JOCO São Paulo 2025

IAA AAI SECTIONS

IBA INSTITUTO BRASILEIRO DE ATUÁRIA

Explainable Least Square Monte Carlo for Solvency Capital Requirement Evaluation

Salvatore Scognamiglio

University of Naples/ Senior Assistant Professor

ABOUT THE SPEAKER

Salvatore Scognamiglio – *Senior Assistant Professor in Financial and Actuarial Mathematics.*

- **Department of Management and Quantitative Studies (DISAQ)**
- **University of Naples Parthenope**

SOLVENCY II

The European Directive 2009/138 changes the management style of insurance undertakings, changes the logic of the evaluation process of the fundamental measures and requires insurance undertakings to evaluate the values and risks in ”market consistent way”.

Some measures gained prominence:

- Net Asset Value (*NAV*),
- Probability Distribution Forecast (*PDF*),
- Solvency Capital Requirement (*SCR*).

SOLVENCY II

The European Directive 2009/138 changes the management style of insurance undertakings, changes the logic of the evaluation process of the fundamental measures and requires insurance undertakings to evaluate the values and risks in ”market consistent way”.

Some measures gained prominence:

- Net Asset Value (*NAV*),
- Probability Distribution Forecast (*PDF*),
- Solvency Capital Requirement (*SCR*).

To evaluate these measures according to the Solvency II principles could be Complex.

Let $(\Omega, F, (F_t)_t \in [0, T], \mathbb{P})$be a filtered probability space, and $\left(B_{0,t}\right)_{t\in[0,T]}$ be the risk-free asset, such that $B_{0,t} = e^{\int_0^t r_u du}$.

The Net Asset Value of an insurance company at time t ∈ [0, T], denoted as NAV_t, is defined as:

$$NAV_t = V(t, \boldsymbol{A}) - V(t, \boldsymbol{L}).$$

where:

$V(t, \boldsymbol{A})$ is the market-consistent value of the assets $\boldsymbol{A} = \{A_t, t \in [0, T]\}$;

$V(t, \boldsymbol{L})$ is the market-consistent value of the assets $\boldsymbol{L} = \{L_t, t \in [0, T]\}$.

The cash flows **A** and **L** depend on some risk drivers denoted as $\boldsymbol{X} = \{\boldsymbol{X_t} \in \mathbb{R}^{q_0}, t \in [0, T]\}$.

SOLVENCY CAPITAL REQUIREMENT

The Solvency Capital Requirement (SCR) determines the amount of capital ensuring that an undertaking will be able to meet its obligations over 1 year with a probability of 99.5 %.

It can be mathematically formalized as:

$$SCR_{0.995} = (VaR_{0.995}(NAV_1) - \mathbb{E}[NAV_1])v(0,1)$$

where $v(0,1)$ is the price of a one-year ZCB, and $VaR_\tau(NAV_1)$ is:

$$VaR_\tau(NAV_1) = \inf\{ x \in \mathbb{R} : F_{NAV}(x) \geq \tau\}$$

At the security level $\tau \in]0,1[$.

The SCR calculation involves F_{NAV}^1 that is generally unknown.

[1] that is called the Probability Distribution Forecast (PDF) in the Directive (art. 13).

SOLVENCY CAPITAL REQUIREMENT EVALUTATION

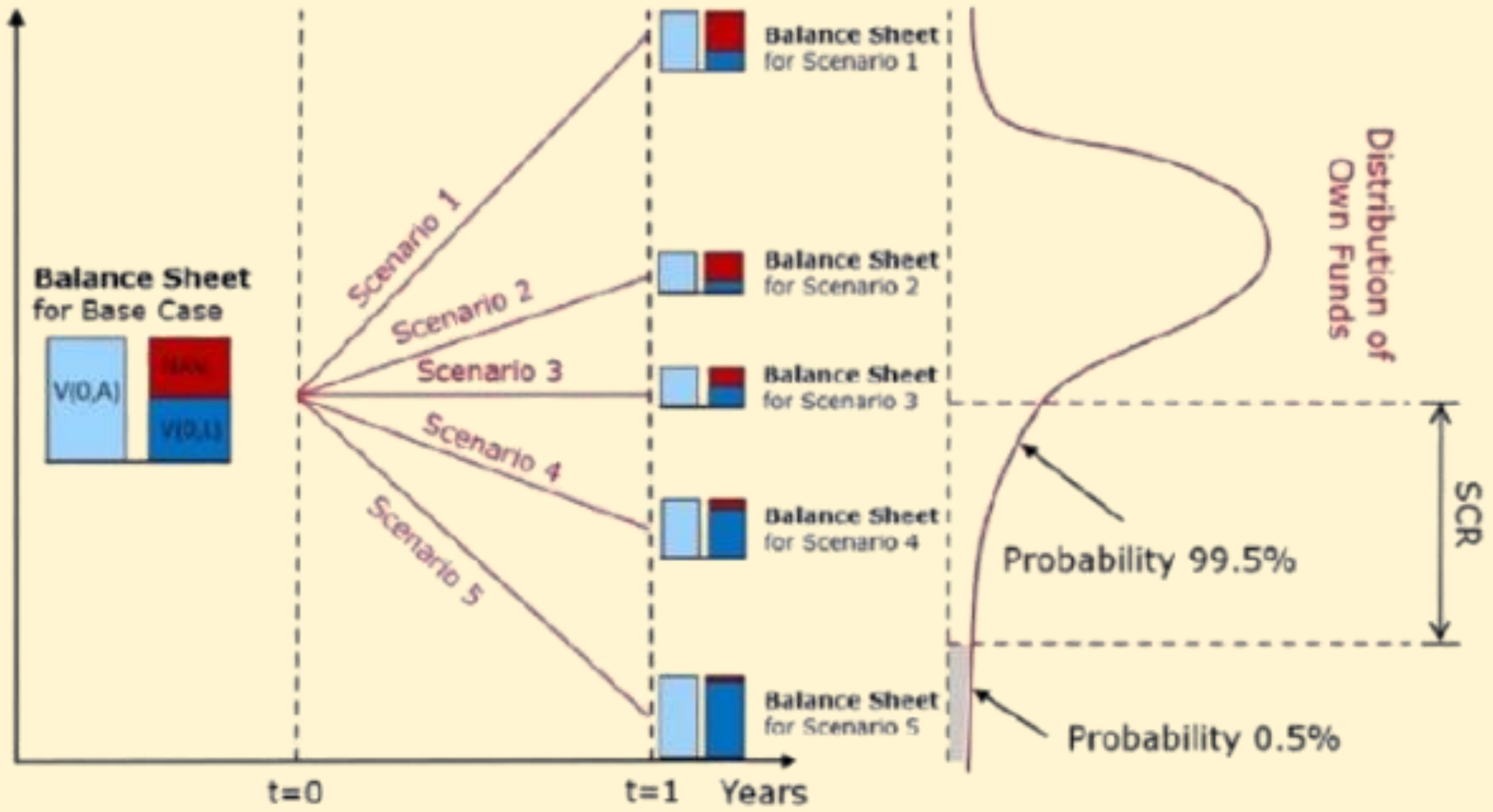


Figure: Solvency Capital Requirement Evaluation.

Source: Jonen, C., Meyhofer, T., & Nikolic, Z. (2023). Neural networks meet least squares Monte Carlo at internal model data. European Actuarial Journal, 13(1), 399-425.

THE NESTED SIMULATION APPROACH

Bauer et al. (2013) suggests a two-step procedure:

1. Simulating under the real-world measure $\mathbb{P}$, sample paths $\left(\boldsymbol{X}_t^{(i)}\right)_{t\in[0,1]}, i = 1, \dots, n_{\mathbb{P}}$

THE NESTED SIMULATION APPROACH

Bauer et al. (2013) suggests a two-step procedure:

1. Simulating under the real-world measure $\mathbb{P}$, sample paths $\left(\boldsymbol{X}_t^{(i)}\right)_{t\in[0,1]}, i = 1, \dots, n_{\mathbb{P}}$
2. Evaluating the NAV1 in each scenario as:

$$NAV_1^{(i)} = V^{(i)}(1, \boldsymbol{A}) - V^{(i)}(1, \boldsymbol{L}) \qquad i = 1, \dots, n_{\mathbb{P}}$$

THE NESTED SIMULATION APPROACH

Bauer et al. (2013) suggests a two-step procedure:

1. Simulating under the real-world measure $\mathbb{P}$, sample paths $\left(\boldsymbol{X}_t^{(i)}\right)_{t\in[0,1]}, i = 1, \dots, n_{\mathbb{P}}$
2. Evaluating the NAV1 in each scenario as:

$$NAV_1^{(i)} = V^{(i)}(1, \boldsymbol{A}) - V^{(i)}(1, \boldsymbol{L}) \qquad i = 1, \dots, n_{\mathbb{P}}$$

