SHGR: A Generalized Maximal Correlation Coefficient

Samuel Stocksieker
CNRS, I2M
Aix Marseille University
Marseille, France

Denys Pommeret
CNRS, I2M
Aix Marseille University
Marseille, France

Abstract

Traditional correlation measures, such as Pearson's and Spearman's coefficients, are limited in their ability to capture complex relationships, particularly nonlinear and multivariate dependencies. The Hirschfeld–Gebelein–Rényi (HGR) maximal correlation offers a powerful alternative by measuring the highest Pearson correlation achievable through nonlinear transformations of two random variables. However, estimating the HGR coefficient remains challenging due to the complexity of optimizing arbitrary nonlinear functions. We introduce a new coefficient, satisfying Rényi's axioms, based on the extension of HGR with Spearman's rank correlation: the Spearman HGR (SHGR). We propose a neural network-based estimator tailored to estimate (i) the bivariate correlation matrix, (ii) the multivariate correlations between a set of variables and another one, and (iii) the full correlation between two sets of variables. This estimate effectively detects nonlinear dependencies and demonstrates robustness to noise, outliers, and spurious correlations (*hallucinations*). Additionally, it achieves competitive computational efficiency through designed neural architectures. Comprehensive numerical experiments and feature selection tasks confirm that SHGR outperforms existing state-of-the-art methods.

1 Introduction

Understanding variable dependencies is a central challenge in machine learning, statistics, and data science, with critical applications including feature selection, dimensionality reduction, fairness assessment, causal inference, and multimodal learning. Classical measures, especially Pearson's and Spearman's correlations, are widely used but limited to linear, monotonic, and bivariate relationships, often failing to capture complex multivariate or higher-order dependencies. To address these limitations, some generalized dependence measures have been proposed. Among them, the Hirschfeld-Gebelein-Rényi (HGR) maximal correlation stands out as a theoretically principled tool for quantifying nonlinear dependence between random variables, whether univariate or multivariate. Introduced by Hirschfeld [21], extended by Gebelein [10], and formalized by Rényi [37], HGR defines correlation as the maximum linear correlation between transformed versions of the variables. Despite its strong theoretical appeal, estimating the HGR coefficient remains challenging due to the complexity of identifying optimal transformations. Existing alternative methods suffer from similar drawbacks, including limited interpretability, high computational complexity, or difficulty in extending to multivariate configurations.

In this paper, we introduce an extension of the Hirschfeld-Gebelein-Rényi (HGR) maximal correlation that we shall call *Spearman HGR* (SHGR). It is based on Spearman instead of the Pearson coefficient. SHGR offers a scalable and flexible way to estimate nonlinear dependencies in both bivariate and multivariate settings: by estimating pairwise, multivariate (set-to-target), and groupwise correlation matrices. This extension builds on two key components: (i) a neural approximation tailored for simultaneous estimation of correlations, and (ii) a copula-based formulation that operates on ranks, ensuring invariance to monotonic transformations while improving robustness and stability against noise and extreme values. Our main contributions are summarized as follows:

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

- We revisit HGR maximal correlation by proposing a Spearman-based extension: `SHGR`
- We present a first estimator of `SHGR` that is differentiable, fast, efficient, and robust to noise, outliers, and spurious dependencies. Unlike some existing methods, it recovers the optimal nonlinear transformations and enables formal significance testing.
- We introduce a stacked cross-encoder architecture specifically designed to estimate multiple correlations simultaneously in both bivariate, multivariate, and groupwise contexts.
- We establish a comprehensive evaluation protocol for assessing maximal correlation estimators in terms of performance, robustness to noise, hallucinations (respecting Rényi's Axiom 4; see Appendix A), extreme values, as well as the estimation of bivariate, multivariate, and full correlations, significance testing, and computational efficiency.
- We validate `SHGR` on synthetic and real-world tabular datasets. We demonstrate that `SHGR` outperforms existing state-of-the-art methods in terms of performance and robustness.

The remainder of the paper is organized as follows: Section 2 reviews prior work on maximal correlation. Section 3 introduces `SHGR`, defining the proposed coefficient and its neural estimator based on stacked encoders. Section 4 presents a comprehensive empirical evaluation, comparing `SHGR` to state-of-the-art methods, especially in terms of detection power and robustness. Section 5 shows the application of `SHGR` to feature selection tasks on real datasets. Section 6 concludes with a discussion of the strengths and limitations of our approach and future directions. Code and data are available at: `https://github.com/sstocksieker/SHGR`

2 Background

2.1 Related Work

Numerous nonlinear dependence measures have been proposed to extend classical coefficients such as Pearson's r, Kendall's τ, and Spearman's ρ. Canonical Correlation Analysis (CCA) [23] is a foundational method for extracting linear dependencies between sets of variables, but its linearity limits its capacity to capture complex relationships. Its extensions ([42, 49]) include Kernel CCA [26, 20], which applies CCA to nonlinear feature spaces, and Deep CCA (DCCA) [1, 45, 9], which uses deep networks to model nonlinear projections. The Randomized Dependence Coefficient (RDC) [31] offers a flexible and efficient alternative using random nonlinear projections followed by CCA. In contrast, the Hirschfeld–Gebelein–Rényi (HGR) maximal correlation, originally developed by [21, 10, 37], aims to quantify dependence via maximal Pearson correlation after applying measurable transformations. HGR has been widely studied in machine learning, notably through the Alternating Conditional Expectations (ACE) algorithm [6], which iteratively estimates transformations via conditional regressions. While partially interpretable, ACE is sensitive to noise and the quality of the estimators [25]. Recently, HGR-based methods have been applied in fairness-aware learning [36, 33, 16, 13], multimodal learning [30, 44], regression [47], a linear-time independence criterion [48], a generalized ACE for unlabeled data [46] or in feature selection [24]. Hellinger Correlation [11] is a recent alternative, though limited to two variables. Other notable dependence measures include Distance Correlation (dCor) or Brownian Correlation [28], the Hilbert-Schmidt Independence Criterion (HSIC and CHSIC) [19, 18], and the Maximum Mean Discrepancy (MMD) [17]. These kernel-based methods, widely used in hypothesis testing, capture a broad range of dependencies, but are sensitive to kernel choice and do not yield interpretable scores in $[0, 1]$. The Mutual Information Coefficient (MIC) [38] can detect various functional relationships but lacks axiomatic properties like those of HGR and is sensitive to binning heuristics. GeDI [14], recently proposed for fairness applications, offers an interpretable, transparent measure. MMD and its copula-based variant CMMD [35], as well as GeDI, are not included in our evaluation as they yield scale-free measures not confined to $[0, 1]$, limiting interpretability and comparability. Table 1 in Appendix B) compares the different methods, theoretically and empirically.

2.2 Hirschfeld–Gebelein–Rényi Correlation coefficient

The Hirschfeld–Gebelein–Rényi maximum correlation (HGR) is a statistical measure theoretically designed to quantify the relationship between two (univariate or multivariate) random variables. Unlike the standard correlation coefficients, the HGR is capable of detecting both linear and nonlinear associations. It is computed as the maximum correlation coefficient obtained after transforming the variables using measurable functions. This makes the HGR particularly useful in situations where the relationship between variables is complex. The HGR coefficient has the advantage of being between 0 (independence) and 1 (strong correlation). As defined by Rényi ([37]), the maximal coefficient presents interesting properties, described in Appendix A. While the HGR correlation coefficient holds great potential in theory, its estimation can be quite challenging due to the limitless possibilities for transformations. Moreover, the relationships between variables can be highly complex and require substantial computational resources, making its application difficult.

Definition 1 (HGR maximal correlation coefficient). Let U ($\sim \mathcal{D}_U$) and V ($\sim \mathcal{D}_V$) be two continuous random variables taking values in $\mathcal{U}$ and $\mathcal{V}$, respectively. Let $\mathcal{E}(\mathcal{U})$ (resp. $\mathcal{E}(\mathcal{V})$) denote the set of measurable functions from $\mathcal{U}$ (resp. $\mathcal{V}$) to $\mathbb{R}$. Let $r()$ denote the Pearson correlation coefficient. The Hirschfeld-Gebelein-Renyi (HGR) maximal correlation coefficient is defined as follows ([21], [10], [37]):

$$HGR(U,V) := \sup_{\substack{f_U \in \mathcal{E}(\mathcal{U}) \\ f_V \in \mathcal{E}(\mathcal{V})}} r(f_U(U), f_V(V)) = \sup_{\substack{f_U \in \mathcal{E}(\mathcal{U}), f_V \in \mathcal{E}(\mathcal{V}) \\ \mathbb{E}(f_U(U))=0, \mathbb{E}(f_V(V))=0 \\ \mathbb{E}(f_U^2(U))=1, \mathbb{E}(f_V^2(V))=1}} \mathbb{E}_{U \sim \mathcal{D}_U, V \sim \mathcal{D}_V}(f_U(U) f_V(V))$$

2.3 Neural HGR

In the context of fairness analysis, [15] proposed estimating the HGR transformations using neural networks to capture nonlinear relationships. This estimation can be formulated as a general optimization problem, where the goal is to map two random variables into a space where their linear correlation is maximized. The algorithm takes u, a sample of U, and v,a sample of V, as inputs, and returns as output the estimated Pearson correlation[1] $r(f_u(u), f_v(v))$, where f_u and f_v are parameterized by a compact domain Θ. This measure is then estimated using a neural network by minimizing the following loss function:

$$\mathcal{L}_{NHGR} = -\sup r(f_{\theta_u}(u), f_{\theta_v}(v)),$$

where f_{θ_u} (resp. f_{θ_v}) denotes a neural estimator of f_u (resp. f_v). This estimator of the maximal correlation coefficient $HGR(u,v)$, denoted $NHGR$ (for Neural-HGR) is defined as:

$$NHGR_\Theta(u,v) = r(f^*_{\theta_u}(u), f^*_{\theta_v}(v)), \text{ with } (f^*_{\theta_u}, f^*_{\theta_v}) = \operatorname*{arg\,max}_{f_{\theta_u}, f_{\theta_v} \in \Theta} r(f_{\theta_u}(u), f_{\theta_v}(v))$$

This method enables the approximation of the maximal correlation between two variables, or between one variable and a set of variables ([15], [31]). Nevertheless, by construction, this approach is based on the Pearson correlation coefficient and therefore inherits its known limitations.

3 SHGR: A Robust and Efficient Maximal Correlation Coefficient

3.1 Neural Spearman HGR

The HGR coefficient is a valuable tool for identifying potential correlations but may, in practice, exhibit two key limitations similar to those of Pearson's correlation: (i) sensitivity to extreme values, and (ii) restricted to capturing linear dependencies of transformed variables (illustration in Appendix C). In practice, the constraints of centering and unit variance are insufficient to exclude outliers, which could bias the HGR estimator completely. Using neural networks could lead to overfitting behaviors, such as learning extreme values to inflate the correlation score artificially. The sensitivity of neural HGR estimators to outliers can lead to overfitting behaviors and artificially inflated correlation scores. To address these limitations, we propose a natural extension of the HGR coefficient. Our approach is inspired by the Spearman coefficient and copula-based methods, which rely on cumulative distribution functions and are inherently robust to both extreme values and monotonic transformations. This substitution offers two main advantages: (i) it improves robustness to extreme values and distributional shifts, and (ii) it extends the scope of measurable dependence to include monotonic but nonlinear relationships. Importantly, it remains consistent with the HGR principle of maximizing correlation between nonlinear transformations of the variables, but reframes the notion of dependence in terms of rank-based monotonicity rather than raw linearity. This could help reduce transformation efforts and optimize calibration to avoid seeking linear correlation, just monotonic correlation. Recall here

[1] Its estimate is also designated by r to simplify notation.

that the ith rank statistic from a n-sample, $(z_1, \cdots, z_n)$, drawn from a random univariate variable Z, is given by:

$$rank(z_i) = \sum_{j=1}^{n} \mathbb{1}_{z_j \leq z_i} = n\widehat{F}_z(z_i), \text{ with } \widehat{F}_z(z_i) := \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_{z_j \leq z_i},$$

where $\widehat{F}_z$ denotes the empirical cumulative distribution function. By extension, the rank of a random vector yields the vector of all ranks. We recall that the Spearman correlation coefficient between two iid paired samples $u = (u_1, \cdots, u_n)$ and $v = (v_1, \cdots, v_n)$ from U and V, denoted by $\rho(u, v)$, is defined as the Pearson correlation coefficient based on the ranks:

$$\rho(u, v) = r\left(n\widehat{F}_U(u), n\widehat{F}_V(v)\right) = r\left(\widehat{F}_U(u), \widehat{F}_V(v)\right) \text{ , with } \widehat{F}_U(u) = (\widehat{F}_U(u_1), \cdots, \widehat{F}_U(u_n)).$$

Definition 2. (Spearman-HGR (SHGR) coefficient). Let U and V be two paired continuous random variables taking values in $\mathcal{U}$ and $\mathcal{V}$, respectively. Let $\mathcal{E}(\mathcal{U})$ (resp. $\mathcal{E}(\mathcal{V})$) denote the set of measurable functions from $\mathcal{U}$ (resp. $\mathcal{V}$) to $\mathbb{R}$. The Spearman-HGR (SHGR) coefficient associated to (U, V) is defined by

$$SHGR(U, V) := \max_{\substack{f_u \in \mathcal{E}(\mathcal{U}), f_v \in \mathcal{E}(\mathcal{V}) \\ \mathbb{E}(f_u(U))=0, \mathbb{E}(f_v(V))=0 \\ \mathbb{E}(f_u^2(U))=1, \mathbb{E}(f_v^2(V))=1}} r(F_{f_u(U)}(f_u(U)), F_{f_v(V)}(f_v(V))).$$

The SHGR is related to the notion of grade correlation between two random variables U and V (see [12]), which is the limit of their Spearman coefficients and is defined as the correlation between their copula transformations:

$$\gamma(U, V) = r(F_U(U), F_V(V)) = \lim_{n \mapsto \infty} \mathbb{E}(\rho(u, v)),$$

where $(u, v) = (u_i, v_i)_{i=1,\cdots,n}$ are n-iid paired samples from (U, V). So, we have:

$$SHGR(U, V) = \max_{f_u \in \mathcal{E}(\mathcal{U}), f_v \in \mathcal{E}(\mathcal{V})} \gamma(f_u(U), f_v(V)).$$

It is important to note that the copula transformation preserves the dependence between the original vectors U and V (see for instance, [34]).

Using the empirical estimator $\widehat{F}$ of F, we obtain an estimator of the SHGR and its Neural version:

$$\widehat{SHGR} = \max_{\substack{f_u \in \mathcal{E}(\mathcal{U}), f_v \in \mathcal{E}(\mathcal{V}) \\ \mathbb{E}(f_u(U))=0, \mathbb{E}(f_v(V))=0 \\ \mathbb{E}(f_u^2(U))=1, \mathbb{E}(f_v^2(V))=1}} r(\widehat{F}_{f_u(U)}(f_u(U)), \widehat{F}_{f_v(V)}(f_v(V))),$$

$$SHGR_\Theta(u, v) = \max_{f_{\theta_u}, f_{\theta_v} \in \Theta} \rho(f_{\theta_u}(u), f_{\theta_v}(v)) = \max_{f_{\theta_u}, f_{\theta_v} \in \Theta} r(\widehat{F}_{f_{\theta_u}(U)}(f_{\theta_u}(u)), \widehat{F}_{f_{\theta_v}(V)}(f_{\theta_v}(v))).$$

Note that the rank transformation is applied only to the correlation computation, not to the input variables themselves. This preserves the information of the original inputs while benefiting from the robustness of rank-based objectives.

Consistency of SHGR estimates

SHGR is empirically motivated as a rank-based reformulation of the HGR coefficient ensuring robustness to outliers and extreme values. It theoretically relies on copula transformations to preserve dependency structures. We now prove consistency guarantees through convergence results and Rényi's axioms. The proofs are given in Appendix D.

Proposition 1. For all $\epsilon > 0$, there exists a family of continuous neural networks parametrized by a compact domain Θ, such that

$$|SHGR_\Theta(u, v) - \widehat{SHGR}(u, v)| \leq \epsilon.$$

Proposition 2. Let (u, v) be independent sequences of iid samples of size n drawn by (U, V). We have the following convergence in law as n tends to infinity:

$$\widehat{SHGR}(u, v) \to SHGR(U, V).$$

Proposition 3. Let U, V be two continuous random variables. All the following Rényi axioms for nonlinear dependence measures (modified by [40]) are satisfied by the SHGR coefficient.

1. $SHGR(U, V)$ is defined for all pairs of non-constant continuous random variables U, V.
2. $SHGR(U, V) = SHGR(V, U)$.
3. $0 \leq SHGR(U, V) \leq 1$.
4. $SHGR(U, V) = 0$ if and only if U and V are independent.
5. For all Borel-measurable bijective functions $f : \mathbb{R}^p \to \mathbb{R}$ and $g : \mathbb{R}^q \to \mathbb{R}$, $SHGR(U, V) = SHGR(f(U), g(V))$.
6. $SHGR(U, V) = 1$ if $U = f(V)$ or $V = g(U)$, for some Borel-measurable function f or g.
7. If $(U, V) \sim \mathcal{N}(\mu, \Sigma)$, then $SHGR(U, V)$ is a strictly increasing function of $|r(U, V)|$.

