



## UvA-DARE (Digital Academic Repository)

### The Search for Causality: A Comparison of Different Techniques for Causal Inference Graphs

Kossakowski, J.J.; Waldorp, L.J.; van der Maas, H.L.J.

**DOI**

[10.1037/met0000390](https://doi.org/10.1037/met0000390)

**Publication date**

2021

**Document Version**

Final published version

**Published in**

Psychological Methods

**License**

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/policies/open-access-in-dutch-copyright-law-taverne-amendment>)

[Link to publication](#)

**Citation for published version (APA):**

Kossakowski, J. J., Waldorp, L. J., & van der Maas, H. L. J. (2021). The Search for Causality: A Comparison of Different Techniques for Causal Inference Graphs. *Psychological Methods*, 26(6), 719-742. <https://doi.org/10.1037/met0000390>

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

*UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)*

# The Search for Causality: A Comparison of Different Techniques for Causal Inference Graphs

Jolanda J. Kossakowski, Lourens J. Waldorp, and Han L. J. van der Maas  
Department of Psychology, University of Amsterdam

## Abstract

Estimating causal relations between two or more variables is an important topic in psychology. Establishing a causal relation between two variables can help us in answering that question of why something happens. However, using solely observational data are insufficient to get the complete causal picture. The combination of observational and experimental data may give adequate information to properly estimate causal relations. In this study, we consider the conditions where estimating causal relations might work and we show how well different algorithms, namely the Peter and Clark algorithm, the Downward Ranking of Feed-Forward Loops algorithm, the Transitive Reduction for Weighted Signed Digraphs algorithm, the Invariant Causal Prediction (ICP) algorithm and the Hidden Invariant Causal Prediction (HICP) algorithm, determine causal relations in a simulation study. Results showed that the ICP and the HICP algorithms perform best in most simulation conditions. We also apply every algorithm to an empirical example to show the similarities and differences between the algorithms. We believe that the combination of the ICP and the HICP algorithm may be suitable to be used in future research.

## Translational Abstract

Psychologists study the (possible) causal relation between psychological constructs, like sleep, concentration, and feelings of guilt. For example, does sleep deprivation lead to concentration problems? And could sleep deprivation be caused by increased feelings of guilt? Knowing what the cause is of something so intrusive as sleep problems may in turn lead to finding the solution to help an individual with sleep problems. If we know what causes a problem, we can help to solve it. The type of data that is most often used to estimate causal relations between variables are observational data. These are (empirical) data in which no manipulations have taken place. Although one can use observational data to estimate some causal relations, this alone is not enough to properly estimate all relationships between variables. We also need so-called experimental data to estimate causal relations. These are (empirical) data where some perturbation or manipulation has taken place. Here, we provide an overview of a set of algorithms, namely the Peter and Clark algorithm, the Downward Ranking of Feed-Forward Loops algorithm, the Transitive Reduction for Weighted Signed Digraphs algorithm, the Invariant Causal Prediction (ICP) algorithm and the Hidden Invariant Causal Prediction (HICP) algorithm, and investigate how well each of these algorithms estimates causal relations by means of a simulation study. We also apply these algorithms to an empirical dataset. Our results showed that two algorithms, the ICP and the HICP-algorithms, perform best in most simulation conditions. We expect that the combination of these algorithms may be suitable to be used in future research.

**Keywords:** causal inference, perturbation, transitive reduction, invariant causal prediction, experimental design

**Supplemental materials:** <https://doi.org/10.1037/met0000390.supp>

Some tens of thousands of years ago, humans began to realize that certain things cause other things and that tinkering with the former can change the latter. No other species grasps this, certainly not to the extent that we do. From this discovery came organized societies, then towns and cities, and eventually the science- and technology-based

civilization we enjoy today. All because we asked a simple question: Why? (Pearl & Mackenzie, 2018).

The quest for causality is one that people have been striving for decades. Establishing a causal relation between two phenomena or

This article was published Online First July 29, 2021.

Jolanda J. Kossakowski  <https://orcid.org/0000-0002-6946-1732>

Lourens J. Waldorp  <https://orcid.org/0000-0002-5941-4625>

Han L. J. van der Maas  <https://orcid.org/0000-0001-8278-319X>

Correspondence concerning this article should be addressed to Jolanda J. Kossakowski, who is now at Research and Documentation Centre (WODC), Dutch Ministry of Security and Justice, Koningskade 4, 2596 AA The Hague, the Netherlands. Email: [jolanda.kossakowski@gmail.com](mailto:jolanda.kossakowski@gmail.com)

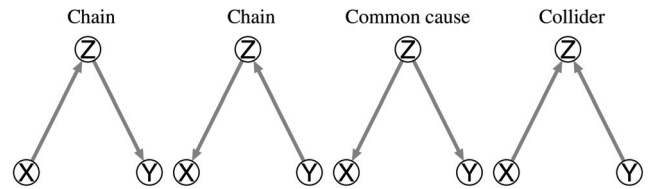
variables can help us in answering that big question of why something happens. In psychology, we study the (possible) causal relation between psychological constructs, like sleep, concentration, or feelings of guilt. For example, does sleep deprivation lead to concentration problems? And could sleep deprivation be caused by increased feelings of guilt? Knowing what the cause is of something so intrusive as sleep problems may in turn lead to finding the solution to help an individual with sleep problems. If we know what causes a problem, we can help to solve it.

