Sortability of Time Series Data

Christopher Lohse *lohsec@tcd.ie*
School of Computer Science and Statistics
University of Dublin Trinity College
IBM Research Europe, Dublin

Jonas Wahl *jonas.wahl@dfki.de*
Deutsches Forschungszentrum für künstliche Intelligenz (DFKI)

Reviewed on OpenReview: *https://openreview.net/forum?id=OGumCpcHdV*

Abstract

Evaluating the performance of causal discovery algorithms that aim to find causal relationships between time-dependent processes remains a challenging topic. In this paper, we show that certain characteristics of datasets, such as varsortability Reisach et al. (2021) and R^2-sortability Reisach et al. (2023), also occur in datasets for autocorrelated stationary time series. We illustrate this empirically using four types of data: simulated data based on SVAR models and Erdős-Rényi graphs, the data used in the 2019 causality-for-climate challenge (Runge et al., 2019), real-world river stream datasets, and real-world data generated by the Causal Chamber of Gamella et al. (2024). To do this, we adapt var- and R^2-sortability to time series data. We also investigate the extent to which the performance of continuous score-based causal discovery methods goes hand in hand with high sortability. Arguably, our most surprising finding is that the investigated real-world datasets exhibit high var-sortability and low R^2-sortability indicating that scales may carry a significant amount of causal information.

1 Introduction

Inferring causal relationships between variables in a multivariate time series setting is an ongoing topic of research in many fields, including economics/econometrics (Varian, 2016), climate science (Runge et al., 2019; Runge, 2020), medicine (Yazdani & Boerwinkle, 2015) and neuroscience (Bergmann & Hartwigsen, 2021). The process of inferring causal structure from data is often referred to as causal discovery or causal structure learning. Most of the literature on causal discovery is devoted to the following two types of methods: Constraint-based causal structure learning algorithms, such as the PC-algorithm (Spirtes & Glymour, 1991) and PCMCI (Runge, 2020) for time series data, use tests of conditional independence to iteratively learn a causal graph. Score-based method such as GES (Chickering, 2002) fit a directed acyclic graph (DAG) to the data by optimising a score function, but need to search a large discrete space of DAGs which is an NP-hard problem.

More recently, Zheng et al. (2018) have proposed to embed the discrete DAG-search space into a continuous one through a differentiable acyclicity constraint to make the problem amenable to continuous (gradient based) optimisation. The method introduced in Zheng et al. (2018), NOTEARS, showed impressive performance on data simulated with linear additive noise models (LANMs). However, as shown by Reisach et al. (2021) the strong performance of NOTEARS and similar methods on LANM data vanished after the data was normalized. Reisach et al. (2021) noticed that, before normalization, additive noise model data is highly varsortable, meaning that, on average, the causal order of the system can be recovered well by sorting the variables by the amplitude of their estimated variances. Since high varsortability and good NOTEARS performance were highly correlated, they therefore conjectured that NOTEARS implicitly made

use of variance sorting in its optimisation. The issue has recently been revisited by Ng et al. (2024), who pointed out that (a) NOTEARS does not necessarily perform well in the presence of high varsortability and that (b) normalisation of data generated by LANMs moves the data far away from the assumption of equal noise variances that underlies the NOTEARS methodology. Thus, according to Ng et al. (2024), NOTEARS should not have been expected to perform well on normalized LANM data in the first place as one of its fundamental assumptions is not satisfied.

Rather than weighing in on this discussion, in this contribution we explore how much var- and R^2-sortability (the coefficient of determination which acts as a proxy for the fraction of the variance a variable that is explained by its causal parents introduced in Reisach et al. (2023)) can be seen in data commonly used in method validation of *time series* causal discovery methods. We evaluate the degree of var-/R^2-sortability in different datasets as well as the performance of different algorithms for time series causal discovery in the presence/absence of var-/R^2-sortability and before/after normalisation. One of the main questions underlying the discussions on sortability is this: how much var-/R^2-sortability do we expect to see in real-world data? The answer to this question is likely highly context-dependent, and the question can be notoriously difficult to settle even for a single dataset, since one needs to know the causal ground truth to compute sortability. We investigate sortability scores in two of the rare cases where a causal ground-truth is available, the river flow dataset used in Tran et al. (2024) and the Causal Chamber datasets recently published by Gamella et al. (2024). In both cases, we find sortability values quite far from 0.5 (the case where there is no information in the sortability criterion), meaning that there is information in the variance or R^2-values of the data. In the river stream datasets, marginal variances tend to decrease along the causal order (varsortability close to zero) while in the Causal Chamber case, marginal variances increase along the causal order (varsortability close to one). For the R^2-score we find the exact opposite: values close to one for some rivers in the river dataset, and values close to zero for some of the causal chamber datasets.

These observations illustrate that discarding scales in causal discovery as arbitrary may be premature, as scales may encode significant causal information. We hypothesise plausible physical explanations for the observed varsortability in the investigated datasets which also call into question the validity of an equal noise variance assumption in these cases. For the river data we know that the the width of the rivers decrease from the source to the mouth which has potentially an influence on the variance of extreme flows.

In more detail, our main contributions are:

1. We extend var- and R^2-sortability and the simple benchmarking algorithm of Reisach et al. (2021; 2023) to the time series setting.

2. We show empirically that simulated data typically used to evaluate causal discovery algorithms for time series data is varsortable, meaning the amplitude of the marginal variance increases the lower the variable is in the causal ordering of the ground truth summary graph. We also show that, in this case, varsortability is largely driven by contemporanous dependencies.

3. We demonstrate that there is a positive correlation between the performance of continuous score-based causal discovery algorithms for time series and the varsortability of data generated by structural autoregressive processes.

4. We investigate sortability of the data used in the 2019 causal discovery challenge[1] by Runge et al. (2019) and show that some data sets are highly varsortable and simple benchmark perform well on these.

5. We calculate var- and R^2-sortability of the river flow dataset used in Tran et al. (2024), and of the recently published datasets generated by the Causal Chamber(Gamella et al., 2024). We find that in these real-live datasets, the scale plays an important role for potential causal discovery algorithms.

[1] `https://causeme.uv.es/neurips2019/`

2 Preliminaries

2.1 Causal discovery for time series

A *stationary time series graph* (ts-graph) is a directed graph $\mathcal{G} = (V \times \mathbb{Z}, \mathcal{D}),\ V = \{1.\ldots, d\}$, whose edges $(i, t-k) \to (j, t)$ are assumed invariant under translation of the time component. In addition, it is typically required that there is a finite maximal lag $\tau_{max} = \max_{i,j \in V}\{k|((i, t-k), (j, t)) \in \mathcal{D}\} < \infty$ and that the contemporaneous component of $\mathcal{G}$ is acyclic, where $k \in \mathbb{N}$. Any stationary ts-graph induces a directed, potentially cyclic *summary graph* $\mathcal{G}_{\text{sum}}$ over V that contains a directed edge (i, j) if $(i, t-k) \to (j, t) \in \mathcal{D}$. The adjacency matrices of $\mathcal{G}, \mathcal{G}_{\text{sum}}$ will be denoted by $\mathbf{W}, \mathbf{W}_{\text{sum}}$ respectively.

The aim of most causal discovery methods for time series is to recover either $\mathcal{G}$ or $\mathcal{G}_{\text{sum}}$ from observational or interventional data. Often the data is assumed to be generated by a discrete multivariate process $(\mathbf{X}_t)_{t\in\mathbb{Z}},\ \mathbf{X}_t = (X_t^1, ..., X_t^{(d)})$ compatible with $\mathcal{G}$. The process $(\mathbf{X}_t)_{t\in\mathbb{Z}}$ is typically modelled as a structural vector autoregressive (SVAR) process(Hyvärinen et al., 2010), in which case, compatibility with $\mathcal{G}$ means that $(\mathbf{X}_t)_{t\in\mathbb{Z}}$ follows the evolution rule

$$X_t^j = \sum_{i \in \text{pa}_{\mathcal{G}_{\text{sum}}}(j)} \sum_{k=0}^{\tau_{max}} a_{i,t-k}^j X_{t-k}^i + \eta_t^j \tag{1}$$

for $j \in \{1, ..., d\}$, where η^j are Gaussian white noise processes, $\text{pa}_{\mathcal{G}_{\text{sum}}}(j)$ denotes the parents of node j in the summary graph, and $a_{i,t-k}^j$ is only allowed to be non-zero if $(i, t-k) \to (j, t)$ is an edge in $\mathcal{G}$. SVAR-processes can be considered the time series analogue of additive linear noise models for which sortability was discussed by Reisach et al. (2021; 2023). When generating data from such a model, coefficients are typically randomly drawn, sometimes with a pre-specified proportion of contemporaneous links, and the process is being run until it has converged to a stationary distribution (or is discarded if the distribution is non-stationary).