3.2 Generalized Neural SHGR

While pairwise correlation analysis is useful, many real-world applications require assessing dependencies between a group of variables and one or more target variables. In a similar manner to the Canonical Correlation Analysis (CCA) and the Randomized Dependence Coefficient (RDC), we propose here a natural generalization of our rank-based HGR estimator to this multivariate setting. For instance, it is often necessary to analyze several pairwise correlations rather than focusing on a single variable pair. Since training neural networks can be time-consuming, estimating each correlation independently may be inefficient. We address this limitation by introducing a neural architecture specifically designed for this purpose. To construct a matrix of SHGR estimates, we extend the neural SHGR estimator using a stacked cross-encoder design. This architecture is illustrated in Figure 1a for pairwise (1-vs-1) analysis and in Figure 1b for full (p-vs-q) correlation matrices. An architecture for multivariate (p-vs-1) settings is provided in Appendix (Figure 9). Typically, for bivariate estimation, the algorithm considers p input variables $u_1, \ldots, u_p$, and aims to optimize all pairwise entries in the $p \times p$ correlation matrix. The stacked encoder design enables marginal transformations that capture complex, nonlinear dependencies between variables. The objective function for bivariate estimations (pairwise correlations) is defined as follows:

$$\mathcal{L}_{SHGR}(\boldsymbol{u}) := -\sum_{\substack{i,j=1 \\ i \neq j}}^{p} \left[\rho^2 \left(f_{\theta_{u_i}}(u_i), f_{\theta_{u_j}}(u_j) \right)^{1/2} \right]^{\alpha}, \text{ with } \alpha > 0.$$

The objective function for multivariate analysis (correlations of any order) is thus:

$$\mathcal{L}_{SHGR}(\boldsymbol{u}) := -\sum_{i=1}^{p} \left[\rho^2 \left(f_{\theta_{u_i}}(u_i), g_{\theta_{u_i}}(\{u_j\}_{j \neq i}) \right)^{1/2} \right]^{\alpha}, \text{ with } g_{\theta_{u_i}} : \mathbb{R}^{p-1} \to \mathbb{R}.$$

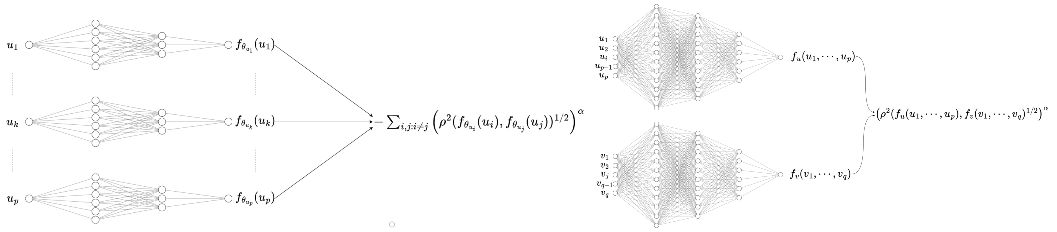
We also extend this approach to obtain full correlation between two sets of variables $\mathbf{u}$ and $\mathbf{v}$ of dimensions p and q respectively (for example, two datasets):

$$\mathcal{L}_{SHGR}(\mathbf{u}, \mathbf{v}) := -\left[\rho^2 \left(f_{\theta_{\mathbf{u}}}(\mathbf{u}), f_{\theta_{\mathbf{v}}}(\mathbf{v}) \right)^{1/2} \right]^{\alpha}, \text{ with } f_{\theta_{\mathbf{u}}} : \mathbb{R}^p \to \mathbb{R}, f_{\theta_{\mathbf{v}}} : \mathbb{R}^q \to \mathbb{R}.$$

Similar to the absolute value, we apply a square root transformation of the squared coefficient to obtain a differentiable operation. We introduce a hyperparameter α that allows weighting the coefficients: a weight greater than 1 gives more importance to stronger correlations, and their maximization will be prioritized during training, enabling faster maximization of potential correlations. It is interesting to note that the Canonical Correlation Analysis (CCA) is a special case of SHGR, corresponding to a single linear layer without activation and rank transformation.

Practical Remarks (1) Since our loss function depends on rank-based estimation, we use *TorchSort* approach [4] to obtain a differentiable approximation of the sorting operation[2]. (2) Empirically, SHGR exhibits convergence, with no substantial gain beyond a threshold (illustrated in Figure 12a in

[2] We used the implementation available at `https://github.com/teddykoker/torchsort`.


u_1
u_k
u_p
$f_{\theta_{u_1}}(u_1)$
$f_{\theta_{u_k}}(u_k)$
$f_{\theta_{u_p}}(u_p)$
$-\sum_{i,j:i\neq j}\left(\rho^2(f_{\theta_{u_i}}(u_i), f_{\theta_{u_j}}(u_j))^{1/2}\right)^{\alpha}$
u_1 u_2 u_i u_{p-1} u_p
v_1 v_2 v_j v_{q-1} v_q
$f_u(u_1, \cdots, u_p)$
$f_v(v_1, \cdots, v_q)$
$(\rho^2(f_u(u_1, \cdots, u_p), f_v(v_1, \cdots, v_q)^{1/2})^{\alpha}$

(a) for bivariate correlations estimation

(b) for full correlation estimation

Figure 1: Cross-Encoder Architecture

Appendix F.3.4), motivating the use of an early stopping to balance accuracy and efficiency. (3) Our estimator supports significance testing via a Spearman-based test. To improve robustness, we apply a masking rule: correlations not significant with the test are set to zero. (4) An extended version of SHGR handles mixed-type data by incorporating the correlation ratio (numeric-to-categorical) and Cramér's V (categorical-to-categorical) (illustration in Appendix F.3.4).

4 Experiments

We propose to evaluate the SHGR and its competitors with a complete protocol following the analysis below (defining the *multivariate power of correlation measures*): (4.2.1) Performance: ability to capture complex nonlinear correlations without noise ; (4.2.2) Robustness to noise: ability to identify complex nonlinear correlations with noise ; (4.2.3) Robustness to hallucination (spurious correlations): null correlation in the case of independence ; (4.2.4) Robustness to extreme values: null correlation in the case of independence, in the presence of extreme values; (4.2.6) *Bivariate power of a dependence measure* as proposed by [31] ; (4.2.7) Computation time: fast to estimate correlations ; (4.2.5) Significance test analysis: possibility of performing a significance test on the coefficient and ability to reject the null hypothesis (of null correlation) in the presence of noise.

4.1 Experimental Protocol

We apply the *multivariate power of correlation measures* protocol in three settings: (i) bivariate (pairwise) correlations (1-vs-1), (ii) multivariate correlations (p-vs-1), and (iii) full (groupwise) correlations (p-vs-q). For (i), we generate six independent Gaussian variables and five others that depend nonlinearly on them. For (ii), we simulate 20 variables, including nonlinear dependencies involving more than two variables. For (iii), we generate two datasets with varying global correlation structures. Details on data generation are provided in Appendix F.3. Figure 2a presents the pairplot for bivariate correlations. We estimate nonlinear correlations using SHGR and compare them with the following alternative methods [3] (some methods are unavailable in the multivariate and full settings):

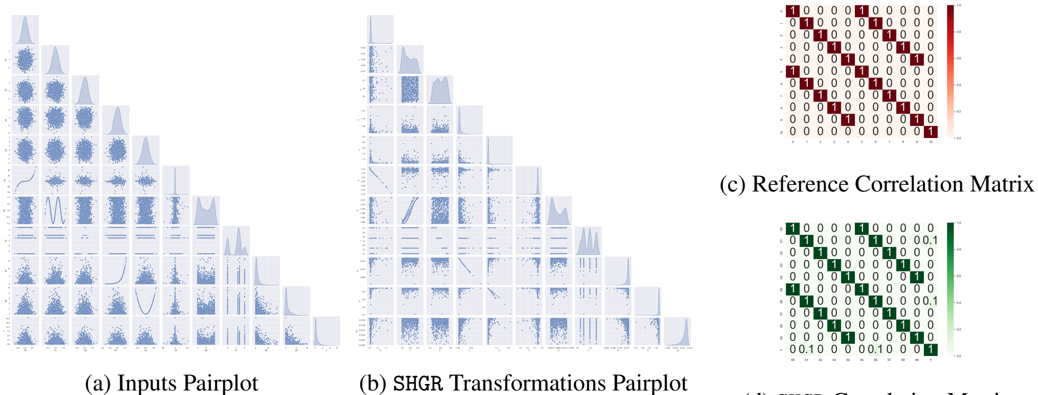
- The Pearson correlation coefficient: *Pearson*
- The Spearman correlation coefficient: *Spearman*
- The Kendall correlation coefficient: *Kendall*
- The Randomized Dependence Coefficient ([31]): *RDC*
- The Mutual Information Criterion ([38]): *MIC*
- The Distance Correlation ([41]): *dCor*
- Canonical Correlation Analysis ([23]): *CCA*
- Kernel Canonical Correlation Analysis ([2]): *kCCA*[4]
- Alternating Conditional Expectations ([7]): *ACE*
- HGR estimation with kernel ([33]): *HGRkde*
- HGR estimation with Neural Net ([16]): *HGRnn*
- HGR estimation with Lattice ([14]): *HGRlat*
- HGR estimation with double kernel ([14]): *dk*

[3] The used libraries are listed in Appendix F.4

[4] This approach was not integrated because it was unstable and too time-consuming to calculate

- HGR estimation with simple kernel ([14]): *sk*
- The Normalized Hilbert-Schmidt Independence Criterion ([18]): *NHSIC*
- The Hellinger Correlation ([11]): *HR*[5]

We perform our SHGR model for 100, 200, and 500 epochs and with early stopping. A sensitivity analysis of the SHGR architecture and hyperparameters was performed (details are provided in Appendix F.2.3). A single architecture and hyperparameter configuration were then used consistently across all illustrations and experiments. In the analyses below, we generate $K = 10$ synthetic datasets and compare the estimated correlations from each method to the true, known, correlations, referred to as reference correlations. We evaluate the deviation from these reference correlations using: the pairwise correlation matrix for bivariate correlation, the vector of multiple correlations for multivariate correlation, and the correlation coefficient for full correlation analysis.



(a) Inputs Pairplot

(b) SHGR Transformations Pairplot

(c) Reference Correlation Matrix

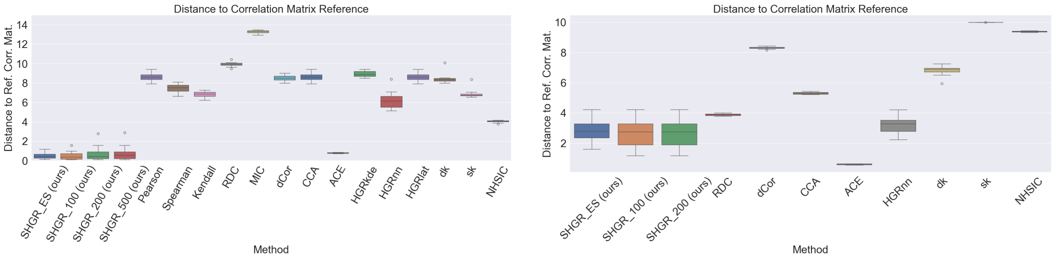
(d) SHGR Correlation Matrix

Figure 2: Illustration of SHGR on bivariate correlations

4.2 Results

4.2.1 Performance

Our goal in this experiment is to evaluate the ability of the different approaches to detect perfect nonlinear correlations in a noise-free setting. Figure 3 shows the results for both bivariate and multivariate settings. We observe that SHGR closely matches the reference correlation values and performs on par with the strong baseline *ACE*. Moreover, we find that varying the number of training epochs for SHGR (early stopping, 100, 200, or 500 epochs) has little to no effect on performance, suggesting that the method converges efficiently. In the full-correlation setting, SHGR, together with *RDC*, *CCA*, *dk*, and *sk*, successfully captures nonlinear dependencies, whereas other approaches such as *HGRnn*, *NHSIC*, and *dCor* fail to do so (see Figure in Appendix F.5.3). Further implementation details and numerical results are provided in Appendix F.5.



(a) Bivariate Correlation

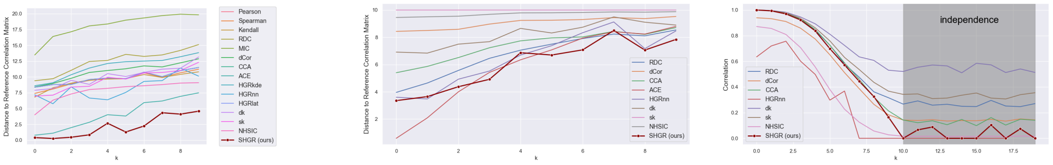
(b) Multivariate Correlation

Figure 3: Performance Results: distance to Reference Correlations (lower is better)

[5] This approach was not integrated because inefficient, limited to pairwise correlations and too time-consuming

4.2.2 Robustness to Noise

We now evaluate the robustness of correlation measures with respect to additive white noise (zero mean), assessing their ability to detect significant dependencies as noise increases. To this end, we reproduce the previous simulations while gradually introducing higher levels of noise in each trial. In this setting, the true correlation is unknown. Following the approach of [31], we consider a method more robust if it yields higher correlation values under increasing noise. We therefore adopt the same evaluation metric as in the previous section: the distance to the reference correlations. Results are shown in Figure 4, detailed experimental settings and additional visualizations are given in Appendix F.6. As noise increases (left to right on the x-axis), `SHGR` consistently outperforms state-of-the-art baselines across bivariate, multivariate, and full-correlation scenarios. Notably, while *ACE* performs well in noise-free settings, its accuracy drops significantly under noise. In the full-correlation setting, some methods even exhibit hallucinated correlations when applied to independent variables, which is the case when the value of k on the x-axis exceeds 10.



(a) Bivariate Correlation (b) Multivariate Correlation (c) Full Correlation

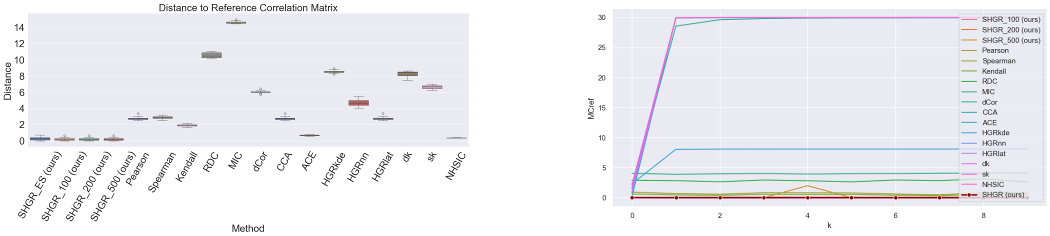
Figure 4: Robustness to Noise (a) and (b): distance to Reference Correlations (lower is better) ; (c): Decreasing correlation context on the left side (higher is better) and independence context on the right side (lower is better)) - Results with `SHGR` are red

4.2.3 Robustness to Hallucination

We evaluate here the ability of correlation measures to avoid hallucinations, that is, to refrain from reporting spurious correlations on fully independent data. For each scenario, we simulate independent variables and compute the corresponding estimated correlation coefficients. As illustrated in Figure 5a, our method, `SHGR`, remains stable and does not produce artificial correlations, unlike several competing approaches that yield inflated values. Additional details and experimental results are provided in Appendix F.7.

4.2.4 Robustness to Extreme Values

We evaluate the robustness of correlation measures to hallucinated dependencies induced by extreme values. Following the previous setup, we simulate independent variables and inject extreme values in all iterations except the first, which serves as a baseline reference. We then apply various correlation estimators to assess their stability under such perturbations. As shown in Figure 5b, `SHGR` remains stable in the presence of extreme values, while several competing methods, including *HGRnn*, exhibit inflated or unstable correlation estimates. Additional results and implementation details are provided in Appendix F.8.



(a) Robustness to hallucination (b) Robustness to Extreme Values

Figure 5: Analysis of the distances to the Bivariate Reference Correlation Matrix (lower is better)

4.2.5 Significance Test

SHGR optimizes the Pearson correlation coefficient over nonlinear transformations of the input variables, in line with the theoretical definition of HGR. As such, it is possible to perform an asymptotic significance test on the obtained correlation coefficient (when n is sufficiently large or by applying a Gaussian quantile transformation, that is monotonic) [29]. A non-parametric permutation test can also be applied. Specifically, we test the null and alternative hypotheses:

$$H_0 : \ SHGR_\Theta(u, v) = 0 \quad \text{versus} \quad H_1 : \ SHGR_\Theta(u, v) \neq 0$$

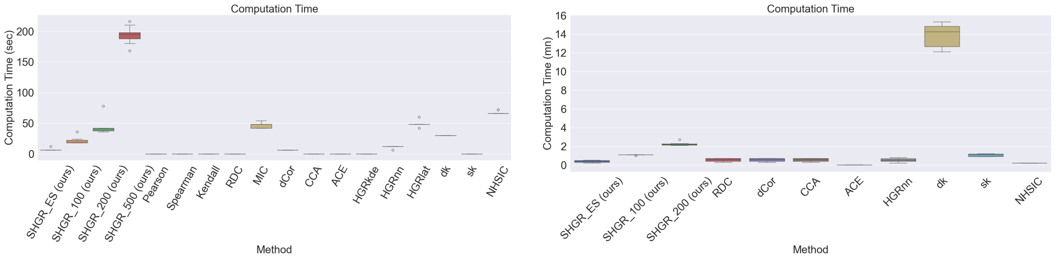
We assess the effectiveness of this test across various bivariate and multivariate relationships, each perturbed with increasing levels of noise. The results are compared to the significance tests implemented for *MIC*, *RDC*, and *dCor*, over nine representative types of dependencies. Full details are provided in Appendix F.9. Overall, the SHGR-based test consistently outperforms the alternatives, demonstrating greater robustness and a higher ability to correctly reject the null hypothesis of independence.