What is a causal relation? We confine ourselves here to an interventional (or, equivalently, a counterfactual) definition of causality. The idea is that, if one changes (perturbs) variable  $X$ , then this should have effects only on variables  $Y$  with which  $X$  has a causal relation. For instance, if we consider the structure  $X \rightarrow Y \leftarrow Z$ , then we expect that changing  $X$  will change  $Y$ , but this change in  $X$  will not change  $Z$ . Therefore, we use the following definition of a unidirectional causal relation: “a relation between two variables ( $X \rightarrow Y$ ) where, when one changes one variable ( $X$ ), one observes a change in the other variable ( $Y$ ), and if we change variable  $Y$  we observe no change in variable  $X$ . In general, when more than three variables are involved, we require that conditioned on all other variables, a direct cause  $X$  is one that changes the distribution of  $Y$ . If we go back to the example of sleep, our definition of a causal relation states that, when there is an increase in sleep problems, there should be a change in the level of concentration as well, or any other aspect of the distribution of the effect variable. The definition of a causal relation that we use here is also a counterfactual relationship (Pearl, 2009; Peters et al., 2017). It may be argued that our definition is not sufficient to capture all aspects of a causal relation. For instance, we ignore the question of what kinds of events could have a causal relation; thereby, interpreting the causal relations.

To infer causal relations from the data (and hence the probability distribution obtained from the data) we require that any change in a causal relation in the graph implies a corresponding change in the probability distribution. This is known as the causal Markov assumption. Reversely, a change in the probability distribution implies a change in the graph, known as the faithfulness assumption. The relations between variables in the graph are referred to as d-separation (Pearl, 2009). Two variables are d-separated if the path between the variables in the graph is blocked by a third variable. The causal Markov assumption then implies that the set of d-separations in the graph implies a set of conditional independence relations in the probability distribution. The faithfulness assumption implies that a change in the conditional independencies implies a change in the set of d-separations. To illustrate, consider Figure 1 In the left panel we observe a chain structure  $X \rightarrow Z \rightarrow Y$ , where  $X$  and  $Y$  are d-separated if we block (or condition on)  $Z$ . The causal Markov assumption then implies that we should find variables  $X$  and  $Y$  conditionally independent given variable  $Z$ . Conversely, the faithfulness assumption implies that if  $X$  and  $Y$  are conditionally independent given  $Z$  in the probability distribution, then  $X$  and  $Y$  should also be d-separated in the graph.

The type of data that is most often used to estimate causal relations between variables are *observational data*. These are (empirical) data in which no perturbations have taken place. Observational data includes cross-sectional data that one collects with questionnaires for example. The most widely used technique to estimate causal relations with observational data are the algorithm developed

**Figure 1**  
The Different Causal Structures That Can be Detected With the Peter and Clark (PC)-Algorithm



*Note.* The chain structures and the common cause structure are statistically equivalent, whereas the collider structure is statistically unique.

by Pearl (2009) and Spirtes et al. (2000) or variations thereof. Pearl uses the notion of (*conditional*) dependence and independence between sets of three variables to determine a causal relation. The ideas from Pearl and Verma (1991) and Spirtes et al. (2000) indicate that, if one were to solely use multivariate normal observational data, we can infer causal relations using the notion of conditional (in)dependence. Based on the raw (simple, Pearson) and partial correlations, four different causal structures can be obtained for an example with three variables, as shown in Figure 1 In the first three situations (the two chain structures and the common cause structure), nodes  $X$  and  $Y$  have a nonzero correlation, but their partial correlation is zero when conditioning on node  $Z$ . Nodes  $X$  and  $Y$  are then said to be separated in the graph by  $Z$ . In the fourth structure (collider structure), nodes  $X$  and  $Y$  have zero correlation, but a nonzero partial correlation when conditioning on node  $Z$ . The set of conditional independence relations in the probability distribution is different for the collider structure in the right panel of Figure 1 in comparison with the other three structures. The three structures in the left three panels in Figure 1 (two chains and a common cause structure) cannot be distinguished in terms of their conditional independence, nor in terms of their d-separations; they are Markov equivalent (see, e.g., Peters et al., 2017, p. 102).

As the rules for conditional independence are equal for the first three causal structures, they are statistically equivalent and one cannot distinguish them from one another. It is only possible to identify the fourth (collider) structure from the other three (Pearl, 2000; but see Mooij et al., 2016, for some interesting cases). These ideas have been used in different methods to obtain causal relations. Tetrad (Glymour & Scheines, 1986) applied a conditional independence test to each possible alternative path, an implementation in R called ggm (short for Gaussian Graphical Models; Drton & Richardson, 2004) uses a likelihood based method for a complete set of conditional independencies. Temporal ordering has also been used (Hamaker et al., 2015; Usami et al., 2019; Zyphur et al., 2019). However, using observational data exclusively will not resolve all causal relations.

This led Granger (1980) to state that an “observed relationship does not allow one to say anything about causation between the variables,” and Holland (1986) argued that there can be “no causation without manipulation.” Although one can use observational data to estimate some causal relations, this alone is not enough to properly estimate all relationships between variables. As implied by our definition of a causal relation, one needs to perturb one variable and observe its effect to establish causal relations between variables. This means that we also need so-called *experimental*

data to estimate causal relations. These are (empirical) data where some perturbation has taken place. Real-world examples include a pre-posttest comparison with therapy as a perturbation (see, e.g., Kossakowski et al., 2021), or experimental designs in which participants are presented with hypothetical scenarios to change their attitude toward a construct (see Hoekstra et al., 2018, for an empirical example).

