Department of Mathematics and Computer Science
Statistics Group

Structure Learning in High-Dimensional Time Series Data

Master Thesis

Martin de Quincey

Supervisors:
dr. Rui Castro
dr. Alex Mey

Assessment Committee Members:
dr. Rui Castro
dr. Alex Mey
dr. Jacques Resing

version 0.4

Eindhoven, June 2022

Contents

Chapter 7

Evaluation

Throughout Chapters 4, 5, and 6, we have presented several methods for learning an acyclic structure that characterizes the linear dependencies between the variables in time series data $\mathbf{X} \in \mathbb{R}^{T \times p}$. We have already highlighted the advantages and disadvantages of most methods in their respective chapters using toy examples, but the methods have not been objectively and quantitatively compared to each other. As no theoretical results have been provided, the methods will be evaluated using both simulated and real-life data.

Methods that will be evaluated. We have developed several methods. However, some methods will not be evaluated for several reasons. For example, the exhaustive approach in Section 4.1 is not tractable for more than ten variables. Furthermore, the method where we relaxed the set of permutation matrices to the Birkhoff polytope in Section 5.1 will not be evaluated, as the method did not properly enforce acyclicity and convergence was rather slow.

~~We will be evaluating the following methods~~. We will be evaluating the random walk approach as discussed in Section 4.2, where we can only transition to the permutations that are one transposition away, as defined in Equation 4.30 and Equation 4.31. Furthermore, we will be evaluating the regular Metropolis-Hastings approach discussed in Section 4.3, as well as the greedy Metropolis-Hastings variant, where we only transition to permutations that are able to achieve a strictly larger likelihood than the current permutation. The decision rule for the greedy Metropolis-Hastings approach was given in Equation 4.44. As a stopping criterion, we will terminate the algorithm after one thousand permutations have been evaluated.

From the continuous-based methods, we will be evaluating the NO TEARS approach modified for VAR(1) models as discussed in Section 5.2, as well as the DAG-LASSO approach as discussed in Section 5.3. Lastly, from the iterative approaches, we will be evaluating the DAG-OMP approach from Section **??**, and the DAG-OLS-V approach, where we will be using the "violators-first" approach as discussed in Subsection 6.3.1.

This yields a total of seven methods that we will be evaluating throughout this chapter.

7.1 Performance Criteria

We will evaluate the aforementioned methods based on several performance criteria. These performance criteria can be split into two categories. The first category consists of structural performance criteria, where we will compare the estimated matrix W to the ground truth W^* from a structural point of view. The second category consists of predictive performance criteria, where we will compare how well W can be used to predict $\mathbf{X}$ compared to W^*.

7.1.1 Structural Performance Criteria

Throughout this thesis, the focus has mainly been on *predictive performance*. We are not necessarily trying to recover W^*, but we are trying to find a matrix W that is able to accurately predict $X_{t,\cdot}$ using $X_{t-1,\cdot}W$.

Nevertheless, the structural performance criteria are widely used to assess the quality of methods in the structure learning community, for example in [21, 25, 66]. Furthermore, these structural performance criteria give insights into how similar the structures of W and W^* are. Therefore, we will also employ several structural performance criteria. Lastly, as we are also interested in recovering structures that are easy to interpret, we must verify that the coefficient matrix is sparse. As predictive performance criteria do not capture sparsity, analyzing structural performance criteria may prove useful to us as well.

Consider the setting where we want to objectively assess how closely the matrix W resembles the true matrix W^* from a structural point of view. For this purpose, we do not consider any data, but merely compare the coefficient matrices. We consider each entry in W^* to be equally important. Whether the coefficient in W^* is equal to 0.01 or 1.00, from a structural point of view, failing to recover any of these coefficients is considered equally problematic.

True Positive Rate (TPR). A first structural performance criterion is the percentage of non-zero coefficients we managed to recover. This metric is called the *true positive rate* (TPR). From a graphical perspective, the TPR corresponds to the ratio of arcs in W^* that are also in W. More formally, given the true matrix W^*, the true positive rate of a matrix W is

$$\mathrm{TPR}(W^*, W) = \frac{|\mathrm{supp}(W) \cap \mathrm{supp}(W^*)|}{|\mathrm{supp}(W^*)|}, \tag{7.1}$$

where where $\mathrm{supp}(W)$ is defined as the support of W, the indices of W that correspond to non-zero entries,

$$\mathrm{supp}(W) = \{(i, j) \mid w_{i,j} \neq 0\}.$$

Furthermore, $|\cdot|$ represents the cardinality of the set.

True Negative Rate (TNR). In a similar manner, we can look at the coefficients that the method correctly identified to be zero, which is called the *true negative rate* (TNR). From a graphical perspective, the TNR represents the ratio of arcs that were not in W^* and were also not in W. More formally, the true negative rate is defined as

$$\mathrm{TNR}(W^*, W) = \frac{|\overline{\mathrm{supp}}(W) \cap \overline{\mathrm{supp}}(W^*)|}{|\overline{\mathrm{supp}}(W^*)|}, \tag{7.2}$$

where $\overline{\mathrm{supp}}(W)$ is defined as the complement of the support of W, containing are all the indices in W corresponding to zero entries,

$$\overline{\mathrm{supp}}(W) = \{(i, j) \mid w_{i,j} = 0\}.$$

Structural Hamming Distance (SHD). The Structural Hamming Distance (SHD) is a performance metric that captures both the number of false positives and the number of false negatives. The SHD has first been defined in [62] to compare adjacency matrices of directed graphs. The structural hamming distance between two adjacency matrices A and B is defined as the smallest number of arc additions, deletions, and reversals in order to transform the graph $G(A)$ into the graph $G(B)$.

The SHD can be seen as a metric that on one hand verifies how many arcs of W^* are contained in W, and on the other hand how many missing arcs of W^* are also not contained in W. This combination is useful as we often want a trade-off between the number of true positives and the number of true negatives. We always obtain an optimal TPR by naive estimating all coefficients, and we can always obtain an optimal TNR by estimating no coefficients. To get an optimal SHD, however, we require W to contain exactly all arcs of W^* and no more. As the SHD considers an incorrect arc direction as only one mistake, we see that the SHD is also quite lenient with respect to arc discovery, as another metric such as the accuracy regards this as two mistakes.

In mathematical notation, the structural hamming distance is equal to

$$\begin{aligned}\text{SHD} &= \#\text{arc additions} + \#\text{arc deletions} + \#\text{arc reversals} \\ &= |\overline{\text{supp}}(W) \cap \text{supp}(W^*)| + |\text{supp}(W) \cap \overline{\text{supp}}(W^*)| - |\text{supp}(\tilde{W}^T) \cap \text{supp}(W^*)|, \end{aligned} \tag{7.3}$$

where the $\tilde{W}$ corresponds to the coefficient matrix W, where we have set the diagonal entries to zero,

$$\tilde{w}_{ij} = \begin{cases} w_{ij} & \text{if } i \neq j \\ 0 & \text{if } i = j. \end{cases}. \tag{7.4}$$

Note that the first two components of Equation 7.3 ensure that an arc reversal is counted as two mistakes. Therefore, the third component of Equation 7.3 subtracts one mistake for each incorrectly directed off-diagonal arc.

7.1.2 Predictive Performance Criteria

Predictive performance criteria quantify how useful a coefficient matrix W is for prediction. For our VAR(1) model, we are looking for a matrix W that is good at predicting $X_{t,\cdot}$ using $X_{t-1,\cdot}W$. Rather than looking at how close W is to the true data generating matrix W^* in terms of structure, in predictive performance we consider how close $X_{t-1,\cdot}W$ is to $X_{t,\cdot}$, or rather $X_{t-1,\cdot}W^*$.

Empirical risk We can consider the *empirical risk*. The empirical risk of a coefficient matrix W on a data matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$ is defined as

$$\begin{aligned} R_{\text{emp}}(W) &= \frac{1}{T-1} \|\mathbf{X}_{2:T,\cdot} - \mathbf{X}_{1:T-1,\cdot}W\|_F^2 \\ &= \frac{1}{T-1} \sum_{t=2}^{T} \|X_{t,\cdot} - X_{t-1,\cdot}W\|_2^2 \end{aligned} \tag{7.5}$$

This is in fact equivalent to the mean squared error, as was defined in Equation 4.25.

A lower the empirical risk of W indicates that it was more likely that $\mathbf{X}$ has been generated by a VAR(1) model characterized by W. Therefore, we can say that the method that provides a coefficient matrix W which achieves the smallest empirical risk performs best on $\mathbf{X}$.

True Risk. Although the empirical risk adequately assesses how suitable a coefficient matrix W predicts $\mathbf{X}$, we also want to assess whether W achieve a low empirical risk on similar data $\mathbf{X}'$.

Therefore, we can also consider the *true risk*. Such a performance criterion can only be evaluated when the data generating model is available. For example, when we generate data according to a VAR(1) model,

$$X_{t,\cdot} = X_{t-1,\cdot}W^* + \varepsilon_t, \tag{7.6}$$

we can calculate the expected risk of W as

$$\begin{aligned} R(W) &= \mathbb{E}\left[\|X_{t,\cdot} - X_{t-1,\cdot}W\|_2^2\right] \\ &= \text{Tr}\left((W^* - W)^T \mathbb{V}(X_{t,\cdot})(W^* - W) + \mathbb{V}(\varepsilon_t)\right), \end{aligned} \tag{7.7}$$

where we can compute the variance of $\mathbb{V}(X_{t,\cdot})$ as

$$\text{vec}\left(\mathbb{V}(X_{t,\cdot})\right) = \left(I_{p^2} - (W^* \otimes W^*)^T\right)^{-1} \text{vec}\left(\mathbb{V}(\varepsilon_t)\right). \tag{7.8}$$

Note that the true risk is lower bounded by $\text{Tr}\left(\mathbb{V}(\varepsilon_t)\right)$, which will be equal to p throughtout all simulations.

7.2 Time Series Experiments

Now that we have defined the necessary performance criteria to objectively evaluate the methods discussed in this thesis, we will propose the following three types of time series experiments.

First and foremost, we will simulate the optimal setting where the data $\mathbf{X}$ has been generated by a VAR(1) model with an *acyclic* coefficient matrix W^*. This is the exact setting we are assuming, and therefore we expect our methods to perform quite well. The results of these experiments will be discussed in Subsection 7.2.1.

Secondly, we will simulate a slightly sub-optimal setting where the data $\mathbf{X}$ has been generated by a VAR(1) model, but the coefficient matrix W^* is *cyclic*. Therefore, we cannot expect to find an acyclic coefficient matrix W that exactly resembles W^*. Nevertheless, we hope our methods are still able to retrieve a reasonably suitable acyclic coefficient matrix W. The results of these experiments will be discussed in Subsection 7.2.2.

Lastly, we will verify our methods on *real-life* data. In real-life settings, the VAR(1) modeling assumption is most likely violated. However, verifying these methods on real-life data will provide interesting findings in the directed relations between the variables. These experiments will be discussed in Subsection 7.2.3.

Generating W^* and $\mathbf{X}$. To generate the true coefficient matrix W^*, we first specify the number of variables p. For the time-series experiments, the p auto regressive coefficients on the diagonal will be set to 0.5. This value is rather low, but necessary to ensure stationarity when there are numerous off-diagonal entries, especially for large values of p.