2.2 NOTEARS and Derivatives

Zheng et al. (2018) propose the continuous score-based causal discovery method NOTEARS, which embeds the discrete search space of DAGs into a continuous one by using the differentiable function $h(\mathbf{W}) = tr\ e^{\mathbf{W}\circ\mathbf{W}} - d$. This function is 0 if and only if $\mathbf{W}$ is the adjacency matrix of an acyclic graph and hence measures the *DAGness* of $\mathbf{W}$ (Zheng et al., 2018). By combining this function with a score evaluating how well the estimated weight matrix $\mathbf{W}$ fits the data, Zheng et al. (2018) formulate the constrained optimisation problem to find

$$\min_{\mathbf{W}} \frac{1}{n}||\mathbf{X} - \mathbf{X}\mathbf{W}||_2^2 \tag{2}$$

s.t. $\mathbf{W}$ is acyclic, which is modelled by h. $||\cdot||_2^2$ is the Frobenius norm.

The DYNOTEARS algorithm Pamfil et al. (2020) modifies the NOTEARS algorithm to work with time-lagged and auto-correlated dependencies by redefining the optimisation problem to $\min_{\mathbf{W_l}} \ell(\mathbf{W_c}, \mathbf{W_l})$ s.t. $\mathbf{W}_c$ is acyclic, where $\mathbf{W_l} \in \mathbb{R}^{d\times d\times \tau_{max}}$ is the lagged adjacency matrix and $\mathbf{W}_c$ is the contemporaneous adjacency of the underlying time series process. As the underlying time series graph to estimate is acyclic if and only if $\mathbf{W}_c$ is acyclic, it suffices to enforce the acyclicity constraint only on $\mathbf{W}_c$ (Pamfil et al., 2020). To ensure sparsity of $\mathbf{W}$, Pamfil et al. (2020) also add an ℓ_1 penalty term, leading to the constraint optimisation problem

$$\min_{\mathbf{W}} \frac{1}{2n}||\mathbf{X} - \mathbf{X}\mathbf{W}_c - \mathbf{X_l}\mathbf{W}_l||_2^2 + \lambda_1||\mathbf{W}_c||_1 + \lambda_2||\mathbf{W}_l||_1 \tag{3}$$

s.t. $\mathbf{W}_c$ is acyclic. Here λ_1 and λ_2 are two regularisation parameters and $\mathbf{W}_l$ is the lagged adjacency matrix.

The continuous optimisation problems of NOTEARS and DYNOTEARS can be solved efficiently by rewriting the problem using the augmented Lagrangian method and using a numerical solver such as L-BFGS (Zheng et al., 2018; Pamfil et al., 2020). After applying the numerical optimisation algorithm in both of the algorithms, a threshold t is applied to remove weights close to zero (Zheng et al., 2018; Pamfil et al., 2020).

2.3 Sortability Criteria

In the following, we briefly reiterate the two sorting criteria varsortability and R^2-sortability introduced by Reisach et al. (2021) and Reisach et al. (2023) respectively. In essence, both of these approaches calculate a score s of sortability in a comparable manner; s is the measurement of the degree of agreement between the true causal order and the increasing order of a sortability criterion cri.

For any causal model containing the variables $\{X^{(1)}, \ldots, X^{(d)}\}$ with a (non-degenerate) adjacency matrix $\mathbf{W}$, the sortability score is the fraction of directed paths that start from a node with a strictly lower sortability criterion than the node they end in. Thus the sortability for one selected criterion cri can be calculated as

$$s := \frac{\sum_{k=1}^{d-1} \sum_{i \to j \in \mathbf{W}^k} increasing(cri(X^i), cri(X^j))}{\sum_{k=1}^{d-1} \sum_{i \to j \in \mathbf{W}^k} 1} \in [0, 1], \tag{4}$$

where

$$increasing(a, b) = \begin{cases} 1 & a < b \\ 1/2 & a = b \\ 0 & a > b. \end{cases} \tag{5}$$

The matrix $\mathbf{W}$ in Equation 4 is set to the power k, since the (i, j) entry of the k-th power of an adjacency matrix of a DAG exactly counts the number of directed paths from i to j. In the case of varsortability Reisach et al. (2021), the criterion $cri(X^i) = \mathrm{Var}(X^i)$ is the marginal variance, whereas in the case of R^2-sortability, it is the coefficient of determination $cri(X^i) = R^2(X^i)$ which acts as a proxy for the fraction of the variance of X_i that is explained by its causal parents, (see Reisach et al. (2023) for details). Varsortalility v is defined by using the marginal variance as cri. R^2-sortability r is defined by using the obtained R^2-coefficients as the sorting criterion cri (Reisach et al., 2021; 2023).

Reisach et al. (2021) showed that varsortability is usually high in data generated by LANMs (see the analytical and empirical proof in their paper). Based on their sortability criteria, Reisach et al. (2021) and Reisach et al. (2023) introduced the baseline methods varsortnregress and R^2-sortnregress respectively. These algorithms sort the system variables based on their variances or R^2-scores which are estimated by fitting a regression model; these simple benchmark methods are then shown to have similar performance to some state of the art causal discovery algorithms (Reisach et al., 2021; 2023) on LANM data.

3 Modified Sortability Criteria for Summary Graphs

In a time series causal discovery setting, the summary graph $\mathcal{G}_{\text{sum}}$ describing the relationship between the different time-evolving processes may contain cycles. Since the marginal variances $\mathrm{Var}(X_t^i)$ do not depend on the time index t due to the assumed stationarity of the processes, to compute varsortability in this situation, in principle, one could still use Equation 4 as is.

However, any pair of processes X^i and X^j that belong to the same strongly connected component of the summary graph (i.e. that are connected by a cycle) would always contribute a 1 to the numerator and a 2 to the denominator of Equation 4.

Therefore the presence of cycles would dilute the sortability signal and would naturally push it closer to 1/2. In other words, R^2-, and varsorting of cyclicly connected processes is meaningless, and we are only interested in whether nodes that are not cyclicly connected can be sorted. Our sortability criterion thus becomes

$$s := \frac{\sum_{(i,j) \in \mathcal{AP}(\mathcal{G}_{\text{sum}})} increasing(cri(X_i), cri(X_j))}{\sum_{(i,j) \in \mathcal{AP}(\mathcal{G}_{\text{sum}})} 1} \in [0, 1], \tag{6}$$

where $\mathcal{AP}(\mathcal{G}') = \{(i, j) \in V' \times V' \mid i \implies j,\ j \not\Longrightarrow i\}$ is the set of admissible node pairs (the long double arrow indicates the existence of a directed path).

For the example in Figure 1, the sortability score s is $\frac{1+1+1+0}{1+1+1+1} = 0.75$. The contribution of the pairs (B, C) and (C, B) is ignored since the two nodes belong to the black cycle. Cycle-free directed paths that connect admissible node pairs are depicted in colour. If we would calculate it including cyclicly connected pairs, it would result in $\frac{4}{6} \approx 0.67$.

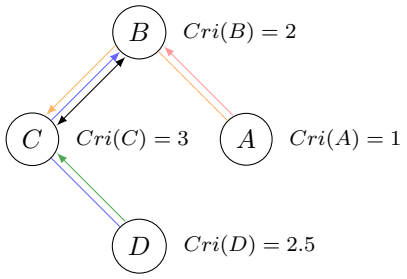


Figure 1: Example of the calculation of s with cycles.

High varsortability and the performance of DYNOTEARS Based on heuristics and empirical findings, Reisach et al. (2021) argue that high variability causes gradient-based optimisation algorithms such as NOTEARS(Zheng et al., 2018) to favour graphs whose edges point in the direction of increasing marginal variances during their first optimisation step. Since DYNOTEARS is a time-series adaptation of NOTEARS, that essentially coincides with NOTEARS on the contemporaneous components of a ts-graph, Reisach et al. (2021)'s arguments might be applicable to (the contemporaneous part of) DYNOTEARS as well.

However, in the recent work Ng et al. (2024) provide examples in which continuous optimization-based methods can not perform well in the presence of high varsortability. Ng et al. (2024) also give an alternative explanation for why continuous structure learning performs significantly worse after standardisation of LANM data. Continuous structure learning assumes equal noise variances for all variables in the system, which is violated in the standardised model leading to poor performance.

4 Sortnregress for Time Series Graphs

In order to have simple algorithms that exploit high R^2- and varsortability, we present our time series adapted sortnregress algorithm based on sortnregress from Reisach et al. (2021; 2023). To estimate contemporaneous dependencies, we use the standard sortnregress algorithm, which consists of two steps:

1. Sort nodes by increasing marginal variance or R^2-score.
2. Each node is regressed on its predecessor, determined by order, using a penalised regression technique. As described in Reisach et al. (2021), LASSO regression is used, using the Bayesian Information Criterion (BIC) for hyperparameter selection.