In an experimental study, one needs a control and an experimental condition to see if a manipulation significantly changes an outcome variable. Just like an experiment, to estimate causal relations, we need both observational data that serves as a baseline measurement, and experimental data that may show us which causal relations survive the manipulation and which ones change. We assume here that a perturbation does not alter the underlying causal structure. Thus, the combination of observational and experimental data gives us a complete picture of the causal relations between variables, which in turn may be used to set up a treatment plan where the causes of constructs like concentration problems are intervened upon, instead of the effect. We need both observational and experimental data to determine the difference after some perturbation compared with a baseline (observation, without some perturbation).

We selected four algorithms for this study that are potentially suitable for psychological data using both observational and experimental data. Two of these assume a variable-specific perturbation, meaning that a perturbation take place on each variable individually. Although this approach may work in theory, in practice it is difficult to single out symptoms of psychological disorders and perturb them accordingly. It is more likely that perturbations in psychology occur in a “fat finger” fashion, which means that multiple variables or symptoms are perturbed simultaneously. The two other algorithms do not assume that only a single variable can be perturbed at a time and so may be more useful in psychology. For comparison we also considered an algorithm that uses only observational data.

The goal of this article is threefold: (a), we want to provide an overview of a set of algorithms that stem from different fields, describing and illustrating each algorithm using both observational and experimental data; (b) we want to investigate how well each of these algorithms can estimate causal relations by means of a simulation study; and (c) we want to show how these algorithms perform when empirical data are used. First, we will describe the algorithms that can be used to estimate causal relations. For each algorithm we use a simulated dataset as an illustration. Then we will describe the simulation study that we have set up to test not only the performance of these techniques individually, but also in comparison to one another. Lastly, we will apply each algorithm to an empirical example to show how the algorithms work in practice.

## Methods of Causal Inference

The goal of this study is to compare different algorithms for inferring causal graphs. The algorithms we use all work nodewise, that is, we consider each variable (node associated with that variable) in turn and determine the variables directly connected to it; a regression basically. Within each regression we assume that if nodes are separated in the graph then this corresponds to a conditional independence in the probability distribution (Markov condition). Also, we

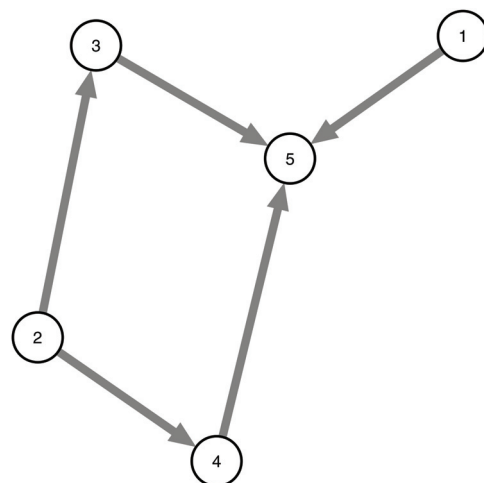
assume that a conditional independence (partial correlation for multivariate normal data) translates to a separation of nodes in the graph (faithfulness condition). See introduction above and Appendix A for more details on this.

Identifying causal relations is not an easy task. Take for example the causal graph shown in Figure 2, where one can see that there is no direct relation between variables 2 and 5 due to the chain structures  $2 \rightarrow 3 \rightarrow 5$  and  $2 \rightarrow 4 \rightarrow 5$ . So, there are three possible paths when one does not know the true graph. The trick is then to remove the path  $2 \rightarrow 5$  in this case. Here, we focus on two different types of methods. The first is called *transitive reduction* (Klamt et al., 2010; Pinna et al., 2013). Here, a causal graph is set up, and direct connections are removed if there is enough evidence to suggest that two variables are not directly connected. When there is a direct causal relation between two variables, any alternative path between these variables should be removed with transitive reduction. However, when a direct causal relation is small, algorithms that use transitive reduction may erroneously remove the direct connection in favor of the alternative paths. Transitive reduction may not always work in practice. We expand on this more in the section where we discuss the Down-Ranking of Feed-Forward Loops algorithm and the Transitive Reduction for Weighted Signed Digraphs algorithm as well as Appendix B.

The second method is by conditioning on the remaining variables (Meinshausen et al., 2016; Peters et al., 2017). In our example shown in Figure 2 this is guaranteed to work, since conditioning on both 3 and 4 will remove the correlation between variables 2 and 5. Note that even if we were to perturb the variable associated with node 2, then conditioning on nodes 3 and 4 would still lead to no change in the variable associated with node 5, allowing the correct inference that there is no direct relation from node 2 to node 5.