Lastly, a total of s off-diagonal arcs will be set to 0.5 such that W^* corresponds to an acyclic structure in Subsection 7.2.1, and to a cyclic structure in Subsection 7.2.2.

Given our coefficient matrix W^*, we can generate our data matrices $\mathbf{X} \in \mathbb{R}^{T \times p}$ by generating T samples according to a VAR(1) model as defined in Definition 2.3. The noise variables are all Gaussian random variables with mean zero and identity covariance. To ensure stationarity, $X_{1,\cdot}$ will have a covariance corresponding to its stationary distribution.

Experimental Setups. There are many parameters with respect to the data generating process that we can consider. As we are predominantly interested in high-dimensional time series, we will carefully investigate the influence of p. Furthermore, we will consider the setting where T is small and where T is large. Thirdly, we can change the sparsity of the coefficient matrix W by altering s, the number of off-diagonal arcs. Therefore, we will consider the following range of parameters.

- The number of variables $p \in \{5, 10, 15, 25, 50\}$, ranging from low-dimensional to high-dimensional.
- The number of time steps $T \in \{100, 1000\}$, corresponding to few time steps and many time steps.
- The number of arcs $s \in \{3p, 5p\}$, corresponding to three outgoing arcs per variable (sparse setting) and five outgoing arcs per variable (dense setting), respectively. The number of arcs will be thresholded to $p(p-1)/2$ as that is the maximum number of off-diagonal arcs in a directed acyclic graph.

For each tuple (p, T, s), we generate a total of $N = 10$ data sets to obtain reliable estimates. We will compute the mean, as well as the corresponding standard errors to express uncertainty.

For all methods, we will first compute the mean squared error or empirical risk $R_{\text{emp}}(W)$ as defined in Equation 7.5. Subsequently, we will threshold all coefficients in W with an absolute value smaller than $\epsilon = 0.30$ to zero in order to obtain a suitable number of arcs. Based on this thresholded matrix, we will compute the TPR, TNR, and SHD according to Equation 7.1, Equation 7.2, and Equation 7.3, respectively. Lastly, we will first reestimate the non-zero coefficients of the thresholded matrix to compute the true risk $R(W)$ as defined in Equation 7.7.

7.2.1 Simulated VAR(1) data with an acyclic coefficient matrix W^*.

Let us consider the setting where we have a sparse coefficient matrix, meaning that each variable has an average of three incoming arcs. For the results on dense acyclic coefficient matrices, we refer the interested reader to Section B.2 in the appendix.

The results for the true risk $R(W)$ are given in Table 7.1. Furthermore, the results with the corresponding standard errors have been plotted as a function of p for $T = 100$ in Figure 7.1, and for $T = 1000$ in Figure 7.2. For the Structural Hamming Distance, the results are given in Table 7.2, and the corresponding plots are given in Figure 7.3 and Figure 7.4. For readers interested in the empirical risk, we refer to Table B.1, Figure B.1, and Figure B.2 in Section B.1 of the appendix.

Table 7.1: Average true risk $R(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W corresponds to an acyclic structure. A lower true risk indicates a better predictive performance.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**5.26**	12.70	20.85	41.30	109.70	**5.02**	11.92	19.41	37.87	89.57
MH-Regular	**5.26**	12.47	20.85	41.28	109.54	**5.02**	11.87	20.35	37.91	95.08
MH-Greedy	5.38	11.41	17.02	28.95	68.01	**5.02**	10.46	15.47	25.67	54.62
NO TEARS	**5.26**	10.59	**16.08**	**27.08**	**61.03**	**5.02**	**10.04**	15.07	**25.10**	**50.19**
DAG-LASSO	12.32	51.51	68.16	148.08	353.65	10.72	46.02	69.44	136.20	290.74
DAG-OMP	**5.26**	11.45	17.51	27.88	63.64	**5.02**	10.52	15.85	25.34	50.62
DAG-OLS-V	5.28	**10.55**	16.15	27.61	128.07	**5.02**	**10.04**	**15.06**	**25.10**	**50.19**

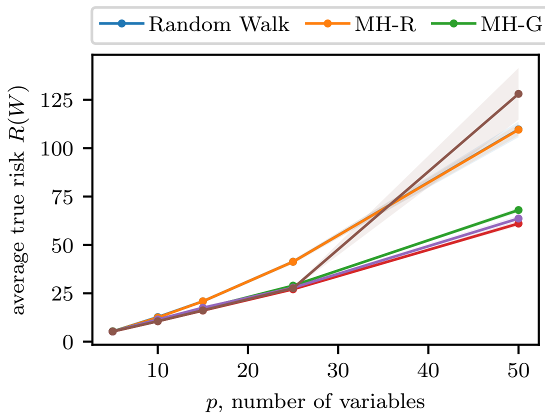

Figure 7.1: Plot of the average true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.1, excluding DAG-LASSO.

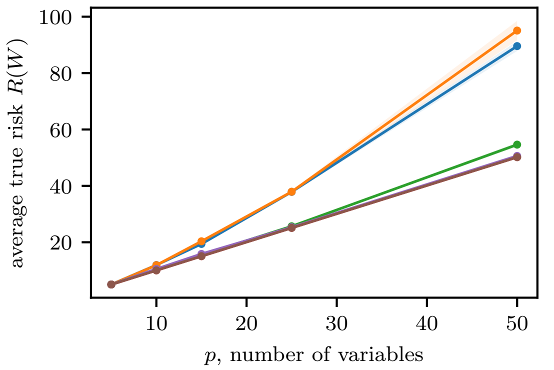

Figure 7.2: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.1, excluding DAG-LASSO.

Inspecting the true risk, we see immediately that DAG-LASSO achieves by far the largest true risk, indicating that it is the least suitable method of all. As its large values will skew the plots, we have decided not to include the DAG-LASSO results.

Secondly, the random walk and the regular Metropolis-Hastings approach also achieve a relatively okay true risk for $p \in \{5, 10, 15\}$, after which the predictive performance becomes quite poor. This is most likely due to the exponential increase of the search space. When only 1000 permutations can be tried, this is enough to exhaustively try all permutations for $p = 5$, and a reasonable subset for $p = 10$ and $p = 15$. However, for $p = 25$, we can only cover a minuscule portion of $25!/1000 \approx 10^{-20}\%$ of the search space, and therefore it is reasonable to expect that these permutation-based approaches will decrease in performance as p gets larger. Interestingly, the regular Metropolis-Hastings approach does not seem to be a significant improvement over the random walk, which may be because the coefficient matrix is relatively sparse, as also discussed in Example 4.6. On the other hand, the greedy Metropolis-Hastings approach seems to be performing surprisingly well, even though only such a minuscule fraction of the search space has been explored. It seems just slightly poorer than the NO TEARS and DAG-OMP approach. Apparently, the exploitative decision rule of the greedy Metropolis-Hastings approach can efficiently traverse the search space of permutation matrices using only a small number of transitions.

The NO TEARS, DAG-OMP, and DAG-OLS all three seem be performing quite good. Note that, although the true risks increase, note that the optimal coefficient matrix will still yield a true risk of p due to the noise. Therefore, we see that NO TEARS and DAG-OMP seem to be performing close to optimal here. DAG-OLS-V, quite surprisingly, seems to on par with NO TEARS and DAG-OMP. However, DAG-OLS-V seems to be performing quite poorly for $T = 100$ and $p = 50$. As the number of possible arcs have increased fourfold compared to $p = 25$, the spurious correlations with the noise seem to be quite troublesome for DAG-OLS-V.

All methods seem to benefit from having a larger sample size T. Interestingly, the permutation-based approaches do not seem to have a large increase in performance when the sample size increases. The reason for this is that the bottleneck was not that there were not enough samples, but that there was not enough time to find a suitable permutation matrix. Having more samples only helps slightly in a permutation-based approach, whereas having more samples helps greatly in finding suitable arcs in an iterative approach.

Table 7.2: Average structural hamming distance (SHD) as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W corresponds to an acyclic structure. A lower SHD indicates a better structural performance.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**0.3**	5.4	11.2	29.5	97.5	**0.0**	4.5	9.1	22.6	57.7
MH-Regular	**0.3**	5.2	11.2	30.6	99.6	**0.0**	4.3	10.8	20.8	58.4
MH-Greedy	0.7	2.3	2.4	6.1	33.2	**0.0**	1.1	0.8	1.2	6.0
NO TEARS	**0.3**	0.4	**0.7**	**1.9**	**20.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
DAG-LASSO	12.1	37.2	54.5	92.8	188.1	9.1	36.1	55.0	91.9	184.6
DAG-OMP	**0.3**	1.6	1.4	3.6	22.3	**0.0**	0.6	0.9	0.4	0.3
DAG-OLS-V	0.4	**0.3**	0.8	2.9	58.1	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

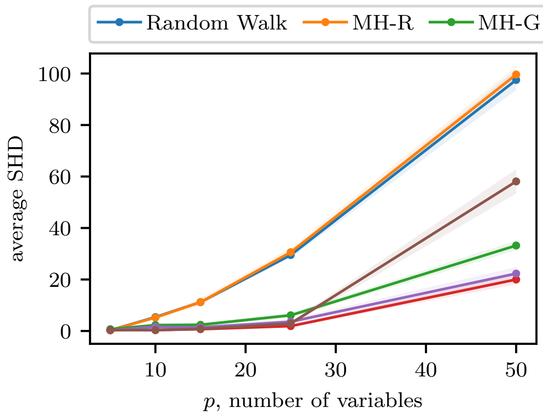

Figure 7.3: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.2, excluding DAG-LASSO.

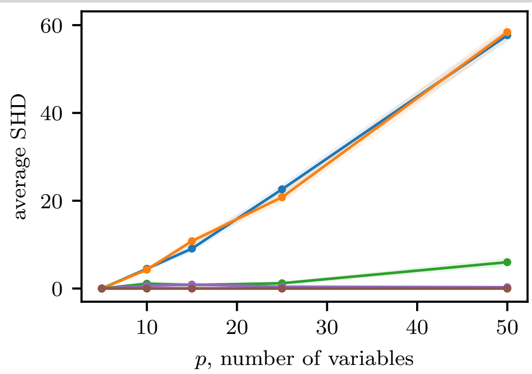

Figure 7.4: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.2, excluding DAG-LASSO.

Considering the structural hamming distance, we see comparable results as with the true risk. Firstly, all methods improve as we have more samples, most notably the DAG-OLS-V approach, most likely because it relies on a suitable initial ordinary least squares estimate. We also see a quite sharp increase in SHD for all methods for $T = 100$ at $p = 50$, indicating that the difficulty of recovering a suitable structure increases when we have many variables yet few time steps.

Again, we see that the random walk and the regular Metropolis-Hastings approach perform quite poor compared to the other methods. The greedy Metropolis-Hastings approach seems to be performing quite well, but NO TEARS and DAG-OMP both seem to be performing slightly better, especially for larger values of p. Interestingly, these three methods either almost always exactly recover the true coefficient matrix W^* when we have enough samples. Again, DAG-OLS-V seems to be performing surprisingly well apart from the scenario where $T = 100$ and $p = 50$.

7.2.2 Simulated VAR(1) data with a cyclic coefficient matrix W^*.

Let us consider the setting where we have the coefficient matrix W^* is cyclic. Therefore, we can never achieve a structural hamming distance of zero, or a true risk as low as when W^* was acyclic. Nevertheless, it is interesting to see which methods cope best with this difficulty.