This gives us an estimated contemporaneous adjacency matrix $\hat{\mathbf{W}}_c$. A random sortnregress algorithm is also used, where we determine the order of the variables randomly using i.i.d. Bernoulli trials. To estimate lagged dependencies between variables, we use the same first step and change the second step: We now regress each node on each of its predecessors $p_{i,t}$, where $t \in [1, \tau_{max}]$ indicates the time lag. After this step we have an estimated lagged adjacency matrix $\hat{\mathbf{W}}_l$.

5 Numerical Experiments & Results

In the following section, we first give an overview of the evaluation metrics which are used for all the considered datasets. After that we outline the setup and the results for each dataset. We conduct our experiments in Python, using the TIGRAMITE library[2] for simulating data with SVAR models. When assessing the performance of different algorithms across a range of sortability values, the hyperparameters of the DYNOTEARS algorithm are set to $\lambda_1 = \lambda_2 = 0.05$. The weight threshold is set to 0.1. As a constraint-based comparison algorithm, PCMCI$^+$(Runge, 2020) is run with $\alpha = 0.01$ and the ParCorr conditional independence test. We further use the varsortnregress, R^2-sortnregress and random regress algorithms as described in Section 4.

We assess the $F1$-score using the formula:

$$F1 = \frac{TP}{TP + 0.5(FP + FN)}$$

[2] `https://github.com/jakobrunge/tigramite`

to gauge the performance of the selected algorithms concerning the comparison between the estimated binary time series adjacency matrices $\hat{\mathbf{W}}$ and the ground truth $\mathbf{W}$.

Here, TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives for edge detection.

We refer to this metric as the overall $F1$-score. Additionally, we calculate the $F1$-scores comparing the estimated contemporaneous adjacency matrix $\hat{\mathbf{W}}_c$ with the true contemporaneous adjacency matrix $\mathbf{W}_c$, and the $F1$-scores comparing the estimated lagged adjacency matrix $\hat{\mathbf{W}}_l$ with $\mathbf{W}_l$. These metrics are denoted as $F1$-contemp and $F1$-lagged, respectively. In cases where only information about the summary graph is available, we calculate the $F1$-score between $\mathbf{W}_{sum}$ and the estimated summary adjacency matrix $\hat{\mathbf{W}}_{sum}$.

5.1 NEURIPS Competition Data

Setup We also assess var- and R^2-sortability on the 2019 Causality-for-Climate-competition(Runge et al., 2019) data. This dataset is relevant not only because it was used in the competition, but also because it follows the same structure as the CauseMe platform (Munoz-Marí et al., 2020), which is widely used to evaluate causal discovery algorithms (Bussmann et al., 2021; Runge, 2020). The dataset includes simulated and partially simulated datasets of varying complexity, including high-dimensional datasets and non-linear dependencies, with 100, 150, 600 or 1000 realisations for each dataset specification. We excluded the datasets with missing values.

We set the maximal time-lag in our methods τ_{max} following the description of the respective dataset (ranging from 3 to 5).

Results As illustrated in Table 1, we observe a varsortability above 0.5 for all data except the logistic model, which has a varsortability of 0, meaning that each causal child has a lower marginal variance than its parent. In particular, the realistic climate and weather models have a high varsortability, with a mean of over 0.86 for all of them. For R^2-sortability, we observe a value around 0.5 for most of the realistic models, with the exception of the FinalCLIM models with values of 0.25 and 0.16 for the 5 and 40 variable datasets respectively. The linear and logistic models have scores between 0.5 and 0.6. For the data sets with high varsortability (var > 0.8 varsortnregress outperforms or is en par with PCMCI$^+$.

Table 1: Mean Sortability criteria and F1-scores for different benchmark algorithms over different realisations on the NeurIps competition data. We set $\tau_{max} = 3$.

Dataset	var	R^2	PCMCI$^+$	varsortnregress	R^2-sortnregress	random
Testlinear-VAR_N-10_T-150	0.59 ± 0.17	0.61 ± 0.19	-	-	-	-
Testlinear-VAR_N-100_T-150	0.65 ± 0.11	0.64 ± 0.12	-	-	-	-
Testnonlinear-VAR_N-20_T-600	0.56 ± 0.13	0.57 ± 0.14	0.25 ± 0.07	0.10 ± 0.07	0.10 ± 0.06	0.15 ± 0.06
Finallinear-VAR_N-10_T-150	0.62 ± 0.18	0.62 ± 0.18	0.15 ± 0.10	0.17 ± 0.09	0.16 ± 0.09	0.20 ± 0.09
Finallinear-VAR_N-100_T-150	0.66 ± 0.11	0.63 ± 0.11	-	-	-	-
FinalCLIM_N-5_T-100	0.91 ± 0.13	0.29 ± 0.25	0.40 ± 0.13	0.51 ± 0.23	0.23 ± 0.18	0.31 ± 0.19
FinalCLIM_N-40_T-100	0.9 ± 0.09	0.19 ± 0.11	-	-	-	-
FinalCLIMnoise_N-5_T-100	0.91 ± 0.13	0.3 ± 0.25	0.32 ± 0.16	0.43 ± 0.22	0.21 ± 0.18	0.28 ± 0.21
FinalCLIMnoise_N-40_T-100	0.9 ± 0.09	0.23 ± 0.13	-	-	-	-
Finallogistic-largenoise_N-5_T-150_medium	0.21 ± 0.32	0.58 ± 0.35	0.42 ± 0.23	0.02 ± 0.10	0.04 ± 0.13	0.04 ± 0.13
FinalWEATHnoise_N-5_T-1000	0.77 ± 0.21	0.5 ± 0.3	0.32 ± 0.15	0.25 ± 0.20	0.20 ± 0.20	0.26 ± 0.18
FinalWEATHnoise_N-10_T-1000	0.27 ± 0.11	0.23 ± 0.15	0.17 ± 0.12	0.21 ± 0.13		
FinalWEATH_N-10_T-1000	0.84 ± 0.16	0.52 ± 0.22	0.34 ± 0.10	0.32 ± 0.14	0.22 ± 0.12	0.27 ± 0.12
FinalWEATH_N-5_T-1000	0.81 ± 0.2	0.47 ± 0.31	0.36 ± 0.14	0.34 ± 0.18	0.22 ± 0.17	0.31 ± 0.17

5.2 Data Generation with Erdős–Rényi Graphs and SVAR Models

Setup In order to investigate var- and R^2-sortability for datasets used to evaluate continuous score-based causal discovery methods, we replicate one of the two data generation methods used by Pamfil et al. (2020).

Following Pamfil et al. (2020); Zheng et al. (2018), when generating random time series graphs, we use Erdős–Rényi Graphs (ER Graphs) (Newman, 2018) to draw the contemporaneous edges with i.i.d Bernoulli trials. Sampling only lower triangle entries of the contemporaneous adjacency matrix and then permuting the node order ensures that the contemporaneous adjacency matrix $\mathbf{W}_c$ is acyclic. Pamfil et al. (2020) sample the graph to ensure a pre-specified

mean degree d_c for the contemporaneous dimension and d_l for the lagged dependencies for a total number of variables d.

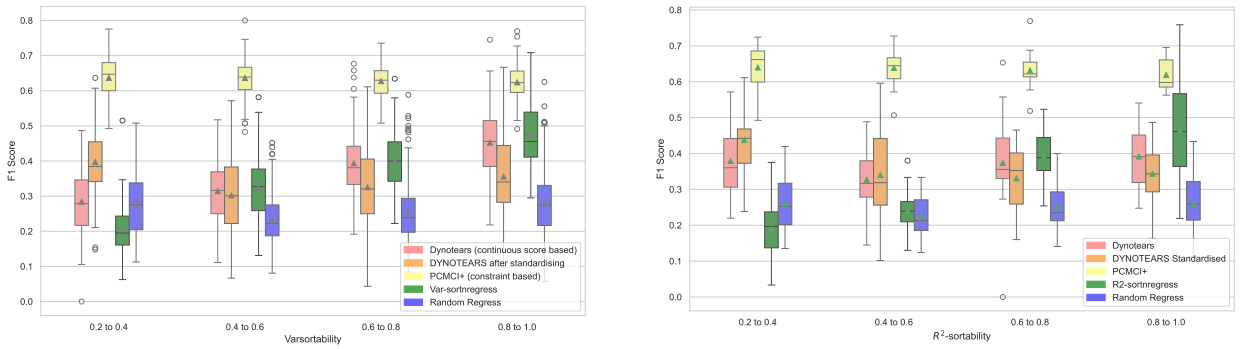
In order to ensure that the expected mean degree is d_c the probability of each Bernoulli trial is set to $d_c/(d-1)$, where d. The edge coefficients are sampled uniformly at random from $[-2, -0.5] \cup [0.5, 2.0]$ (Pamfil et al., 2020). Again following Pamfil et al. (2020), the edge weights for lagged variables are sampled depending on t from $[-0.5\alpha, -0.3\alpha] \cup [0.3\alpha, 0.5\alpha]$, where $\alpha = 1/\delta^{t-1}$. The weight decay $\delta > 1$ reduces the influence (weights) of variables further back in time. We randomly sample Erdős-Rényi graphs with degree $d_c = 4$ for the contemporaneous dimension, and for each lag we set $d_l = 1$; δ is set to 1.1.