However $v^{(i)}(1, \boldsymbol{L}) = \mathbb{E}^{\mathbb{Q}}[\sum_{t=2}^{T} \frac{L_t}{B_{1,t}} \mid \boldsymbol{X}_1^{(i)}]$ required to be computed numerically by simulating, under the risk-neutral measure $\mathbb{Q}$, sample paths $\left(\boldsymbol{X}_t^{(j)}\right)_{t\in[1,T]}, j = 1, \dots, n_{\mathbb{Q}}$ and evaluating:

$$\hat{V}_{n_{\mathbb{Q}}}^{(i)}(1, \boldsymbol{L}) = \frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}}, i = 1, \dots, n_{\mathbb{P}}$$

THE NESTED SIMULATION APPROACH

Bauer et al. (2013) suggests a two-step procedure:

1. Simulating under the real-world measure $\mathbb{P}$, sample paths $\left(\boldsymbol{X}_t^{(i)}\right)_{t\in[0,1]}, i = 1, \ldots, n_{\mathbb{P}}$
2. Evaluating the NAV1 in each scenario as:

$$NAV_1^{(i)} = V^{(i)}(1, \boldsymbol{A}) - V^{(i)}(1, \boldsymbol{L}) \qquad i = 1, \ldots, n_{\mathbb{P}}$$

However $v^{(i)}(1, \boldsymbol{L}) = \mathbb{E}^{\mathbb{Q}}[\sum_{t=2}^{T} \frac{L_t}{B_{1,t}} \mid \boldsymbol{X}_1^{(i)}]$ required to be computed numerically by simulating, under the risk-neutral measure $\mathbb{Q}$, sample paths $\left(\boldsymbol{X}_t^{(j)}\right)_{t\in[1,T]}, j = 1, \ldots, n_{\mathbb{Q}}$ and evaluating:

$$\hat{V}_{n\mathbb{Q}}^{(i)}(1, \boldsymbol{L}) = \frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}}, i = 1, \ldots, n_{\mathbb{P}}$$

Nested Simulations! The computational cost is proportional to $n_{\mathbb{P}} \times n_{\mathbb{Q}}$

COMPUTATIONAL COST: AN EXAMPLE

For example, if we consider:

1. $n_{\mathbb{P}} = 100000$;
2. $n_{\mathbb{Q}} = 100$;

The computational time of the procedure for the SCR calculation is:

COMPUTATIONAL COST: AN EXAMPLE

For example, if we consider:

1. $n_{\mathbb{P}} = 100000$;
2. $n_{\mathbb{Q}} = 100$;

The computational time of the procedure for the SCR calculation is:

$$100000 \times 100 \times 1\, sec \approx 115\, days.$$

LEAST SQUARE MONTE CARLO (LONGSTAFF AND SCHWARTZ, 2001)

If the conditional expectation function belongs to the L^2-space, it can be expressed as

$$V^{(i)}(1, \boldsymbol{L}) = \mu(\boldsymbol{X}_1) = \sum_{k=0}^{\infty} \beta^{(k)} \psi^{(k)}(\boldsymbol{X}_1)$$

here $\{\psi^{(k)}(\cdot), k = 1, \ldots, \infty\}$ form an orthonormal basis of L^2 and $\{\beta^{(k)}(\cdot), k = 1, \ldots, \infty\}$ are some coefficients. An approximation can be obtained by considering a finite set of *K* basis

$$\hat{\mu}^{(OP)}(\boldsymbol{X}_1) = \sum_{k=0}^{K} \beta^{(k)} \psi^{(k)}(\boldsymbol{X}_1)$$

and estimating the parameters by solving

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta \in \mathbb{R}^k} \sum_{i=1}^{n_{\mathbb{P}}} \left[\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \sum_{k=0}^{\infty} \beta^{(k)} \psi^{(k)}(\boldsymbol{X}_1) \right]^2$$

Nested Simulations Vs Least Squares Monte Carlo

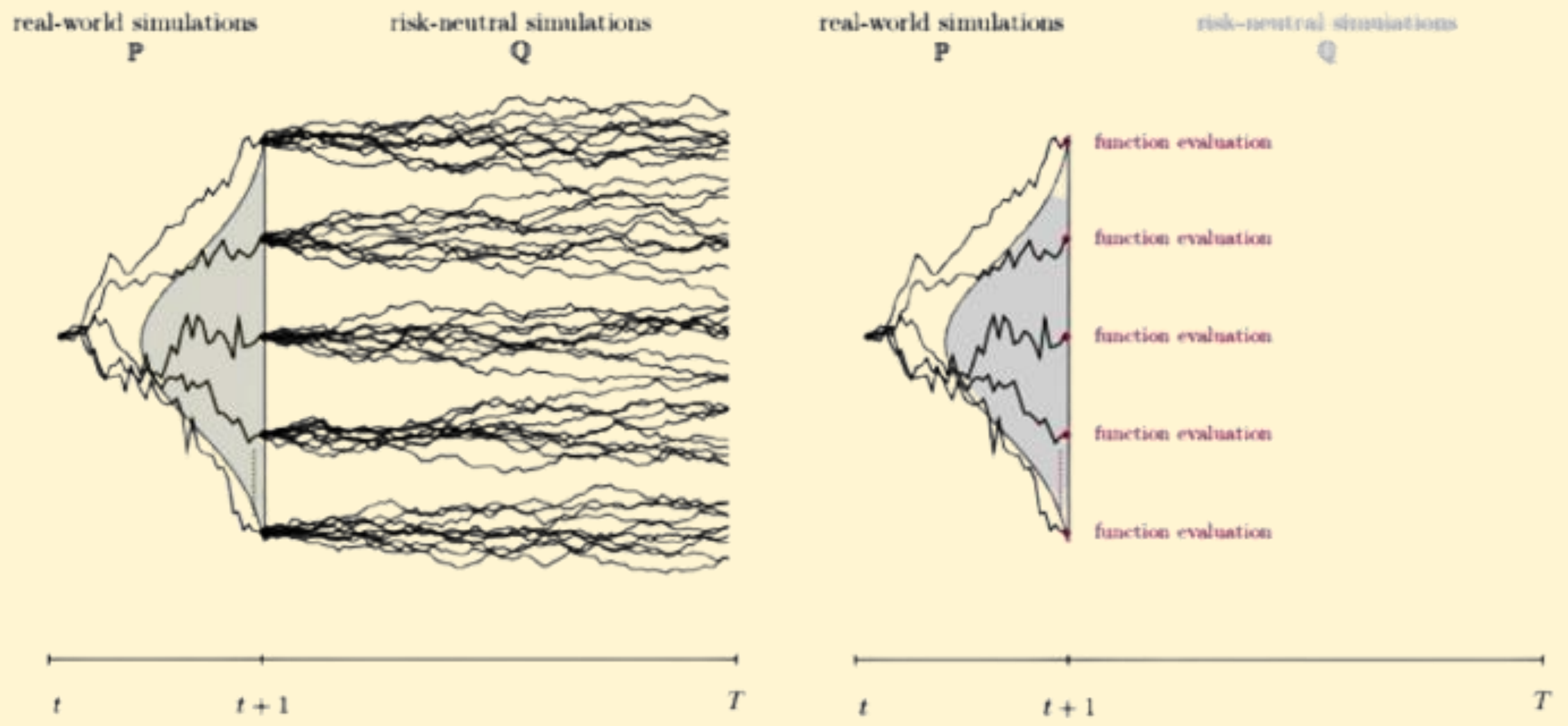


Figure: Graphical representation of the Nested simulations and the Least Square Monte Carlo approaches.

LEAST SQUARES MONTE CARLO AND THE CURSE OF DIMENSIONALITY

The number of terms in the polynomial regression grows with the number of the risk drivers and the maximum degree of the polynomials m:

$$\binom{m+q_0}{m}$$

Table: Number of tems for polynomial given m and q_0

m	q_0							
	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	5	9	14	20	27	35	44
3	3	9	19	34	55	83	119	164
4	4	14	34	69	125	209	329	494
5	5	20	55	125	251	461	791	1286
6	6	27	83	209	461	923	1715	3002
7	7	35	119	329	791	1715	3431	6434
8	8	44	164	494	1286	3002	6434	12869
9	9	54	219	714	2001	5004	11439	24309

NEURAL NETWORKS

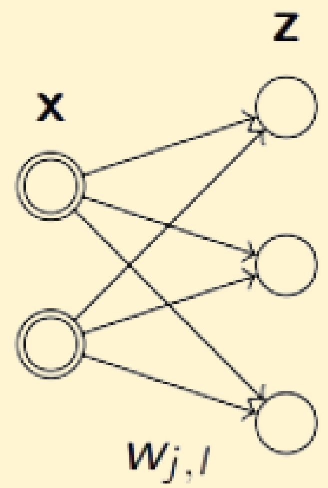
Let be $\boldsymbol{x} \in \mathbb{R}^{q_0}$ the vector of features, a fully connected (FC) layer of size $q_1 \in \mathbb{N}$ is a function

$$\boldsymbol{z}: \mathbb{R}^{q_0} \to \mathbb{R}^{q_1}, \qquad \boldsymbol{x} \mapsto \boldsymbol{z}(\boldsymbol{x}) = \left(z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), \dots, z_{q_1}(\boldsymbol{x})\right)^T$$

Each component $z_j(x)$ is a non-linear function of $\boldsymbol{x}$

$$\boldsymbol{x} \mapsto \boldsymbol{z_j}(\boldsymbol{x}) = \phi\left(w_{j,0} + \sum_{l=1}^{q_0} w_{j,I} x_l\right) = \phi\left(w_{j,0} + < \boldsymbol{w}_j, \boldsymbol{x} >\right), \qquad j = 1, \dots, q_1,$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function, $w_{j,l} \in \mathbb{R}$ represent the network parameters and $< , >$ denotes the scalar product in $\mathbb{R}^{q_0}$.