The results for the true risk $R(W)$ are given in Table 7.3. Furthermore, these results accompanied by standard errors have been plotted as a function of p for $T = 100$ in Figure 7.5, and for $T = 1000$ in Figure 7.6. For the Structural Hamming Distance, the results are given in Table 7.4, and the corresponding plots are given in Figure 7.7 and Figure 7.8. For readers interested in the empirical risk, we refer to Table B.7, Figure B.13, and Figure B.14 in Section B.3 of the appendix.

Table 7.3: Average true risk $R(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W^* corresponds to a cyclic structure. A lower true risk indicates a better predictive performance.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**6.79**	13.88	21.32	40.23	95.62	**6.61**	13.10	19.97	36.75	77.33
MH-Regular	**6.79**	**13.80**	20.64	41.22	93.31	**6.61**	13.29	20.12	37.06	79.32
MH-Greedy	7.01	13.85	19.94	35.72	81.01	6.75	13.02	18.39	**31.50**	63.20
NO TEARS	6.83	13.47	**19.57**	**34.58**	74.41	6.66	**12.76**	**18.27**	32.87	**60.35**
DAG-LASSO	20.03	42.85	67.68	129.41	334.85	19.63	42.49	64.84	127.45	268.56
DAG-OMP	7.38	14.35	20.44	35.86	77.72	7.08	13.35	18.65	31.78	60.93
DAG-OLS-V	7.89	14.82	21.21	36.25	**58.18**	7.49	14.02	19.39	33.99	63.09

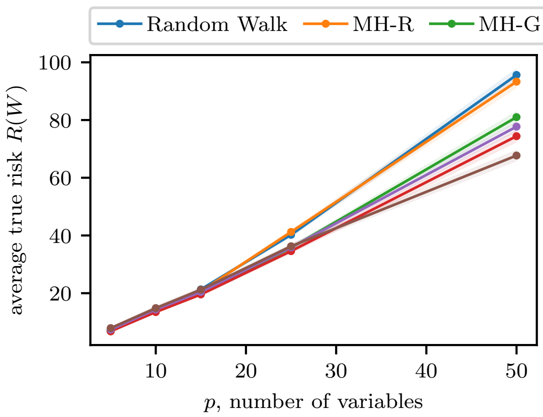

Figure 7.5: Plot of the average true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.3, excluding DAG-LASSO.

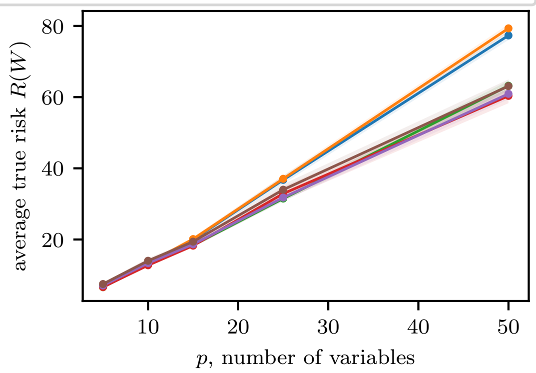

Figure 7.6: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.3, excluding DAG-LASSO.

Interestingly, the predictive performance of the six methods seem to be closer than in the acyclic setting, with DAG-LASSO being the only outcast who performs poor. However, we already knew that DAG-LASSO would perform poorly in the cyclic setting, as we had also encountered in Example 5.9.

Especially for $p \in \{5, 10, 15\}$, the remaining six methods seem very close, with the iterative methods just slightly poorer. However, we see that as the number of variables increases, the permutation-based approached fall slightly behind, as the number of permutations grows exponentially, and we can therefore only explore a miniscule portion of the search space. This is especially visile in Figure, where $T = 1000$. The non permutation-based approaches all seem to achieve a similar true risk, which is most likely close to optimal. However, the permutation-based approaches who are not able to efficiently travel the search space seem to benefit very little of this larger sample size.

What is also quite peculiar is that arguably the simplest method, the DAG-OLS-V algorithm, performs well in the $T = 100$ setting, outperforming a state of the art method such as NO TEARS when $p = 50$.

Table 7.4: Average structural hamming distance (SHD) as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W^* corresponds to a cyclic structure. A lower SHD indicates a better structural performance.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**3.5**	7.1	12.7	25.0	73.9	**3.3**	5.5	9.7	18.0	37.2
MH-Regular	**3.5**	6.7	11.5	26.7	78.4	**3.3**	5.0	9.9	17.7	40.1
MH-Greedy	3.7	5.8	9.6	15.7	52.7	3.8	5.6	6.1	**8.2**	15.6
NO TEARS	3.7	**5.7**	**8.2**	**13.0**	42.1	**3.3**	**4.9**	**5.8**	9.2	**11.3**
DAG-LASSO	14.3	28.2	41.8	69.7	140.3	14.2	27.9	42.0	69.4	134.9
DAG-OMP	4.6	8.1	10.4	16.2	49.8	4.3	6.3	7.0	10.2	13.8
DAG-OLS-V	4.5	6.8	10.0	14.7	**23.5**	4.6	6.5	7.4	11.8	17.4

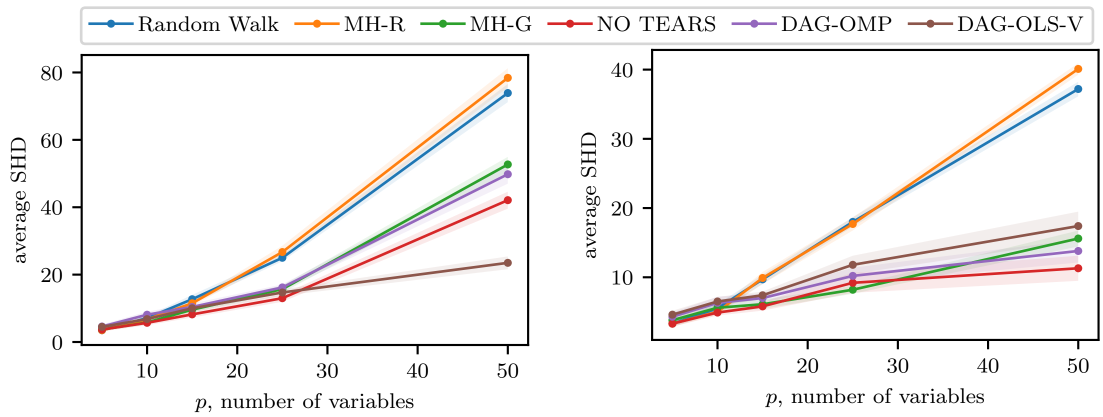

Figure 7.7: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.4, excluding DAG-LASSO.

Figure 7.8: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.4, excluding DAG-LASSO.

From a structural perspective, the random walk and the regular Metropolis-Hastings achieve quite a large structural hamming distance, indicating that their recovered coefficient matrix deviates quite a lot from the true coefficient matrix. For small sample sizes, the DAG-OLS-V approach interestingly seems to recover the structure of W^* best, as NO TEARS, DAG-OMP, and the greedy Metropolis-Hastings approach seem to struggle when $T = 100$ and $p = 50$.

For larger sample sizes, the structural performance increases for all methods, with NO TEARS being slightly better than DAG-OMP, the greedy Metropolis-Hastings approach, and DAG-OLS-V. The remaining two permutation-based methods recover the structure of W^* quite poorly, most likely again because they have not been able to sufficiently traverse the search space of permutation matrices.

7.2.3 Real Life Time Series Data.

In the previous two subsections, we have generated data according to a VAR(1) model, where all noise components were all independently and identically distributed with mean zero and an identity covariance matrix. This setting is quite optimistic, as it perfectly aligns with our model assumptions, apart from the acyclicity assumption in Subsection 7.2.2.

However, more often than not, real-life data does not perfectly align with the model assumptions. Nevertheless, we want to see how our developed methods perform on real-life data where some assumptions of the model are possibly violated. Hopefully, these violations are not too problematic and we can still obtain interesting results. Furthermore, these real-life datasets allow us to give meaning to the directed relationships. Rather than $X_1 \to X_2$, these variables have a physical meaning, such as how the sales of one product affect the sales of another product. Therefore, let us consider the Dominick's Finer Foods data.

Dominick's Finer Foods Data. Dominick's Finer Foods was a well-established supermarket chain in Chicago which was declared defunct in 2013 [38, 46]. Information regarding the prices, sales and promotions of all their available products was gathered from 1989 until 1999, as part of a partnership with Chicago Booth, the graduate business school of the University of Chicago. For each Dominick's Finer Foods store, the weekly number of units sold has been recorded. Many more attributes have been recorded, such as the demographics of customers, prices of the products, profit margin, deal codes, etc. Nevertheless, we will only focus on the sales of the products.

Over those seven years, more than 3,500 different Universal Product Codes (UPCs) have been tracked, corresponding to specific products sold by Dominick's Finer Foods. These UPCs have been categorized into 29 different categories, ranging from foods such as "cheese" and "crackers", beverages such as "beer" and "bottled juice", and general commodities such as "dish detergent" and "toothbrushes". We have decided to focus on the sixteen consumable categories, which correspond to: "beer", "bottled juice", "canned soup", "canned tuna", "cereal", "cheese", "cookies", "cracker", "front end candy", "frozen dinner", "frozen entree", "frozen juice", "oatmeal", "refrigerated juice", "snack cracker", and "soft drink".

We follow the same approach as [26] and [57] by only considering data from January 1993 until July 1994, yielding 77 weeks of sales numbers for sixteen categories in total. Next the log-differences of the sales have been considered rather than the actual sales to ensure stationarity of the time series. This results in sixteen time series of 76 time steps for each Dominick's Finer Foods store. Now, we are interested in seeing how the *sales* of one category influence the *sales* of another category. We conjecture that similar product categories, such as "bottled juice" and "refrigerated juice" will have some relation, as they are similar categories. We argue that when quite a lot of bottled juice is sold, that this might be at the expensive of refrigerated drinks. Similarly, if the sales of beer increases, this might be associated with an increase of snacks such as cheese. Furthermore, we expect no relation to exist between dissimilar categories, such as "canned tuna" and "beer". If customers purchase more beer, there is no reason to expect more canned tuna to be sold in the near future.

We have first used `DAG-OMP` from Section 6.1 to estimate a dense directed acyclic graph, after which we have used leave-one-out cross-validation to determine a suitable number of arcs. This resulted in a sparse coefficient matrix W containing nine off-diagonal arcs. The structure has been visualized in Figure 7.9. Note that we have only drawn variables that had at least one incoming or outgoing arc as to clutter the structure as little as possible.