To explain these algorithms, we will use one simulated causal graph and associated dataset that contains five variables, visualized in Figure 2. For illustration purposes we simulated data for 1,000

**Figure 2**  
*Visualization of the Causal Graph That We Use to Illustrate the Different Algorithms*



*Note.* Arrows represent causal relations between individual variables, which are depicted as circles.

measurements. We will compare five algorithms: the *Peter and Clark* algorithm (PC; Kalisch & Bühlmann, 2007); the *Down-Ranking of Feed-Forward Loops* algorithm (DR-FFL; Pinna et al., 2013), the *Transitive Reduction for Weighted Signed Digraphs* algorithm (TRANSWESD; Klamt et al., 2010); the *Invariant Causal Prediction* algorithm (ICP; Meinshausen et al., 2016) and the *Hidden Invariant Causal Prediction* algorithm (HICP; Peters et al., 2017). We chose to include the PC-algorithm, even though it only uses observational data, to compare its results to algorithms that include experimental data next to observational data. Other algorithms that use observational data include a directional dependence model using copulas (Sungur, 2005), a linear causal acyclic model (Shimizu et al., 2006), or a directional dependence analysis with possible confounding variables (Wiedermann & Sebastian, 2019). We chose to restrict our study to these algorithms because we were interested in combining observational and experimental data and different types of perturbations.

The data used to illustrate the different algorithms are publicly available, so that the reader may use the data to replicate our examples. Throughout this section we use the example graph in Figure 2 with  $p = 5$  variables and  $n = 1,000$  observations (independent and identically distributed). Edge  $e_{ij}$  denotes a directed edge  $i \rightarrow j$ . Symbols that are associated with specific algorithms will be explained when we introduce the symbol for the first time. At the end of this section, we provide a summary table (see Table 1) that gives an overview of the algorithms that are discussed here, and their properties.

**PC-Algorithm**

The PC-algorithm (Spirtes et al., 2000) has a two-step procedure that solely uses observational data. We used the *R*-package *pcalg* (Version 2.6-2; Kalisch et al., 2012) to run the PC-algorithm. The first step in the PC-algorithm is to find the *skeleton* of the causal graph: an undirected graph that shows all possible causal relations. For each node individually, we look at every possible relation with every other node in the graph. The raw correlation between each pair of nodes is calculated (matrix  $r$  in (1)). Then the partial correlations are calculated between every pair of nodes (matrix  $r_p$  in (1)), conditioning on subsets of the remaining variables, increasing in size of the subsets. All possible partial correlations are calculated until either the algorithm has calculated the partial correlation for all possible subsets, or until a partial correlation returns zero when conditioning on a specific subset. In the latter case the correlation in  $r$  is explained away by another variable. This can be seen, for instance, in the partial correlation matrix  $r_p$  below where the partial correlation between nodes 2 and 5 drops from .723 to .038 when conditioning on the remaining three nodes.

In the second step of the PC-algorithm, the direction of the relation is determined by considering *collider* structures (fourth panel, Figure 1). Because the correlational pattern for a collider (nonzero partial correlation between nodes  $X$  and  $Y$ ) is different from the chain and common cause structure (zero partial correlation between nodes  $X$  and  $Y$ ), the collider structure can be distinguished, and hence gives information about the direction of the causal relations. This can be seen from the partial correlation matrix  $r_p$  below where the partial correlation between nodes 1 and 3 is  $-.468$  (conditioning on all three remaining nodes), while

**Table 1**  
*Overview of the Algorithms*

Algorithm	Observational data	Experimental data	$N = 1$	$N > 1$	Within-subjects	Between-subjects	Correction for multiple testing	Cyclic graphs	Limitations	Sensitivity	Specificity
PC	✓	—	—	✓	✓	✓	—	—	Uses only observational data	Partially high	High
DR-FFL	✓	✓	—	✓	—	—	—	—	Resulting graph is unweighted and unsigned	Low	High
TRANSWESD	✓	✓	—	✓	—	—	—	—	Uses arbitrary threshold	Low	High
ICP	✓	✓	—	✓	✓	✓	—	—	Computationally slow with many variables	Partially high	High
HICP	✓	✓	—	✓	✓	✓	✓	—	Contains spurious relations	High	Partially high

*Note.* PC = Peter and Clark; DR-FFL = Down-Ranking of Feed-Forward Loops; TRANSWESD = Transitive Reduction for Weighted Signed Digraphs; ICP = Invariant Causal Prediction; HICP = Hidden Invariant Causal Prediction;  $N$  = number of participants; ✓ = algorithm can handle that specific property; — = algorithm cannot handle that specific property. Partially high means that the sensitivity/specificity is high depending on certain conditions.

without conditioning, the (Pearson) correlation is  $-.019$ . The fact that there is no or a very small correlation without conditioning, but a large partial correlation when conditioning implies that there must be a collider structure (see Figure 2). Note that the PC-algorithm does not automatically correct for multiple testing.

$$r = \begin{pmatrix} 1.000 & & & & \\ -0.028 & 1.000 & & & \\ -0.019 & 0.711 & 1.000 & & \\ -0.073 & 0.717 & 0.517 & 1.000 & \\ 0.298 & 0.723 & 0.759 & 0.748 & 1.000 \end{pmatrix}$$

$$r_p = \begin{pmatrix} 1.000 & & & & \\ -0.006 & 1.000 & & & \\ -0.468 & 0.389 & 1.000 & & \\ -0.515 & 0.392 & -0.467 & 1.000 & \\ 0.686 & 0.038 & 0.685 & 0.693 & 1.000 \end{pmatrix} \quad (1)$$