4.2.6 Bivariate Power of Dependence Measure

In addition to the previous evaluations, we assess the *bivariate power of a dependence measure*. Following the protocol introduced in [31] and adopted in subsequent works such as [32] and [15], we evaluate the robustness of our method by progressively perturbing several nonlinear bivariate and multivariate relationships. An ideal dependence measure should maintain high correlation values even as noise increases. Detailed results are provided in Appendix F.10. Overall, SHGR demonstrates strong robustness, consistently producing higher correlation scores than competing methods across all tested configurations.

4.2.7 Computation time

SHGR exhibits highly competitive computation times. Unlike many existing approaches, it avoids iterative loops for estimating multiple correlations, instead performing joint optimization over all coefficients, which significantly reduces computational overhead. In contrast, baseline methods often rely on sequential estimation, leading to increased runtimes. On a typical illustrative dataset (2,000 observations and 10 variables), SHGR computes the full pairwise correlation matrix in approximately 5 seconds with early stopping. It consistently outperforms *NHSIC*, *MIC*, *dk*, and *HGRlat* for bivariate correlations. In the multivariate setting, its runtime is on par with *RDC* and *CCA*. We also evaluate the method's scalability with respect to the number of variables and sample size. SHGR remains efficient in high-dimensional settings: for instance, it requires only 3.5 minutes to estimate multivariate correlations on a dataset with 10,000 observations, faster than both *RDC* and *CCA*, and just 1 minute to compute bivariate correlations across 1,000 variables. Further computational details are provided in Appendix F.11. Figure 6 presents the computation time for illustration. In the case of bivariate correlation estimates, i.e., the correlation matrix, the computation time of SHGR, with early stopping, is around 6 seconds on average. This is faster than *MIC, HGRlat, dk* and *NHSIC*. In the case of multivariate correlation estimates, the computation time of SHGR is 23 seconds on average and is slightly faster than the *CCA, RDC, and dCor* methods ($\tilde{3}$3 seconds). The *dk* method takes far too long, with almost 14 minutes. The *ACE* method is instantaneous.



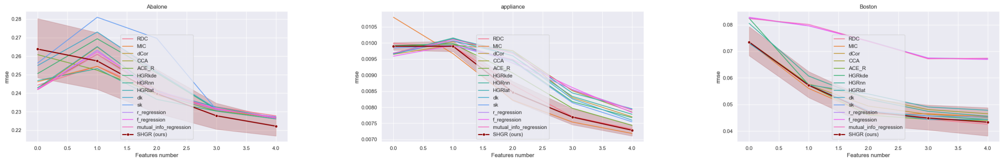
(a) Bivariate correlations

(b) Multivariate correlations

Figure 6: Computational time Comparison

5 Real-World Applications

To evaluate the practical relevance of our method, we apply it to feature selection on nine real-world tabular datasets (see Appendix G). For each method, we select the top k features most correlated with the target y, and assess predictive performance via RMSE on a test set (30%), using a random forest regressor. To test robustness to limited data, all models are trained on at most 500 samples. Some results are shown in Figure 7. Across most datasets, SHGR-selected features consistently yield lower RMSE than competing methods. We also evaluate the multivariate SHGR using a leave-one-out strategy to estimate each variable's contribution to the global X-vs-y correlation. Complete results are reported in Appendix G, confirming the effectiveness of SHGR for real-world feature selection.



(a) Abalone dataset

(b) Appliance dataset

(c) Boston dataset

Figure 7: RMSE for test set prediction with feature selection on real-world datasets (lower is better); Based on maximal (bivariate) correlation - Results with SHGR are red

6 Discussion

The problem of estimating the maximal correlation between two variables has long been a central challenge in statistics and machine learning. Several approaches have been proposed, most of which are based on the Hirschfeld-Gebelein-Rényi (HGR) definition, where the goal is to learn transformations of the input variables that maximize their linear correlation. In this work, we introduce a new maximal correlation coefficient, SHGR, which generalizes the HGR framework using rank-based transformations and satisfies Rényi's axioms. Its estimation relies on a dedicated neural architecture. Our approach offers several key advantages: (1) it directly optimizes a full matrix of correlation coefficients in a non-iterative manner, significantly reducing computation time; (2) the estimator is accurate, robust, and computationally efficient; (3) it enables statistical significance testing of the estimated correlations; (4) it outperforms state-of-the-art methods across multiple evaluation settings and real-world experiments; Interestingly, our approach can be combined with existing methods to improve their results, as shown in Appendix F.14 and, (5) SHGR is fully differentiable, making it easy to integrate into modern deep learning pipelines. As highlighted by [13], HGR-based methods may produce "hallucinated" correlations as a result of overfitting. In our illustrations, we showed that SHGR remains robust to such artifacts. Nevertheless, caution is still required; therefore, it is important to carefully tune the stacked neural network in our approach, sufficiently expressive to capture complex dependencies, but not overly flexible to prevent overfitting and artificially inflated correlation scores. Our experimental design includes a variety of nonlinear dependencies, inspired by the *Power of Dependence Measure* protocol suggested by [31]. We assessed performance in both noise-free and noisy settings.

Further exploration of other types of nonlinear dependencies would be a valuable extension. [13] also points out that HGR is difficult to interpret. Although this criticism is generally valid, SHGR provides access to the learned transformations of the input variables, which can provide information about the nature of the dependency. Indeed, the SHGR approach allows for graphical analysis of the applied transformations by plotting the transformations of stacked neural networks as a function of the inputs, as illustrated in Figure 37. Nevertheless, interpretability is not the main objective of HGR-based measures. Exploring the interpretability of learned functions is an interesting avenue, and a posteriori visualizations could help to better understand the adjusted transformations and therefore the dependencies. Although detaching normalization terms reduces theoretical gradient bias, we found that doing so significantly harmed empirical performance. We thus keep the differentiable form and leave bias-corrected alternatives for future work. Analyzing its behavior in high-dimensional settings remains an open question. Extending the framework to large-scale neural architectures and multimodal learning tasks, as well as applying it to fairness-aware scenarios, represents promising directions for future research.

Acknowledgment

The authors would like to express their sincere gratitude to the reviewers for their helpful comments, which have contributed to improving the manuscript. S. Stocksieker would like to acknowledge the support of the Research Chair DIALog under the aegis of the Risk Foundation, a joint initiative by *CNP Assurance*. D. Pommeret would like to acknowledge the support received from the Research Chair ACTIONS under the aegis of the Risk Foundation, an initiative by BNP Paribas Cardif and the Institute of Actuaries of France.

References

[1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[2] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48, 2002.

[3] Yves Ismaël Ngounou Bakam and Denys Pommeret. Smooth test for equality of copulas. *Electronic Journal of Statistics*, 18(1):895 – 941, 2024.

[4] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

[5] Paula Branco, Luis Torgo, and Rita P Ribeiro. Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing*, 343:76–99, 2019.

[6] L. Breiman and J.H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, page 580–619, 1985.

[7] Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.

[8] Luis Candanedo. Appliances Energy Prediction. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C5VC8G.

[9] Zhiwen Chen, Siwen Mo, Haobin Ke, Steven X Ding, Zhaohui Jiang, Chunhua Yang, and Weihua Gui. Canonical correlation guided deep neural network. *arXiv preprint arXiv:2409.19396*, 2024.

[10] Hans Gebelein. Das statistische problem der korrelation als variations- und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik*, 21:364–379, 1941.

[11] Gery Geenens and Pierre Lafaye de Micheaux. The hellinger correlation. *Journal of the American Statistical Association*, 117(538):639–653, 2022.

[12] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric statistical inference: revised and expanded.* CRC press, 2014.

[13] Luca Giuliani, Eleonora Misino, and Michele Lombardi. Generalized disparate impact for configurable fairness solutions in ML. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11443–11458. PMLR, 23–29 Jul 2023.

[14] Luca Giuliani, Eleonora Misino, and Michele Lombardi. Generalized disparate impact for configurable fairness solutions in ml. In *International Conference on Machine Learning*, pages 11443–11458. PMLR, 2023.

[15] Vincent Grari, Oualid El Hajouji, Sylvain Lamprier, and Marcin Detyniecki. Learning unbiased representations via rényi minimization. In Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 749–764, Cham, 2021. Springer International Publishing.

[16] Vincent Grari, Boris Ruf, Sylvain Lamprier, and Marcin Detyniecki. Fairness-aware neural r\'eyni minimization for continuous features. *arXiv preprint arXiv:1911.04929*, 2019.

[17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[18] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

[19] Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Schölkopf. Kernel methods for measuring independence. *J. Mach. Learn. Res.*, 6:2075–2129, December 2005.

[20] David Roi Hardoon and John Shawe-Taylor. Convergence analysis of kernel canonical correlation analysis: theory and practice. *Machine Learning*, 74:23–38, 2008.

[21] H. O. Hirschfeld. A connection between correlation and contingency. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(4):520–524, 1935.

[22] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, July 1989.

[23] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.

[24] Shao-Lun Huang, Anuran Makur, Lizhong Zheng, and Gregory W Wornell. An information-theoretic approach to universal feature selection in high-dimensional inference. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1336–1340. IEEE, 2017.

[25] Shao-Lun Huang and Xiangxiang Xu. On the sample complexity of hgr maximal correlation functions for large datasets. *IEEE Transactions on Information Theory*, 67(3):1951–1980, 2020.

[26] Su-Yun Huang, Mei-Hsien Lee, and Chuhsing Kate Hsiao. Kernel canonical correlation analysis and its applications to nonlinear measures of association and test of independence. *Institute of Statistical Science: Academia Sinica, Taiwan*, 2006.

[27] Henrik Hult and Filip Lindskog. Multivariate extremes, aggregation and dependence in elliptical distributions. *Advances in Applied Probability*, 34(3):587–608, 2002.

[28] Michael R. Kosorok. Discussion of: Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1270 – 1278, 2009.

[29] Charles J Kowalski. On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 21(1):1–12, 1972.

[30] Yihua Liang, Fei Ma, Yang Li, and Shao-Lun Huang. Person recognition with hgr maximal correlation on multimodal data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1–8, 2021.

[31] David Lopez-Paz, Philipp Hennig, and Bernhard Schölkopf. The randomized dependence coefficient. *Advances in neural information processing systems*, 26, 2013.

[32] Jérémie Mary, Clément Calauzenes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In *International conference on machine learning*, pages 4382–4391. PMLR, 2019.

[33] Jeremie Mary, Clément Calauzènes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4382–4391. PMLR, 09–15 Jun 2019.

[34] Roger B. Nelsen. *An introduction to copulas.* Springer, New York, 2006.

[35] Barnabás Póczos, Zoubin Ghahramani, and Jeff Schneider. Copula-based kernel dependency measures. *arXiv preprint arXiv:1206.4682*, 2012.

[36] Meisam Razaviyayn, Farzan Farnia, and David Tse. Discrete rényi classifiers. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[37] Alfréd Rényi. On measures of dependence. *Acta Mathematica Academiae Scientiarum Hungarica*, 10:441–451, 1959.

[38] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[39] Pablo Romeu-Guallart and Francisco Zamora-Martinez. SML2010. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5RS3S.

[40] B. Schweizer and E. F Wolff. On nonparametric measures of dependence for random variables. *Ann. Statist.*, 9(1):879–885, 1981.

[41] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. 2007.

[42] Viivi Uurtio, João M Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. A tutorial on canonical correlation methods. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.

[43] Saverio Vito. Air Quality. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C59K5F.

[44] Lichen Wang, Jiaxiang Wu, Shao-Lun Huang, Lizhong Zheng, Xiangxiang Xu, Lin Zhang, and Junzhou Huang. An efficient approach to informative feature extraction from multimodal data. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[45] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4590–4594. IEEE, 2015.

[46] Xiangxiang Xu and Shao-Lun Huang. On the asymptotic sample complexity of hgr maximal correlation functions in semi-supervised learning. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 879–886. IEEE, 2019.

[47] Xiangxiang Xu and Shao-Lun Huang. Maximal correlation regression. *IEEE Access*, 8:26591–26601, 2020.

[48] Longfei Yan, W Bastiaan Kleijn, and Thushara Abhayapala. A linear-time independence criterion based on a finite basis approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 202–212. PMLR, 2020.

[49] Xinghao Yang, Weifeng Liu, Wei Liu, and Dacheng Tao. A survey on canonical correlation analysis. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2349–2368, 2021.

[50] I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Contributions are listed in the introduction and summarized in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are presented in the discussion section

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All demonstrations of the proposals in the paper are rigorously detailed in the Appendix, with references where necessary. Unfortunately, due to space constraints, we are unable to give a short proof sketch.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code and datasets are not provided at submission time but will be released upon acceptance, along with the camera-ready version. The synthetic data generation process is fully detailed, and the sources of the real-world datasets used in our experiments are explicitly cited. Additionally, we provide a pseudo-code description of the algorithm and illustrate the construction of the stacked cross-encoder architectures for all three correlation settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No] / [Yes]

Justification: The code and datasets are not provided at submission time but will be released in case of acceptance, along with the camera-ready version. The synthetic data generation process is fully detailed, and the sources of the real-world datasets used in our experiments are explicitly cited. Additionally, we provide a pseudo-code description of the algorithm and illustrate the construction of the stacked cross-encoder architectures for all three correlation settings.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All hyperparameters are documented in the Appendix, including network architectures, training settings, and data simulation parameters. We also conduct sensitivity analyses to assess the robustness of our method with respect to these choices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For each analysis, we performed at least 10 randomized runs (with fixed seeds for reproducibility) to ensure robust and reliable results. We report outcomes using various formats: heatmaps of correlation scores, ranking tables, and visualizations such as boxplots or confidence-interval curves, depending on the context.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational ressources and computation time are detailed in the Appendix. NOus avons également réalisés une étude de sensibilités de temps de calcul par rapport à la taille de l'échantillon et au nombre de vairales.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: the research conducted in the paper is conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Proposition of a new maximal correlation coefficient, without societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets (code, data, and models) are properly referenced, with their original papers. The datasets are sourced from public repositories such as the *UCI Machine Learning Repository*, with URLs provided. The models used in the baselines are implemented using publicly available R or Python libraries, or retrieved from open-access GitHub repositories (with URL). Licenses are indicated wherever applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: details about training, license, limitations, etc. will be communicated in case of acceptance

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The use of large language models (LLMs) was limited strictly to writing assistance, editing, and formatting. No LLMs were used for developing core research ideas, designing experiments, or generating results.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Renyi's axioms

1. General definition: $D(X, Y)$ is defined for all pairs of non-constant random variables X and Y.
2. Symmetry: $D(X, Y) = D(Y, X)$
3. Natural bounds: $0 \leq D(X, Y) \leq 1$
4. Zero under independence: $D(X, Y) = 0$ if and only if X and Y are statistically independent.
5. Invariance under bijective transformations: For all Borel-measurable bijective functions $f, g : \mathbb{R} \to \mathbb{R}$, $D(X, Y) = D(f(X), g(Y))$
6. Unit value under deterministic functional dependence: $D(X, Y) = 1$ if $Y = f(X)$ or $X = g(Y)$, for some Borel-measurable function f or g.
7. Agreement with Pearson correlation in the Gaussian case: If $(X, Y) \sim \mathcal{N}(\mu, \Sigma)$, then $D(X, Y) = |\rho(X, Y)|$
8. where $\rho(X, Y)$ denotes the Pearson correlation coefficient.

B Related Work

Table 1: Comparison of correlation estimation approaches across multiple criteria (*** for good score, * for bad score, based on bivariate illustrations)

Approach	**Non-Lin.**	**Mult.**	**Perf.**	**Noise**	**Halluc.**	**Extreme**	**Rényi**	**Marg. Inv.**
Pearson	✗	✗	*	*	**	✗	✗	✗
Spearman	✗	✗	*	*	**	✓	✗	✓
Kendall	✗	✗	*	*	**	✓	✗	✓
RDC	✓	✓	***	***	*	✓	✓	✓
MIC	✓	✗	***	**	*	✓	✗	✗
dCor	✓	✓	**	**	*	✗	✗	✗
CCA	✗	✓	*	*	**	✗	✗	✗
kCCA	✓	✓	*	*	**	✗	✗	✗
ACE	✗	✓	***	***	***	✗	✓	✗
HGRkde	✗	✗	**	**		✓	✓	✓
HGRnn	✓	✓	**	**	*	✗	✓	✓
HGRlat	✗	✗	*	*	**	✗	✓	✓
dk	✓	✓	**	***	*	✗	✓	✓
sk	✓	✓	**	***	*	✗	✓	✓
SHGR **(ours)**	✓	✓	***	***	***	✓	✓	✓

C Pearson coefficient Analysis

The following figure illustrates the differences between Pearson and Spearman correlation coefficient estimates. In Figure 8b, the relationship is purely monotonic with the exponential function and the Pearson coefficient poorly captures the correlation. It is also observed that the Pearson coefficient is distorted if an extreme value is present (Figure 8d). Finally, both coefficients do not capture nonlinear relationships, such as a quadratic relationship, as shown in Figure 8c.

Sensitivity of Pearson's correlation to extreme values We demonstrate the sensitivity of Pearson's correlation coefficient to extreme values using a simple counterexample. Let X and Y be two independent, centered random variables, such that $\mathbb{E}[X] = \mathbb{E}[Y] = 0$ and $\mathrm{Cov}(X, Y) = 0$. Consider an i.i.d. sample of n pairs $(x_i, y_i)_{i=1}^{n}$ drawn from these variables, with empirical Pearson correlation coefficient ρ_n close to 0 due to independence. Now suppose we add a single extreme data point $(x^*, y^*) = (K, K)$ with $K \gg 1$. Let $\bar{x}_{n+1}$ and $\bar{y}_{n+1}$ denote the empirical means computed over the $n + 1$ observations (including the extreme point). The updated empirical Pearson correlation becomes:

$$\rho_{n+1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}_{n+1})(y_i - \bar{y}_{n+1}) + (K - \bar{x}_{n+1})(K - \bar{y}_{n+1})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x}_{n+1})^2 + (K - \bar{x}_{n+1})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_{n+1})^2 + (K - \bar{y}_{n+1})^2}}$$

Now observe that:

$$\bar{x}_{n+1} = \frac{1}{n+1}\left(\sum_{i=1}^{n} x_i + K\right) \approx \frac{K}{n+1}$$

so that:

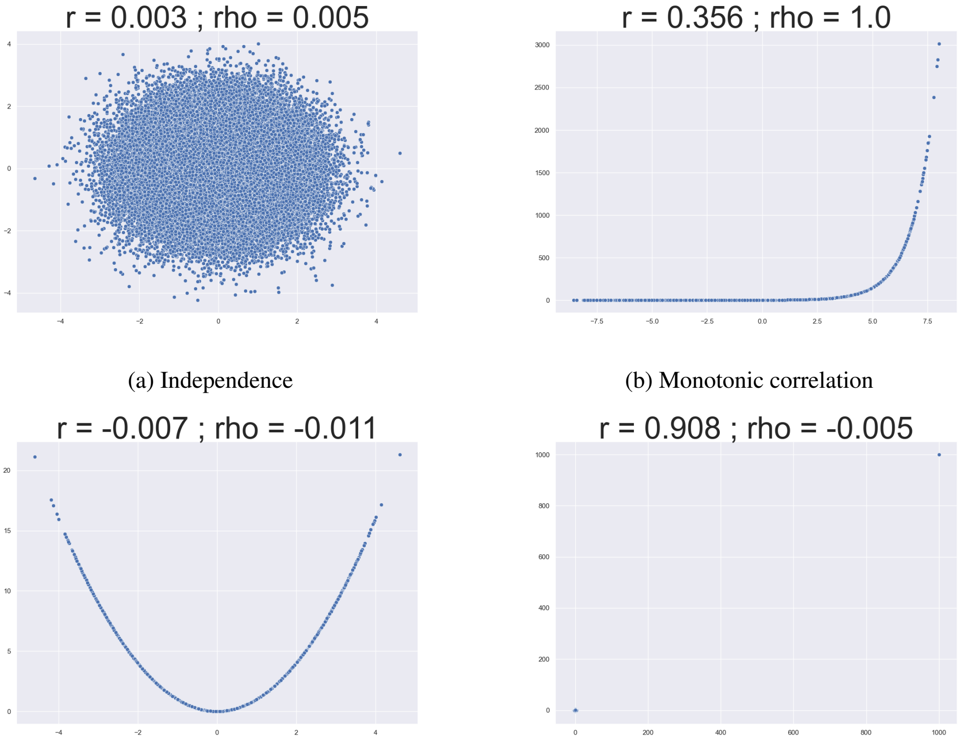
$$K - \bar{x}_{n+1} \approx K - \frac{K}{n+1} = \frac{nK}{n+1} \Rightarrow (K - \bar{x}_{n+1})^2 \approx \left(\frac{nK}{n+1}\right)^2 = \mathcal{O}(K^2)$$

and similarly for Y.

This shows that both the covariance term and the variances are asymptotically dominated by the contribution of the extreme point. Therefore, as $K \to \infty$:

$$\rho_{n+1} \approx \frac{(K - \bar{x}_{n+1})(K - \bar{y}_{n+1})}{\sqrt{(K - \bar{x}_{n+1})^2} \cdot \sqrt{(K - \bar{y}_{n+1})^2}} = 1$$

This confirms that a single well-aligned extreme point can drive the Pearson correlation arbitrarily close to 1, even when the underlying variables are entirely independent. This illustrates the high sensitivity of Pearson's correlation coefficient to outliers and highlights the potential for misleading empirical estimates in the presence of extreme values.



(a) Independence

(b) Monotonic correlation

(c) Quadratic correlation

(d) Independence with an extreme value

Figure 8: Pearson (r) and Spearman (rho) coefficients for different cases

D Proofs

In the proofs, when there is no ambiguity, we simplify notation by writing f in place of f_u, g in place of g_v, F in place of F_U or $F_{f_u(U)}$, G in place of G_V or $G_{g_v(V)}$, depending of the situation.

D.1 Proof of Proposition 1

We have

$$\widehat{SHGR} - SHGR_\Theta \quad = \quad \sup_{f,g} \rho(f(u), g(v)) - \sup_{f^*,g^*} \rho(f^*(u), g^*(v))$$

where f^*, g^* stand for the neural approximation functions.

Let $\xi > 0$. By definition of the supremum, there exist functions f_0, g_0 in $\mathcal{E}(\mathcal{U})$ and $\mathcal{E}(\mathcal{V})$ such that:

$$\rho(f_0(u)g_0(v)) \quad \geq \quad \sup_{f,g} \rho(f(u)g(v)) - \xi.$$

Therefore:

$$\begin{aligned}
\widehat{SHGR} - SHGR_\Theta &\leq \rho(f_0(u), g_0(v)) - \sup_{f^*,g^*} \rho(f^*(u), g^*(v)) + \xi \\
&\leq \rho(f_0(u), g_0(v)) - \rho(f_0^*(u), g_0^*(v)) + \xi \qquad (1) \\
&= r(\widehat{F}_{f_0(u)}, (f_0(u))\widehat{G}_{g_0(v)}(g_0(v))) - r(\widehat{F}_{f_0^*(u)}(f_0^*(u)), \widehat{G}_{g_0^*(v)}(g_0^*(v))) + \xi,
\end{aligned}$$

where f_0^* and g_0^* stand for the neural approximations of f_0 and g_0 respectively. Since the random variables U and V are bounded on the compact supports $\mathcal{E}(\mathcal{U}))$ and $\mathcal{E}(\mathcal{V}))$, we can prove by the universality of the neural approximation [22] that when n tends to infinity,

$$r(\widehat{F}_{f_0(u)}(f_0(u)), \widehat{G}_{g_0(v)}(g_0(v))) - r(\widehat{F}_{f_0^*(u)}(f_0^*(u)), \widehat{G}_{g_0^*(v)}(g_0^*(v))) \quad \rightarrow \quad 0.$$

The argument of such a proof is similar to those used in [15]. Finally, since the inequality (1) holds for every $\xi > 0$, we obtain the result.

D.2 Proof of Proposition 2

We have

$$\widehat{SHGR} - SHGR \quad = \quad \sup_{f,g} r(\widehat{F}(f(u)), \widehat{G}(g(v))) - \sup_{f,g} r(F(f(U)), G(g(V))).$$

Let $\xi > 0$. By definition of the supremum, there exist functions f_0, g_0 in $\mathcal{E}(\mathcal{U})$ and $\mathcal{E}(\mathcal{V})$ such that:

$$r(\widehat{F}(f_0(u)), \widehat{G}(g_0(v))) \quad \geq \quad \sup_{f,g} r(\widehat{F}(f(u)), \widehat{G}(g(v))) - \xi.$$

Therefore:

$$\begin{aligned}
\widehat{SHGR} - SHGR \quad &= \quad r(\widehat{F}(f_0(u)), \widehat{G}(g_0(v))) - \sup_{f,g} r(F(f(U)), G(g(V))) + \xi \\
&\leq \quad r(\widehat{F}(f_0(u)), \widehat{G}(g_0(v))) - r(F(f_0(U)), G(g_0(V))) + \xi \qquad (2) \\
&= \quad \frac{1}{n} \sum_{i=1}^{n} \Big(\widehat{F}'(f_0(u_i))\widehat{G}'(g_0(v_i)) - \mathbb{E}(F'(f_0(U))G'(g_0(V))) \Big) + \xi \\
&:= \quad \frac{1}{n} \sum_{i=1}^{n} A_i + \xi,
\end{aligned}$$

where, for abbreviation, $F'(x) = (F(x) - \mu_F)/\sigma_F$ denotes the normalized version (centered and reduced) of F. We can decompose

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^{n} A_i \quad &= \quad \frac{1}{n} \sum_{i=1}^{n} \Big(\widehat{F}'(f_0(u_i))\widehat{G}'(g_0(v_i)) - F'(f_0(u_i))G'(g_0(v_i)) \Big) \\
&\quad + \frac{1}{n} \sum_{i=1}^{n} (F'(f_0(u_i))G'(g_0(v_i)) - \mathbb{E}(F'(f_0(U))G'(g_0(V)))) \\
&:= \quad \frac{1}{n} \sum_{i=1}^{n} B_i + \frac{1}{n} \sum_{i=1}^{n} C_i.
\end{aligned}$$

We have

$$|\widehat{F}'(f_0(u_i))\widehat{G}'(g_0(v_i)) - F'(f_0(u_i))G'(g_0(v_i))| \leq |\widehat{F}'(f_0(u_i)) - F'(f_0(u_i))| \sup |\widehat{G}'(g_0(v_i))| + |\widehat{G}'(g_0(v_i)) - G'(g_0(v_i))| \sup |F'(f_0(u_i))|.$$

Since $\widehat{G}'$ and F' are bounded, by the Glivenko-Cantelli Theorem we obtain the uniform convergence: $\sup B_i \to 0$, as n tends to infinity and by the Cesàro Lemma we deduce that

$$\frac{1}{n}\sum_{i=1}^{n} B_i \quad \to \quad 0.$$

Since the C_i are iid centered random variables, for $i = 1, , \cdots, n$, by the law of large numbers, we have almost surely

$$\frac{1}{n}\sum_{i=1}^{n} C_i \quad \to \quad 0,$$

and finally

$$\frac{1}{n}\sum_{i=1}^{n} A_i \quad \to \quad 0.$$

Since inequality (2) holds for every $\xi > 0$, we have:

$$\widehat{SHGR} - SHGR \quad \to \quad 0,$$

as n tends to ∞.

D.3 Proof of Proposition 3

- Points 1-3 are immediate.
- Point 4. If U and V are independent, we obviously have $SHGR = 0$. Conversely, let U and V be two continuous dependent random variable. It implies that $C_{U,V}$ the copula associated to (U, V) is not the independent copula. By Proposition 1 in [3] their exists a non null copula coefficient ρ associated to $C_{U,V}$, that is there exists two functions f and g such that $\rho = \mathbb{E}(f(U)g(V)) = \delta \neq 0$. Write $X = f(U)$, $Y = g(V)$. We get

  $$\rho \quad = \quad \int xy f_{X,Y}(x, y)dxdy = \delta \neq 0,$$

  where $f_{X,Y}$ denotes the joint density of (X, Y). Without loss of generality, we can assume $\delta > 0$. Then there exists a set S such that for all $x, y \in S$, $f_{X,Y}(x, y) > 0$ and $F_X(x) > 0$, $F_Y(y) > 0$. It implies that

  $$\begin{aligned} \mathbb{E}(F_X(X)F_Y(Y)) &= \int F_X(x)F_Y(y)f_{X,Y}(x, y)dxdy \\ &\geq \int_S F_X(x)F_Y(y)f_{X,Y}(x, y)dxdy > 0, \end{aligned}$$

  which gives the result.
- Point 5 follows from the construction of the SHGR.
- Point 6. If $U = f(V)$, choosing $f_u = identity$ and $f_v = f$, we obtain $r(F_{f_u(U)}(f_u(U)), F_{f_v(V)}(f_v(V)) = r(F_U(U), F_U(U)) = 1$. By construction of the SHGR, by normalizing with a function g such that $\mathbb{E}(g(U)) = 0$ and $\mathbb{E}(g^2(U)) = 1$, we obtain the result.
- Point 7. In the Gaussian case we have the following relation (see for instance [27]):

  $$\rho_S(X_i, X_j) \;=\; \frac{6}{\pi}\ \arcsin\Big(\tfrac{\rho_{ij}}{2}\Big),$$

  which yields the result.

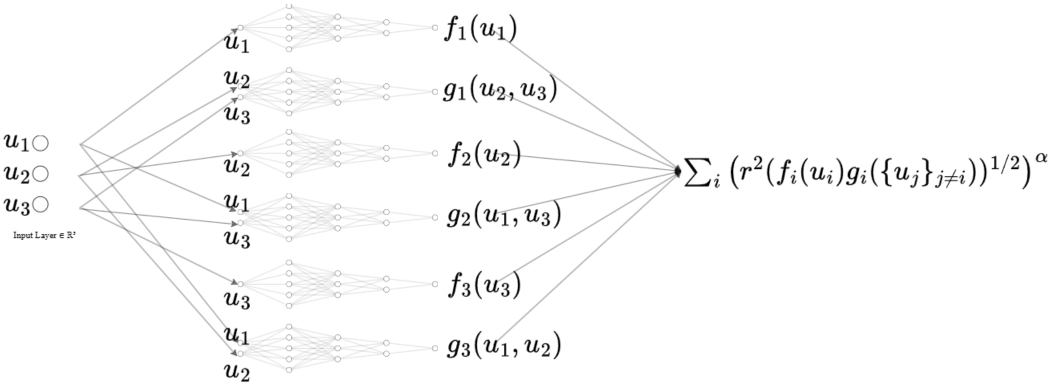


Figure 9: A Cross-Encoder Architecture for multivariate case (illustration with $p = 3$ variables)

E SHGR Algorithm

In addition to figures 1a and 1b, figure 9 shows an example of architecture on a set of 3 variables.

Based on neural networks, the SHGR algorithm is defined in two stages: one for building the architecture and the other for training the model. At each epoch, the correlation is measured on the inputs (in whole), and the model retains the model with the lowest loss (i.e. the highest correlation) on the inputs. If the results no longer improve (to within an epsilon) during a given number of iterations, then learning stops.

```
Input: Input matrices $X$, [$Y$], number of epochs $E$, batch size $B$, learning rate $\eta$, hidden layer
       dimensions $dimHL$, early stopping threshold $\varepsilon$, maximum patience $P_{\max}$, correlation
       type (Spearman or Pearson)
Output: Trained model $M^*$, encoded outputs, loss history
Initialize model $M$ with input dimensions of $X$ [and $Y$], hidden layer dimension $dimDL$;
  Initialize optimizer (AdamW) with learning rate $\eta$;
Convert $X$ [and $Y$] ;
for epoch $= 1$ to $E$ do
    Shuffle $X$ [and $Y$] and create mini-batches;
    foreach batch $(x_b[, y_b])$ do
        Compute encoded representation using $M(x_b[, y_b])$;
        Compute loss $L$:
            • If Spearman: use $L \leftarrow$ SHGR_rank_correlation$(x_b[, y_b])$
            • If Pearson: use $L \leftarrow$ SHGR_linear_correlation$(x_b[, y_b])$
        Perform backpropagation and update model weights;
        Record loss;
    end
    Compute full-batch loss on $(X, Y)$ as $L_{\text{val}}$;
    Apply early stopping if $L_{\text{val}}$ does not improve beyond $\varepsilon$ for $P_{\max}$ epochs;
    If $L_{\text{val}}$ improves, save current model as best model $M^*$;
end
Compute encoded outputs using best model $M^*$ on all inputs;
return $M^*$, encoded outputs, loss history, best epoch
```

Algorithm 1: `train_SHGR`: Training of the SHGR model

Remark: as is classic with neural networks, it's customary to apply a standard scaler to the input data. Even though the algorithm also works with raw data, we noticed an improvement when using a standard scaler. The encoded is then standardized, allowing the HGR constraint to be respected: the transformations are indeed of zero expectation and unit variance.
Examples of possible architecture are defined below in the illustration.

F Experiments Details

F.1 Computational resources

The computations were performed on a personal desktop computer with the following specifications: NVIDIA GeForce RTX 4080 graphics card, 64GB of memory (but the memory usage did not exceed 30GB), Intel i9-14900KF processor.

F.2 Model Architecture and hyperparameters

F.2.1 Hyperparameter

For the illustration, we have chosen the following hyperparameters:

- epoch number: 200 maximum
- batch size: 64
- learning rate: $10e^{-3}$
- hidden layer dimensions : $[64, 32, 16, 8]$
- epsilon for early stopping : 0.5
- iteration max for patience early stopping: 20
- penalization for differentiable ranks (as defined in [4]): 1
- α power parameter in `SHGR` loss function: 2.0

F.2.2 Architectures

The architecture of our `SHGR` for bivariate (pairwise) correlations between variables $u_1, \cdots, u_p$ is defined as follows:

- an encoder for each u_i with $i = 1, \cdots, p$ consisting of 5 hidden Layers:
  - HL_1 of dimensions $(1, 64)$
  - HL_2 of dimensions $(64, 32)$
  - HL_3 of dimensions $(32, 16)$
  - HL_4 of dimensions $(16, 8)$
  - HL_5 of dimensions $(8, 1)$
- activation function are all $Tanh()$ expect the first one that is $Relu$.

The architecture of our `SHGR` for multivariate correlations between variables $u_1, \cdots, u_p$ is defined as follows:

- an encoder for u_i with $i = 1, \cdots, p$ consisting of 5 hidden Layers:
  - HL_1 of dimensions $(1, 64)$
  - HL_2 of dimensions $(64, 32)$
  - HL_3 of dimensions $(32, 16)$
  - HL_4 of dimensions $(16, 8)$
  - HL_5 of dimensions $(8, 1)$
- an encoder for $\{u_j\} \neq u_i$ with $j = 1, \cdots, p$ consisting of 5 hidden Layers:
  - HL_1 of dimensions $(p - 1, 64)$
  - HL_2 of dimensions $(64, 32)$
  - HL_3 of dimensions $(32, 16)$
  - HL_4 of dimensions $(16, 8)$
  - HL_5 of dimensions $(8, 1)$
- activation function are all $Tanh()$ expect the first one that is $Relu$.

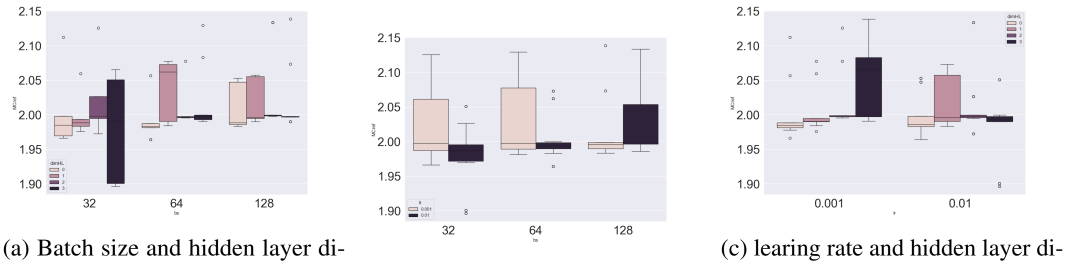
The architecture of our SHGR for full correlations between u (of dimensions p) and v (of dimensions q) is defined as follows:

- an encoder for u consisting of 5 hidden Layers:
  - HL_1 of dimensions $(p, 64)$
  - HL_2 of dimensions $(64, 32)$
  - HL_3 of dimensions $(32, 16)$
  - HL_4 of dimensions $(16, 8)$
  - HL_5 of dimensions $(8, 1)$
- an encoder for v consisting of 5 hidden Layers:
  - HL_1 of dimensions $(q, 64)$
  - HL_2 of dimensions $(64, 32)$
  - HL_3 of dimensions $(32, 16)$
  - HL_4 of dimensions $(16, 8)$
  - HL_5 of dimensions $(8, 1)$
- activation function are all $Tanh()$ expect the first one that is $Relu$.

F.2.3 Sensibilities

Even though this criterion has been analyzed, as the calculation time is very satisfactory, it does not come into play in the following evaluation. Our aim here is rather to analyze and optimize the architecture and hyperparameters of the estimated correlation, which should be as close as possible to the reference correlation. Here we use the simulations shown in the numerical illustration and the bivariate correlations.. For the architecture, we tested the following parameters:

- batch size: 32, 64 and 128
- learning rate: $10e^{-2}$, $10e^{-3}$ and $10e^{-4}$
- hidden layer dimensions: $1 : [5, 5, 5, 5]$, $2 : [10, 10, 10, 10]$, $3 : [64, 32, 16, 8]$ and $4 : [128, 64, 32, 16]$



(a) Batch size and hidden layer dimensions

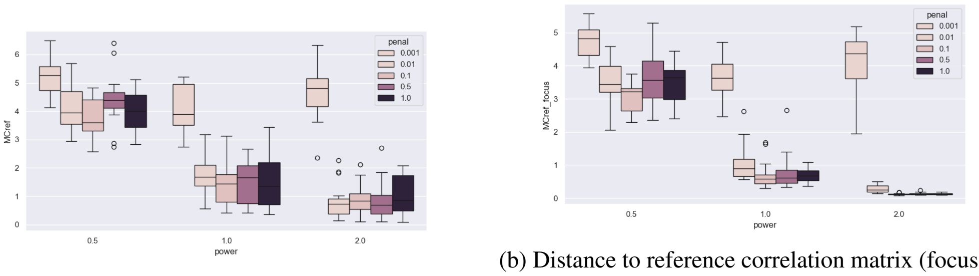
(b) Batch size and learning rate

(c) learing rate and hidden layer dimensions

Figure 10: Sensitivities of architecture and hyperparameter of cross encoders

Concerning the learning rate, except with a batch size of 128, which seems too high to capture correlations on a dataset, using $10e^-3$ seems preferable as it reduces the distance to the reference correlation (Figures 10b and 10c. Concerning batch size, using 64 seems more stable than 32 and 128 (Figures 10a and 10b), even if 32 seems slightly better in median. Finally, for the dimensions of hidden layers, 3:[64,32,16,8] seems more stable than the others. (Figures 10a and 10c).

We also analyzed the sensitivity and impact of the results to the hyperparameter power α in the loss function and to the differentiable rank regularization parameter as defined in [4]. Figure 11 presents the results obtained. We can clearly see the benefits of using a power of 2 for correlations. This allows the networks to quickly focus on the strongest correlations. With regard to differentiable rank regularization, the values 0.1, 0.5 and 1 seem fairly comparable. We choose to use the default value of 1.



(a) Distance to reference correlation matrix according to power (α) and regularization of differentiable ranks

(b) Distance to reference correlation matrix (focus) according to power (α) and regularization of differentiable ranks

Figure 11: Sensitivities of power and regularization of differentiable ranks: with bivariate correlations

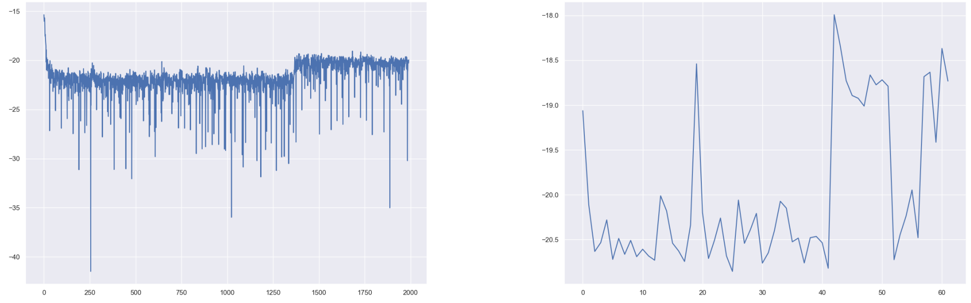
F.3 Synthetic Dataset Generation

F.3.1 Bivariate correlations

We generate a synthetic dataset composed of n samples and 11 input features $(X_0, \ldots, X_{10})$, where the first 5 variables are drawn independently from a standard normal distribution. The remaining features are nonlinearly correlated with the first ones. For performance analysis, we generate noise-free data ($s = 0$). For robustness analysis, the s noise increases progressively. Specifically, the data generation process is as follows:

- X_0 to $X_4 \sim \mathcal{N}(0, 1)$ independently,
- $X_5 \sim \mathcal{N}(X_0^3, 3s)$: cubic relationship with X_0 plus Gaussian noise,
- $X_6 \sim \mathcal{N}(\sin(2X_1), s/5)$: sinusoidal transformation of X_1 plus noise,
- X_7 is a discretized, piecewise-constant function of X_2, mapped into 7 quantile bins, and then perturbed with $\mathcal{N}(0, s/10)$ noise,
- $X_8 \sim \mathcal{N}(\exp(X_3), s)$: exponential transformation of X_3 with additive noise,
- $X_9 \sim \mathcal{N}(X_4^2, s)$: squared transformation of X_4 with noise,
- $X_10 \sim \mathcal{N}(0, 1)^2$: squared standard normal variable, i.e., $\chi^2(1)$-distributed.

Figure 12 gives an illustration of correlations with the pairplot (2a), the SHGR transformation (with 2b,) and loss functions. Figure 2 compare the reference correlation matrix and its estimation with SHGR.



(a) SHGR Loss Function on batch

(b) SHGR Loss Function on input

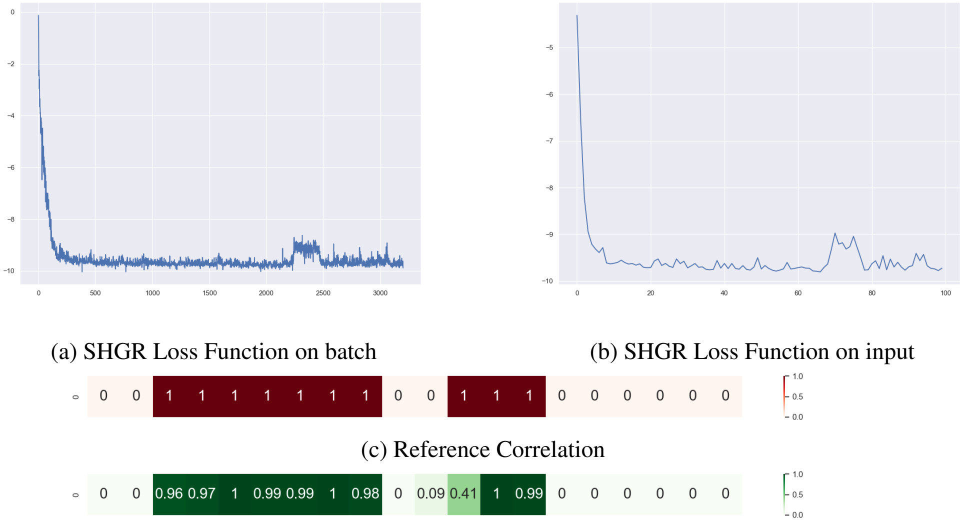
Figure 12: SHGR training: loss function for bivariate correlations

F.3.2 Multivariate correlations

We generate a synthetic dataset composed of n samples and 20 input features $(X_0, \ldots, X_{19})$, where the first 11 variables are drawn independently from a standard normal distribution. The remaining features include nonlinear and additive combinations, introducing structured dependencies. For performance analysis, we generate noise-free data ($s = 0$). For robustness analysis, the s noise increases progressively. Specifically, the data generation process is as follows:

- X_0 to $X_{10} \sim \mathcal{N}(0, 1)$ (independent standard normal variables),
- $X_{11} = X_2^2 + X_3^2 + \varepsilon_{11}$,
- $X_{12} = 3X_4 + 2X_5 + 0.9X_6^2 + \varepsilon_{12}$,
- $X_{13} = \sin(X_7) + 0.5X_8 + \varepsilon_{13}$,
- X_{14} to $X_{17} \sim \mathcal{N}(0, 1)$ (independent noise),
- $X_{18} = Z_{18}^2$, with $Z_{18} \sim \mathcal{N}(0, 1)$,
- $X_{19} = Z_{19}^2$, with $Z_{19} \sim \mathcal{N}(0, 1)$,

where each $\varepsilon_i \sim \mathcal{N}(0, s^2)$ is an optional noise term added to introduce variability. The seed is randomly initialized unless otherwise specified. Figure 13 gives an illustration for multivariate correlations.



(a) SHGR Loss Function on batch

(b) SHGR Loss Function on input

(c) Reference Correlation

(d) SHGR Correlation

Figure 13: Multivariate correlations: reference vs SHGR

F.3.3 Full correlations

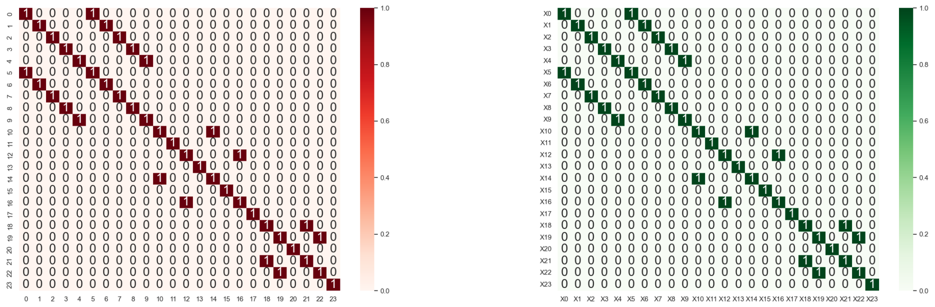
We generate a synthetic dataset consisting of n samples and p input variables $(X_1, \ldots, X_p)$. The data is constructed as a weighted combination of two independent standard Gaussian matrices, controlled by a mixing parameter $\alpha \in [0, 1]$. Specifically, we first generate a base matrix $Z \sim \mathcal{N}(0, I_p)$ of size $n \times p$, then add a perturbation to create X as follows:

$$X = \alpha \cdot Z + (1 - \alpha) \cdot Z',$$

where Z' is an independent Gaussian matrix of the same dimension. When $\alpha = 1$, the data is purely Gaussian with no noise; when $\alpha = 0$, the data is entirely random. This allows control over the signal-to-noise ratio in the generated features. The final output is returned as a labeled DataFrame with columns $(X_1, \ldots, X_p)$.

F.3.4 Mixed bivariate correlations

A mixed-data version was designed to handle both quantitative and categorical variables within a dataset. Specifically, the correlation ratio is used to assess correlations between numerical and categorical variables, while measures such as Cramér's V are employed for correlations between categorical variables. By way of illustration, we generate a dataset with mixed correlations (between numeric, between numeric and categorical, and between categorical). We then apply the `SHGR` approach, applicable to mixed data, and obtain the matrix below, in comparison to the true (reference) matrix. Figure 14 gives an illustration of the mixed correlation matrix estimation.



(a) Reference mixed bivariate correlations

(b) `SHGR` mixed bivariate correlations

Figure 14: Application of `SHGR` on mixed data

F.4 State-of-the-Art Methods

We estimate nonlinear correlations using `SHGR` and compare them with alternative methods; note that some baselines are unavailable in the multivariate and full cases.

- The Pearson correlation coefficient from python library *numpy*: *Pearson*
- The Spearman correlation coefficient from python library *numpy*: *Spearman*
- The Kendall correlation coefficient from python library *numpy*: *Kendall*
- The Randomized Dependence Coefficient ([31]) from R library *AlterCorr*: *RDC*
- The Mutual Information Criterion ([38]) from R library *AlterCorr*: *MIC*
- The Distance Correlation ([41]) from R library *AlterCorr*: *dCor*
- Canonical Correlation Analysis from R library *stats*: *CCA*
- Kernel Canonical Correlation Analysis ([2]) from R library *kernlab*: *kCCA* [but not integrated because unstable and too time-consuming to calculate]
- Alternating Conditional Expectations ([7]) from R library *acepack*: *ACE*
- HGR estimation with kernel ([33]) from python package *maxcorr* [6] and from github repository *HGR_NN* [7] for multivariate correlations: *HGRkde*
- HGR estimation with Neural Net ([16]) from python package *maxcorr* and from python github *HGR_NN* for multivariate and full correlations: *HGRnn*
- HGR estimation with Lattice ([14]) from python package *maxcorr*: *HGRlat*
- HGR estimation with double kernel ([14]) from python package *maxcorr*: *dk*
- HGR estimation with simple kernel ([14]) from python package *maxcorr*: *sk*
- The Normalized Hilbert-Schmidt Independence Criterion ([18]) from python library *HSIC*[8]: *NHSIC*

[6] `https://pypi.org/project/maxcorr/0.1.1/`

[7] `https://github.com/fairml-research/HGR_NN`

[8] `https://github.com/amber0309/HSIC`

- The Hellinger Correlation ([11]) from R library *HellCor*: *HR* [but not integrated because inefficient, limited to pairwise correlations and too time-consuming to calculate] We perform our `SHGR` model for 100, 200, and 500 epochs with early stopping. A sensitivity analysis of the

The python package *maxcorr* is available from this github repository with the following license: *MIT License, Copyright (c) 2024 Luca Giuliani, Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.*

The python package *numpy* is available with the following license:*Copyright (c) 2005-2024, NumPy Developers. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: i) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. ii) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. iii) Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.*

The License of R package *AlterCorr* is *GPL2* The github repository *HGR_NN* is public.

The license of R package *acepack* is as follow : *License: MIT + file LICENSE Copyright (c) <YEAR>, <COPYRIGHT HOLDER> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.*

F.5 Performance analysis details

In this section, we analyze the performance of the methods, i.e. their ability to identify perfectly non-linear correlations. These performances are analyzed for the estimation of bivariate, multivariate and full correlations.

F.5.1 Bivariate Correlation

The objective is that:

i) the estimated correlation matrix is as close as possible to the reference correlation matrix. We therefore analyze the results based on the distance to the reference correlation matrix:

$$d\_CM_{ref} := \sum_i \sum_j |CM_{est}[i,j] - CM_{ref}[i,j]|$$

where $CM_{est}[i,j] \in [0,1]$ (resp. $CM_{ref}[i,j] \in [0,1]$) is the estimated (resp. reference) correlation between variables i and j.

ii) the 5 nonlinear correlations are exactly identified. We therefore analyze the results based on the focused distance to the reference correlation matrix:

$$d\_CM_{refFocus} := \sum_i \sum_j |CM_{est}[i,j] - CM_{ref}[i,j]| \times mask$$

where $mask$ filters the 5 coefficients of interest. It is indeed important to measure performance on these 5 correlations, as some coefficients introduce spurious correlations in their estimates, which can degrade overall results.

In Figure 15 we can observe that the methods *ACE* and SHGR achieve the best scores regarding the distance to the reference correlation matrix (d_CM_ref. When focusing on the five non-linear correlations, we notice that the methods *ACE, RDC, MIC* and our approach *SHGR* yield the best results. This can be explained by the fact that these methods are able to effectively identify non-linear correlations. However, the methods *RDC* and *MIC* tend to introduce false correlations where none exist, which degrades their performance relative to the reference matrix.

F.5.2 Multivariate Correlation

In Figure 16, we observe that SHGR gives good results but not as good as the *ACE* method. It is, however, better than the other methods *RDC, dCor, CCA, HGRnn, dk, sk* and *NHSIC*.