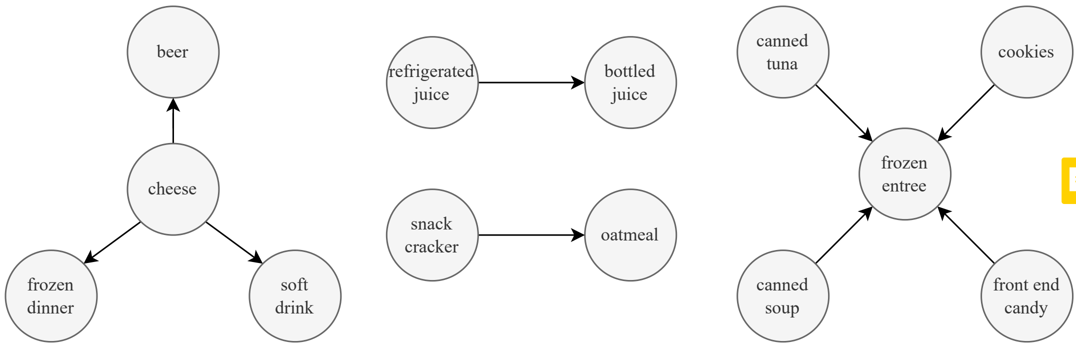

Figure 7.9: Acyclic structure corresponding to the Dominick's Finer Foods Dataset. The structure has been inferred using `DAG-OMP`, and the number of arcs was chosen using leave-one-out cross-validation. Variables with no incoming or outgoing arcs have been omitted.

We see some interesting relations between the product categories. First of all, we see that the sales of cheese seem to affect the future sales of beverages such as soft drinks and beer, as well as frozen dinner. This is in line with the conjecture that snacks such as cheese are frequently bought either together with or in quick succession of other unhealthy consumables such as frozen pizzas, beer, or soft drinks.

Furthermore, we also see a directed relationship from refrigerated juice to bottled juice, and from snack cracker to oatmeal. The former could be explained that when customer purchase more refrigerated juice, they see no need to purchase bottled juice in the near future. For the latter directed relationship, no reasonable explanation could be deduced.

Lastly, the future sales of frozen entrees is affected by canned tuna, canned soup, cookies, and front end candy. It seems that when customers purchase canned food or sweet snacks such as cookies and candy, then this will affect how much frozen entrees are purchased in the near future.

Note that these assumptions should be taken with a grain of salt, as there is no way to verify whether our findings are indeed correct. Nevertheless, it is interesting to see that recovering such an acyclic structure can provide useful insights into how the sales of consumables affect each other.

7.3 Time-Independent Experiments

So far, all methods and concepts introduced assumed the time-series setting of a VAR(1) model. However, recall that most structure learning methodologies assume *instantaneous* relations, for example through a linear structural equation model as defined in Definition 2.2.

Although all methods and examples have been centered around time dependent data, we will use this section to briefly investigate the performance of our discussed methods on time-independent data, as the structure learning research in time-independent experiments is more well-established.

We will first evaluate the methods based on simulated data in Subsection 7.3.1, after which we will also evaluate our methods on real-life biological data in Subsection 7.3.2.

Modifications to the methods. Luckily, our methods need to be adjusted only slightly. For the permutation-based and iterative approaches, we only need to align the index of the response and explanatory variable, rather than shifting one time index. Furthermore, we must fix the diagonal entries of W to zero, as we cannot use the value of a variable to predict itself.

For the continuous-based methods, NO TEARS now becomes the regular method as the authors have proposed in [71]. The DAG-LASSO algorithm, unfortunately, was not suitable for these models as it shrinks all coefficients until no cycle remains.

7.3.1 Simulated Time-Independent Data

Generating W^* and X. For the linear structural equation model, we will similarly generate an acyclic coefficient matrix W^* by setting s coefficients of W to non-zero, such that the structure W^* remains acyclic. Then, the values of these s coefficients will be sampled uniformly from the range $(-2.0, -0.5) \cup (0.5, 2.0)$.

Given this coefficient matrix W^*, we generate T independent p-dimensional vectors X according to a linear structural equation model (SEM)

$$X = XW^* + \varepsilon, \tag{7.9}$$

where ε is an independent Gaussian random variable with mean zero and as covariance matrix the identity matrix. Note that for we first need to sample the variables with no incoming arcs, and continue only sampling variables when all its parents have been sampled first, a so-called *ancestor-first* sampling.

Now, generating T of these independent and identically distributed variables X_t, $t = 1, \ldots, T$ yields a time-independent data matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$.

Experimental Setups. We will again simulated $N = 10$ data matrices $\mathbf{X} \in \mathbb{R}^{T \times p}$ for each tuple (p, T), where $p \in \{5, 10, 15, 25, 50\}$ and $T \in \{100, 1000\}$. Furthermore, the number of arcs in W^* will be equal to $3p$, thresholded to a complete directed acyclic graph if $3p > p(p-1)/2$ to ensure acyclicity. We will again use a threshold value of $\epsilon = 0.30$ to obtain a suitable number of arcs.

The results for the true risk $R(W)$ are given in Table 7.5. Furthermore, the results with the corresponding standard errors have been plotted as a function of p for $T = 100$ in Figure 7.10, and for $T = 1000$ in Figure 7.11. For the Structural Hamming Distance, the results are given in Table 7.6, and the corresponding plots are given in Figure 7.12 and Figure 7.13. For readers interested in the empirical risk, we refer to Table B.10, Figure B.19, and Figure B.20 in Appendix B.

Table 7.5: Average true risk $R(W)$ for the aforementioned methods for several values of p and T, where $s = 3p$ and the data has been generated according to a linear structural equation model. A lower true risk indicates a better predictive performance

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**5.13**	13.12	29.26	62.28	335.55	**5.01**	14.08	30.09	54.89	498.40
MH-Regular	**5.13**	12.68	25.57	66.41	290.89	**5.01**	12.13	25.31	58.90	462.91
MH-Greedy	**5.13**	**10.41**	**15.82**	**27.35**	**67.77**	**5.01**	**10.03**	**15.04**	**25.11**	**56.25**
NO TEARS	5.20	10.82	16.36	30.27	152.6	5.06	10.44	22.55	28.38	153.32
DAG-OMP	5.98	15.34	19.71	37.92	77.2	5.95	15.35	19.08	33.74	74.26
DAG-OLS-V	7.04	26.88	63.67	182.5	2154.0	7.11	34.43	98.66	194.27	2468.56

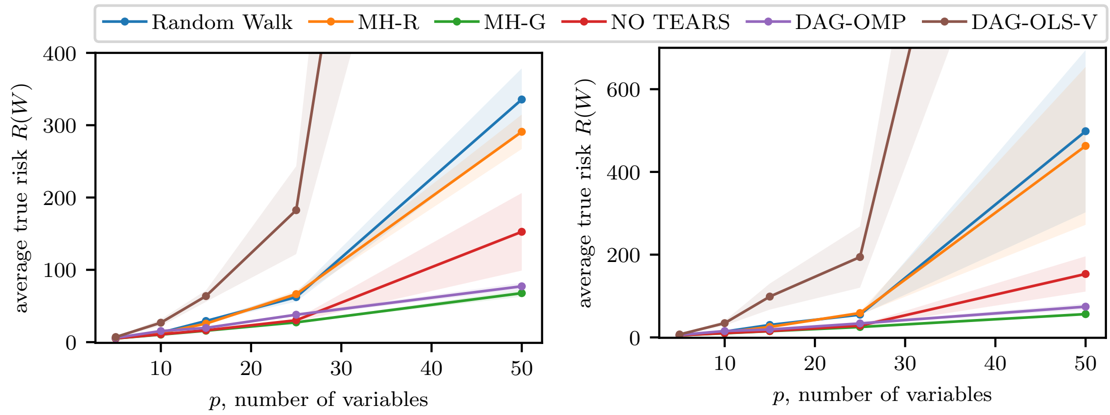

Figure 7.10: Plot of the average true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.5.

Figure 7.11: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.5.

Comparing the several methods, we interestingly see that DAG-OLS-V algorithm achieves the poorest true risk of all six methods. The results were so poor for $p = 50$ that the plots needed to be adjusted. Apparently, its ordinary least squares estimate is not a suitable starting point for linear SEMs.

After this, we see that the two explorative permutation-based approaches, the random walk and the regular Metropolis-Hastings approach, achieve quite a good performance for $p \in \{5, 10, 15\}$, but this performance drops as p grows larger. Again, the number of permutations scales so fast that such an explorative approach does not seem tractable.

Interestingly, the exploitative permutation-based approach seems to be achieving the smallest true risk of all methods, even significantly smaller than the state of the art method NO TEARS. Furthermore, the DAG-OMP approach also seems to be a more suitable approach than NO TEARS, especially when p is large.

Table 7.6: Average structural hamming distance for the aforementioned methods for several values of p and T, where $s = 3p$ and the data has been generated according to a linear structural equation model. A lower SHD indicates a better structural perfomance.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	**0.2**	13.9	43.4	112.0	375.8	**0.0**	12.5	37.9	87.4	261.7
MH-Regular	**0.2**	12.5	40.7	104.9	372.6	**0.0**	10.7	42.3	95.8	260.3
MH-Greedy	**0.2**	**3.1**	**10.4**	**45.7**	270.1	**0.0**	**0.0**	**0.0**	**1.0**	**39.1**
NO TEARS	1.5	6.6	15.3	49.3	220.1	1.2	3.6	6.6	9.9	54.6
DAG-OMP	2.9	15.3	25.8	78.3	272.4	2.6	15.4	25.2	39.0	93.8
DAG-OLS-V	1.9	13.5	30.5	54.7	**138.7**	1.4	15.8	29.5	50.5	125.5

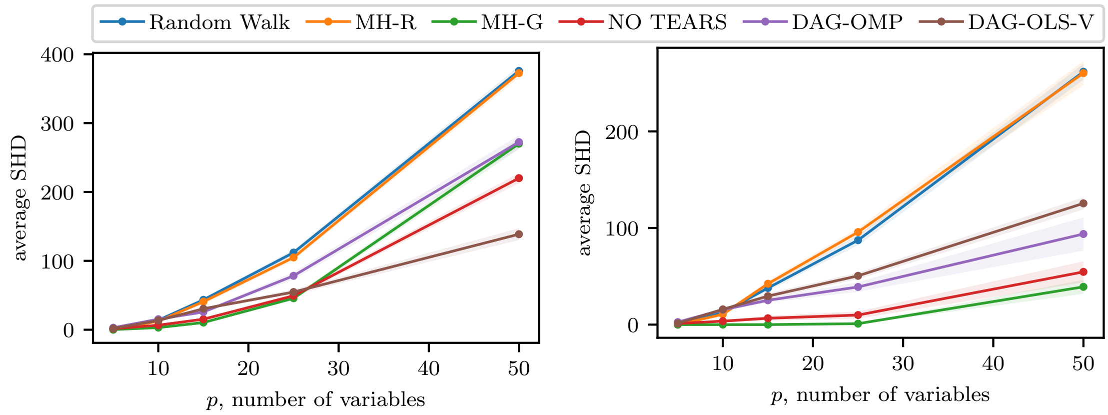

Figure 7.12: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table 7.6.

Figure 7.13: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table 7.6.

Interestingly, we see that the DAG-OLS-V algorithm achieves the lowest SHD, which would imply that its structure corresponds closely to W^*. However, from a predictive point of view, DAG-OLS-V performs the poorest. An explanation for this is that the DAG-OLS-V estimates a coefficient matrix W that is much too sparse. In fact, one can achieve an average SHD of $s = 3p$ by simply returning the zero-matrix. However, this would result in a poor predictive performance, just as is the case for DAG-OLS-V. Therefore, we need to compare the methods on both predictive and structural performance criteria to ensure that the method is adequate.