DEEP NEURAL NETWORKS

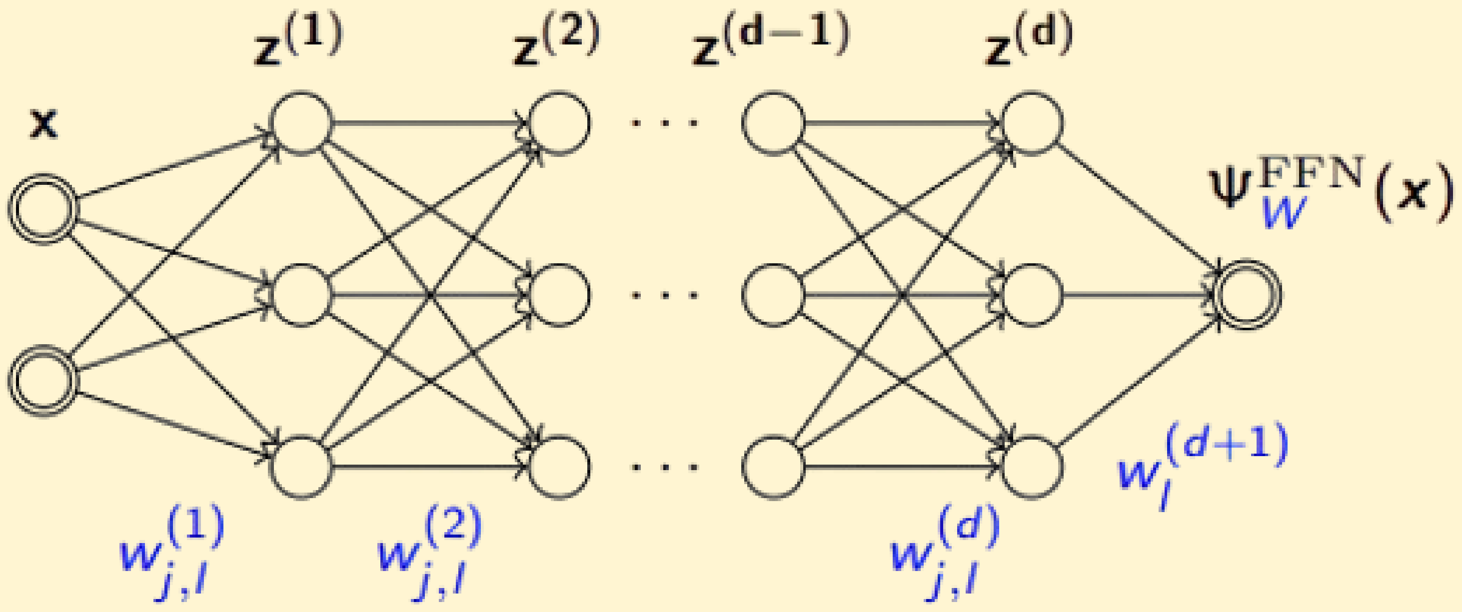
In the case of d layers of size $q = \{q_k\}_{1 \le k \le d} \in \mathbb{N}^d$, the mapping reads:

$$x \mapsto z^{(d:1)} \stackrel{\text{def}}{=} \left(z^{(d)} \circ \ldots \circ z^{(1)}\right)(x) \in \mathbb{R}^{q_d}$$

Where $z^{(k)}: \mathbb{R}^{q_{k-1}} \to \mathbb{R}^{q_k}$. In the case of univariate response, the output of the network is:

$$x \mapsto \mu_w(x) \stackrel{\text{def}}{=} \psi_W^{FFN}(x) \stackrel{\text{def}}{=} g^{-1}\left(w_0^{(d+1)} + \sum_{l=1}^{q_d} w_l^{(d+1)} z_l^{(d:1)}(x)\right),$$

g^{-1} is an inverse link function.

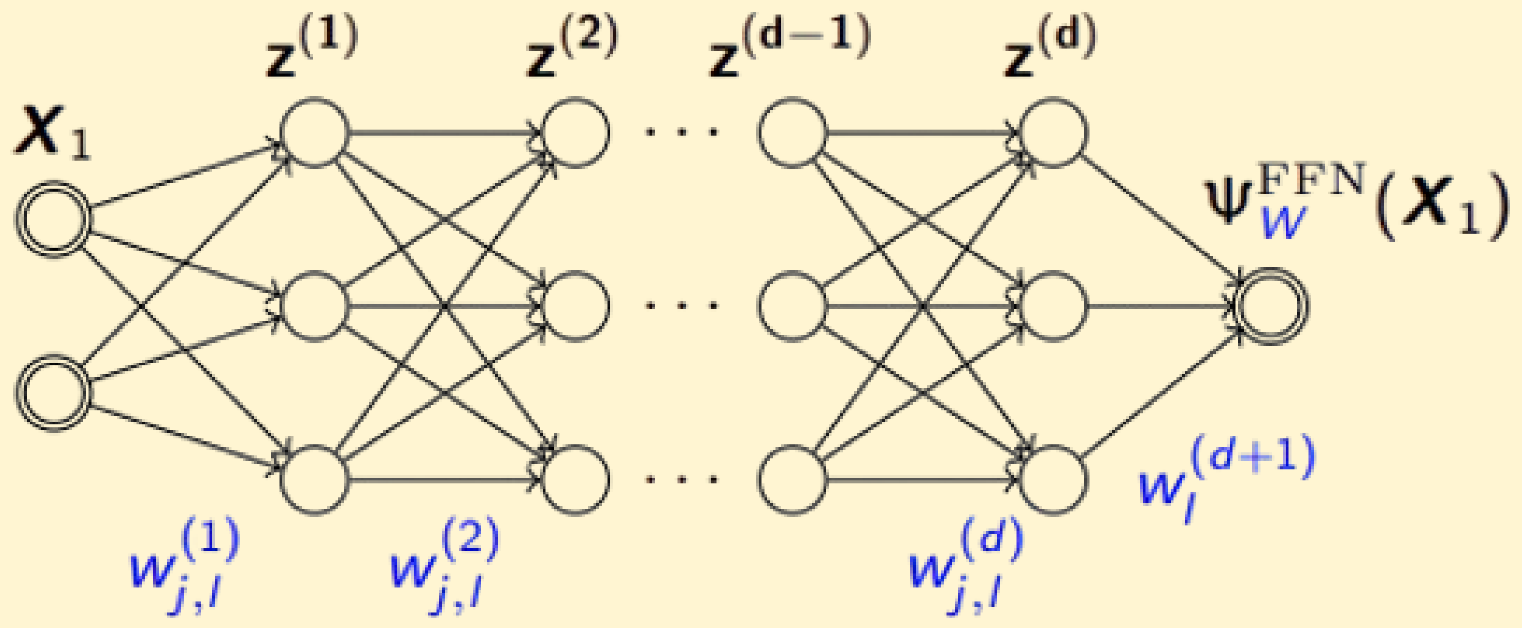


THE LSMC-DL METHOD

Since our aim consists of approximating a conditional expectation function we use the MSE as loss function. The training of the network requires the optimisation:

$$\widehat{W}_{n_{\mathbb{Q}}} = \underset{W \in \mathbb{R}^M}{argmin} \sum_{i=1}^{n_{\mathbb{P}}} \left[\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \psi_W^{FFN}(\boldsymbol{X_1}) \right]^2$$

where W is the vector of the neural network parameters.