F.5.3 Full Correlation

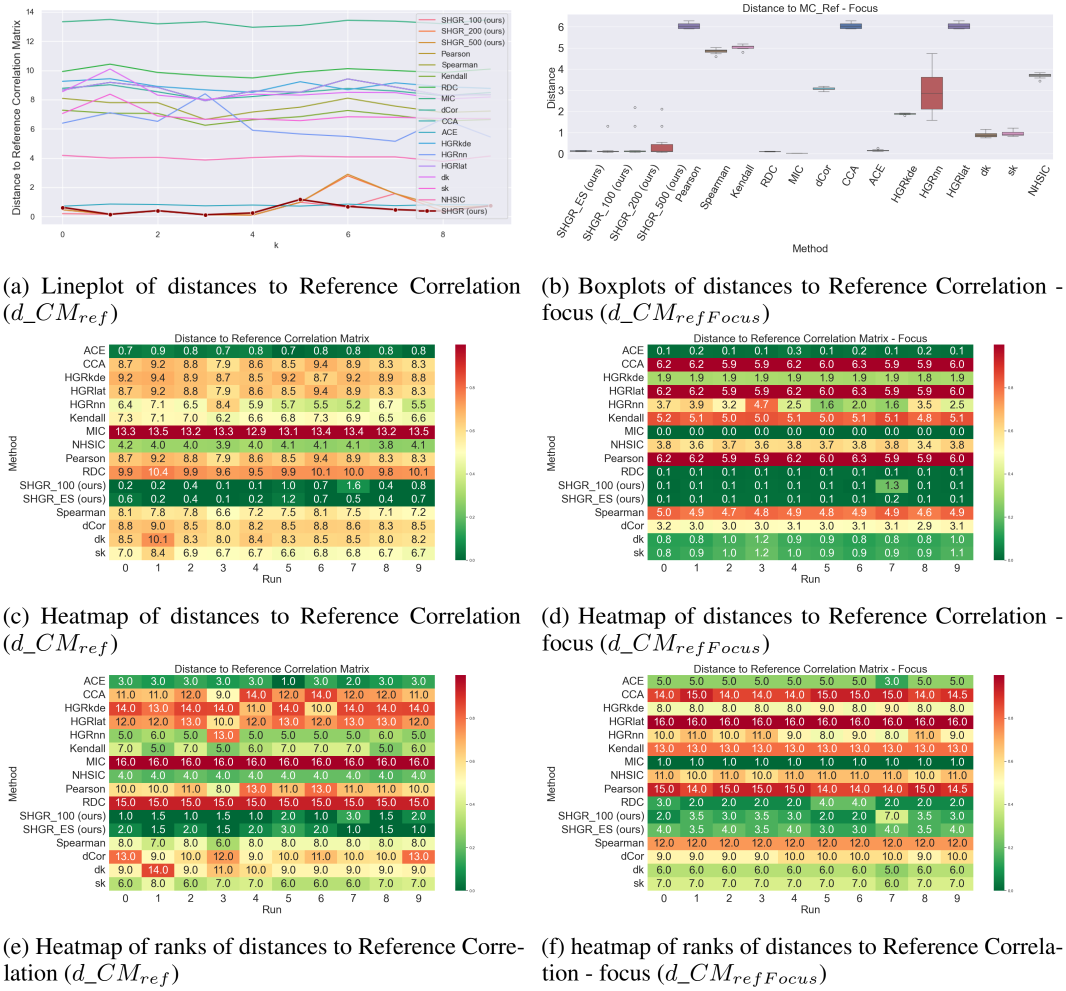
In Figure 17, we see that only the methods SHGR, *CCA, RDC, dk* and *sk* manage to capture perfectly non-linear correlations between the two datasets. The methods *HGRnn, NHSIC and dCor* show deviations from the reference correlation.

F.6 Robustness to Noise

Here, we analyze the robustness to noise of methods: their performance in the presence of white noise (null average). We therefore propose to reproduce the previous simulations while introducing noise that increases with each simulation. Under these conditions, the true correlation is unknown. However, following the analysis of [31], the metrics producing the highest coefficients will be considered the most robust. For this reason, we retain the same metric as in the previous section: the distance to reference correlations. In this scenario, the noise is progressively introduced (the x-axis represents increasing noise from left to right).

F.6.1 Bivariate Correlation

For the analysis of bivariate correlations, through the correlation matrix, we use the same two metrics as for performance: distance to the reference correlation matrix and distance to the 5 non-linear correlations only. In Figure 18, we can see that the SHGR approach outperforms the other methods. Indeed, it gives the lowest distances to reference correlations, even though the *ACE* method still delivers good results. However, it's interesting to note that the performance of *ACE* deteriorates rapidly with the addition of noise: the SHGR approach remains below, i.e., closer to the reference correlations. By analyzing the distances to the focus matrix, i.e. to the 5 non-linked correlations, we can see that the *MIC* method excels in the absence of noise, but its performance deteriorates sharply with increasing noise.



(a) Lineplot of distances to Reference Correlation (d_CM_{ref})

(b) Boxplots of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

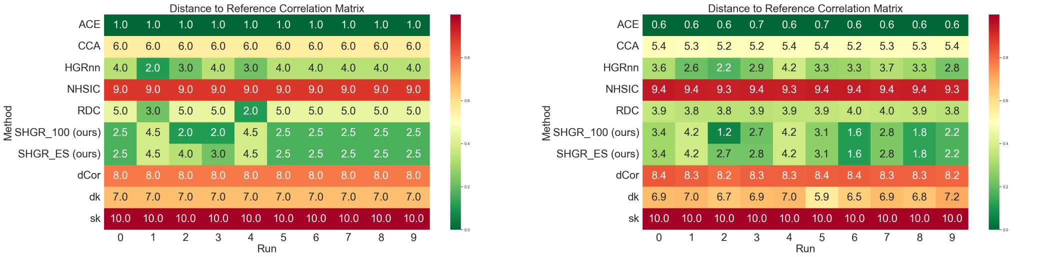
(c) Heatmap of distances to Reference Correlation (d_CM_{ref})

(d) Heatmap of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

(e) Heatmap of ranks of distances to Reference Correlation (d_CM_{ref})

(f) heatmap of ranks of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

Figure 15: Performance: distance to Reference Correlation Matrix (lower is better) ; results for bivariate correlations



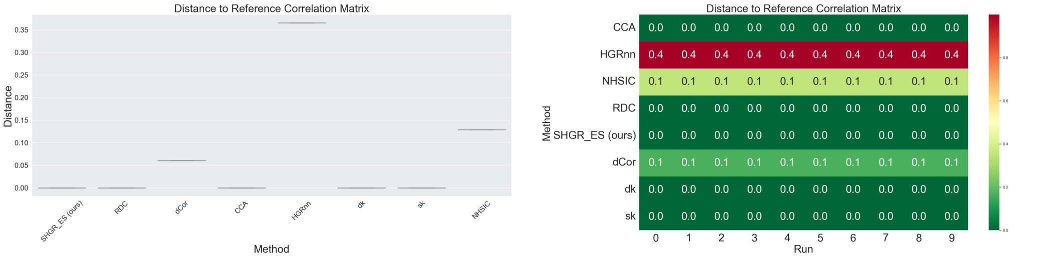
(a) Heatmap of ranks of distances to Reference Correlation (d_CM_{ref})

(b) Heatmap of distances to Reference Correlation (d_CM_{ref})

Figure 16: Performance: distance to Reference Correlation Matrix (lower is better) ; results for multivariate correlations

F.6.2 Multivariate Correlation

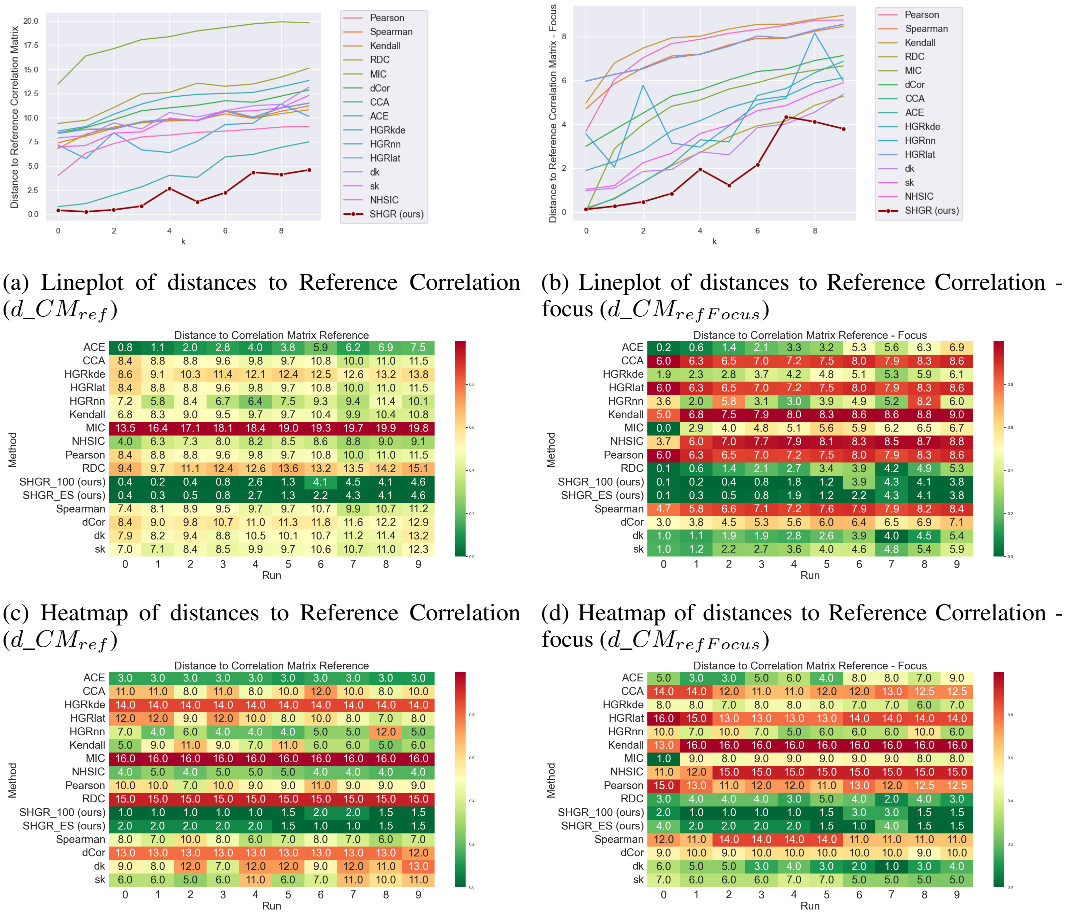
In Figure 19, we see that the SHGR method gives the best results: smallest distance to reference correlations. Once again, the *ACE* method is significantly better in the absence of noise, but results deteriorate with the addition of noise; our SHGR approach remains more stable.



(a) Boxplots of distances to Reference Correlation

(b) Heatmap of distances to Reference Correlation

Figure 17: Performance: distance to Reference Correlation Matrix (lower is better) ; results for full correlations



(a) Lineplot of distances to Reference Correlation (d_CM_{ref})

(b) Lineplot of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

(c) Heatmap of distances to Reference Correlation (d_CM_{ref})

(d) Heatmap of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

(e) Heatmap of ranks of distances to Reference Correlation (d_CM_{ref})

(f) heatmap of ranks of distances to Reference Correlation - focus ($d\_CM_{refFocus}$)

Figure 18: Robustness to noise: distance to Reference Correlation Matrix (lower is better) ; results for bivariate correlations

F.7 Robustness to Hallucination

In this section, we evaluate the correlation measures to avoid hallucination, indicating correlation on independent data (ten independent Gaussian variables). For each context, we simulate independent data and apply the correlation coefficients. In Figure 20, we observe that our approach SHGR remains very robust to hallucinations, whatever the type of correlations estimated.

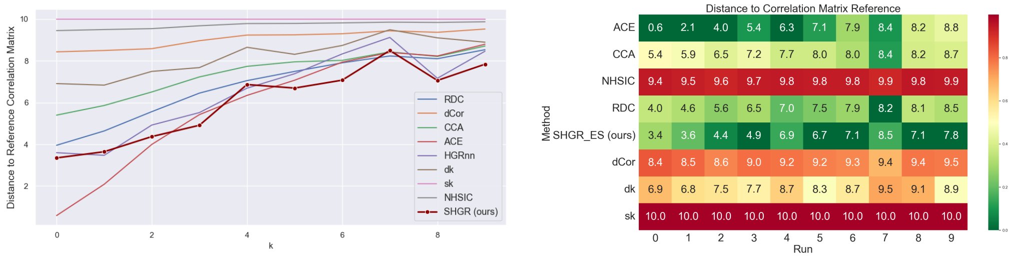


Figure 19: Robustness to noise: distance to Reference Correlation Matrix (lower is better) ; results for multivariate correlations

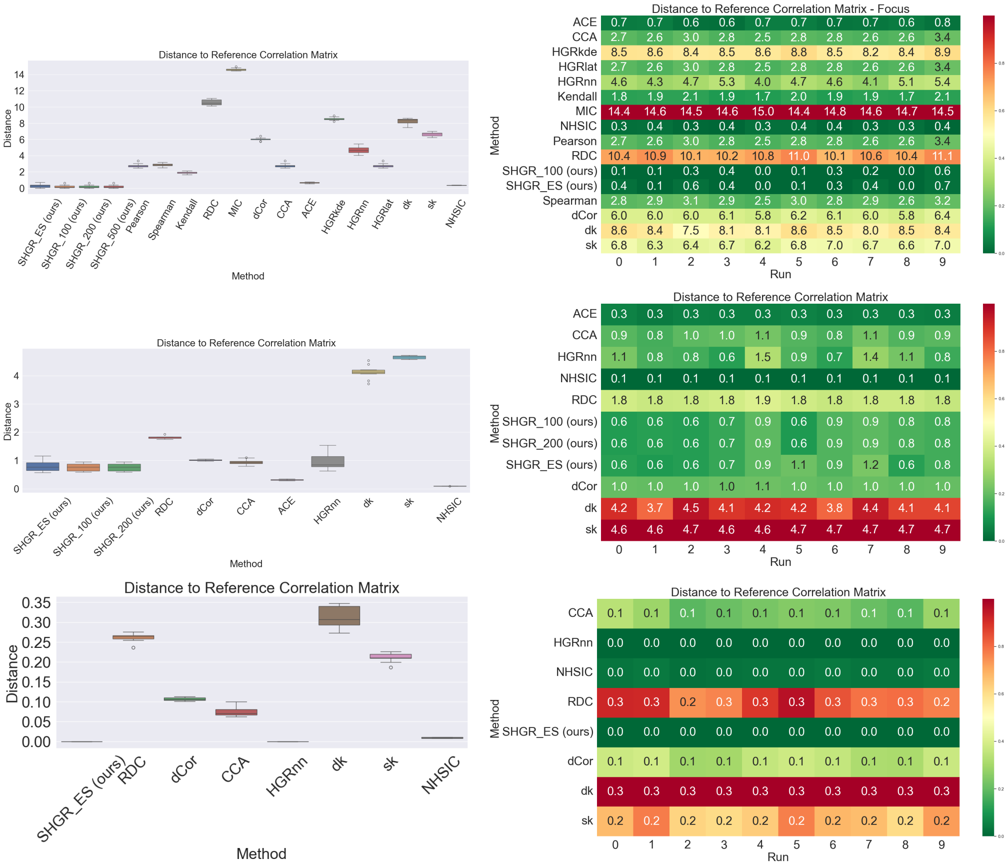
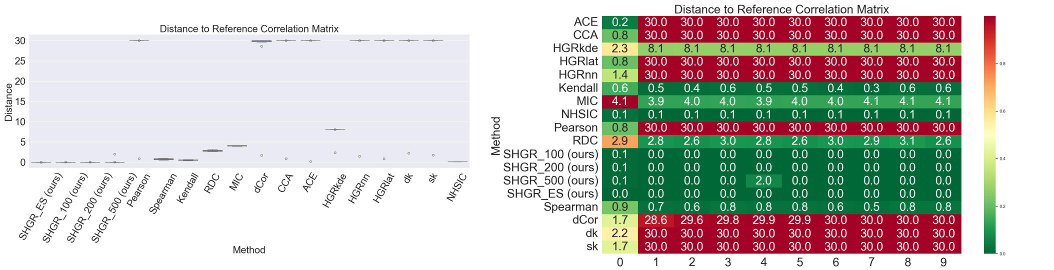


Figure 20: Robustness to hallucination: distance to Reference Correlation Matrix (lower is better) ; for bivariate correlations (up), multivariate correlations and full correlations (down)

F.8 Robustness to Extreme Values

In this section, we evaluate the correlation measures to avoid hallucinations due to extreme values. As above, we simulate independent data with extreme values (except for the first iteration, to measure the impacts). In Figure 21, we see that the approaches *ACE, CCA, HGRnn, HGRlat, Pearson, dk, sk* and *dCor* obtain values of 30 and are therefore very sensitive to extreme values. Methods *HGRkde, MIC* and *RDC* show values significantly different from 0 (due to hallucinations). Methods with low values are *Spearman, Kendall, NHSIC* and SHGR.