We examine sortability values for different numbers of nodes $d \in \{10, 20, 50, 100\}$. For each number of nodes we randomly generate 500 different graphs and generate $n = 500$ samples per graph. The samples are taken after a burn-in period to ensure stationarity. We then calculate the overall varsortability, the varsortability of only the contemporanous dependencies and the varsortability of all all the lagged dependencies.

We also want to investigate whether varsortability is driven by contemporaneous or lagged dependencies. This is why we compute var- and R^2-sortability over a grid of $d_c, d_l \in [0, 0.5, 1, 2, 3, 4, 6, 8]$. We do this for $d = 10$ and $d = 20$ nodes.

We further investigate the influence of the two sortability criteria on the performance of score and constraint-based algorithms. In order to do so, we generate data for $d = 10$ variables, which has varying sortability values. We then randomly draw $m = 30$ samples per sortability interval, which we set to $[0, 0.2]$, $[0.2, 0.4]$, $[0.4, 0.6]$, $[0.6, 0.8]$, $[0.8, 1]$. We report the performance of the following algorithms for varying var- and R^2-sortability: DYNOTEARS, DYNOTEARS standardised (run after standardising the data), PCMCI$^+$, varsortnregress/R^2-sortnregress and randomregress. This means that for DYNOTEARS, the data has a varsortability as defined by the respective bin before standardising (after standardisation the varsortability is always 0.5).

Results We observe that the overall mean varsortability for the ER-SVAR data used by Pamfil et al. (2020) ranges from 0.58 for ten nodes to 0.54 for 100 nodes. We do not see a trend in varsortability for different numbers of nodes. The varsortability of the contemporaneous dimension is around 0.7 for all numbers of nodes. The lagged varsortability is around 0.54 for all numbers of nodes except 10 nodes where it is 0.56. The R^2 sortability is around 0.5 for all numbers of nodes. Varsortability of the contemporaneous component of $\mathcal{G}$ is always above 0.7 and higher than the overall varsortability. Consequently, lagged varsortability is always lower than contemporanous and overall varsortability. The detailed results can be found in Appendix A.



(a) Performance of different algorithms for varying var-sortability.

(b) Performance of different algorithms for varying R^2-sortability.

Figure 2: Comparison of algorithm performance under different sortability conditions: (a) Varying varsortability, (b) Varying R^2-sortability.

Figure 2a shows that the higher the varsortability, the higher the F1-score of DYNOTEARS. The varsortnregress benchmark model also improves with higher varsortability and seems to outperform the DYNOTEARS algorithm for varsortability values higher than 0.6. The constraint-based PCMI$^+$ algorithm does not seem to be as affected by varsortability as the randomregress algorithm. The DYNOTEARS algorithms perform better on standardised data for low varsortability values before standardisation.

For high R^2-sortability values, we observe a different behaviour: Both DYNOTEARS and PCMCI$^+$ seem unaffected by varying values. Again the R^2-sortnregress algorithm seems to outperform DYNOTEARS for R^2-sortability values of 0.6 or higher (see Figure 2b).

We also investigate whether high varsortability leads to a higher F1-score of DYNOTEARS due to better identification of contemporaneous edges or lagged dependencies. As we can see in Figure 3, the effect of increasing F1-scores with increasing varsortability is even higher for contemporaneous dependencies. The F1-score for lagged dependencies seems to be unaffected. Moreover, we observe in our experiments that the contemporaneous F1-score has a Pearson correlation of more than 0.6 with the varsortability score. Furthermore, higher weighting thresholds of the DYNOTEARS algorithm seem to increase the influence of varsortability on the performance of DYNOTEARS as measured by the F1-score.

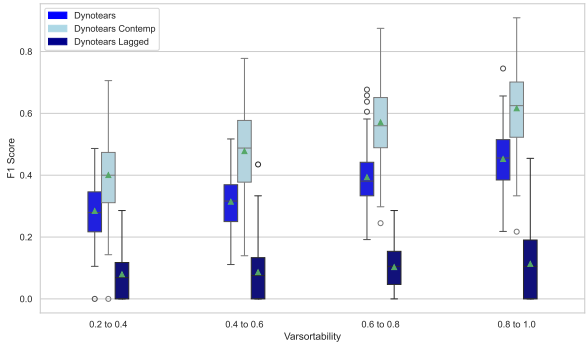


Figure 3: F1-score of Contemporaneous and lagged Dependencies DYNOTEARS for different varsortability values.

We also want to determine if different degrees of contemporaneous and lagged dependencies influence the overall varsortability of the generated data. In general we observed that a higher value for d_c increases the varsortability and higher values of d_l seem to decrease it. If there are more nodes the varsortability seems to be higher. If there are no contemporaneous edges we observe varsortability values of around 0.5. For R^2-sortability, the mean score is not affected by the degree or number of nodes. The detailed results can be found in Appendix B.

5.3 Extremal River Flow Problem

Setup Next, we investigate the two sortability criteria on four real-world datasets modelling extremal river flow previously used in (Tran et al., 2024). The aim of the problem is to recover the direction and connections of a river network with only extreme flow measurements at certain stations, without knowing the location of these stations. This is a time-dependent process, as an extreme measurement recorded at one station at time $t-1$ may lead to an extreme measurement at another station at time t (Tran et al., 2024). Having the ground truth river network allows us to report R^2 and varsortability on this real-world dataset and investigate whether high values can also occur on real time series data. We also include the $F1$-scores of selected benchmark algorithms for a τ_{max} value of 3 as this gave the best results. Following Tran et al. (2024), we treat the downstream river direction as the causal ground truth $\mathcal{G}_{sum}$, which perhaps may be controversial due to the fact that downstream extreme events might have damming effects further upstream. Nevertheless, even if one is not comfortable calling the flow direction causal, the results below show that varsorting is able to cover the 'flow graph'.

As we only know the ground truth/flow graph on the summary level, i.e. we know $\mathcal{G}_{sum}$, we calculate the $F1$-score between the ground truth for $\mathbf{W}_{sum}$ and the estimated adjacency matrix $\hat{\mathbf{W}}_{sum}$ for the following algorithms: varsortnregress, R^2-sortnregress, randomregress, PCMCI$^+$ and reversed varsortnregress. We use reversed varsortnregress as we observed very low varsortability values and wanted to investigate if an algorithm that orders by decreasing variance could exploit this fact.

Results As illustrated in Table 2, varsortability for the entire river network is below 0.5, with a varsortability value for the Danube close to 0. This indicates that as the river network follows a tree structure, the variance of the extremal river velocities for nodes decreases with increasing distance from the river source. In other words, the real-life dynamics of the river flow systems entail sortable marginal variances. It can be observed that for high R^2-sortability, the R^2-sortnregress algorithm performs as well as PCMCI$^+$. Conversely, for low varsortability values, the reversed

varsortability algorithm, while being superior to random, is unable to match the performance of PCMCI$^+$. Notably, the varsortnregress algorithm outperforms the reverse varsortnregress algorithm on the upper Colorado dataset even though the varsortability is below 0.5. We also tried to run DYNOTEARS on this dataset. However, the algorithm

Table 2: Varsortability(var) and R^2-sortability (R^2)and $F1$-scores of different algorithm reported on the extremal river flow datasets. We set $\tau_{max} = 3$ as this leads to the best results.

Dataset	var	R^2	PCMCI$^+$	varsortnregr.	varsortnregr. rev.	R^2-sortnregr.	randomregr.
danube	0.07	0.82	0.56	0.08	0.20	0.18	0.11
lower-colorado	0.34	0.61	0.36	0.08	0.08	0.08	0.10
middle-colorado	0.29	0.94	0.26	0.09	0.23	0.26	0.22
upper-colorado	0.43	0.95	0.41	0.26	0.11	0.30	0.22

did not converge after a couple of hours run time as it did not manage to satisfy the acyclity constraint, at least for our choices of hyperparameters.

5.4 Causal Chamber Data

Setup We now investigate the values of the two sortability criteria for data generated by the recently introduced Causal Chamber (Gamella et al., 2024), which provides a toolbox consisting of real physical systems that can be used to evaluate causal discovery or other AI algorithms on real data.

We use each dataset contained in their PYTHON library and calculate the var- and R^2-sortability for each dataset and each experiment in the dataset[3]. We then calculate the mean and standard deviation of the different datasets, where one value is one experiment performed on the dataset. We again run benchmarks on varsortnregress, R^2-sortnregress, randomregress, PCMCI$^+$ and evaluate the $F1$-score between $\mathbf{W}_{sum}$ and $\hat{\mathbf{W}}_{sum}$.