Figure 3 (left panel) shows the skeleton based on our illustration data, using a significance level of  $.05$ . The right panel of Figure 3 shows the final result of our illustration. Four out of six edges that are present are correctly identified. What is interesting in this example is the edge  $e_{23}$ . This edge is undirected, the PC-algorithm could not determine a direction. This makes sense when we look at the causal structure in Figure 2 that is formed between nodes 2, 3, and 5, and between nodes 3, 2, and 4. No matter the direction of the edge between nodes 2 and 3, the causal structures between nodes 2, 3, and 5, and between nodes 3, 2, and 4 will remain statistically equivalent. It is impossible for the PC-algorithm to determine a direction. This illustration shows one of the prime disadvantages of the PC-algorithm, in that it obtains an equivalence class of graphs that are all equally likely to be true, and so some directions of edges cannot be resolved. The other edge that stands out is  $e_{14}$ . In this case the correlation between the variables associated with nodes 1 and 4 exists because of the induced path  $4 - 3 - 1$  when conditioning on node 5. Hence it shows up in the skeleton and in the final graph. Overall, in this illustration the PC-algorithm performs reasonably well, only one edge is incorrectly estimated, and one edge is left undirected. Most of the edges that are present in the true causal graph are correctly estimated.

### DR-FFL-Algorithm

The Downward-Ranking of Feed-Forward Loops algorithm (DR-FFL; Pinna et al., 2013) has an advantage over the PC-algorithm in that it uses both observational data and experimental data to estimate a causal graph. The DR-FFL-algorithm (Pinna et al., 2013) originates from the field of gene biology and estimates unweighted (no edge weights), unsigned (edge can be positive or negative, there is no information on this) causal graphs for single subjects and single measurements (where each node was perturbed once). The DR-FFL-algorithm uses a two-step procedure. In the first step, the algorithm compares the effect of perturbing a node to the average effect that includes the observational data as well to create a *perturbation graph* (PG). In the second step, the DR-FFL-algorithm applies *transitive reduction* to remove direct causal relations from the perturbation graph where indirect effects are in order.

The DR-FFL-algorithm needs two components to infer the causal graph: observational data for each of the nodes ( $G^{wt}$ ; also known as wild-type data) and experimental data ( $G^{ko}$ ; also called knock-out data) where each node in the data are perturbed. The observational data are given in (3) for the example data based on Figure 2. The experimental data in (3) consists of results of a particular node being perturbed. For example, row 1 of the matrix in (3) depicts the new values that the nodes in the graph have after perturbing node 1.

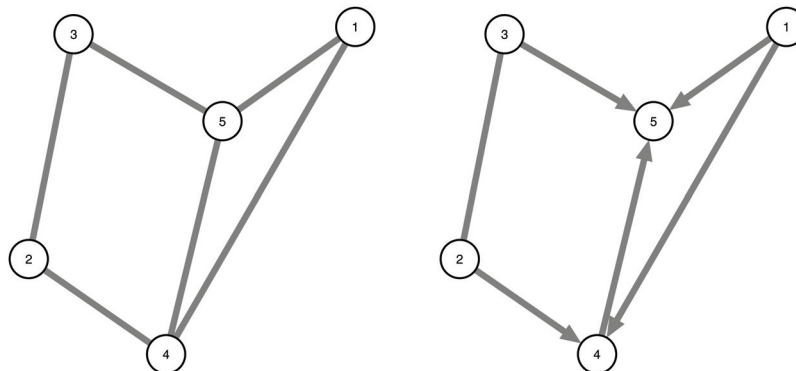
$$G^{wt} = (-0.015, 0.025, -0.001, 0.013, 0.015) \quad (2)$$

$$G^{ko} = \begin{pmatrix} -0.018 & 0.115 & -0.003 & 0.064 & 0.072 \\ -0.070 & -0.012 & -0.004 & 0.063 & 0.073 \\ -0.079 & 0.128 & 0.134 & 0.067 & 0.075 \\ -0.073 & 0.108 & -0.004 & 0.082 & 0.080 \\ -0.093 & 0.105 & -0.003 & 0.079 & -0.032 \end{pmatrix} \quad (3)$$

The first step is to obtain a PG where an effect is determined by normalizing and comparing the perturbed effect using a  $z$ -score:

$$|z_{ij}| = \left| \frac{G_{ij}^{ko} - \mu_j}{\sigma_j} \right| \quad (4)$$

**Figure 3**  
Visualization of the Skeleton (Left Panel), and the Causal Graph (Right Panel)  
Estimated With the Peter and Clark (PC)-Algorithm (Kalisch et al., 2012)



where  $\mu_j$  is the mean, and  $\sigma_j$  the standard deviation of node  $j$  across different perturbations.

Both include the observation for node  $j$ . The PG is then generated by selecting those edges whose  $|z|$ -score (shown in (5)) is larger than a prespecified threshold  $\beta$ . The resulting PG with the edges that survive a threshold of  $\beta = .60$  is seen in the left panel of Figure 4. This threshold  $\beta = .60$  is arbitrary in that no heuristic is known for sensible values. Note that we averaged over the sample to highlight the differences between the DR-FFL and the TRANSWESD-algorithm (discussed in the next section).

$$|z| = \begin{pmatrix} 0.000 & 0.651 & 0.405 & 0.090 & 0.542 \\ 0.370 & 0.000 & 0.425 & 0.072 & 0.567 \\ 0.641 & 0.872 & 0.000 & 0.233 & 0.611 \\ 0.452 & 0.513 & 0.423 & 0.000 & 0.718 \\ 1.047 & 0.463 & 0.416 & 0.693 & 0.000 \end{pmatrix} \quad (5)$$

The second step of the DR-FFL-algorithm is transitive reduction. In this step, the algorithm narrows its search to edges that connect strongly connected components. A strongly connected component is a (sub)set of nodes where any node can be reached (i.e., there must be a directed path) from any other node in the component. Such a subset is called transitive. The DR-FFL-algorithm only focuses on edges between strongly connected components because cycles exist between the nodes within a strongly connected component. For each edge  $e_{ij}$  that connects two strongly connected components, the DR-FFL-algorithm searches for alternative paths, and removes the direct edge  $e_{ij}$  if the alternative path satisfies two criteria (1), edge  $e_{ij}$  can only be removed when  $e_{ij}$  connects different strongly connected components in the PG and (2), edge  $e_{ij}$  can only be removed when there is an alternative route from node  $i$  to node  $j$  without using  $e_{ij}$ .