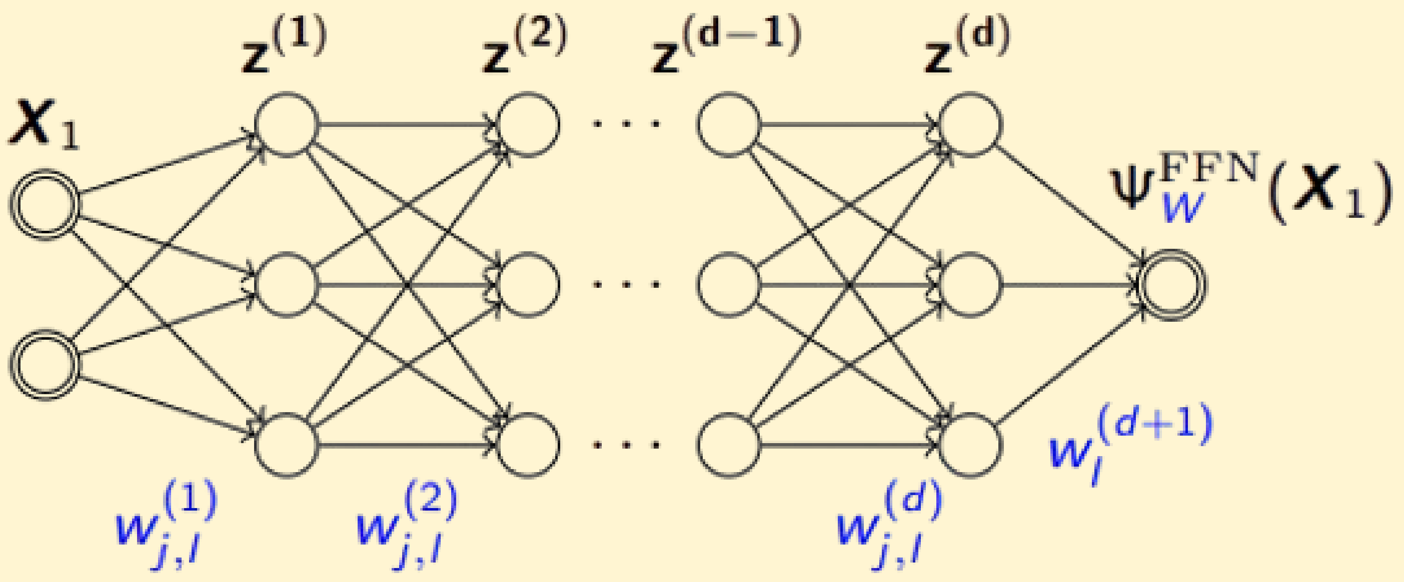


THE LSMC-DL METHOD

Since our aim consists of approximating a conditional expectation function we use the MSE as loss function. The training of the network requires the optimisation:

$$\widehat{W}_{n_{\mathbb{Q}}} = \underset{W \in \mathbb{R}^M}{argmin} \sum_{i=1}^{n_{\mathbb{P}}} \left[\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \psi_W^{FFN}(\boldsymbol{X_1}) \right]^2$$

where W is the vector of the neural network parameters.



However, there is lack of explainability!

THE LOCALGLMNET MODEL OF RICHMAN AND WUTHRICH (2023)

Let ψ_W be a neural network with output dimension equal to the input dimension q_0:

$$\psi_W : \mathbb{R}^{q_0} \to \mathbb{R}^{q_0}, \qquad x \mapsto \psi_W(x),$$

having network weights W . The *LocalGLMnet* regression function is defined by

$$x \mapsto \mu_{W,\beta_0}(x) \stackrel{\text{def}}{=} g^{-1}(\beta_0 + \beta(x)^T X),$$

where $g : \mathbb{R} \to \mathbb{R}$ is the link function, $\beta_0 \in \mathbb{R}$, and $\beta(x) = \psi_W(x)$.

1. If $\beta_j(x) \equiv \beta_j$ is not feature dependent.
2. If $\beta_j(x) \equiv 0$, term $\beta_j(x)x_j$ is dropped altogether.
3. If $\beta_j(x) = \beta_j\left(x_j\right)$, term $\beta_j(x)x_j$ does not interact with any other terms $x_{j'}, j' \neq j$.
4. Interactions can be studied by considering the gradient of $\beta_j(x)$

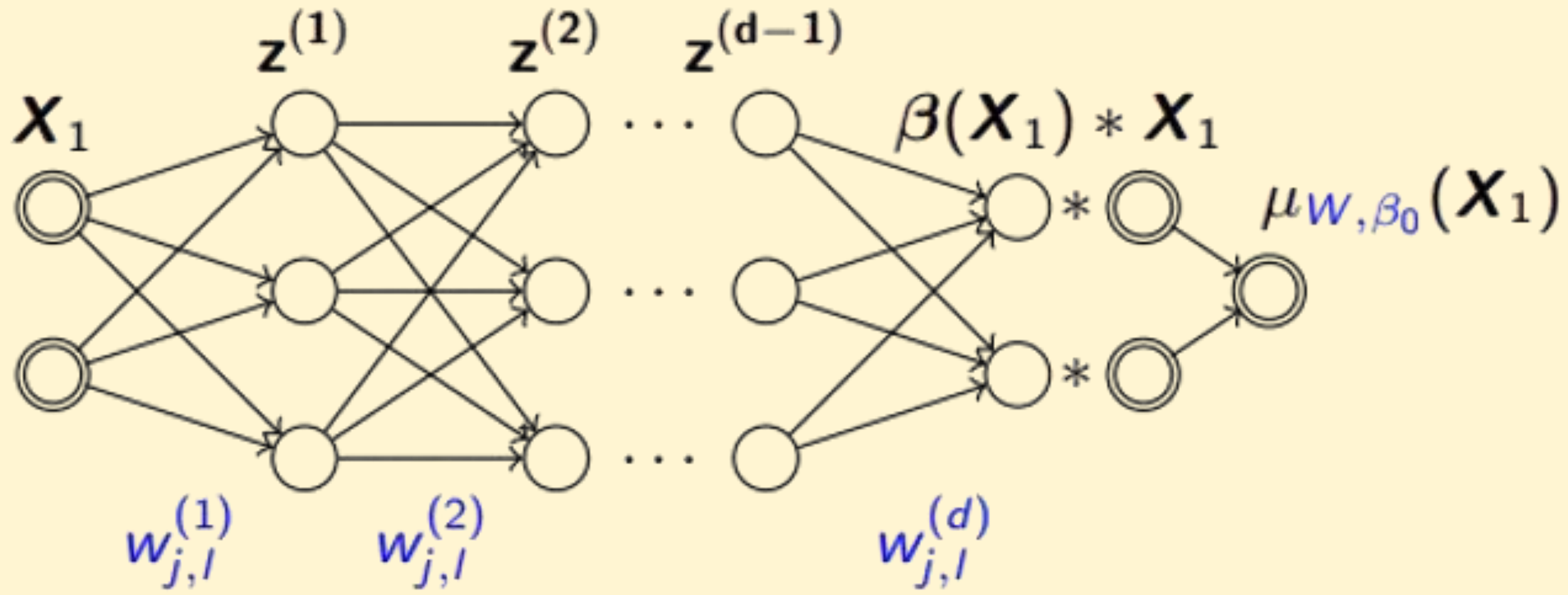
$$\nabla\beta_j(x) = \left(\partial_{x_1}\beta_j(x), \dots, \partial_{x_{q_0}}\beta_j(x)\right)^T \in \mathbb{R}^{q_0}$$

THE LSMC-LGN METHOD

The training of the localGLMnet induces the following optimisation:

$$(\beta_{0,n_{\mathbb{Q}}},\widehat{W}_{n_{\mathbb{Q}}}) = \underset{W\in\mathbb{R}^{M},\beta_0\in\mathbb{R}}{argmin} \sum_{i=1}^{n_{\mathbb{P}}} \left[\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \mu_{W,\beta_0}(\boldsymbol{X_1}) \right]^2$$