Table 3: Mean and standard deviation of var- and R^2-sortability as well as F1-score of benchmark algorithms obtained on each dataset with different experiments. The standard deviation is given after $\pm$. We set $\tau_{max} = 2$.

Dataset	var	R^2	PCMCI$^+$	varsortnregress	R^2-sortnregress	random
lt_camera_test_v1	0.94 ± 0.01	0.25 ± 0.24	0.05	0.30	0.17	0.14
lt_camera_validate_v1	0.99 ± 0.02	0.01 ± 0.03	0.08	0.37	0.05	0.25
lt_camera_walks_v1	0.95 ± 0.0	0.23 ± 0.07	0.31	0.34	0.13	0.22
lt_color_regression_v1	0.94 ± 0.02	0.15 ± 0.06	0.19	0.25	0.18	0.19
lt_interventions_standard_v1	0.94 ± 0.03	0.46 ± 0.04	0.26	0.18	0.12	0.08
lt_malus_v1	0.98 ± 0.02	0.02 ± 0.01	0.20	0.24	0.25	0.19
lt_test_v1	0.96 ± 0.03	0.01 ± 0.02	-	-	-	-
lt_validate_v1	0.99 ± 0.02	0.02 ± 0.02	0.34	0.30	0.29	0.13
lt_walks_v1	0.9 ± 0.08	0.24 ± 0.04	0.16	0.38	0.22	0.33
wt_bernoulli_v1	0.97 ± 0.06	0.06 ± 0.05	-	-	-	-
wt_changepoints_v1	0.94 ± 0.01	0.14 ± 0.02	0.12	0.14	0.09	0.06
wt_intake_impulse_v1	1.00 ± 0.00	0.32 ± 0.08	-	-	-	-
wt_pc_validate_v1	0.78 ± 0.00	0.14 ± 0.0	-	-	-	-
wt_pressure_control_v1	0.92 ± 0.0	0.36 ± 0.0	0.21	0.29	0.26	0.25
wt_test_v1	0.94 ± 0.08	0.14 ± 0.13	-	-	-	-
wt_validate_v1	0.97 ± 0.05	0.03 ± 0.04	-	-	-	-
wt_walks_v1	0.92 ± 0.03	0.28 ± 0.06	-	-	-	-

Results We observe very high varsortability values for all datasets; almost all datasets have values over 0.9 with the wt_pc_validate_v1 dataset beeing the only exception at 0.78 as shown in Table 3. All R^2-sortability values are below 0.25. Most of them are between 0.2 and 0.32. The lt_camera_validate_v1, lt_malus_v1, lt_test_v1, wt_bernoulli_v1 and wt_validate_v1 have values very close to 0. The values for each individual experiment contained in one dataset can be found in Appendix C. As shown in Table Table 3 varsortnregress outperforms other

[3] `https://github.com/juangamella/causal-chamber`

causal discovery algorithms on 9 out of 10 of the selected Causal Chamber datasets. We only report F1-scores for 10 of the data sets since for For wt_pc_validate_v1 there are not enough samples to select the hyperparameter for sortnregress and the other 6 data sets have over $50,000$ samples resulting in very long execution times.

6 Discussion and Conclusion

In general, both varsortability and R^2-sortability are present in both simulated and real datasets for benchmarking causal discovery algorithms. In line with Reisach et al. (2021), we observe that DYNOTEARS, which is a NOTEARS based algorithm, seems to perform better when varsortability is higher, which is in line with our hypothesis in section 3.

Furthermore, we observe that the data used in the NEURIPS competition(Runge et al., 2019) are highly varsortable, especially the realistic datasets, by our defined time series varsortability metric. This is also reflected by the good performance of sortnregress, which should be too simple to perform as good as the more sophisticated PCMCI$^+$ algorithm.

The low varsortability in the simulated var models could potentially be due to the fact that these datasets do not have contemporaneous dependencies, and as we showed earlier, contemporaneous dependencies seem to be the driver for high varsortability in simulated time series data. This is hardly surprising given that a team exploiting this effect won the competition(Weichwald et al., 2020). As for R^2-sortability, we see that high values can lead to better performance of simple benchmark algorithms. However, R^2-sortability values observed on the NEURIPS and ER Datasets seem to be too low to be exploited by the sorting algorithm.

The low varsortability value observed for the river data, particularly for the Danube, may be attributed to the fact that the width and catchment area of a river increase from the source to the mouth, resulting in a reduction in the impact of extreme flows on river velocity closer to the river's mouth. Consequently, the marginal variance decreases. This serves to illustrate the importance of scales in real dataset. The data for the other rivers only covers parts of the river, which probably results in a less intense effect and higher values for varsortability.

In addition, we can see that the Causal Chamber data is highly varsortable overall while having low R^2-values. This could be due to the fact that the variables controlled by the user are high in the causal order. We conjecture that deeper in the system and further from the user-controlled variables, dynamic turbulence and other sources of noise start having a larger and larger influence. Thus, unexplained noise is higher the lower we are in the causal order. This is in line with the low R^2-sortability values as the causal parents explain less and less down the causal order. The increase in noise variance also drives the increase in total marginal variance and affects varsortability in this way.

Limitations This paper is an empirical study and we do not determine analytically why high varsortability leads to better performance of score based causal discovery algorithms for time series data. While we believe that our explanations for varsortability in the considered real-world datasets are plausible, they should be treated cautiously as hypotheses only. The only thing that we can say with certainty that sortability is highly context-dependent and therefore discarding scales as arbitrary for causal discovery seems premature.

Conclusion In conclusion, our paper represents an empirical extension of the work of Reisach et al. (2021; 2023). We demonstrate that high var-sortability occurs in SVAR-simulated time series data, resulting in enhanced performance of continuous score-based causal discovery algorithms assuming equal noise variance. Moreover, in some settings our simple benchmark algorithms outperformed or were en par with more sophisticated algorithms in the presence of high sortability. Consequently, we advise caution when assuming equal noise variance for time series causal discovery algorithms. Furthermore, it may be advisable to examine simulated data for high varsortability before using them as benchmark data, as was done in the 2019 NEURIPS competition(Runge et al., 2019). Finally, we observe high and low varsortability, as well as R^2-sortability, in two different types of real-life datasets: the Causal Chamber data is generated in a controlled environment while the river flow dataset is measured in an uncontrolled environment. This indicates that var- and R^2-sortability in auto-correlated time series data is not solely a phenomenon observed in simulated data. For this reason, we believe that marginal variances may contain relevant causal information and exploiting variance or inverse variance sorting may be justified in some situations, if one can combine it with physical reasons for one or the other (even though these physical reasons might already eliminate the need for causal discovery in the first place). Furthermore, the observed R^2-sortability scores indicate that the assumption of equal noise variance is equally tricky as the relative fraction of unexplained variance may change throughout the graph and unequal noise variances are one possible reason for this.

Acknowledgments

JW was supported by the European Research Council (ERC) Starting Grant CausalEarth under the European Union's Horizon 2020 research and innovation program (Grant AgreementNo. 948112, PI Prof. Jakob Runge) as well as by the European Regional Development Fund (ERDF) and the German Federal State of Saarland as part of the project (To)CERTAIN. During the initial phase of this project, JW was employed at the Technical University of Berlin and affiliated with the DLR Institute of Data Science as a guest researcher.

References

Til Ole Bergmann and Gesa Hartwigsen. Inferring causality from noninvasive brain stimulation in cognitive neuroscience. *Journal of cognitive neuroscience*, 33(2):195–225, 2021.

Bart Bussmann, Jannes Nys, and Steven Latré. Neural additive vector autoregression models for causal discovery in time series. In *Discovery Science: 24th International Conference, DS 2021, Halifax, NS, Canada, October 11–13, 2021, Proceedings 24*, pp. 446–460. Springer, 2021.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Juan L. Gamella, Peter Bühlmann, and Jonas Peters. The causal chambers: Real physical systems as a testbed for ai methodology, 2024.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

J Munoz-Marí, G Mateo, J Runge, and G Camps-Valls. Causeme: An online system for benchmarking causal discovery methods. *In Preparation*, 2020.

Mark Newman. *Networks*. Oxford university press, 2018.

Ignavier Ng, Biwei Huang, and Kun Zhang. Structure Learning with Continuous Optimization: A Sober Look and Beyond. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, pp. 71–105. PMLR, March 2024. URL `https://proceedings.mlr.press/v236/ng24a.html`.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1595–1605. PMLR, 2020.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021.

Alexander Gilbert Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1388–1397. PMLR, 2020.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Ngoc Mai Tran, Johannes Buck, and Claudia Klüppelberg. Estimating a directed tree for extremes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. qkad165, 2024.

Hal R Varian. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, 113 (27):7310–7315, 2016.