When we look at the other methods, we see that the random walk and the regular Metropolis-Hastings algorithm both perform quite poorly. Interestingly, the greedy Metropolis-Hastings approach outperforms the state of the art NO TEARS method also with respect to the structural hamming distance, which is rather surprising. The DAG-OMP algorithm seems to be slightly worse from a structural perspective, although it performed quite well from a predictive perspective.

7.3.2 Real-Life Time-Independent Data

Sachs In this section, we will consider the dataset introduced by Sachs et al. in [53]. It contains a total of 7,466 measurements of 11 different types of phosphorylated proteins and phospholipids. Sachs. et al used, among others, this dataset to learn the causal pathways between these different proteins and phospholipids.

This dataset is widely used as a benchmark in the structure learning community because that these pathway linkings are already known from existing literature. Figure 7.14 depicts a network that is widely accepted by biologists as a ground truth. The eleven variables that Sachs. et al have considered are colored in orange. Having such a ground truth widely accepted by biologists, researchers can benchmark their methods against real-life data. The model reported by Sachs et al is shown in Figure 7.15.

Some authors use the model reported by Sachs as the ground truth [71, 68], whereas others authors use the original biologists' view depicted in Figure as the ground truth [51, 20]. We will be using the same ground truth as NO TEARS has used, as that is the state of the art method that we will compare our methods to, which corresponds to Figure 7.15. Furthermore, several versions of the dataset exists, such as discretised versions or where outliers have been removed. For clarity, we have selected the original dataset, which can be retrieved here.

We have applied our five methods, as well as the state of the art NO TEARS method on this biological dataset. We have used the model aligns with the biologists view in Figure 7.14 as the ground truth. The results are shown in Table 7.7.

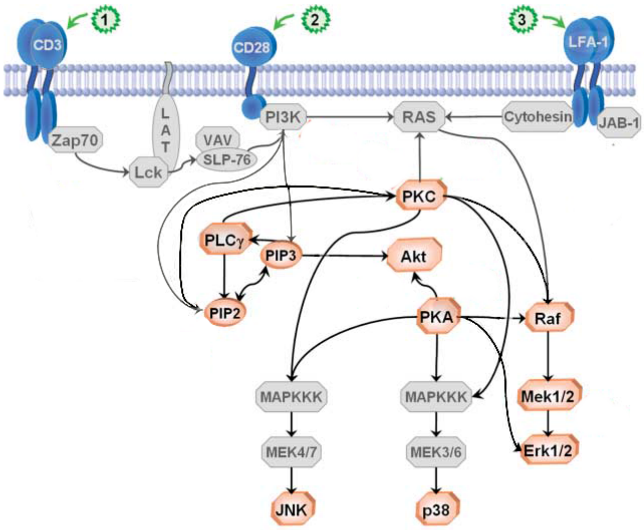

Figure 7.14: Biological overview of the causal pathways widely accepted by biologists. Variables that were included in the dataset are colored in orange. Retrieved from [53].

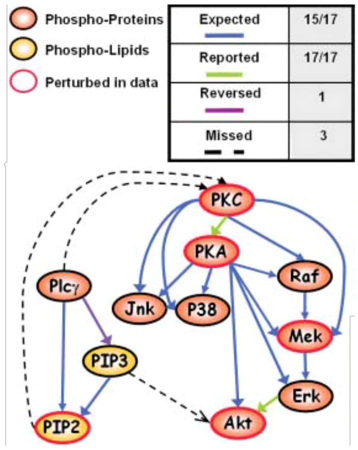

Figure 7.15: The network obtained by Sachs et al. in [53]. Fourteen expected pathways were correctly recovered, one pathway was recovered in the reversed direction, and three pathways were missed. Furthermore, two pathways were reported by Sachs et al., but were not among the widely accepted pathways.

Table 7.7: Results of applying five of our methods as well as the NO TEARS approach on the time-independent protein dataset of Sachs et al. [53]. We have reported the total number of predicted edges, as well as the true positives (TP) out of 20, the structural hamming distance, and the empirical risk. We consider the graph in Figure 7.15 to be the ground truth.

Method	Predicted Edges	TP (out of 20)	SHD	$R_{\text{emp}}(W)$
Random Walk	13	6	21	$5.037 \cdot 10^5$
MH-Regular	15	7	21	$5.051 \cdot 10^5$
MH-Greedy	17	**8**	21	$\mathbf{4.998 \cdot 10^5}$
NO TEARS	16	**8**	22	$5.03 \cdot 10^5$
DAG-OMP	17	**8**	21	$5.000 \cdot 10^5$
DAG-OLS-V	14	7	**20**	$5.156 \cdot 10^5$

From Table 7.7, we conclude that all methods seem to achieve similar performance. All methods correctly recover either six, seven, or eight of the causal pathways. Furthermore, the corresponding structural hamming distance is either 20 or 21 for all methods, indicating that their structural performance is similar. Furthermore, all methods achieve a similar empirical risk, with the greedy Metropolis-Hastings approach attaining the lowest empirical risk at $4.999 \cdot 10^5$, and DAG-OLS-V attaining the highest empirical risk at $5.156 \cdot 10^5$.

As the number of variables p is quite low, it is not unexpected that the permutation-based approaches achieve a similar performance to the other methods. Even a naive random walk that which has tried 1000 permutations attains a similar performance as a state of the art method such as NO TEARS. Furthermore, as the number of samples is quite large, the iterative approaches also achieve a similar performance to NO TEARS.

Bibliography

[1] Scipy api reference for `optimize.minimize`. 22

[2] Scipy api reference for the `L-BFGS-B` optimization method. 22

[3] *Vector Autoregressive Models for Multivariate Time Series*, pages 385–429. Springer New York, New York, NY, 2006. 54

[4] Awad H. Al-Mohy and Nicholas J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2010. 74, 93

[5] M. Andrle, L. Rebollo-Neira, and E. Sagianos. Backward-optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 11(9):705–708, 2004. 91

[6] Mark Bartlett and James Cussens. Integer linear programming for the bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017. Combining Constraint Solving with Mining and Learning. 20

[7] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics Data Analysis*, 120:70–83, 2018. 104

[8] Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946. 51

[9] Thomas Blumensath and Mike Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. 03 2007. 76

[10] Graham Brightwell and Peter Winkler. Counting linear extensions is #p-complete. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, page 175–181, New York, NY, USA, 1991. Association for Computing Machinery. 32

[11] Prabir Burman, Edmond Chow, and Deborah Nolan. A cross-validatory method for dependent data. *Biometrika*, 81:351–358, 1994. 107

[12] Nancy Cartwright. Are rcts the gold standard? *BioSocieties*, 2(1):11–20, 2007. 4

[13] Rui Castro and Robert Nowak. Likelihood based hierarchical clustering and network topology identification. In Anand Rangarajan, Mário Figueiredo, and Josiane Zerubia, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 113–129, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 39

[14] S. CHEN, S. A. BILLINGS, and W. LUO. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989. 76

[15] David Maxwell Chickering. *Learning Bayesian Networks is NP-Complete*, pages 121–130. Springer New York, New York, NY, 1996. 15

[16] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990. 4

[17] James Cussens. Bayesian network learning with cutting planes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 153–160, Arlington, Virginia, USA, 2011. AUAI Press. 20

[18] Aramayis Dallakyan and Mohsen Pourahmadi. Learning bayesian networks through birkhoff polytope: A relaxation method. *CoRR*, abs/2107.01658, 2021. 63

[19] Ivan Damnjanovic, Matthew E. P. Davies, and Mark D. Plumbley. Smallbox - an evaluation framework for sparse representations and dictionary learning algorithms. In Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent, editors, *Latent Variable Analysis and Signal Separation*, pages 418–425, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 86

[20] Tristan Deleu, António Góis, Chris Chinenye Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 127

[21] Vera Djordjilović, Monica Chiogna, and Jiří Vomlel. An empirical comparison of popular structure learning algorithms with a view to gene network inference. *International Journal of Approximate Reasoning*, 88:602–613, 2017. 116

[22] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979. 97

[23] Kim Esbensen and Paul Geladi. Principles of proper validation: use and abuse of re-sampling for validation. *Journal of Chemometrics*, 24:168 – 187, 03 2010. 104

[24] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman amp; Co., USA, 1990. 76

[25] Maxime Gasse, Alex Aussem, and Haytham Elghazel. An experimental comparison of hybrid algorithms for bayesian network structure learning. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 58–73, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 116

[26] Sarah Gelper, Ines Wilms, and Christophe Croux. Identifying demand effects in a large network of product categories. *Journal of Retailing*, 92(1):25–39, 2016. 123

[27] M. Gharavi-Alkhansari and T.S. Huang. A fast orthogonal matching pursuit algorithm. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 3, pages 1389–1392 vol.3, 1998. 76

[28] Hemant S. Goklani, Jignesh N. Sarvaiya, and A. M. Fahad. Image reconstruction using orthogonal matching pursuit (omp) algorithm. In *2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking*, pages 1–5, 2014. 77

[29] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969. 5

[30] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001. 72

[31] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 39

[32] Sander Hofman. Making euv: From lab to fab, Mar 2022. 3

[33] Guoxian Huang and Lei Wang. High-speed signal reconstruction with orthogonal matching pursuit via matrix inversion bypass. pages 191–196, 10 2012. 86

[34] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 2nd edition, 2018. 26

[35] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975. 92

[36] Hidde De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *JOURNAL OF COMPUTATIONAL BIOLOGY*, 9:67–103, 2002. 3

[37] A. B. Kahn. Topological sorting of large networks. *Commun. ACM*, 5(11):558–562, nov 1962. 75

[38] Wagner A. Kamakura and Wooseong Kang. Chain-wide and store-level analysis for cross-category management. *Journal of Retailing*, 83(2):159–170, 2007. 123

[39] Mahdi Khosravy, Nilanjan Dey, and Carlos Duque. *Compressive Sensing in Health Care*. 10 2019. 76

[40] S. N. Lahiri. *Bootstrap Methods*, pages 17–43. Springer New York, New York, NY, 2003. 97

[41] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988. 4

[42] Hanxi Li, Yongsheng Gao, and Jun Sun. Fast kernel sparse representation. In *2011 International Conference on Digital Image Computing: Techniques and Applications*, pages 72–77, 2011. 86

[43] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 76

[44] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. , 21(6):1087–1092, June 1953. 39

[45] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1595–1605. PMLR, 26–28 Aug 2020. 23

[46] Koen Pauwels. How retailer and competitor decisions drive the long-term effectiveness of manufacturer promotions for fast moving consumer goods. *Journal of Retailing*, 83(3):297–308, 2007. 123

[47] Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986. 4

[48] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. v, 1, 2, 4

[49] Judea Pearl and Thomas Verma. A theory of inferred causation. In *KR*, 1991. 16

[50] K. B. Petersen and M. S. Pedersen. The matrix cookbook, October 2008. Version 20081110. 57

[51] Joseph Ramsey and Bryan Andrews. Fask with interventional knowledge recovers edges from the sachs model. *ArXiv*, abs/1805.03108, 2018. 127

[52] R. W. Robinson. Counting unlabeled acyclic digraphs. In Charles H. C. Little, editor, *Combinatorial Mathematics V*, pages 28–43, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg. 31

[53] Karen Sachs, Omar Perez, Dana Pe'er, Douglas Lauffenburger, and Garry Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science (New York, N.Y.)*, 308:523–9, 05 2005. vii, vii, ix, 127, 128