where

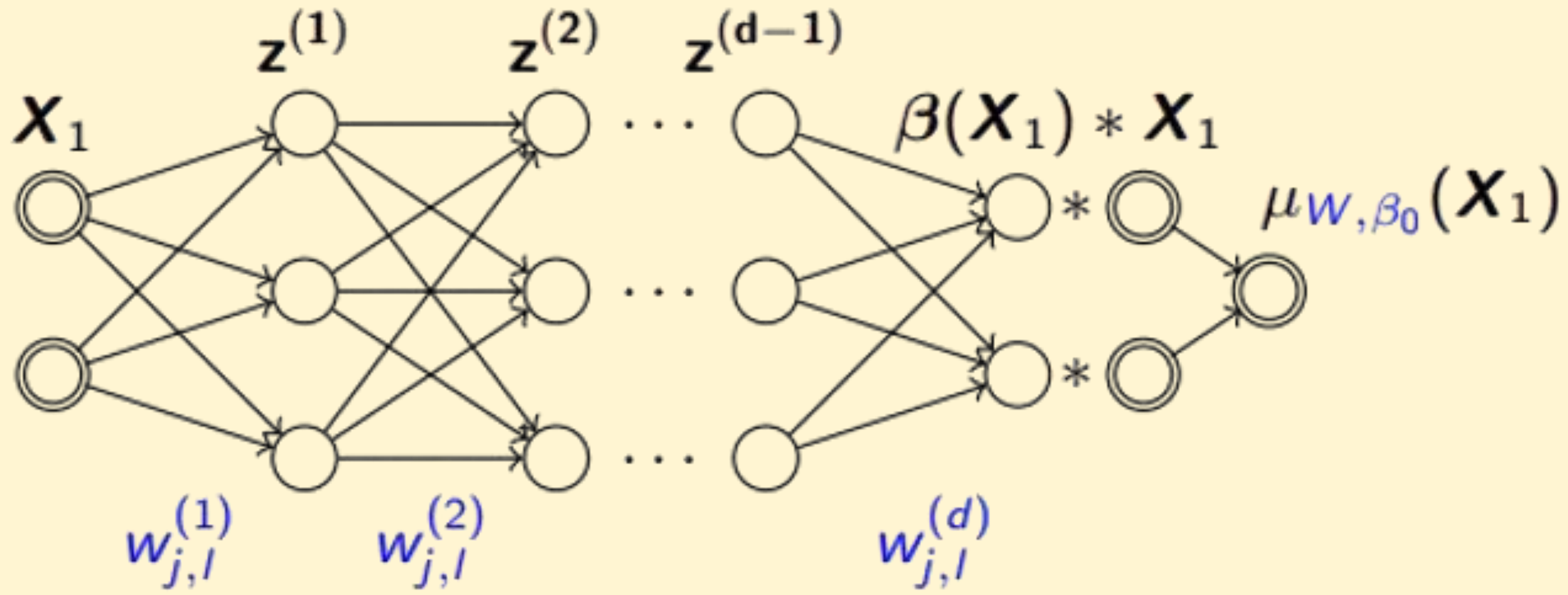


THE LSMC-LGN METHOD

The training of the localGLMnet induces the following optimisation:

$$(\beta_{0,n_{\mathbb{Q}}}, \widehat{W}_{n_{\mathbb{Q}}}) = \underset{w \in \mathbb{R}^M, \beta_0 \in \mathbb{R}}{argmin} \sum_{i=1}^{n_{\mathbb{P}}} \left[\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \mu_{W,\beta_0}(\boldsymbol{X_1}) \right]^2$$

where



Some connections with the local-LSMC proposed by Hainaut & Akbaraly (2023)!

NUMERICAL EXPERIMENTS' SETTING

We consider a simplified insurance portfolio consisting of one with-profit mixed insurance contract affected by 4 risk factors.
Three LSMC-style methods with approximation based on:

- Orthogonal polynomials (LSMC-OP);
- Deep learning (LSMC-DL);
- LocalGLMnet (LSMC-LGN).

We calibrate the methods using data obtained by setting:

- $n_{\mathbb{P}} = 10000$;
- $n_{\mathbb{Q}} = 2^1, 2^2, \dots, 2^{10}$.

The benchmark is the Nested Simulation approach with $n_{\mathbb{P}} = 10000$.