Sebastian Weichwald, Martin E Jakobsen, Phillip B Mogensen, Lasse Petersen, Nikolaj Thams, and Gherardo Varando. Causal structure learning from time series: Large regression coefficients may predict causal links better in practice than small p-values. In *NeurIPS 2019 Competition and Demonstration Track*, pp. 27–36. PMLR, 2020.

Afsaneh Yazdani and Eric Boerwinkle. Causal inference in the age of decision medicine. *Journal of data mining in genomics & proteomics*, 6(1), 2015.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

A Var and R2-sortability for different sizes of ER Graphs

Table 4: Mean of and standard deviation denoted by ± of varsortability(var) and R^2-sortability (R^2) for ER Graphs with a different number of nodes. For each number of nodes 500 random graphs have been created and 500 samples have been generated.

	10 Nodes		20 Nodes		50 Nodes		100 Nodes	
	var	R^2	var	R^2	var	R^2	var	R^2
Contemp	0.71 ± 0.16	0.54 ± 0.22	0.72 ± 0.12	0.52 ± 0.15	0.71 ± 0.08	0.50 ± 0.12	0.72 ± 0.06	0.50 ± 0.08
Lagged	0.56 ± 0.18	0.49 ± 0.18	0.54 ± 0.13	0.49 ± 0.13	0.54 ± 0.11	0.50 ± 0.11	0.54 ± 0.10	0.49 ± 0.09
Overall	0.58 ± 0.09	0.51 ± 0.10	0.56 ± 0.06	0.51 ± 0.06	0.54 ± 0.03	0.50 ± 0.03	0.54 ± 0.02	0.50 ± 0.02

B Investigation on Influence of Different Degrees

B.1 On Varsortability

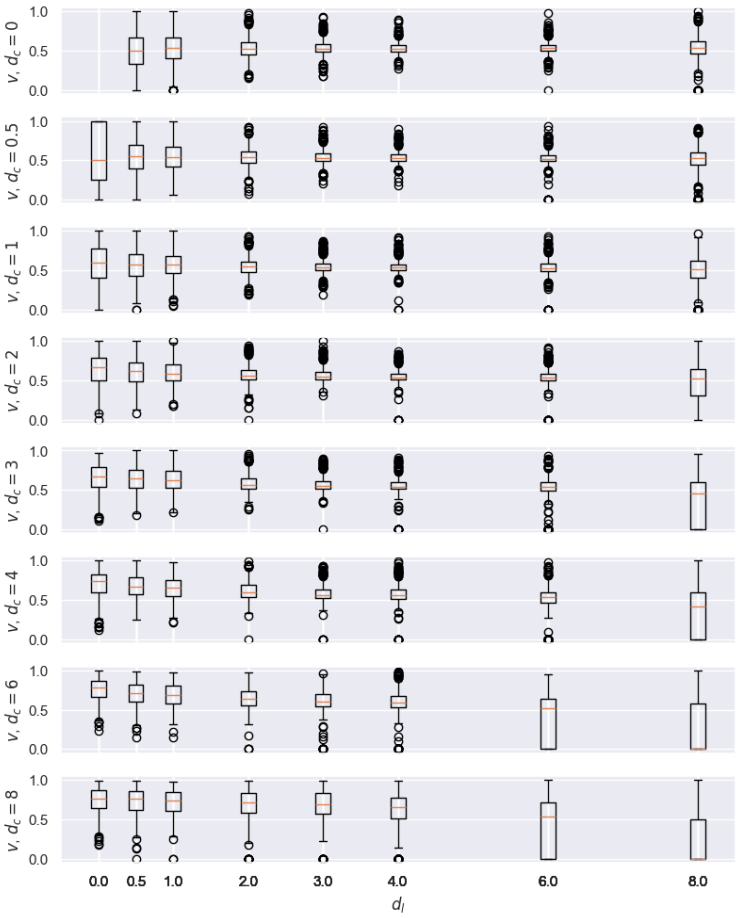


Figure 4: Varsortability for $d = 10$ nodes for different contemporaneous degrees d_c and lagged degrees d_l

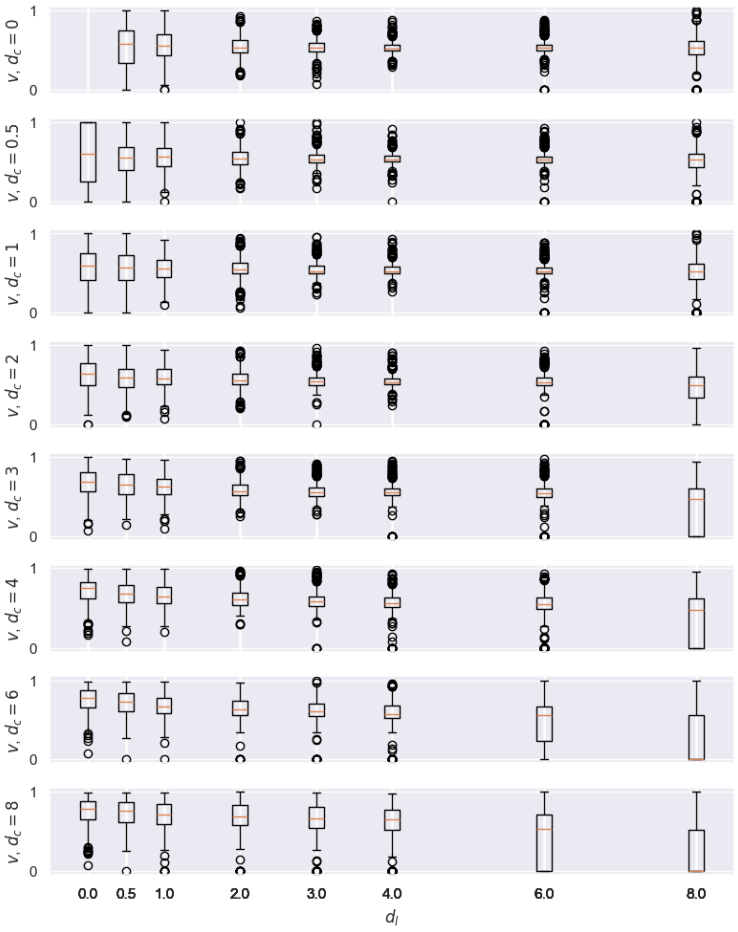


Figure 5: Varsortability for $d = 20$ nodes for different contemporaneous degrees d_c and lagged degrees d_l

B.2 On R2-sortability

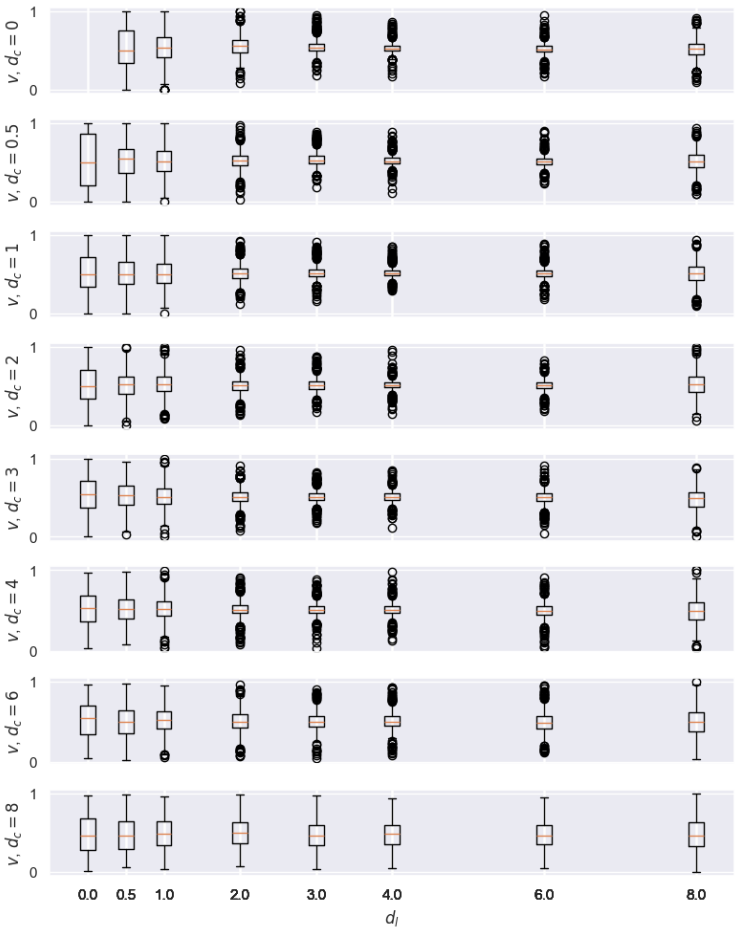


Figure 6: R^2-sortability for $d = 10$ nodes for different contemporaneous degrees d_c and lagged degrees d_l