[54] Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. 2018. 15

[55] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyv228;rinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006. 18, 87

[56] Konstantinos Skianis, Nikolaos Tziortziotis, and Michalis Vazirgiannis. Orthogonal matching pursuit for text classification. *ArXiv*, abs/1807.04715, 2018. 77

[57] Shuba Srinivasan, Koen Pauwels, Dominique M. Hanssens, and Marnik G. Dekimpe. Do promotions benefit manufacturers, retailers, or both? *Management Science*, 50(5):617 – 629, 2004. Cited by: 177; All Open Access, Green Open Access. 123

[58] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974. 104

[59] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 67

[60] Ryan Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39, 05 2010. 69

[61] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 86

[62] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 10 2006. 116

[63] Alexander L. Tulupyev and Sergey I. Nikolenko. Directed cycles in bayesian belief networks: Probabilistic semantics and consistency checking complexity. In Alexander Gelbukh, Álvaro de Albornoz, and Hugo Terashima-Marín, editors, *MICAI 2005: Advances in Artificial Intelligence*, pages 214–223, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. 6

[64] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. 22

[65] John Von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, 2(0):5–12, 1953. 51

[66] Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Comput. Surv.*, mar 2022. Just Accepted. 15, 17, 116

[67] Norbert Wiener. The theory of prediction. *Modern mathematics for engineers*, 1956. 5

[68] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. 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 7154–7163. PMLR, 09–15 Jun 2019. 127

[69] Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10(19):555–568, 2009. 76

[70] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018. 22

[71] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc. 50, 63, 125, 127

[72] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997. 22

[73] Hufei Zhu, Wen Chen, and Yanpeng Wu. Efficient implementations for orthogonal matching pursuit. *Electronics*, 9:1507, 09 2020. 86

Appendix B

Additional tables

In Chapter 7, the methods discussed in this thesis have been compared using several performance criteria. Furthermore, we have experimented with several settings. The following settings were considered:

- Sparse acyclic VAR(1) models, where the number of off-diagonal arcs was $3p$. Furthermore, we have generated data matrices consisting of few time steps $T = 100$ and many time steps $T = 1000$. The three corresponding tables and figures of the empirical risk, true risk, and structural hamming distance are given in Section B.1.
- Dense acyclic VAR(1) models, where the number of off-diagonal arcs was $5p$. Furthermore, we have generated data matrices consisting of few time steps $T = 100$ and many time steps $T = 1000$. The three corresponding tables and figures of the empirical risk, true risk, and structural hamming distance are given in Section B.2.
- Sparse cyclic VAR(1) models, where the number of off-diagonal arcs was $2p$. Furthermore, we have generated data matrices consisting of few time steps $T = 100$ and many time steps $T = 1000$. The three corresponding tables and figures of the empirical risk, true risk, and structural hamming distance are given in Section B.3.
- Sparse linear structural equation models, where the number of off-diagonal arcs was $2p$. Furthermore, we have generated data matrices consisting of few time steps $T = 100$ and many time steps $T = 1000$. The three corresponding tables and figures of the empirical risk, true risk, and structural hamming distance are given in Section B.4.

B.1 Sparse acyclic VAR(1) models

Table B.1: Average empirical risk $R_{\text{emp}}(W)$ for the aforementioned methods for several values of p and T, where $s = 3p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p=5$	$p=10$	$p=15$	$p=25$	$p=50$	$p=5$	$p=10$	$p=15$	$p=25$	$p=50$
Random Walk	4.81	10.67	17.22	29.62	57.53	4.95	11.57	18.55	34.69	76.29
MH-Regular	4.81	10.74	16.95	30.06	58.18	4.95	11.55	19.42	35.11	79.12
MH-Greedy	4.91	9.64	14.00	21.47	38.03	4.95	10.25	15.15	25.01	51.48
NO TEARS	4.81	9.18	13.46	20.89	34.79	4.95	9.87	14.79	24.55	48.57
DAG-LASSO	10.75	46.94	62.76	123.03	305.72	9.19	38.92	62.57	112.58	270.07
DAG-OMP	4.81	9.76	13.90	21.09	35.85	4.95	10.18	15.39	24.71	48.59
DAG-OLS-V	4.81	9.31	14.30	24.08	110.21	4.95	9.88	14.84	24.75	49.58

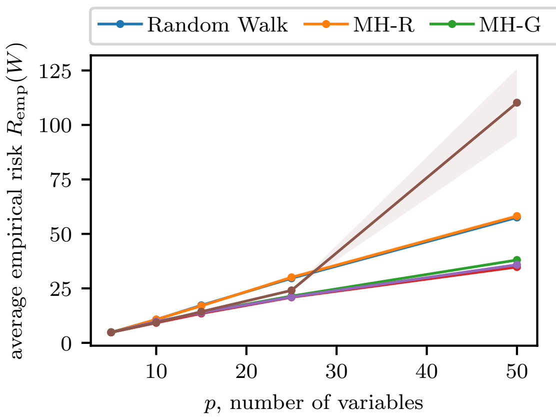

Figure B.1: Plot of the average empirical risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.1, excluding DAG-LASSO.

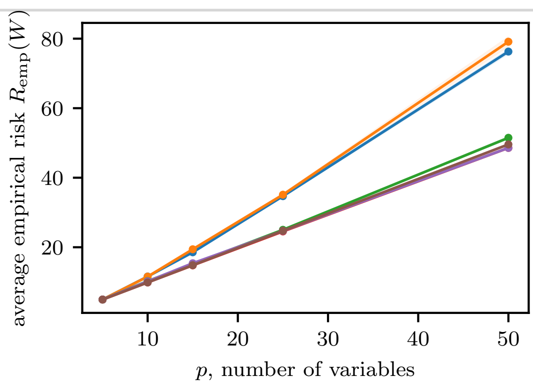

Figure B.2: Plot of the average empirical risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.1, excluding DAG-LASSO.

Table B.2: Average true risk $R(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	5.26	12.70	20.85	41.30	109.70	5.02	11.92	19.41	37.87	89.57
MH-Regular	5.26	12.47	20.85	41.28	109.54	5.02	11.87	20.35	37.91	95.08
MH-Greedy	5.38	11.41	17.02	28.95	68.01	5.02	10.46	15.47	25.67	54.62
NO TEARS	5.26	10.59	16.08	27.08	61.03	5.02	10.04	15.07	25.1	50.19
DAG-LASSO	12.32	51.51	68.16	148.08	353.65	10.72	46.02	69.44	136.20	290.74
DAG-OMP	5.26	11.45	17.51	27.88	63.64	5.02	10.52	15.85	25.34	50.62
DAG-OLS-V	5.28	10.55	16.15	27.61	128.07	5.02	10.04	15.06	25.1	50.19

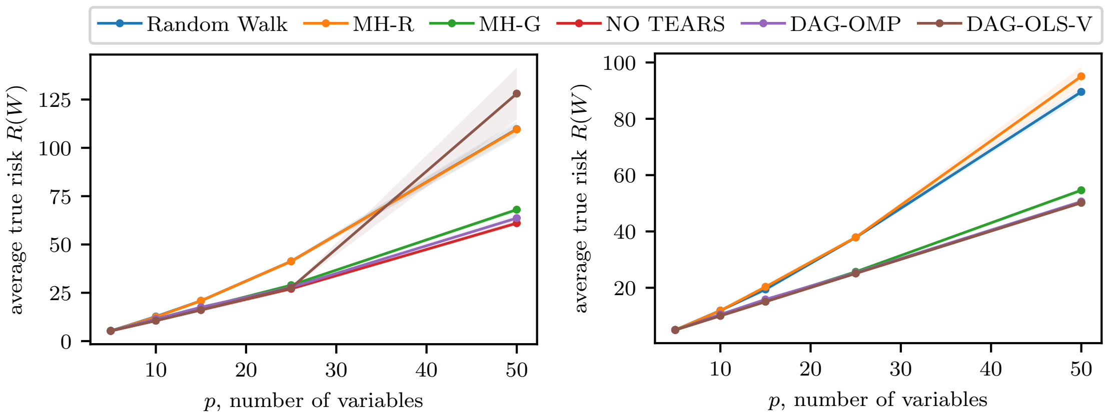

Figure B.3: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table B.2, excluding DAG-LASSO.

Figure B.4: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.2, excluding DAG-LASSO.

Table B.3: Average structural hamming distance (SHD) as a function of p for $T = 100$ and $T = 1000$, where $s = 3p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	0.3	5.4	11.2	29.5	97.5	0.0	4.5	9.1	22.6	57.7
MH-Regular	0.3	5.2	11.2	30.6	99.6	0.0	4.3	10.8	20.8	58.4
MH-Greedy	0.7	2.3	2.4	6.1	33.2	0.0	1.1	0.8	1.2	6.0
NO TEARS	0.3	0.4	0.7	1.9	20.0	0.0	0.0	0.0	0.0	0.0
DAG-LASSO	12.1	37.2	54.5	92.8	188.1	9.1	36.1	55.0	91.9	184.6
DAG-OMP	0.3	1.6	1.4	3.6	22.3	0.0	0 .6	0.9	0.4	0.3
DAG-OLS-V	0.4	0.3	0.8	2.9	58.1	0.0	0.0	0.0	0.0	0.0

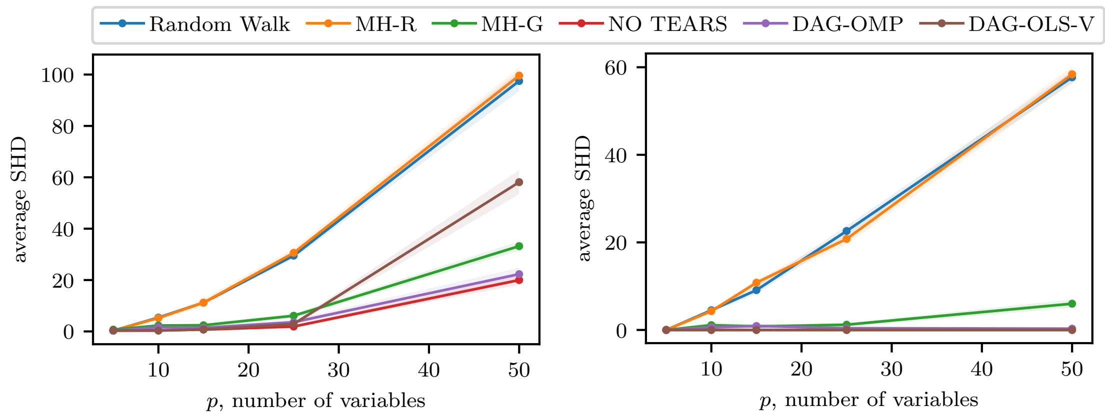

Figure B.5: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table B.3, excluding DAG-LASSO.

Figure B.6: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.3, excluding DAG-LASSO.