THE ROLE OF $n_{\mathbb{Q}}$ IN THE LSMC METHODS

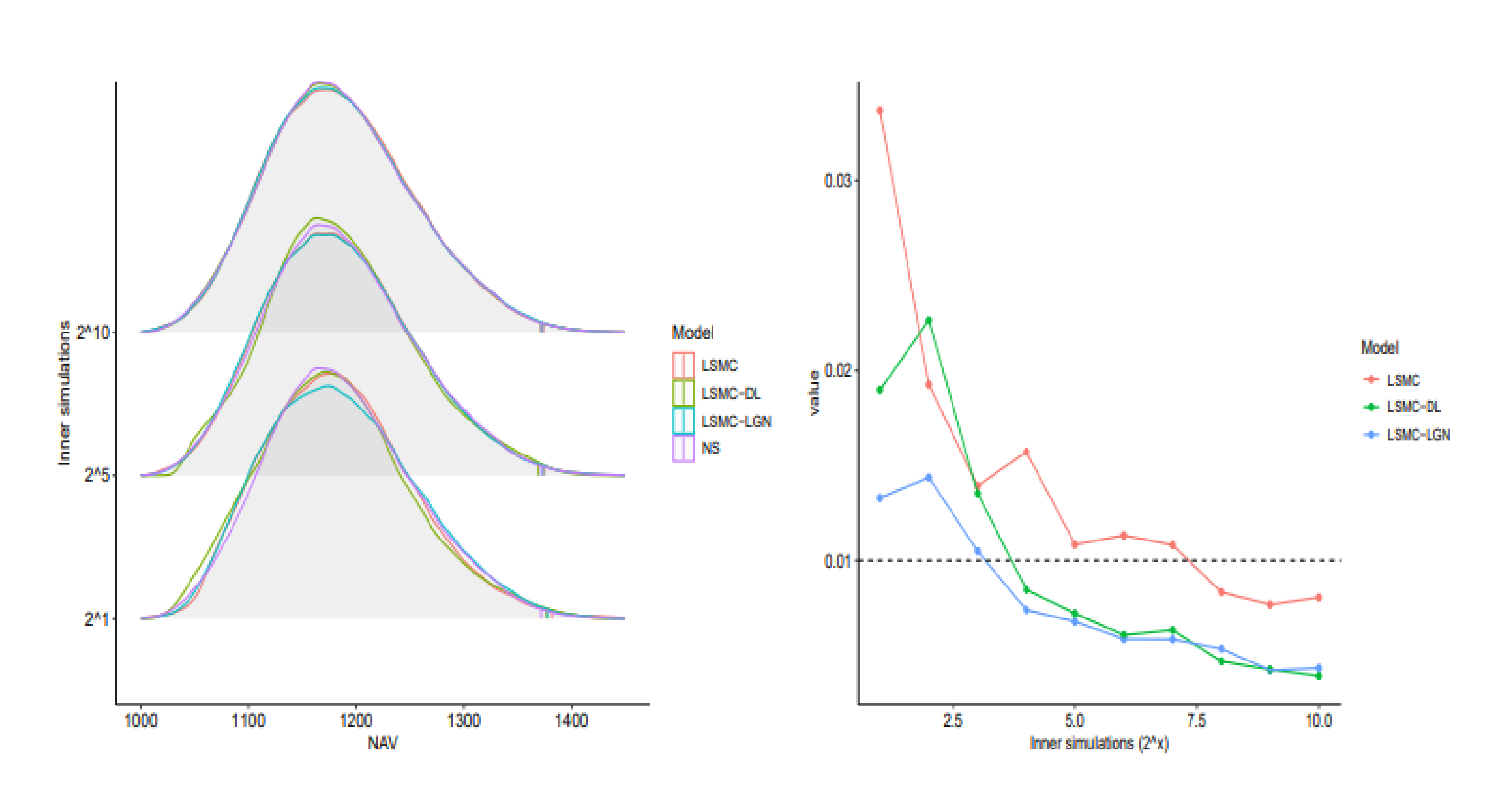


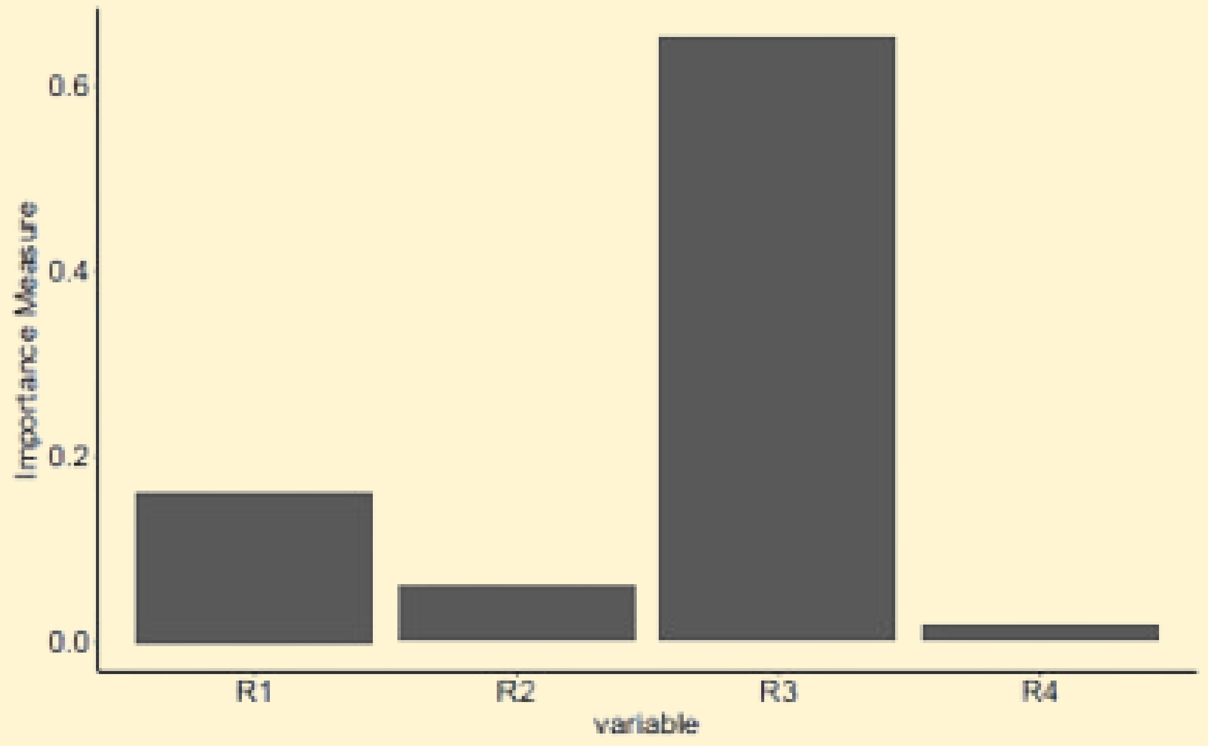
Figure: *Left:* estimated NAV distributions obtained with the different approaches (NS, LSMC-OP, LSMC-DL, LSMC-LGN) for $n_{\mathbb{Q}} \in \{2^1, 2^5, 2^{10}\}$.
Right: Normalised Root Mean Squared Error produced by the LSMC-style methods for $n_{\mathbb{Q}} = 2^I, I = 1,2,\dots,10$.

IMPORTANCE VARIABLE

The estimated regression attention $1 \leq k \leq q_0$, allow us to quantity variable importance. A simple measure can be defined by:

$$VI^{(k)} = \frac{1}{n}\sum_{i} \left| \widehat{\beta_{n_{\mathbb{Q}}}}^{(k)} (\boldsymbol{X}_1^i) \right|$$

- A large $VI^{(k)}$ value suggests that the *k*-th component has a notable effect on the response;
- A small $VI^{(k)}$ value suggests that the *k*-th component has a limited effect on the response.



THE CONTRIBUTION VALUE $\widehat{\beta}_{0,n_{\mathbb{Q}}}(X_1^i)$

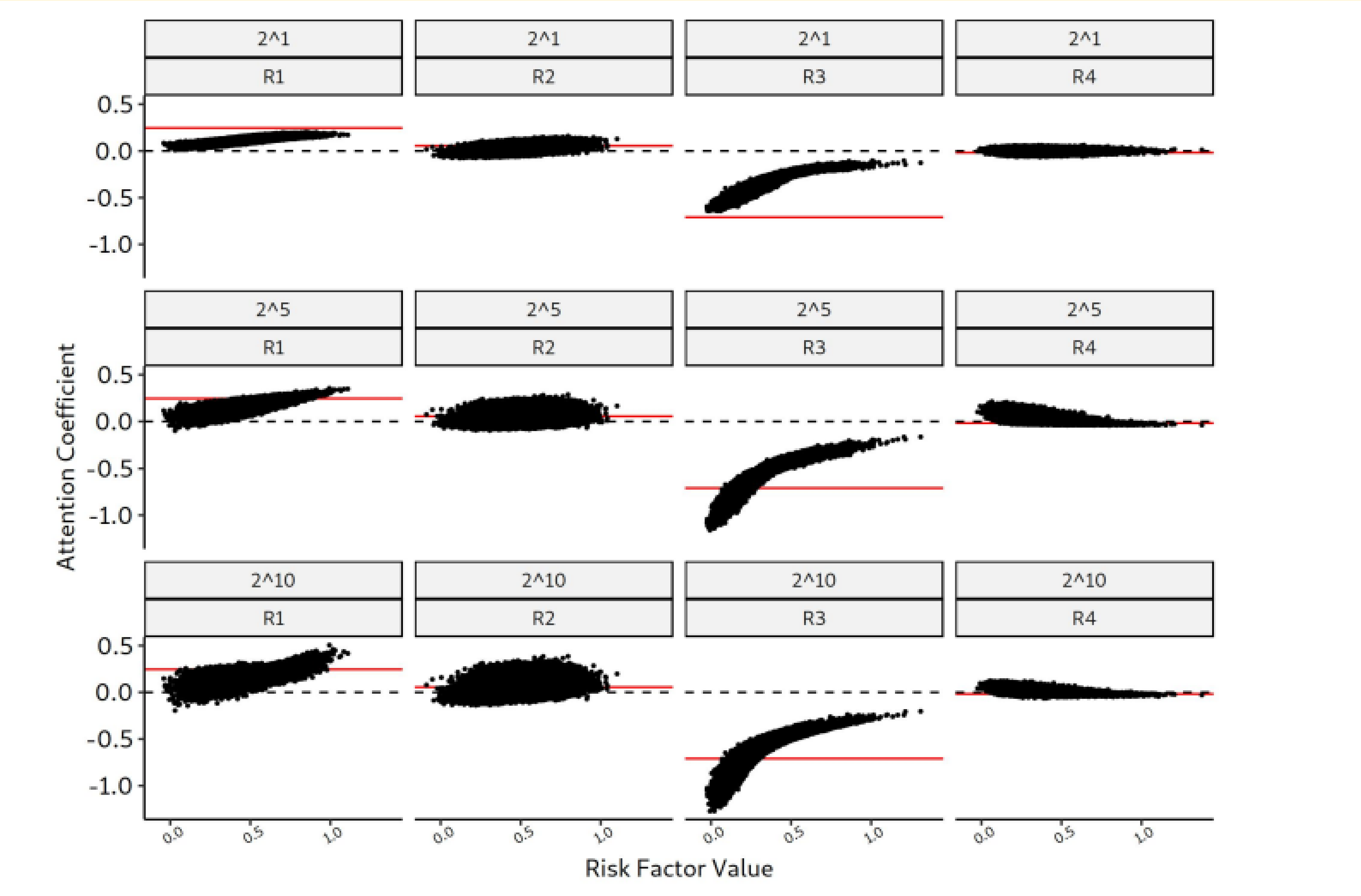


Figure: Attention coefficients $\hat{\beta}_{0,n_{\mathbb{Q}}}(X_1^i), 1 \leq i \leq n_{\mathbb{P}}$, of the LSMC-LGN model.