In this illustration based on the example graph in Figure 2, four strongly connected components exist (see the middle panel of Figure 4): nodes 4 and 5 form a strongly connected component (component A), and nodes 1, 2, and 3 each from their own individual component (components B, C, and D, respectively). There are

only five edges that connect these strongly connected components, shown by the middle panel in Figure 4. For each of these five edges, the DR-FFL-algorithm determines whether an alternative path exists to connect these two components. There are no alternative paths between components A and B, components A and C, and components A and D. There is an alternative path between components D and B ( $D \rightarrow A \rightarrow B$ ) and components D and C ( $D \rightarrow A \rightarrow B \rightarrow C$ ). Thus, the edges that directly connect components D and B ( $e_{31}$ ) and components D and C ( $e_{32}$ ) are removed from the causal graph, resulting in the graph shown in Figure 4 (right panel).

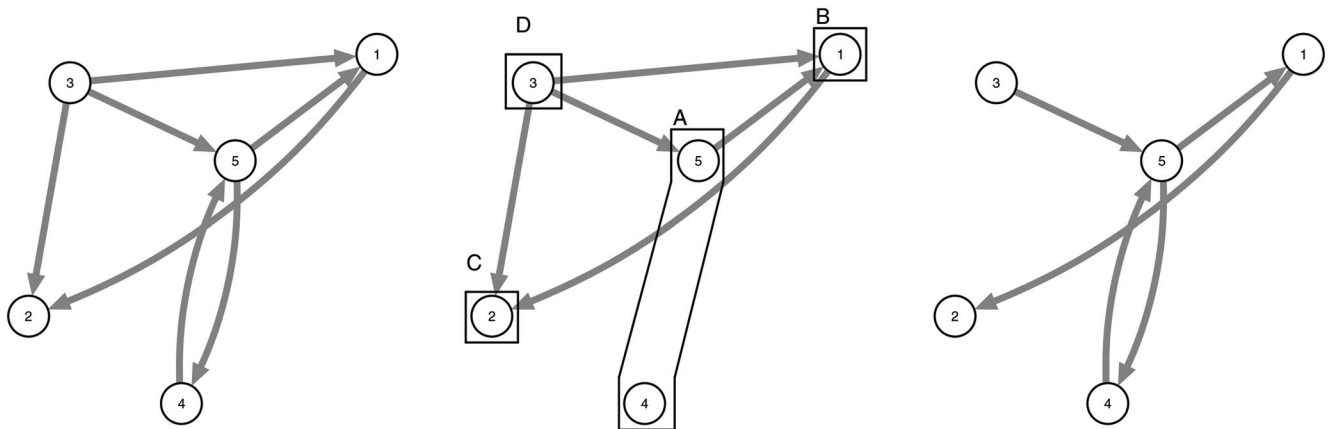
Overall, the DR-FFL-algorithm does not perform well in this illustration. Only two edges that exist in the true causal graph (see Figure 2) are also estimated here. Two edges are estimated in the wrong direction, one edge is incorrectly estimated and one edge is incorrectly absent from the graph.

### Transitive Reduction for Weighted Signed Digraphs

The TRANSitive reduction for WEighted Signed Digraphs (TRANSWESD; Klamt et al., 2010; Pinna et al., 2013) returns a causal graph with weighted edges that indicate a positive or a negative relationship, while applying transitive reduction to estimate a causal graph at the same time. Furthermore, where the DR-FFL algorithm mostly handles single-subjects data, the TRANSWESD-algorithm can be solely applied to between-subjects data.

As a first step, we generate the PG. Like the DR-FFL-algorithm, we calculate  $|z|$ -scores. In addition to the  $|z|$ -score, we calculate an absolute change score  $c$  (shown in (6)) between  $G^{wt}$  and  $G^{ko}$  that shows the absolute effect of perturbing a node. Edges are retained in the PG when their associated  $|c|$ -scores exceed a prespecified threshold  $\gamma$ . Each edge in the PG gets a sign  $s_{ij}$  that reflects the direction of the change that node  $j$  has made after node  $i$  was perturbed: if the change score is positive, then the edge will be blue, and when the change score is negative, the edge will be red. Each edge also has a weight  $w_{ij}$  that reflects the uncertainty of the causal relation, where a higher weight indicates a lower certainty. The weight  $w_{ij}$  is determined by  $1 - |\rho_{ij}|$ , where  $\rho_{ij}$  is the *conditional*

**Figure 4**  
Visualization of the Down-Ranking of Feed-Forward Loops (DR-FFL) Process



*Note.* The left panel denotes the perturbation graph in which present edges represent potential causal relations whose effect were strong enough. The middle panel depicts the edges that can be removed in the transitive reduction step of the DR-FFL-algorithm. The black boxes around the nodes in the left panel indicate the strongly connected components. The right panel depicts the final causal graph that results from the DR-FFL-algorithm.