B.2 Dense acyclic VAR(1) models

Table B.4: Average empirical risk $R_{\text{emp}}(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 5p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	4.81	12.13	23.05	45.40	100.85	4.95	13.02	26.04	52.87	132.65
MH-Regular	4.81	12.49	23.47	45.49	103.04	4.95	13.65	26.50	53.92	145.3
MH-Greedy	4.81	9.25	14.06	23.36	49.01	4.95	9.87	15.32	27.83	64.72
NO TEARS	4.81	9.25	13.60	21.14	36.04	4.95	9.87	14.83	24.62	50.37
DAG-LASSO	10.75	65.91	332.95	565.14	3004.02	9.19	65.92	255.99	435.44	2661.91
DAG-OMP	4.81	9.63	17.41	23.75	45.55	4.95	9.94	19.37	28.36	65.94
DAG-OLS-V	4.81	9.25	13.89	24.31	212.65	4.95	9.87	14.83	24.71	49.5

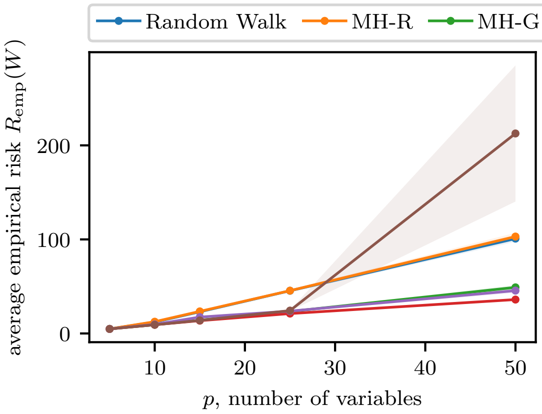

Figure B.7: Plot of the average empirical risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.4, excluding DAG-LASSO.

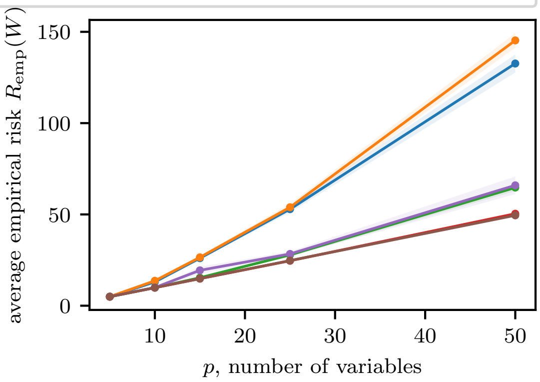

Figure B.8: Plot of the average empirical risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.4, excluding DAG-LASSO.

Table B.5: Average true risk $R(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 5p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	5.26	14.65	31.45	73.38	270.40	5.02	14.57	29.22	64.67	212.57
MH-Regular	5.26	15.30	30.46	76.33	278.67	5.02	14.58	28.75	68.29	218.29
MH-Greedy	5.26	10.74	17.13	31.30	117.50	5.02	10.06	15.81	29.46	78.15
NO TEARS	5.26	10.74	16.32	27.83	65.44	5.02	10.06	15.09	25.16	50.32
DAG-LASSO	12.32	69.26	481.20	643.96	5183.68	10.72	76.76	459.05	515.09	4645.04
DAG-OMP	5.26	11.39	23.82	39.79	158.59	5.02	10.13	20.82	31.63	100.15
DAG-OLS-V	5.28	10.74	16.32	28.8	301.54	5.02	10.06	15.09	25.16	50.32

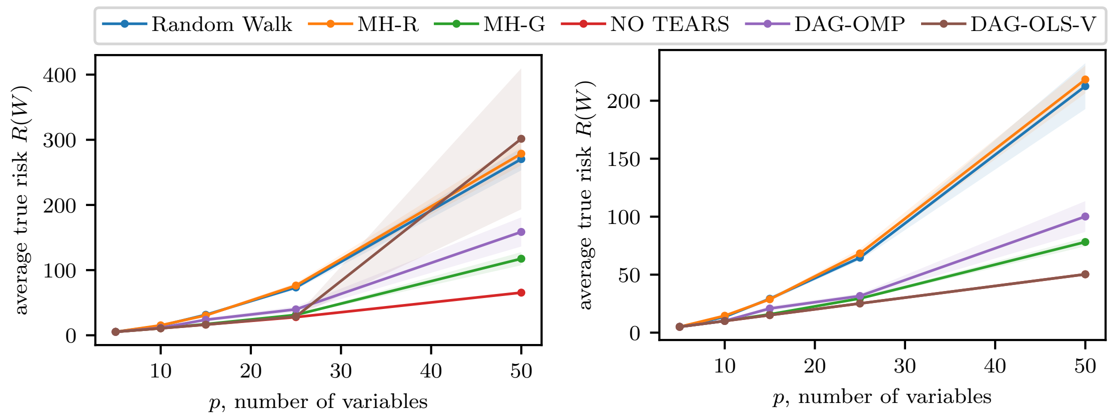

Figure B.9: Plot of the average true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.5, excluding DAG-LASSO.

Figure B.10: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.5, excluding DAG-LASSO.

Table B.6: Average structural hamming distance (SHD) as a function of p for $T = 100$ and $T = 1000$, where $s = 5p$ and W corresponds to an acyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	0.3	9.2	24.5	61.4	177.4	0.0	8.5	24.6	53.4	118.5
MH-Regular	0.3	10.9	22.9	61.6	183.7	0.0	9.7	22.7	50.8	135.0
MH-Greedy	0.3	0.5	2.2	9.0	67.9	0.0	0.0	1.5	7.3	29.3
NO TEARS	0.3	0.5	0.6	2.0	15.5	0.0	0.0	0.0	0.0	0.0
DAG-LASSO	12.1	50.9	86.8	145.6	294.3	9.1	51.0	85.4	143.5	291.7
DAG-OMP	0.3	1.4	6.4	7.9	62.8	0.0	0.1	4.6	5.5	18.7
DAG-OLS-V	0.4	0.5	0.6	3.1	74.9	0.0	0.0	0.0	0.0	0.0

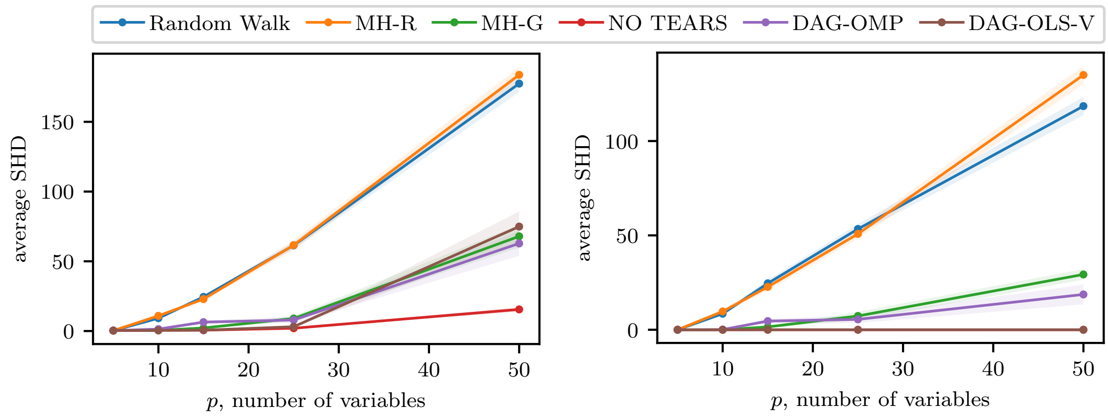

Figure B.11: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table B.6, excluding DAG-LASSO.

Figure B.12: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.6, excluding DAG-LASSO.

B.3 Sparse cyclic VAR(1) models

Table B.7: Average empirical risk $R_{\text{emp}}(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 2p$ and W corresponds to a cyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	6.48	11.90	17.42	28.92	51.43	6.49	12.54	19.02	33.04	66.88
MH-Regular	6.48	11.77	17.30	29.30	52.08	6.49	12.66	19.15	33.28	67.82
MH-Greedy	6.68	11.47	16.30	25.55	42.77	6.64	12.48	17.72	29.36	57.42
NO TEARS	6.58	11.28	15.99	24.53	40.05	6.51	12.26	17.6	29.52	54.58
DAG-LASSO	20.56	42.37	54.78	103.97	288.88	16.77	35.97	54.48	107.46	222.84
DAG-OMP	7.07	12.03	16.29	25.78	42.25	6.93	13.0	17.87	29.87	55.62
DAG-OLS-V	7.57	13.28	18.96	31.22	58.18	7.24	13.68	18.93	32.96	60.77

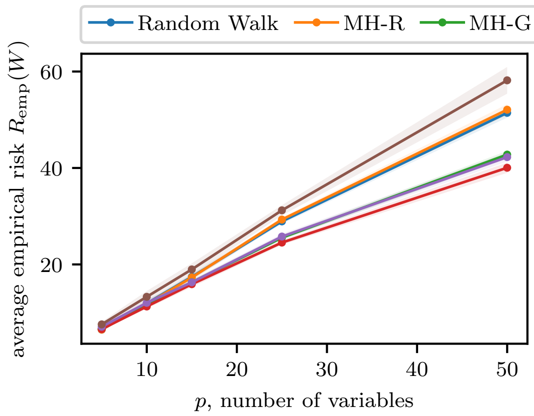

Figure B.13: Plot of the average empirical risk as a function of p, the number of variables, where $T = 100$ for the methods in Table **??**, excluding DAG-LASSO.

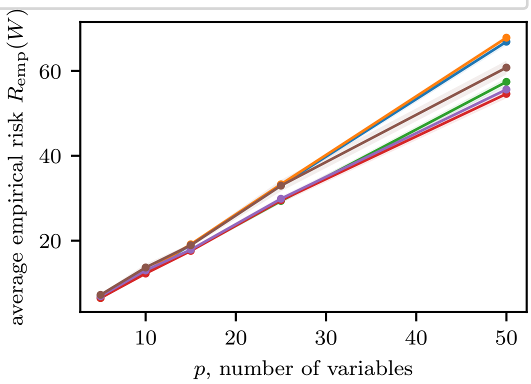

Figure B.14: Plot of the average empirical risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table **??**, excluding DAG-LASSO.

Table B.8: Average true risk $R(W)$ as a function of p for $T = 100$ and $T = 1000$, where $s = 2p$ and W corresponds to a cyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	6.79	13.88	21.32	40.23	95.62	6.61	13.10	19.97	36.75	77.33
MH-Regular	6.79	13.80	20.64	41.22	93.31	6.61	13.29	20.12	37.06	79.32
MH-Greedy	7.01	13.85	19.94	35.72	81.01	6.75	13.02	18.39	31.5	63.2
NO TEARS	6.83	13.47	19.57	34.58	74.41	6.66	12.76	18.27	32.87	60.35
DAG-LASSO	20.03	42.85	67.68	129.41	334.85	19.63	42.49	64.84	127.45	268.56
DAG-OMP	7.38	14.35	20.44	35.86	77.72	7.08	13.35	18.65	31.78	60.93
DAG-OLS-V	7.89	14.82	21.21	36.25	58.18	7.49	14.02	19.39	33.99	63.09

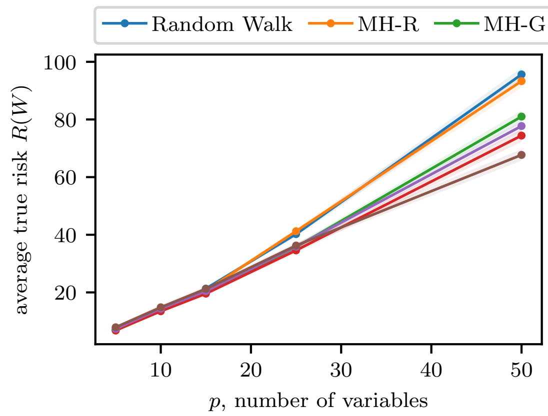

Figure B.15: Plot of the average true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.8, excluding DAG-LASSO.

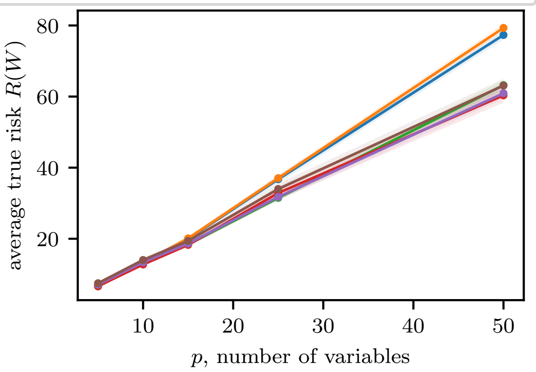

Figure B.16: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.8, excluding DAG-LASSO.

Table B.9: Average structural hamming distance (SHD) as a function of p for $T = 100$ and $T = 1000$, where $s = 2p$ and W corresponds to a cyclic structure.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	3.5	7.1	12.7	25.0	73.9	3.3	5.5	9.7	18.0	37.2
MH-Regular	3.5	6.7	11.5	26.7	78.4	3.3	5.0	9.9	17.7	40.1
MH-Greedy	3.7	5.8	9.6	15.7	52.7	3.8	5.6	6.1	8.2	15.6
NO TEARS	3.7	5.7	8.2	13.0	42.1	3.3	4.9	5.8	9.2	11.3
DAG-LASSO	14.3	28.2	41.8	69.7	140.3	14.2	27.9	42.0	69.4	134.9
DAG-OMP	4.6	8.1	10.4	16.2	49.8	4.3	6.3	7.0	10.2	13.8
DAG-OLS-V	4.5	6.8	10.0	14.7	23.5	4.6	6.5	7.4	11.8	17.4

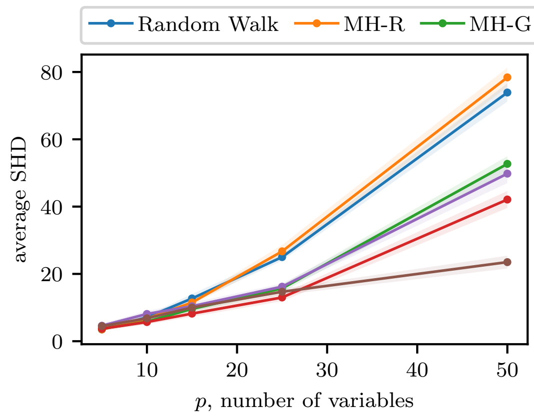

Figure B.17: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table B.9, excluding DAG-LASSO.

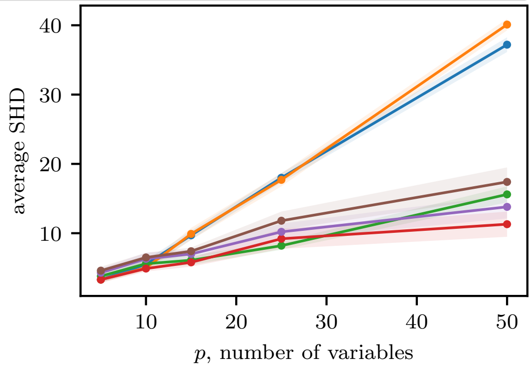

Figure B.18: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.9, excluding DAG-LASSO.

B.4 Linear structural equation models.

Table B.10: Average empirical risk $R_{\mathrm{emp}}(W)$ for the aforementioned methods for several values of p and T, where $s = 3p$ and the data has been generated according to a linear structural equation model.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	4.97	11.60	24.44	44.08	143.06	5.02	12.30	24.20	46.22	172.68
MH-Regular	4.97	11.45	21.28	45.60	147.32	5.02	11.73	23.18	48.07	160.71
MH-Greedy	4.97	9.59	14.07	22.35	39.35	5.02	9.93	14.91	24.70	53.35
NO TEARS	5.03	17.09	395.64	136.79	2667.6	5.07	11.86	326.30	32.91	1209.85
DAG-OMP	5.65	13.39	16.62	27.65	44.74	5.87	14.12	18.24	30.16	62.24
DAG-OLS-V	6.87	23.62	58.98	175.61	1985.58	7.10	34.19	96.83	191.95	2384.91

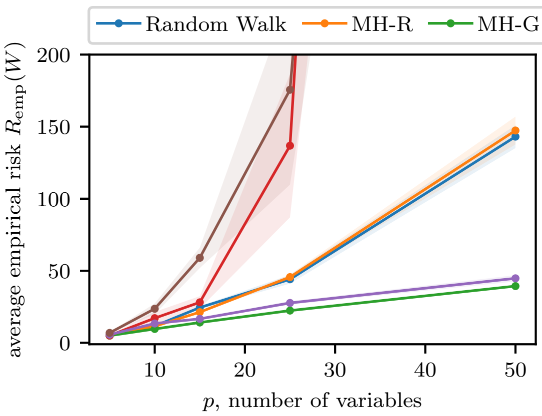

Figure B.19: Plot of the average empirical risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.10, excluding DAG-LASSO.

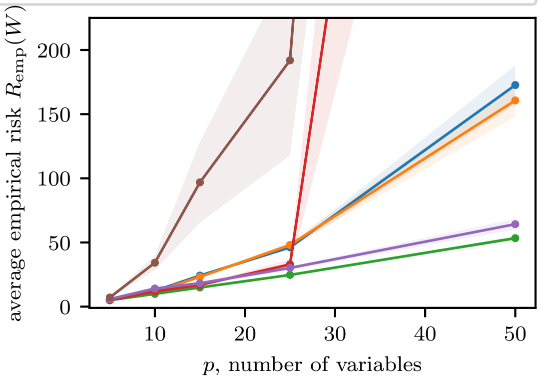

Figure B.20: Plot of the average empirical risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.10, excluding DAG-LASSO.

Table B.11: Average true risk $R(W)$ for the aforementioned methods for several values of p and T, where $s = 3p$ and the data has been generated according to a linear structural equation model.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	5.13	13.12	29.26	62.28	335.55	5.01	14.08	30.09	54.89	498.40
MH-Regular	5.13	12.68	25.57	66.41	290.89	5.01	12.13	25.31	58.9	462.91
MH-Greedy	5.13	10.41	15.82	27.35	67.77	5.01	10.03	15.04	25.11	56.25
NO TEARS	5.2	10.82	16.36	30.27	152.6	5.06	10.44	22.55	28.38	153.32
DAG-OMP	5.98	15.34	19.71	37.92	77.2	5.95	15.35	19.08	33.74	74.26
DAG-OLS-V	7.04	26.88	63.67	182.5	2154.0	7.11	34.43	98.66	194.27	2468.56

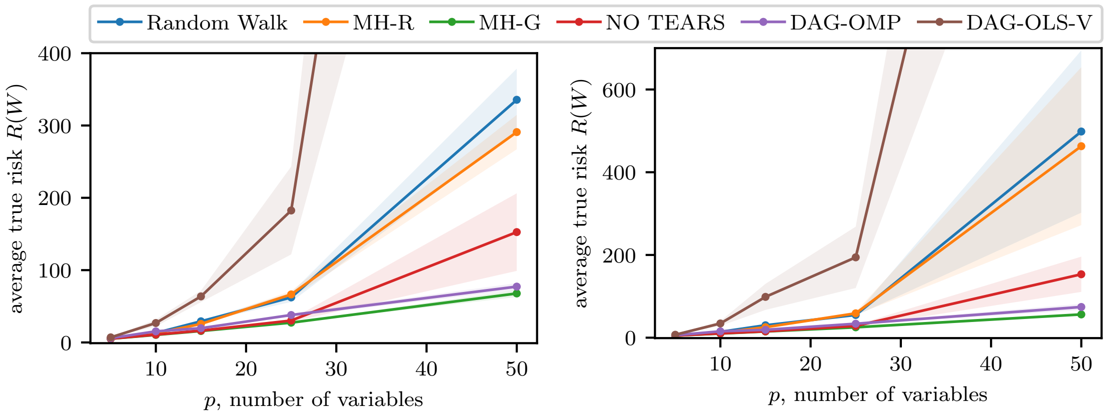

Figure B.21: Plot of the true risk as a function of p, the number of variables, where $T = 100$ for the methods in Table B.11, excluding DAG-LASSO.

Figure B.22: Plot of the average true risk as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.11, excluding DAG-LASSO.

Table B.12: Average structural hamming distance (SHD) for the aforementioned methods for several values of p and T, where $s = 3p$ and the data has been generated according to a linear structural equation model.

	$T = 100$					$T = 1000$				
Method	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$	$p = 5$	$p = 10$	$p = 15$	$p = 25$	$p = 50$
Random Walk	0.2	13.9	43.4	112.0	375.8	0.0	12.5	37.9	87.4	261.7
MH-Regular	0.2	12.5	40.7	104.9	372.6	0.0	10.7	42.3	95.8	260.3
MH-Greedy	0.2	3.1	10.4	45.7	270.1	0.0	0.0	0.0	1.0	39.1
NO TEARS	1.5	6.6	15.3	49.3	220.1	1.2	3.6	6.6	9.9	54.6
DAG-OMP	2.9	15.3	25.8	78.3	272.4	2.6	15.4	25.2	39.0	93.8
DAG-OLS-V	1.9	13.5	30.5	54.7	138.7	1.4	15.8	29.5	50.5	125.5

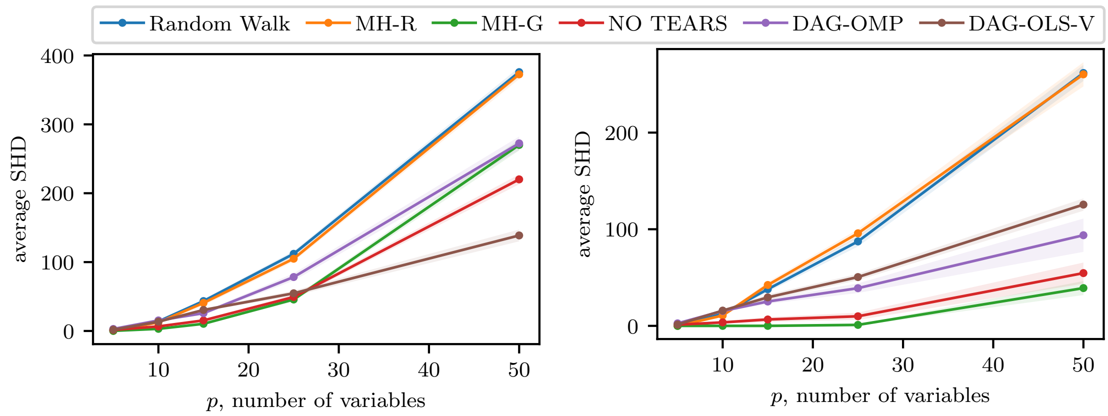

Figure B.23: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 100$ for the methods in Table B.12, excluding DAG-LASSO.

Figure B.24: Plot of the average structural hamming distance as a function of p, the number of variables, where $T = 1000$ for the methods in Table B.12, excluding DAG-LASSO.