THE CONTRIBUTION VALUE $\widehat{\beta}_{0,n_{\mathbb{Q}}}(X_1^i)X_{1,k}^{(i)}$

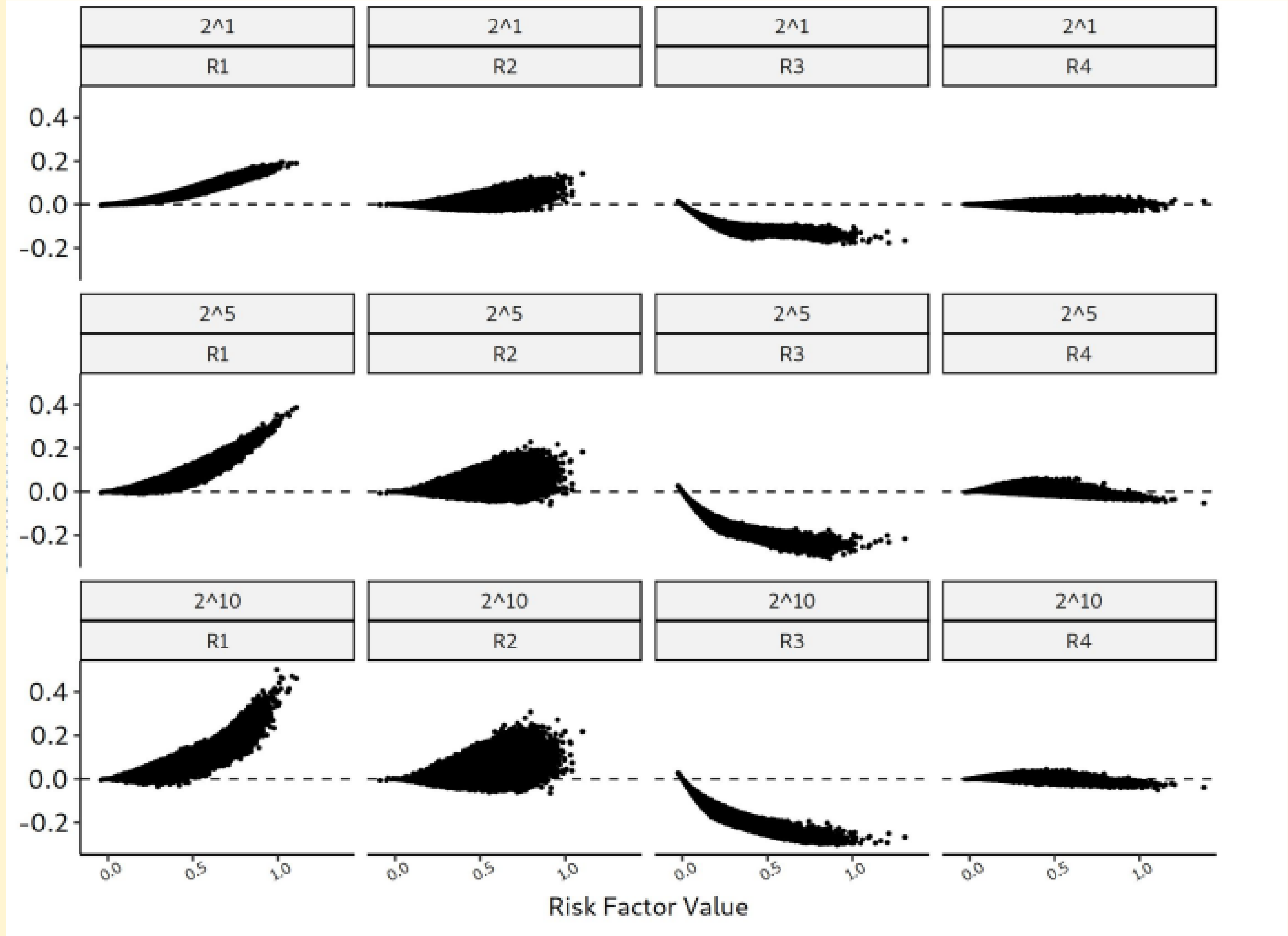


Figure: Attention coefficients $\hat{\beta}_{0,n_{\mathbb{Q}}}(X_1^i)X_{1,k}, 1 \leq i \leq n_{\mathbb{P}}$, of the LSMC-LGN model.

MEASURING THE INTERACTIONS

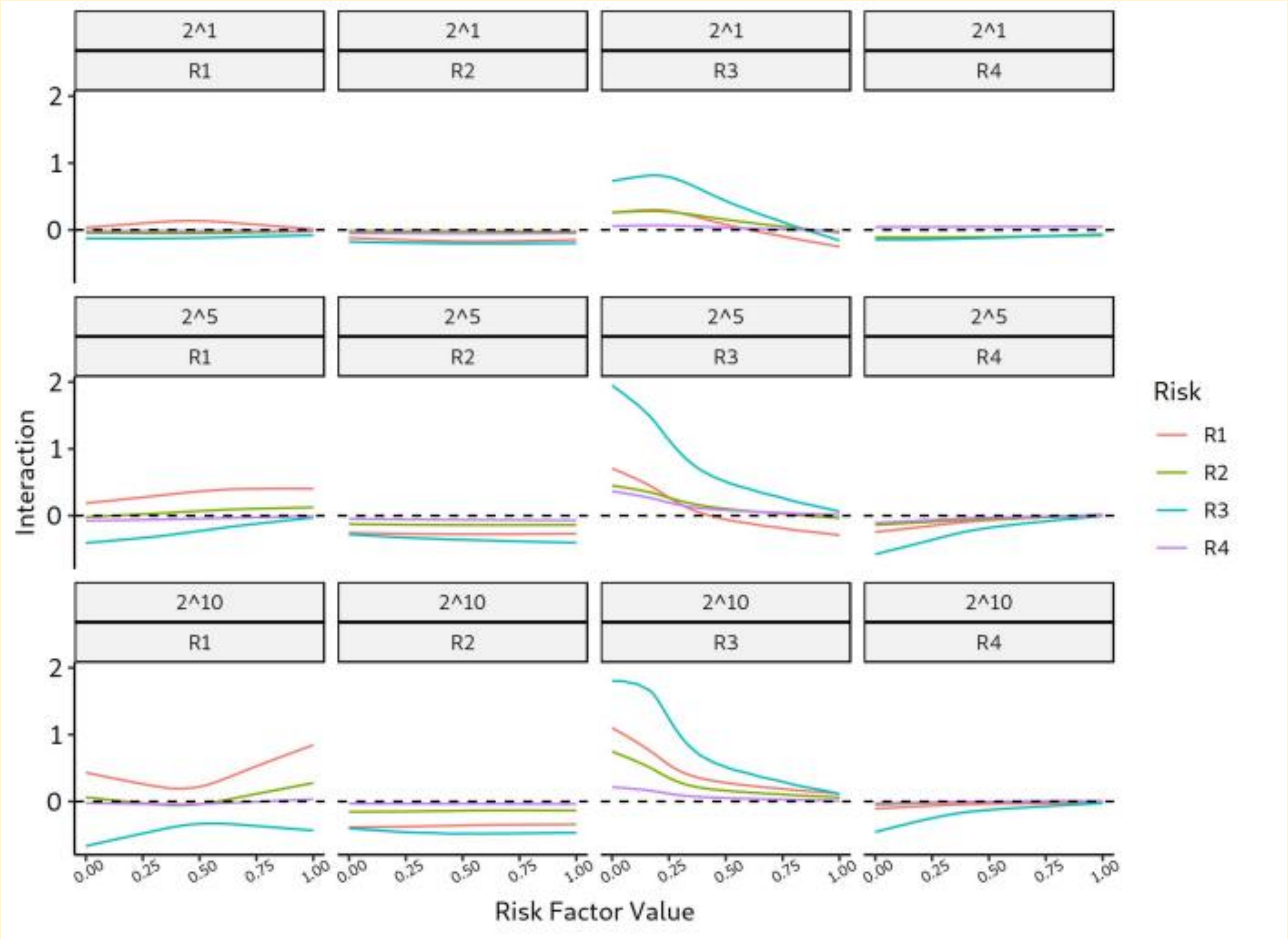


Figure: Spline fits to the sensitivities $\partial_{x_1,u}\hat{\beta}_{0,n_{\mathbb{Q}}}(X_1^i), 1 \leq u, k \leq 4$ over the scenarios , $i = 1, \ldots, n_{\mathbb{P}}$

NUMERICAL EXPERIMENTS' SETTING

We consider a more realistic insurance portfolio consisting of several insurance contracts affected by 23 risk factors.

Two simulated samples:

# outer simulations	# inner simulations	Execution time (hh:mm:ss)
10000	1000	2:13:06
100000	10000	219:30:52

Table: Execution time of nested simulations with different values of $n_{\mathbb{P}}$ and $n_{\mathbb{Q}}$. The values refer to a parallel computing system, consisting of 152 cores.

REGULARISATION WITHIN THE LSMC-LGN MODEL

ElasticNet regularisation could be introduced in the LSMC-LGN to encourage sparsity in the attention coefficients and perform feature selection.

In this case, the network training aims to minimize

$$(\beta_0, \widehat{W}) = \underset{\beta_0, W}{argmin} \sum_{i=1}^{n_{\mathbb{P}}} \left[\left(\frac{1}{n_{\mathbb{Q}}} \sum_{j=1}^{n_{\mathbb{Q}}} \sum_{t=2}^{T} \frac{L_t^{(i,j)}}{B_{1,t}^{(i,j)}} - \beta_0 - \sum_{k=1}^{q_0} \beta_W^{(k)}(x_1) x_{1,k} \right)^2 + \eta \left((1-\alpha) \|\beta_W(\boldsymbol{x}_1)\|_2^2 + \alpha \|\beta_W(\boldsymbol{x}_1)\|_1 \right) \right]$$

- with regularisation parameters $\eta \geq 0$ and $\alpha \in [0, 1]$:
- $\alpha = 0 \rightarrow$ ridge regularisation;
- $\alpha = 1 \rightarrow$ LASSO regularisation.