correlation (Rice et al., 2005). The idea of a conditional correlation is that a variable is influenced by another if it is similar (in terms of correlation) in both the observational and experimental condition. Upon perturbation of one variable the other variable behaves similarly and so the correlation will be high. The assumption is, of course, that the causal relations remain the same in the observed and experimental conditions. A conditional correlation is a correlation between two nodes  $i$  and  $j$ , given that node  $i$  was perturbed, and is calculated as follows:

$$|c| = \begin{pmatrix} 0.000 & 0.131 & 0.013 & 0.079 & 0.087 \\ 0.096 & 0.000 & 0.029 & 0.038 & 0.048 \\ 0.079 & 0.129 & 0.000 & 0.068 & 0.076 \\ 0.086 & 0.094 & 0.017 & 0.000 & 0.067 \\ 0.108 & 0.090 & 0.018 & 0.064 & 0.000 \end{pmatrix} \quad (6)$$

$$\rho_{ij} = \frac{\sum_{a=1}^{2n} (x_{i,a} - \bar{x}_i)(x_{j,a} - \bar{x}_j)}{\left(\sum_{a=1}^{2n} [x_{i,a} - \bar{x}_i]^2\right)^{1/2} \left(\sum_{a=1}^{2n} [x_{j,a} - \bar{x}_j]^2\right)^{1/2}} \quad (7)$$

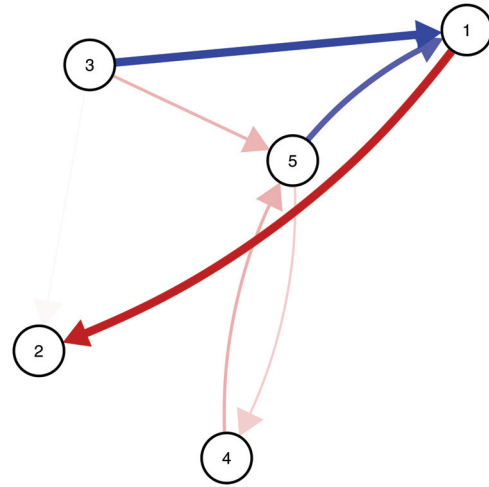
For nodes  $i$  and  $j$  Equation (7) only uses the observational data and the experimental data where node  $i$  was perturbed. This gives us two vectors,  $x_i$  and  $x_j$ , each of length  $2n$ , assuming the same number of data points for observational and experimental data. Parameters  $\bar{x}_i$  and  $\bar{x}_j$  represent the means of these two vectors. To illustrate, row 3 in (8) states that the first node is not really influenced by node 3 as evidenced by the small conditional correlation of  $-0.029$ . In contrast, the second node has a conditional correlation of  $.965$ , suggesting that it is very much influenced by node 3. Due to the design of the conditional correlation, the resulting matrix, is not symmetric. The resulting PG is shown in Figure 5 (with  $\beta = .60$  and  $\gamma = .05$ ).

$$\rho = \begin{pmatrix} 0.000 & -0.015 & -0.034 & -0.008 & 0.032 \\ -0.039 & 0.000 & 0.068 & 0.108 & 0.087 \\ -0.029 & 0.965 & 0.000 & 0.695 & 0.707 \\ -0.034 & 0.953 & 0.676 & 0.000 & 0.700 \\ 0.344 & 0.755 & 0.788 & 0.804 & 0.000 \end{pmatrix} \quad (8)$$

In the second step of the TRANSWESD-algorithm, the algorithm removes an edge  $e_{ij}$  when there is an alternative path between nodes  $i$  and  $j$  and when that alternative path satisfies the following four conditions: (1) the alternative path must not contain a cycle, (2) the alternative path cannot contain the edge  $e_{ij}$  that is under consideration, (3) the overall sign of the alternative path must be equal to that of the edge  $e_{ij}$  under consideration (obtained by multiplying the signs of all edges on the alternative path) and (4), the maximum weight of all edges on the alternative path must be lower than the weight of the edge  $e_{ij}$  under consideration multiplied by a prespecified threshold  $\alpha$ . For all analyses, we set  $\alpha = .95$ , the default value used by Klamt et al. (2010). All edges that exist in the PG are sorted based on their edge weight. The transitive reduction starts with the edge that has the highest weight (and the lowest certainty).

In the PG in Figure 5 that is based on the example graph in Figure 2, four edges have no alternative path ( $e_{15}$ ,  $e_{51}$ ,  $e_{45}$ , and  $e_{54}$ ), and of the remaining five edges, two edges contain a cycle on their

**Figure 5**  
Visualization of the Perturbation Graph Generation



Note. Blue edges indicate positive causal relations, and red edges denote negative causal relations. The thickness and saturation of the edge color indicate the strength of the causal relation. See the online article for the color version of this figure.

alternative paths (condition 1). With three edges left, two of these satisfied the third condition (the product of the signs of the alternative path must match the sign of the edge  $e_{ij}$  that is under consideration). Both of these edges did not meet the final requirement that states that the maximum weight of all edges on the alternative path cannot exceed the weight of the edge  $e_{ij}$  under consideration multiplied by  $\alpha$ . This means that no edges are removed from the causal graph, and that the perturbation graph in Figure 5 will not change after the transitive reduction step.