(a) Boxplots of distance to Reference Correlation Matrix

(b) Heatmap of distance to Reference Correlation Matrix

Figure 21: Robustness to Extreme Values: distance to Reference Correlation Matrix (lower is better) ; for bivariate correlations

F.9 Significance Test

SHGR optimizes the Pearson linear correlation coefficient based on transformations of the original variables, in line with the theoretical definition of HGR. As such, it is possible to perform an asymptotic significance test on the obtained correlation coefficient (when n is sufficiently large) [29]. Specifically, we test the following null and alternative hypotheses:

$$(H_0):\ SHGR_{\Theta}(u,v)=0 \ \text{ versus } \ (H_1):\ SHGR_{\Theta}(u,v)\neq 0.$$

We apply the existing significance test for the Spearman coefficient (more robust than the Pearson test, also sensitive to extreme values), using it on the transformed variables provided by the method. We evaluate the performance of this test across several bivariate and multivariate correlations, which we progressively add noise to. This test is compared to the ones available for *MIC, RDC* and *RDC* for nine types of relationships. Figure 22 shows the p-value (y-axis) and the noise addition (x-axis) (increasing towards the right). We can see that the significance test is fairly low for the SHGR approach, i.e., the test generally rejects the null hypothesis of the correlation coefficient fairly robustly.

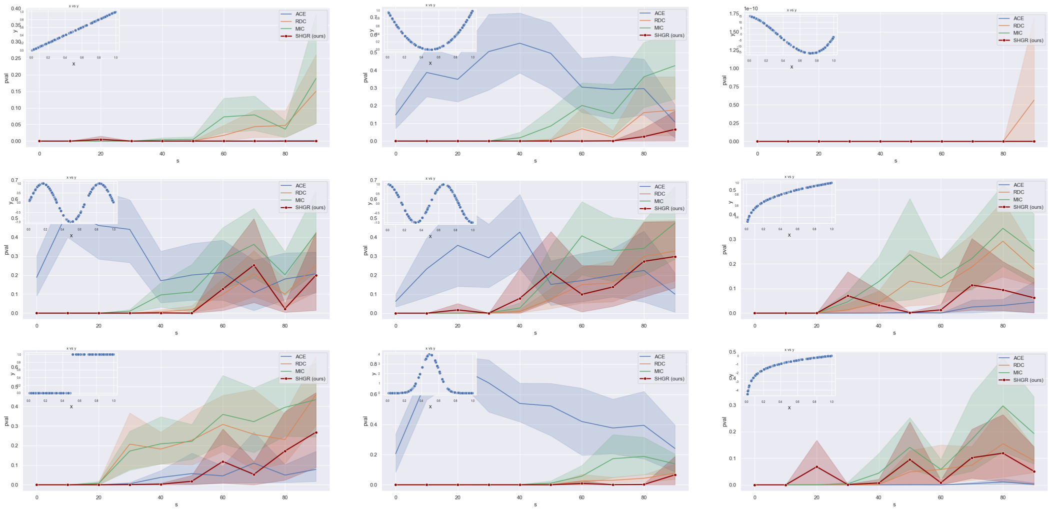


Figure 22: Performance Significance test: for bivariate correlations (lower is better)

F.10 Power of Dependence Measure details

In addition to the previous analyses, we now perform the *bivariate power of a dependence measure*. Following the analyses conducted in [31], [32] and [15], we assess the robustness of our method by progressively perturbing several nonlinear bivariate/multivariate correlations. The coefficient should ideally remain as high as possible.

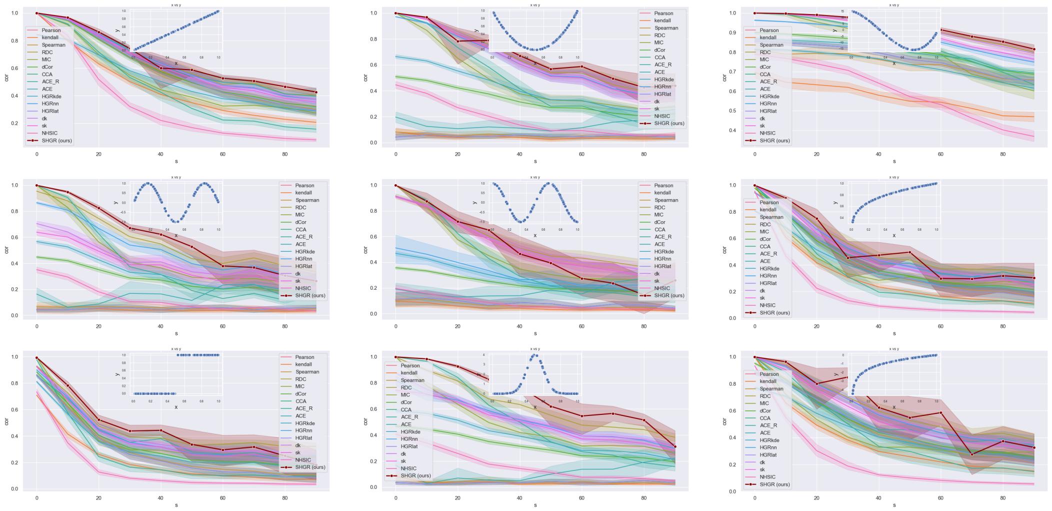


Figure 23: Power of Dependence Measure: results for bivariate correlations (higher is better)

Figure 23 shows correlation coefficients (ordinate) as a function of added noise (abscissa): increasing towards the right. We can see that our approach SHGR outperforms the other methods: higher and manages to capture correlations even in the presence of noise.

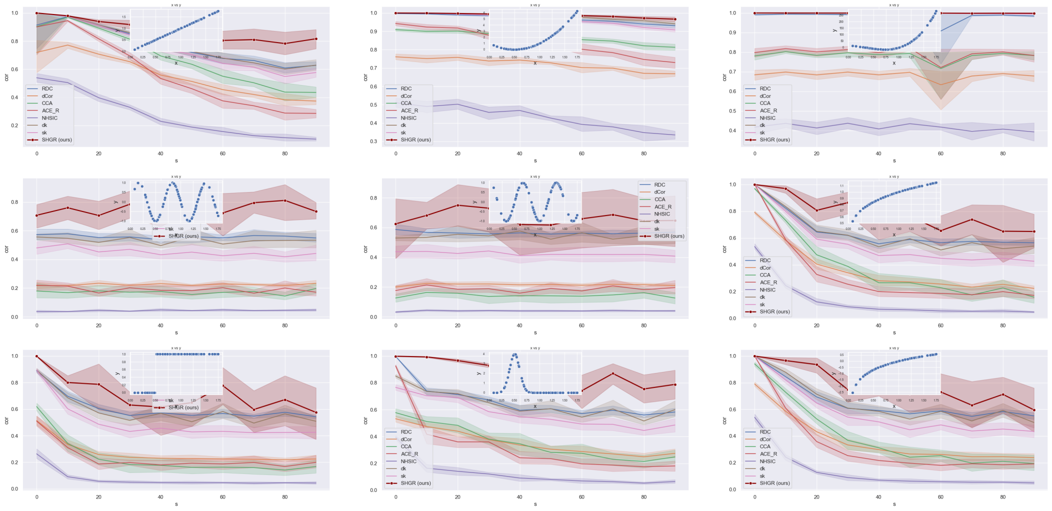
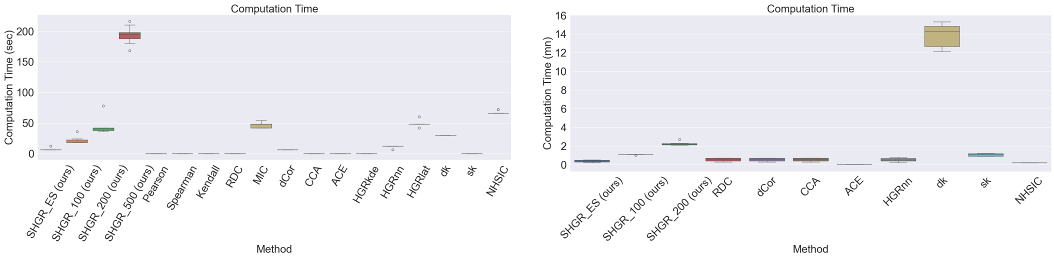


Figure 24: Power of Dependence Measure: results for multivariate correlations (higher is better)

Figure 24 shows correlation coefficients (ordinate) as a function of added noise (abscissa): increasing towards the right. In this scenario, unlike the bivariate case where only 2 variables are generated, we generate 4 variables x_1, x_2, x_3 as reduced-centered Gaussian. We then define y correlated non-linearly with $x_1 + x_2$, x_3 being a spurious variable. We can see that our approach SHGR outperforms the other methods: higher so manages to capture correlations even in the presence of noise.

F.11 Computation time

Here we compare the computation times of SHGR with those of other methods. These analyses are carried out for bivariate and multivariate correlations (the full ones being quite fast).



(a) Bivariate correlations

(b) Multivariate correlations

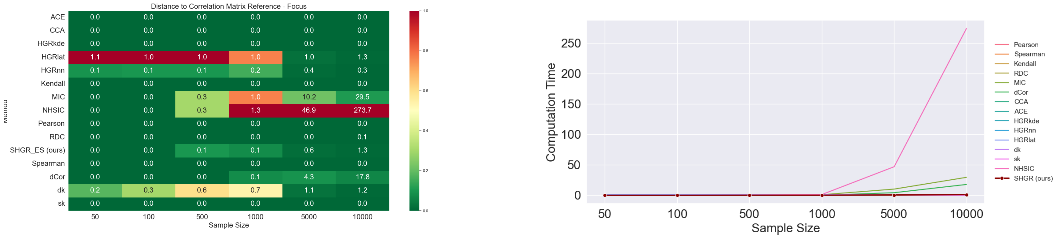
Figure 25: Computational time Comparison

F.12 Sensitivity to sample size

This section presents an analysis of the performance and computation time of the SHGR method, compared with other methods, as a function of sample size. More precisely, we present an analysis that allows us to analyze computation time and performance indicators by varying the number of observations as follows: $[50, 100, 500, 1000, 5000, 10000]$ (resp. $[1000, 10000, 100000, 1000000]$), with the number of variables always fixed at 11. These results are analyzed first on bivariate correlations and then on multivariate correlations.

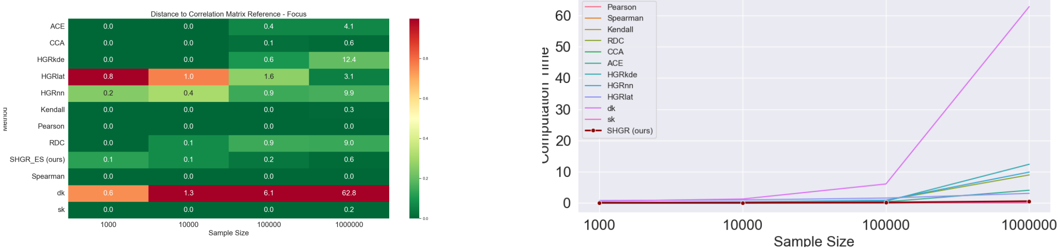
F.12.1 Bivariate correlations

Computation Time We can see in Figure 26 that the *MIC* and *NHSIC* methods are not usable, as they take respectively 10 minutes and 47 minutes for 1000 observations, and 30 minutes and 274 minutes for 10,000 observations. The *dCor* method also takes 18 minutes for $n = 10,000$.


Distance to Correlation Matrix Reference - Focus

	50	100	500	1000	5000	10000
ACE	0.0	0.0	0.0	0.0	0.0	0.0
CCA	0.0	0.0	0.0	0.0	0.0	0.0
HGRkde	0.0	0.0	0.0	0.0	0.0	0.0
HGRlat	1.1	1.0	1.0	1.0	1.0	1.3
HGRnn	0.1	0.1	0.1	0.2	0.4	0.3
Kendall	0.0	0.0	0.0	0.0	0.0	0.0
MIC	0.0	0.0	0.3	1.0	10.2	29.5
NHSIC	0.0	0.0	0.3	1.3	46.9	273.7
Pearson	0.0	0.0	0.0	0.0	0.0	0.0
RDC	0.0	0.0	0.0	0.0	0.0	0.1
SHGR_ES (ours)	0.0	0.0	0.1	0.1	0.6	1.3
Spearman	0.0	0.0	0.0	0.0	0.0	0.0
dCor	0.0	0.0	0.0	0.1	4.3	17.8
dk	0.2	0.3	0.6	0.7	1.1	1.2
sk	0.0	0.0	0.0	0.0	0.0	0.0

Figure 26: Computational time according to sample size: for bivariate correlations


Distance to Correlation Matrix Reference - Focus

	1000	10000	100000	1000000
ACE	0.0	0.0	0.4	4.1
CCA	0.0	0.0	0.1	0.6
HGRkde	0.0	0.0	0.6	12.4
HGRlat	0.8	1.0	1.6	3.1
HGRnn	0.2	0.4	0.9	9.9
Kendall	0.0	0.0	0.0	0.3
Pearson	0.0	0.0	0.0	0.0
RDC	0.0	0.1	0.9	9.0
SHGR_ES (ours)	0.1	0.1	0.2	0.6
Spearman	0.0	0.0	0.0	0.0
dk	0.6	1.3	6.1	62.8
sk	0.0	0.0	0.0	0.2

Figure 27: Computational time according to sample size: for bivariate correlations (bis)

In this second analysis (Figure 27), where time-consuming methods are removed, we estimate bivariate correlations by varying the number of observations as follows: $[1000, 10000, 100000, 1000000]$, with the number of variables always fixed at 11. The SHGR approach is quite long for 100,000 observations but faster for 1,000,000 observations: faster than *RDC* and as fast as *ACE*, for example. The *dk* method is far too long for a large number of observations. It's interesting to note that the SHGR method, being based on neural networks, can be parameterized to optimize computation time (notably on batch size and epochs parameters).

Performance Figures 28 and 29 show the performance results obtained on bivariate correlation estimates. They show that the performance of the SHGR approach outperforms that of competitors, regardless of sample size.

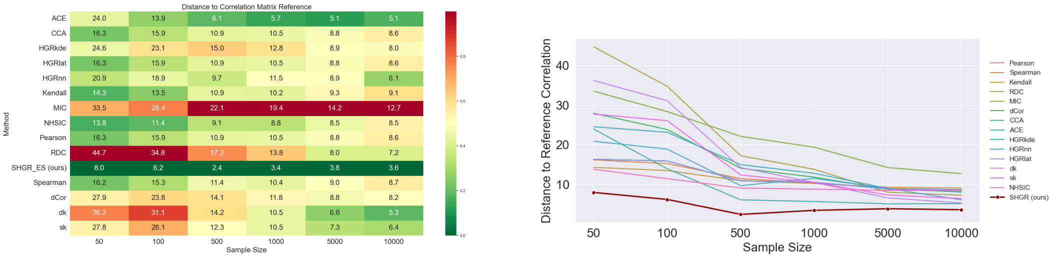


Figure 28: Performance according to sample size: for bivariate correlations: distance to reference correlations (lower is better)

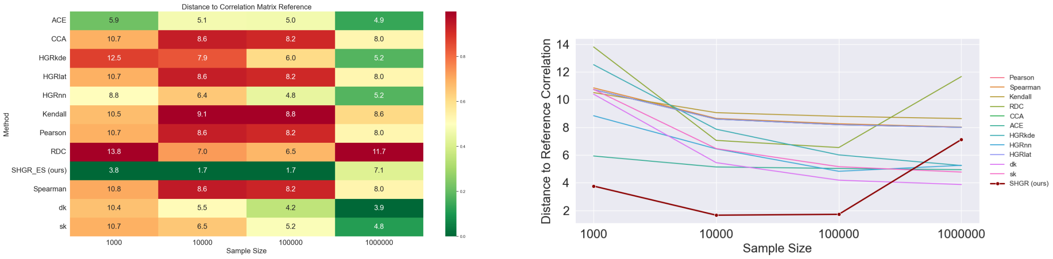


Figure 29: Performance according to sample size: for bivariate correlations (bis): distance to reference correlations (lower is better)

F.12.2 Multivariate correlations

Computation Time Figure 30 presents the computation time to estimate multivariate correlations by varying the number of observations as follows: $[50, 100, 500, 1000, 5000, 10000]$, with the number of variables always fixed at 20. Our method SHGR is quite fast, even faster than *RDC* or *dCor* or *CCA*. The *dk* method takes far too long to calculate.

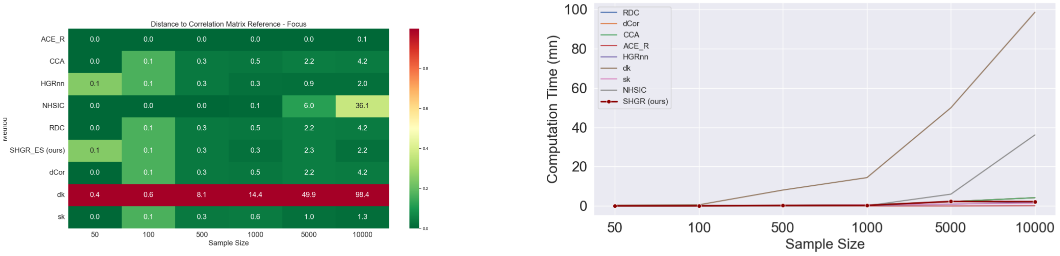


Figure 30: Computational time according to sample size: for multivariate correlations

Performance Figure 31 presents the performance to estimate multivariate correlations by varying the number of observations as follows: $[50, 100, 500, 1000, 5000, 10000]$, with the number of variables always fixed at 20. The results show that the SHGR method gives very good results, whatever the sample size.

F.13 Sensitivity to the number of variables

This section presents an analysis of the performance and computation time of the SHGR method, compared with other methods, as a function of number of variables.