REGULARISATION WITHIN THE LSMC-LGN MODEL

η	train MSE	validation MSE
0	0.0323	0.0328
1.00E-06	0.0318	0.0331
5.00E-06	0.0330	0.0325
1.00E-05	0.0331	0.0323
5.00E-05	**0.0321**	**0.0322**
1.00E-04	0.0332	0.0338
5.00E-04	0.0343	0.0337
1.00E-03	0.0333	0.0334
5.00E-03	0.0336	0.0338
1.00E-02	0.0340	0.0337
5.00E-02	0.0346	0.0340

Table: MSE values on the training and validation sets of the regularised LSMC-LGN for the different values of η

REGULARISATION WITHIN THE LSMC-LGN MODEL

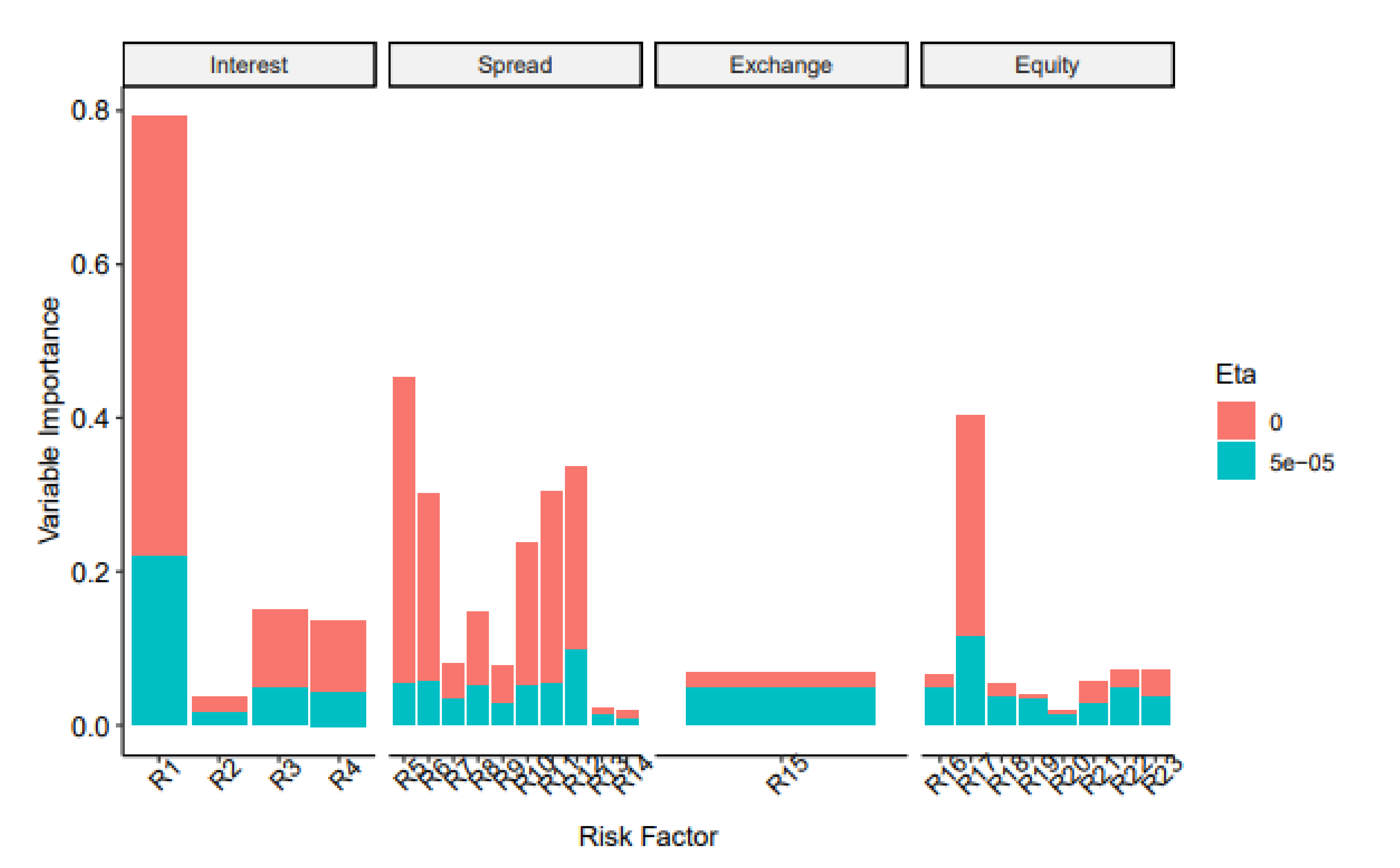


Figure: Importance measures $VI^{(k)}_{n'_{\mathbb{Q}}}(\eta)$ of the different risk factors for $\eta = 0, \hat{\eta}_{opt}$

THE ATTENTION COEFFICIENTS $\widehat{\beta}_{n_{\mathbb{Q}}}(X_1^i)$

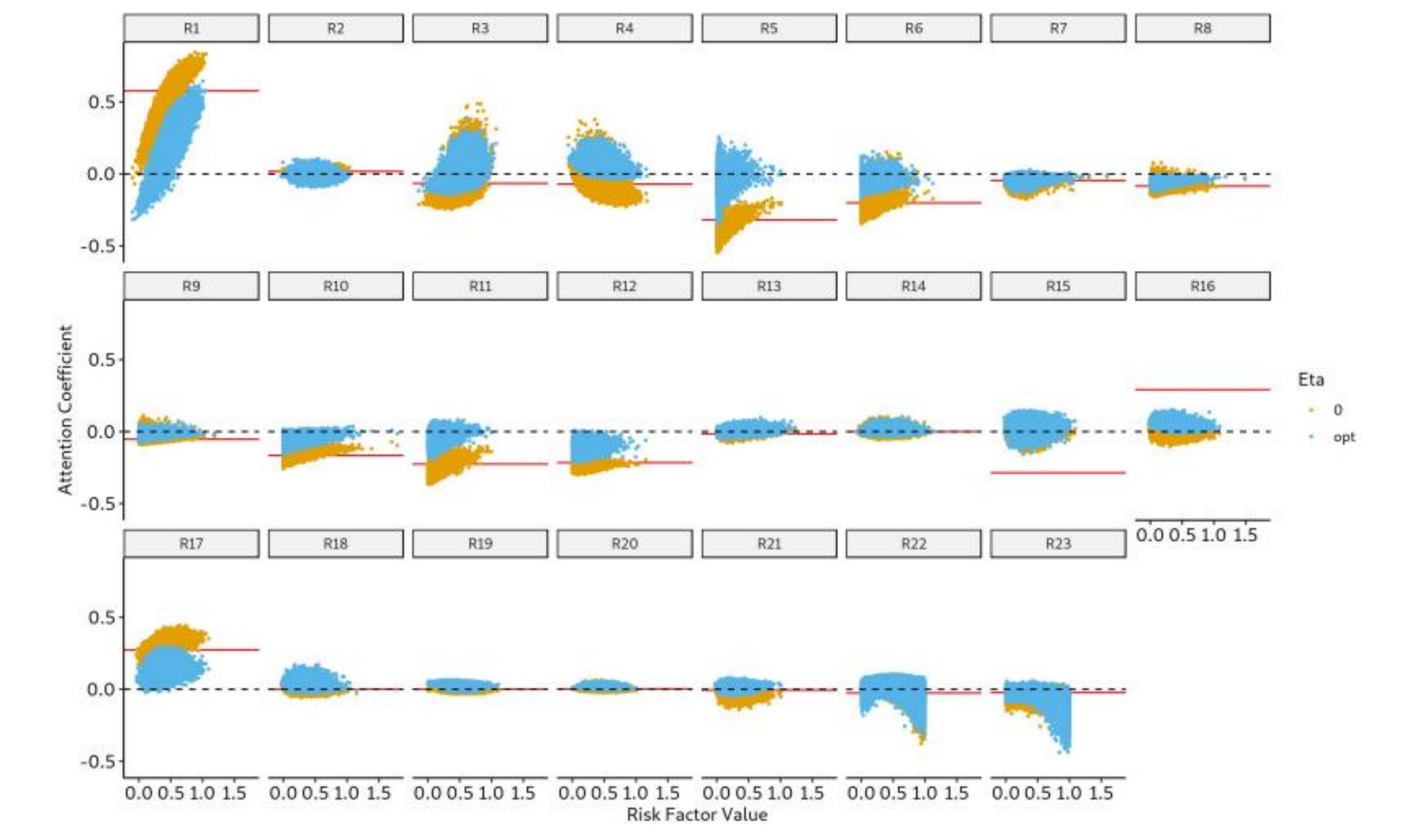


Figure: Attention coefficients $\hat{\beta}\left(X_1^i\right), 1 \leq i \leq n_{\mathbb{P}}$ of the LSMC-LGN model related to the risk factors k = 1, . . . , 23 in the cases of $\eta = 0 \ and \ \eta = \hat{\eta}_{opt}$ The red lines refer to the coefficients of the linear regression model.

RESULTS

Model	NRMSE	RE_{SCR}
LSMC	0.0341	1.4713
LSMC-DL	0.0173	0.8340
$\text{LSMC-LGN}_{\eta=0}$	0.0171	0.8257
$\text{LSMC-LGN}_{\eta=\hat{\eta}_{opt}}$	0.0166	0.7204

Table: Out-of-Sample NRMSE and relative error in the SCR estimation produced by the different LSMC-style methods.

CONCLUSIONS

- Assessing the SCR via nested simulations can pose computational challenges.
- Neural Networks are effective in alleviating the computational cost of SCR calculations, but they operate as black boxes.
- localGLMnet allows for model explainability and yields accurate results.
- Regularisation can improve performance and enhance the robustness of the method.

Thank you! Obrigado!

Questions?

Mail: salvatore.scognamiglio@uniparthenope.it

Linkedin: