588	0.7651
MIWAE	SI	57.7	0.1267	0.6785	0.6658	0.04	0.1617	0.2030
Sinkhorn	MI	41.6	0.1076	0.3278	0.4201	0.73	0.3036	0.2883
MI-NNGP1	MI	4.7	**0.0116**	0.6463	**-0.0147**	**0.94**	0.1492	0.1495
MI-NNGP1-BS	MI	3.4	**0.0145**	0.6188	**-0.0115**	**0.98**	0.1771	0.1436
MI-NNGP2	MI	3.9	**0.0089**	**0.7289**	**0.0126**	0.99	0.1470	0.1247
MI-NNGP2-BS	MI	9.5	**0.0093**	**0.7006**	**-0.0014**	**0.98**	0.1556	0.1258
Complete data	-	-	-	-	0.0156	0.96	0.1119	0.1041
Complete case	-	-	-	-	0.3856	0.70	0.2846	0.2937
ColMean Imp	SI	-	0.1255	0.3278	0.4643	0.71	0.3000	0.2362

Table 11: Gaussian and binary data with $n = 200$ and $p = 1002$ under MAR. Approximately **40%** features and **88%** cases contain missing values. Linear RR and MICE are not included due to running out of RAM. Detailed simulation setup information is in appendix.

decreasing order of correlation. Denote the selected features by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_{10000})$. In the analysis model, the first three features are chosen as predictors and our goal is to fit the regression model $\mathbb{E}[\mathbf{y}|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3] = \beta_0 + \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \beta_3\mathbf{x}_3$ and analyze the first coefficient β_1.

There are no missing values in the original data, so we artificially introduce some missing values, which are separately created in two groups: $\{\mathbf{x}_1, \ldots, \mathbf{x}_{1000}\}$ and $\{\mathbf{x}_{1001}, \ldots, \mathbf{x}_{2000}\}$ by the following logit models for the corresponding missing indicator $\mathbf{R}_1$ and $\mathbf{R}_2$. If the

Models	Style	Time(s)	Imp MSE	Imp accu	Bias($\hat{\beta}_1$)	CR($\hat{\beta}_1$)	SE($\hat{\beta}_1$)	SD($\hat{\beta}_1$)
SoftImpute	SI	24.7	0.0432	0.3328	-0.2771	0.64	0.2154	0.2263
GAIN	SI	95.3	0.8517	0.3328	0.9779	0.12	0.1605	0.7701
MIWAE	SI	57.7	0.1258	0.6706	0.6665	0.15	0.1604	0.2274
not-MIWAE	SI	42.7	1.9305	0.3317	0.6158	0.06	0.0851	0.1876
Sinkhorn	MI	43.6	0.1080	0.3273	0.4175	0.72	0.3033	0.2882
MI-NNGP1	MI	3.6	**0.0117**	0.6477	**-0.0137**	**0.94**	0.1492	0.1514
MI-NNGP1-BS	MI	3.7	**0.0146**	0.6201	**-0.0154**	**0.98**	0.1785	0.1423
MI-NNGP2	MI	4.2	**0.0089**	**0.7277**	**0.0125**	0.99	0.1469	0.1238
MI-NNGP2-BS	MI	10.0	**0.0093**	**0.6971**	**-0.0044**	**0.98**	0.1558	0.1243
Complete data	-	-	-	-	0.0156	0.96	0.1119	0.1041
Complete case	-	-	-	0.3909	0.70	-	0.2854	0.3001
ColMean Imp	SI	-	0.1259	0.3273	0.4620	0.71	0.2996	0.2367

Table 12: Gaussian and binary data with $n = 200$ and $p = 1002$ under MNAR. Approximately **40%** features and **88%** cases contain missing values. Here Linear RR and MICE are not included due to running out of RAM.

missing mechanism is **MAR**:

$$\text{logit}(\mathbb{P}(\mathbf{R}_1 = 1)) = -1 - \frac{3}{100}\sum_{j=2001}^{2100}\mathbf{x}_j + 3\mathbf{y}$$

$$\text{logit}(\mathbb{P}(\mathbf{R}_2 = 1)) = -1 - \frac{3}{100}\sum_{j=2201}^{2300}\mathbf{x}_j + 2\mathbf{y}$$

If the missing mechanism is **MNAR**:

$$\text{logit}(\mathbb{P}(\mathbf{R}_1 = 1)) = -1 - \frac{3}{5}\sum_{j=1001}^{1005}\mathbf{x}_j + 3\mathbf{y}$$

$$\text{logit}(\mathbb{P}(\mathbf{R}_2 = 1)]) = -1 - \frac{3}{5}\sum_{j=1}^{5}\mathbf{x}_j + 2\mathbf{y}$$

We repeat the above procedure for 100 times to generate 100 incomplete datasets. Each incomplete dataset only differs in location of missing values and therefore they are not Monte Carlo datasets (which is the reason that we do not include the SD($\hat{\beta}_1$) in this experiment). We impute the incomplete datasets and present the summarized results.

Models	Style	Time(s)	Imp MSE	$\hat{\beta}_1$	SE($\hat{\beta}_1$)
SoftImpute	SI	991.5	**0.0613**	0.0212	0.0114
Sinkhorn	MI	709.8	0.0866	0.0216	0.0123
MI-NNGP1	MI	7.4	**0.0644**	**0.0155**	0.0101
MI-NNGP1-BS	MI	7.7	**0.0688**	**0.0162**	0.0112
MI-NNGP2	MI	11.6	**0.0622**	**0.0145**	0.0103
MI-NNGP2-BS	MI	18.5	**0.0609**	0.0123	0.0125
Complete data	-	-	-	0.0160	0.0085
Complete case	-	-	-	0.0202	0.0172
ColMean Imp	SI	-	0.1685	0.01776	0.0130

Table 13: Real data experiment with $n = 649$ and $p = 10001$ under MNAR. Approximately **20%** features and **74%** cases contain missing values. Linear RR, MICE, not-MIWAE and GAIN are not included due to running out of RAM.