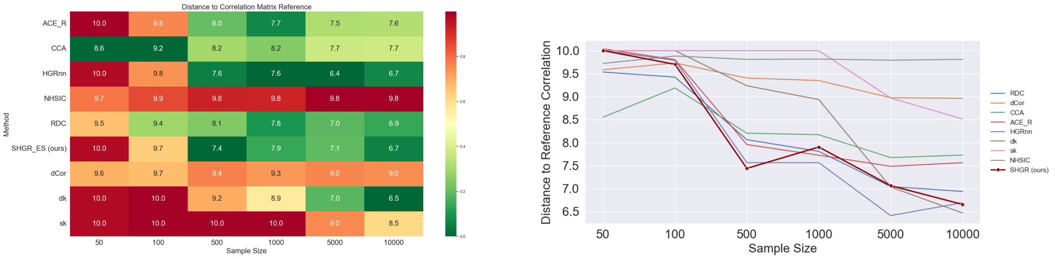


Figure 31: Performance according to sample size: for multivariate correlations: distance to reference correlations (lower is better)

F.13.1 Bivariate correlations

Here we present an analysis that allows us to analyze computation time and performance indicators by varying the number of variables as follows: $[1, 2, 5, 10] \times p = 11$ with the number of observations always fixed at 1000. These results are analyzed first on bivariate correlations and then on multivariate correlations.

Computation Time Figure 32 shows the computation times for estimating multivariate correlations by varying the number of observations. The results show that the SHGR method has very satisfactory levels. The methods *HGRlat, NHSIC* and *dk* present relatively long times.

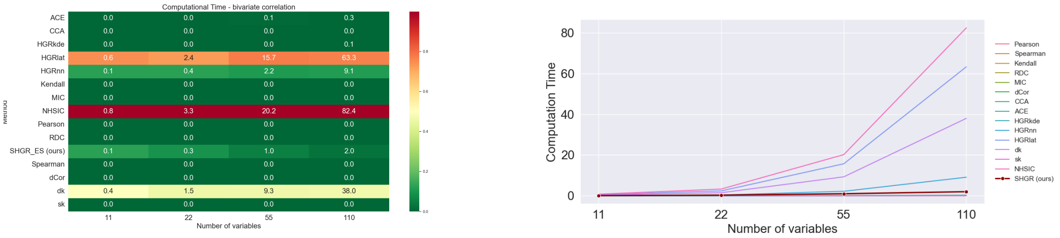


Figure 32: Computational time according to number of variables: for bivariate correlations

Performance Figure 33 shows the performance for estimating multivariate correlations by varying the number of observations. The results show that the SHGR method outperforms its competitors.

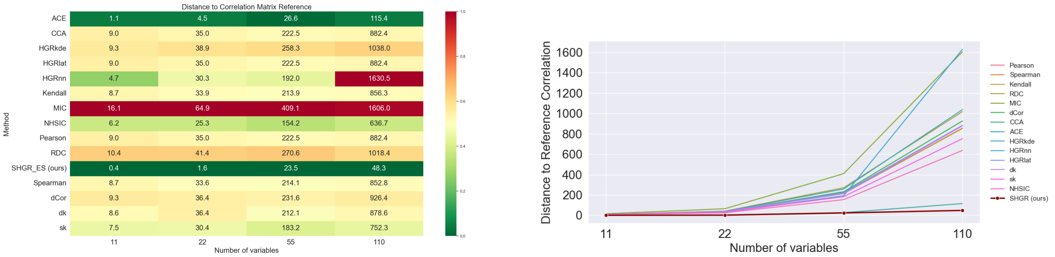


Figure 33: Performance according to number of variables: for bivariate correlations

F.13.2 Multivariate correlations

Computation Time Figure 34 presents the computation time to estimate multivariate correlations by varying the number of variables as follows: $[20, 30, 40, 70, 120]$, with the number of observations always fixed at 2000. SHGR shows fairly long computation times, higher than *ACE, CCA,RDC* and *dCor* but faster than *NHSIC* and *HGRnn*.

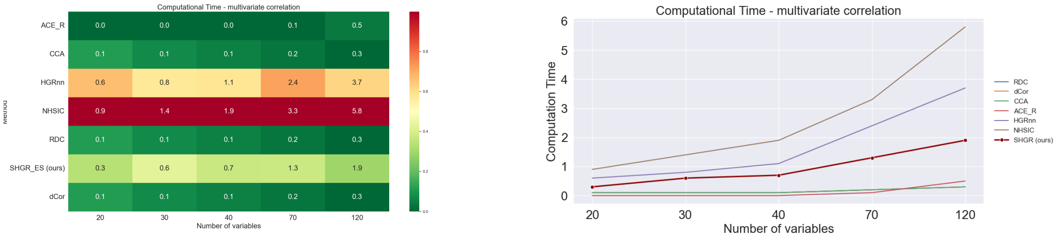


Figure 34: Computational time according to number of variables: for multivariate correlations

Performance Figure 35 presents the performance to estimate multivariate correlations by varying the number of variables as follows: $[20, 30, 40, 70, 120]$, with the number of observations always fixed at 2000. Compared with other methods, the results of SHGR deteriorate slightly as the number of variables increases. To improve results, it might be more efficient to loop over the estimation of correlations, like the other methods, accepting a slight loss in calculation time.

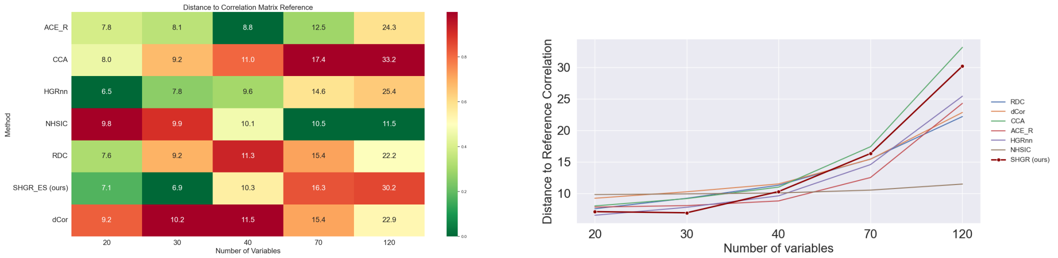


Figure 35: Performance according to number of variables: for multivariate correlations

F.14 Improving existing methods

To guarantee the relevance of our approach, we compare the results obtained with the best competitors: using input data vs. transformed data from the SHGR. Figure 36 shows the gains (correlation on SHGR encoded - correlation on inputs) obtained with the benchmark applied on the SHGR encoded rather than inputs (with the same simulations as *bivariate Power of Dependence Measure above*). All results are enhanced: correlation is greater when combined with SHGR than when applied directly to inputs.

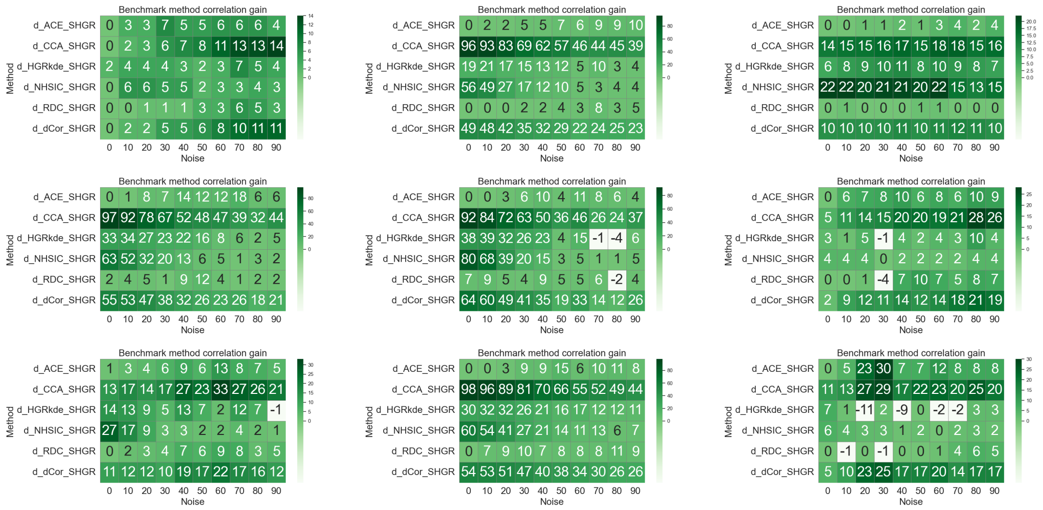


Figure 36: benchmark methods gain (%) when applied on SHGR encoded rather than inputs

F.15 Transformation Visualization

Our `SHGR` approach allows us to recover the transformations of the inputs (encoded by stacked neural nets). This makes it easy to graphically analyze the transformations applied by plotting the outputs of the stacked neural nets as a function of the inputs. Figure 37 illustrates a graphical analysis of the transformations in a bivariate context.

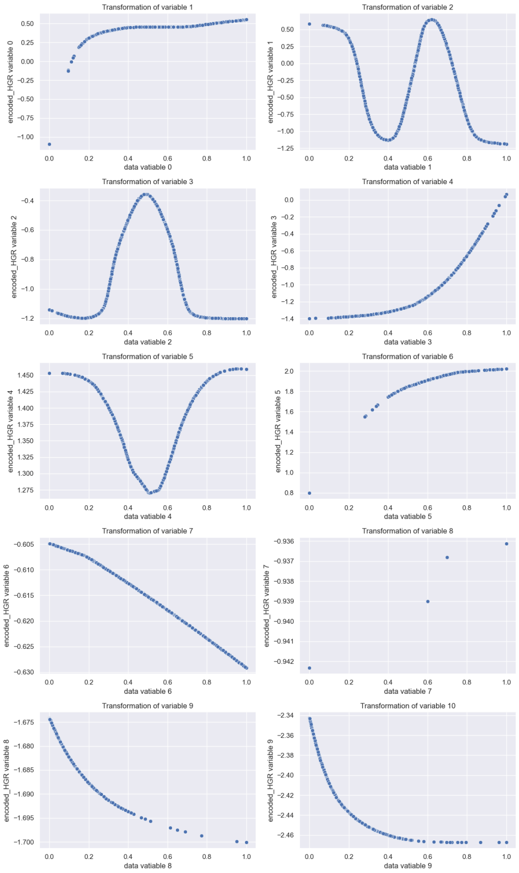


Figure 37: Illustration of the graphical analysis of the transformations: on a bivariate setting

G Real-World Applications details

To assess the relevance of our method, we apply it to 9 real-world datasets for feature selection.

The experiments are conducted on the following datasets:

- *Abalone*, composed of 9 variables and 4177 observations. Available in the imbalanced regression datasets benchmark ([5]) in the public repository: `https://paobranco.github.io/DataSets-IR/`
- *AirQuality*, composed of 15 variables and 9358 observations ([43]). This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. Available at `https://archive.ics.uci.edu/dataset/360/air+quality`
- *Appliance*, composed of 28 variables and 19735 observations ([8]). This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. Available at `https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction`
- *Bank8fm*, composed of 9 variables and 4499 observations. Available in the imbalanced regression datasets benchmark ([5]) in the public repository: `https://paobranco.github.io/DataSets-IR/`
- *Boston*, composed of 14 variables and 505 observations. Available in the imbalanced regression datasets benchmark ([5]) in the public repository: `https://paobranco.github.io/DataSets-IR/`
- *concrete*, composed of 9 variables and 1030 observations ([50]). This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. Available at `https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength`
- *CpuSm*, composed of 13 variables and 8192 observations. Available in the imbalanced regression datasets benchmark ([5]) in the public repository: `https://paobranco.github.io/DataSets-IR/`
- *N02*, composed of 8 variables and 500 observations. Available in the imbalanced regression datasets benchmark ([5]) in the public repository: `https://paobranco.github.io/DataSets-IR/`
- *Temperature*, composed of 24 variables and 4137 observations ([39]). This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. Available at `https://archive.ics.uci.edu/dataset/274/sml2010`

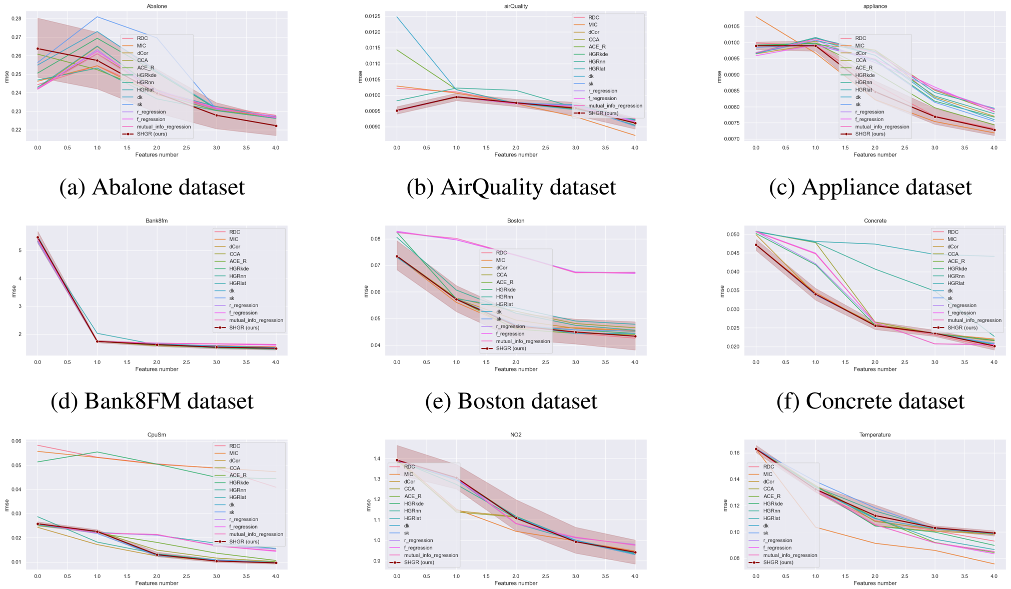
Note that we only consider numerical features.

G.1 Maximal Correlation

We perform regression tasks using the top k features selected by each method, i.e., the k features most correlated with the target variable y. We then analyze the root mean squared error of test set predictions for each value of k and each method. Predictions are made on a test set (30% of the original dataset), randomly sampled, using a random forest model. The training set consists of at most 500 observations for all datasets (to assess the robustness of the methods to sampling). Figures 38 present the obtained results.

G.2 Contribution to Maximal Correlation

We perform regression tasks using the top k features selected by each method, i.e., the k features that contribute most to the maximum correlation of features with the target variable y ($Contr(x_i) := SHGR(y, X) - SHGR(y, X \setminus X_i)$). We then analyze the root mean squared error of test set predictions for each value of k and each method. Predictions are made on a test set (30% of the original dataset), randomly sampled, using a random forest model. The training set consists of at most 500 observations for all datasets (to assess the robustness of the sampling methods).



(a) Abalone dataset

(b) AirQuality dataset

(c) Appliance dataset

(d) Bank8FM dataset

(e) Boston dataset

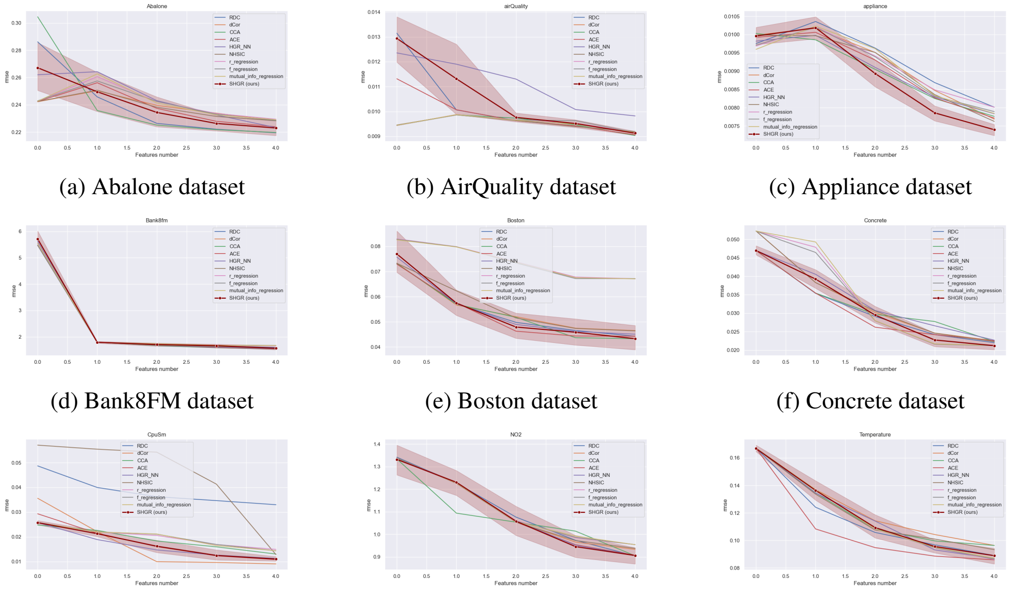
(f) Concrete dataset

(g) CpuSm dataset

(h) NO2 dataset

(i) Temperature dataset

Figure 38: Feature selection on real-world datasets with SHGR shown in red (lower is better): based on maximal correlation (bivariate correlation)



(a) Abalone dataset

(b) AirQuality dataset

(c) Appliance dataset

(d) Bank8FM dataset

(e) Boston dataset

(f) Concrete dataset

(g) CpuSm dataset

(h) NO2 dataset

(i) Temperature dataset

Figure 39: Feature selection on real-world datasets with SHGR shown in red (lower is better): based on maximal correlation contribution (multivariate correlation)