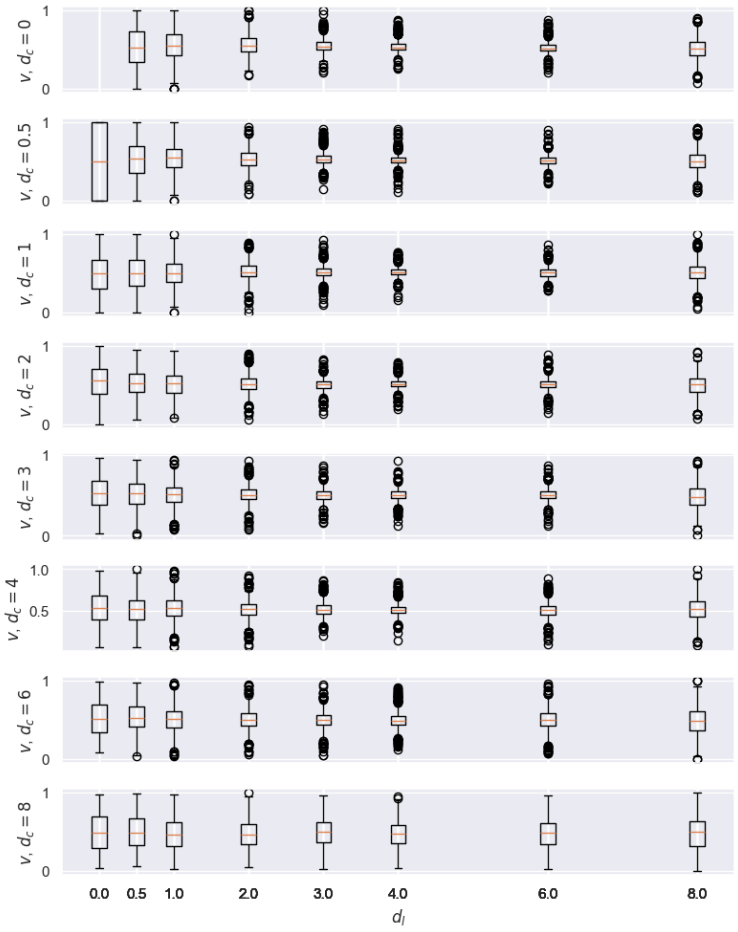


Figure 7: R^2-sortability for $d = 20$ nodes for different contemporaneous degrees d_c and lagged degrees d_l

C Results Causalchamber Data

Dataset	Experiment	varsortability	R^2-sortability
lt_camera_walks_v1	color_mix	0.95	0.28
lt_camera_walks_v1	actuator_mix	0.95	0.18
lt_color_regression_v1	pol_1_90	0.89	0.21
lt_color_regression_v1	aperture_11.0	0.95	0.16
lt_color_regression_v1	iso_1000.0	0.95	0.18
lt_color_regression_v1	bright_colors	0.95	0.00
lt_color_regression_v1	shutter_speed_0.002	0.95	0.19
lt_color_regression_v1	aperture_5.0	0.95	0.16
lt_color_regression_v1	pol_1_45	0.95	0.16
lt_color_regression_v1	reference	0.95	0.18
lt_color_regression_v1	iso_500.0	0.95	0.16
lt_color_regression_v1	shutter_speed_0.001	0.95	0.14
lt_interventions_standard_v1	uniform_t_ir_1_mid	0.95	0.44
lt_interventions_standard_v1	uniform_diode_vis_1_mid	0.95	0.47
lt_interventions_standard_v1	uniform_osr_angle_2_mid	0.95	0.51
lt_interventions_standard_v1	uniform_l_12_mid	0.95	0.49
lt_interventions_standard_v1	uniform_diode_ir_3_mid	0.95	0.49
lt_interventions_standard_v1	uniform_t_ir_1_strong	0.95	0.40
lt_interventions_standard_v1	uniform_diode_ir_3_strong	0.95	0.47
lt_interventions_standard_v1	uniform_red_mid	0.95	0.47
lt_interventions_standard_v1	uniform_diode_ir_2_mid	0.95	0.46
lt_interventions_standard_v1	uniform_osr_angle_2_strong	0.95	0.47
lt_interventions_standard_v1	uniform_t_vis_3_strong	0.82	0.46
lt_interventions_standard_v1	uniform_pol_1_strong	0.86	0.46
lt_interventions_standard_v1	uniform_osr_angle_2_weak	0.95	0.46
lt_interventions_standard_v1	uniform_pol_2_mid	0.95	0.46
lt_interventions_standard_v1	uniform_t_ir_3_weak	0.95	0.49
lt_interventions_standard_v1	uniform_t_ir_2_weak	0.95	0.47
lt_interventions_standard_v1	uniform_diode_vis_2_mid	0.95	0.47
lt_interventions_standard_v1	uniform_t_vis_1_weak	0.95	0.46
lt_interventions_standard_v1	uniform_l_11_mid	0.95	0.53
lt_interventions_standard_v1	uniform_osr_angle_1_mid	0.95	0.44
lt_interventions_standard_v1	uniform_v_angle_1_strong	0.91	0.53
lt_interventions_standard_v1	uniform_osr_c_weak	0.95	0.47
lt_interventions_standard_v1	uniform_t_ir_2_mid	0.95	0.44
lt_interventions_standard_v1	uniform_osr_c_strong	0.95	0.42
lt_interventions_standard_v1	uniform_diode_ir_1_strong	0.95	0.47
lt_interventions_standard_v1	uniform_t_ir_3_mid	0.95	0.47
lt_interventions_standard_v1	uniform_t_ir_3_strong	0.91	0.42
lt_interventions_standard_v1	uniform_pol_1_mid	0.91	0.47
lt_interventions_standard_v1	uniform_diode_vis_3_mid	0.86	0.44
lt_interventions_standard_v1	uniform_diode_ir_1_mid	0.95	0.46
lt_interventions_standard_v1	uniform_blue_mid	0.95	0.37
lt_interventions_standard_v1	uniform_osr_c_mid	0.95	0.47
lt_interventions_standard_v1	uniform_t_vis_1_strong	0.95	0.47
lt_interventions_standard_v1	uniform_green_mid	0.95	0.49
lt_interventions_standard_v1	uniform_diode_ir_2_strong	0.95	0.47
lt_interventions_standard_v1	uniform_osr_angle_1_weak	0.95	0.46
lt_interventions_standard_v1	uniform_v_angle_2_mid	0.95	0.53
lt_interventions_standard_v1	uniform_t_ir_1_weak	0.95	0.46
lt_interventions_standard_v1	uniform_t_vis_2_strong	0.91	0.46
lt_interventions_standard_v1	uniform_blue_strong	0.95	0.32
lt_interventions_standard_v1	uniform_t_vis_1_mid	0.95	0.49
lt_interventions_standard_v1	uniform_l_32_mid	0.95	0.47

lt_interventions_standard_v1	uniform_l_22_mid	0.95	0.49
lt_interventions_standard_v1	uniform_v_c_mid	0.95	0.47
lt_interventions_standard_v1	uniform_green_strong	0.95	0.42
lt_interventions_standard_v1	uniform_v_angle_2_strong	0.91	0.51
lt_interventions_standard_v1	uniform_t_vis_3_mid	0.86	0.46
lt_interventions_standard_v1	uniform_osr_angle_1_strong	0.95	0.47
lt_interventions_standard_v1	uniform_t_ir_2_strong	0.95	0.42
lt_interventions_standard_v1	uniform_v_angle_1_mid	0.95	0.46
lt_interventions_standard_v1	uniform_pol_2_strong	0.86	0.49
lt_interventions_standard_v1	uniform_red_strong	0.95	0.51
lt_interventions_standard_v1	uniform_v_c_strong	1.00	0.46
lt_interventions_standard_v1	uniform_t_vis_2_mid	0.95	0.42
lt_interventions_standard_v1	uniform_t_vis_2_weak	0.95	0.42
lt_interventions_standard_v1	uniform_t_vis_3_weak	0.91	0.47
lt_interventions_standard_v1	uniform_l_31_mid	0.95	0.47
lt_interventions_standard_v1	uniform_l_21_mid	0.95	0.49
lt_interventions_standard_v1	uniform_reference	0.95	0.49
lt_walks_v1	actuators_white	0.96	0.26
lt_walks_v1	color_mix	0.84	0.21
wt_walks_v1	actuators_random_walk_9	0.93	0.26
wt_walks_v1	actuators_random_walk_8	0.93	0.19
wt_walks_v1	loads_hatch_mix_slow_run_2	0.93	0.24
wt_walks_v1	actuators_random_walk_6	0.93	0.36
wt_walks_v1	actuators_random_walk_7	0.88	0.24
wt_walks_v1	loads_hatch_mix_slow_run_3	1.00	0.36
wt_walks_v1	loads_hatch_mix_slow_run_1	0.93	0.29
wt_walks_v1	actuators_random_walk_5	0.95	0.29
wt_walks_v1	actuators_random_walk_4	0.90	0.21
wt_walks_v1	loads_hatch_mix_slow_run_4	0.93	0.29
wt_walks_v1	actuators_random_walk_1	0.86	0.14
wt_walks_v1	loads_hatch_mix_slow_run_5	0.93	0.26
wt_walks_v1	actuators_random_walk_3	0.95	0.24
wt_walks_v1	actuators_random_walk_2	0.88	0.24
wt_walks_v1	actuators_random_walk_11	0.93	0.31
wt_walks_v1	actuators_random_walk_10	0.86	0.21
wt_walks_v1	actuators_random_walk_12	0.93	0.31
wt_walks_v1	loads_hatch_mix_fast_run_5	0.93	0.33
wt_walks_v1	loads_hatch_mix_fast_run_4	0.93	0.33
wt_walks_v1	actuators_random_walk_13	0.90	0.29
wt_walks_v1	loads_hatch_mix_fast_run_1	0.93	0.33
wt_walks_v1	actuators_random_walk_16	0.88	0.24
wt_walks_v1	actuators_random_walk_14	0.93	0.24
wt_walks_v1	loads_hatch_mix_fast_run_3	0.93	0.31
wt_walks_v1	loads_hatch_mix_fast_run_2	1.00	0.36
wt_walks_v1	actuators_random_walk_15	0.93	0.31
lt_malus_v1	red_255	1.00	0.04
lt_malus_v1	white_128	1.00	0.02
lt_malus_v1	green_64	0.96	0.02
lt_malus_v1	blue_255	1.00	0.02
lt_malus_v1	green_128	0.96	0.00
lt_malus_v1	white_64	1.00	0.02
lt_malus_v1	blue_64	0.96	0.02
lt_malus_v1	green_255	1.00	0.00
lt_malus_v1	blue_128	1.00	0.04
lt_malus_v1	white_255	1.00	0.02
lt_malus_v1	red_64	0.96	0.02
lt_malus_v1	red_128	0.96	0.04
wt_bernoulli_v1	random_loads_both	1.00	0.00

wt_bernoulli_v1	fans_off	0.90	0.10
wt_bernoulli_v1	random_loads_intake	1.00	0.07
wt_changepoints_v1	load_in_seed_8	0.94	0.13
wt_changepoints_v1	load_in_seed_9	0.98	0.10
wt_changepoints_v1	load_in_seed_4	0.94	0.13
wt_changepoints_v1	load_in_seed_5	0.94	0.15
wt_changepoints_v1	load_in_seed_7	0.94	0.13
wt_changepoints_v1	load_in_seed_6	0.94	0.15
wt_changepoints_v1	load_in_seed_2	0.94	0.15
wt_changepoints_v1	load_in_seed_3	0.94	0.15
wt_changepoints_v1	load_in_seed_1	0.94	0.15
wt_changepoints_v1	load_in_seed_0	0.94	0.15
wt_intake_impulse_v1	load_out_0.5_osr_downwind_4	1.00	0.29
wt_intake_impulse_v1	load_out_0.5_osr_downwind_2	1.00	0.29
wt_intake_impulse_v1	load_out_1_osr_downwind_4	1.00	0.26
wt_intake_impulse_v1	load_out_0.5_osr_downwind_8	1.00	0.29
wt_intake_impulse_v1	load_out_0.01_osr_downwind_4	1.00	0.45
wt_pressure_control_v1	hatch_0	0.92	0.36
lt_test_v1	current_sensor	1.00	0.04
lt_test_v1	angle_sensors	0.96	0.00
lt_test_v1	analog_calibration	0.93	0.00
lt_test_v1	ir_sensors	0.96	0.00
wt_test_v1	zero_load	1.00	0.21
wt_test_v1	mic_effects	0.93	0.24
wt_test_v1	potis_coarse	0.83	0.12
wt_test_v1	tach_resolution	1.00	0.12
wt_test_v1	osr_mic	1.00	0.00
wt_test_v1	no_load	1.00	0.10
wt_test_v1	potis_fine	0.86	0.12
wt_test_v1	osr_barometers	0.81	0.10
wt_test_v1	analog_calibration	1.00	0.00
wt_test_v1	steps	0.93	0.43
lt_camera_test_v1	polarizer_effect_bright	0.95	0.00
lt_camera_test_v1	pure_colors_bright	0.92	0.50
lt_camera_test_v1	polarizer_effect_dark	0.95	0.07
lt_camera_test_v1	pure_colors_dark	0.95	0.50
lt_camera_test_v1	palette	0.95	0.18
wt_validate_v1	validate_v_2	1.00	0.00
wt_validate_v1	validate_v_in	1.00	0.02
wt_validate_v1	validate_load_out_pressure_intake	1.00	0.00
wt_validate_v1	validate_v_out	1.00	0.00
wt_validate_v1	validate_load_out_current_in	1.00	0.02
wt_validate_v1	validate_v_1	1.00	0.00
wt_validate_v1	validate_osr_1	1.00	0.00
wt_validate_v1	validate_pot_1	1.00	0.00
wt_validate_v1	validate_osr_downwind	0.88	0.12
wt_validate_v1	validate_res_out	1.00	0.00
wt_validate_v1	validate_osr_2	1.00	0.02
wt_validate_v1	validate_pot_2	1.00	0.00
wt_validate_v1	validate_load_out_mic	1.00	0.00
wt_validate_v1	validate_load_in_mic	1.00	0.02
wt_validate_v1	validate_load_in_current_out	1.00	0.02
wt_validate_v1	validate_hatch_rpms	0.90	0.02
wt_validate_v1	validate_osr_out	1.00	0.02
wt_validate_v1	validate_osr_upwind	0.88	0.12
wt_validate_v1	validate_res_in	1.00	0.02
wt_validate_v1	validate_load_in	1.00	0.00
wt_validate_v1	validate_v_mic	1.00	0.00

wt_validate_v1	validate_osr_intake	0.88	0.12
wt_validate_v1	validate_load_out	1.00	0.02
wt_validate_v1	validate_osr_ambient	0.90	0.12
wt_validate_v1	validate_osr_in	1.00	0.00
wt_validate_v1	validate_osr_mic	1.00	0.00
wt_validate_v1	validate_hatch_mic	0.93	0.07
wt_validate_v1	validate_hatch_pressures	0.90	0.02
wt_pc_validate_v1	validate_pressure_downwind_loads	0.78	0.14
lt_validate_v1	validate_l_11	0.96	0.00
lt_validate_v1	validate_osr_c	0.98	0.02
lt_validate_v1	validate_l_12	0.96	0.02
lt_validate_v1	validate_v_c	1.00	0.00
lt_validate_v1	validate_osr_angle_2	0.98	0.02
lt_validate_v1	validate_red	0.96	0.09
lt_validate_v1	validate_osr_angle_1	0.98	0.02
lt_validate_v1	validate_pol_1	0.94	0.06
lt_validate_v1	validate_diode_vis_2	1.00	0.02
lt_validate_v1	validate_diode_vis_3	1.00	0.02
lt_validate_v1	validate_pol_2	1.00	0.02
lt_validate_v1	validate_diode_vis_1	1.00	0.02
lt_validate_v1	validate_diode_ir_1	1.00	0.02
lt_validate_v1	validate_v_angle_1	1.00	0.02
lt_validate_v1	validate_diode_ir_2	1.00	0.02
lt_validate_v1	validate_diode_ir_3	1.00	0.02
lt_validate_v1	validate_v_angle_2	1.00	0.03
lt_validate_v1	validate_green	0.98	0.00
lt_validate_v1	validate_blue	0.98	0.00
lt_validate_v1	validate_l_31	0.96	0.02
lt_validate_v1	validate_l_32	0.96	0.00
lt_validate_v1	validate_t_ir_1	1.00	0.00
lt_validate_v1	validate_t_vis_3	1.00	0.00
lt_validate_v1	validate_t_vis_2	1.00	0.00
lt_validate_v1	validate_l_22	0.96	0.02
lt_validate_v1	validate_t_ir_2	1.00	0.02
lt_validate_v1	validate_t_vis_1	1.00	0.02
lt_validate_v1	validate_t_ir_3	1.00	0.02
lt_validate_v1	validate_l_21	0.98	0.02
lt_camera_validate_v1	validate_shutter_speed	1.00	0.00
lt_camera_validate_v1	validate_iso	1.00	0.00
lt_camera_validate_v1	validate_green	0.98	0.00
lt_camera_validate_v1	validate_pol_2	1.00	0.00
lt_camera_validate_v1	validate_blue	0.98	0.00
lt_camera_validate_v1	validate_aperture	1.00	0.00
lt_camera_validate_v1	validate_red	0.96	0.09
lt_camera_validate_v1	validate_pol_1	1.00	0.00