Similar to the DR-FFL-algorithm, the performance of the TRANSWESD-algorithm seems subpar. Three edges that exist in the true causal graph (see Figure 2) are correctly estimated, two edges are estimated in the wrong direction, two edges are incorrectly estimated and two edges are incorrectly deemed absent from the graph.

### Invariant Causal Prediction

The ICP-algorithm (Meinshausen et al., 2016) combines both the advantage of the PC-algorithm in that it considers a multivariate system, and uses both observational and experimental data in a single analysis. Another advantage of the ICP-algorithm is that the perturbations inflicted on the data do not have to be node-specific: perturbations can be nonspecific and generic for subsets of nodes. The idea of the ICP is somewhat similar to that of the conditional correlation, used in the TRANSWESD-algorithm. If one variable is causally related to another variable, then the correlation in both the observational and the perturbation conditions will be similar. The ICP generalizes this idea to more conditions (called environments here) and uses the distribution of the residuals instead of considering the correlation. Then in a multivariate system, if the direct causes are obtained and conditioned on, then perturbing one of the variables will not lead to a change (is invariant) in the

residuals. Hence, the core assumption of the ICP-algorithm is that the conditional distribution of an individual node, controlling for its direct causes, does not change across perturbations (Peters et al., 2016). In other words, a causal relation between two nodes only exists when the residuals do not change when a perturbation has taken place on any node, except for the dependent node in the regression, called the *target node* here. Recall that we use a node-wise procedure where each node is the dependent variable in turn.

The ICP-algorithm needs two components: the observational and experimental data, and an instrumental variable  $\epsilon$  that distinguishes between different perturbations, which we call *environments*, following Peters et al. (2016). This is similar to the experimental data matrix used in the DR-FFL and TRANSWESD-algorithms, where the rows indicate each separate perturbation. Another example of a situation where multiple environments exist is in data sets where every participant is measured on two or more time points. Every time point is then a unique environment. The minimal requirement is that the data must have at least two environments. Typically, one environment consists of observational data.

The ICP-algorithm first select a *target node* and then uses the remaining nodes to identify all possible subsets, similar to the PC-algorithm. Subsets can range from an empty subset (where the target node had no cause) to a subset that contains all remaining nodes. Figure 6 shows all possible subsets for a regression when node 5 is the target node; the true graph from Figure 2 is shown at the top. The ICP-algorithm regresses the target node onto each subset separately, and obtains its associated residual distribution. For the correct subset (indicated by the black square in Figure 6) the conditional residual distribution (given the predictors) will be the same for any change on one of the other nodes. For if the distribution were to change, then it is clear that an incorrect subset of direct causes is obtained. The residual distribution is tested against the residuals of all remaining environments using a Kolmogorov-Smirnov test. Subsets whose residual distribution is equal across environments are called “invariant”; only those nodes that are part of each invariant subset (intersection) are said to be causes of the target node, and edges are drawn from those nodes to the target node.

To illustrate, Figure 7 shows the residual distribution of two subsets, the empty subset (left panel) and the true subset (right panel). The residual data for the two environments is separated by a dashed vertical line. It is obvious that the residual distributions for the empty subset are not equal (using a significance level of .05). In contrast, the residual distributions of the true subset (right panel) do not show a visual difference between the two environments. Based on this illustration, we conclude that the empty subset holds the incorrect subset of direct causes, while the true subset is “invariant” across environments. In the situation where more than one subset is accepted, the ICP-algorithm will only select the nodes that appear in all accepted subsets (the intersection of the subsets) and will return that set as the set of nodes that have a causal relation for which the target node is on the receiving end. The ICP-algorithm is then repeated for each node in the data, ensuring that every node is the target node once. Because the null hypothesis of the distribution of the residuals must hold for all subsets and all environments, the probability of obtaining a spurious connection is extremely small. This is only true if the test (an  $F$ -test on the residuals) is a level-alpha

test (see Peters et al., 2016; Theorem 1 for details). In our current set-up this implies that the data should be approximately normally distributed. Using the example graph in Figure 2, we end up with the same graph shown in the top row of Figure 6. It is important to note that the direct causes of a target node and its associated residuals must be independent of each other. This assumption guarantees that in this set-up there are no confounding variables. The next section will describe a solution when this assumption cannot be satisfied.

An assumption of the ICP and the HICP-algorithm (discussed in the next section) is that the target node cannot be intervened upon. Depending on the type of research this may or may not hold in practice. We provide two examples to clarify this point. First, suppose that a researcher has data from a therapeutic intervention with only sleep medication. Then it seems reasonable that a target variable like lack of concentration is not intervened upon directly; but is of course affected by sleep indirectly, as is investigated. As a second example, consider a treatment where therapy sessions are guided by psychoanalytical principles. While many aspects are intervened upon at once, it may be difficult to find target variables that satisfy the assumption of the (H) ICP-algorithm. In the empirical data analysis we provide an example where we believe that ICP and HICP-algorithm are valid.

In constructing a graph with the causal relations obtained from different interventions, we may encounter cycles. Similar to Meinshausen et al. (2016), we assume in that case that the interventions were in terms of a shift where the parameters in the cycle are not too large (see Rothenhäusler et al., 2015, for more details). This assumption entails that the causal relation is a solution to a difference equation and so the causal relations are not to be interpreted as being simultaneous.

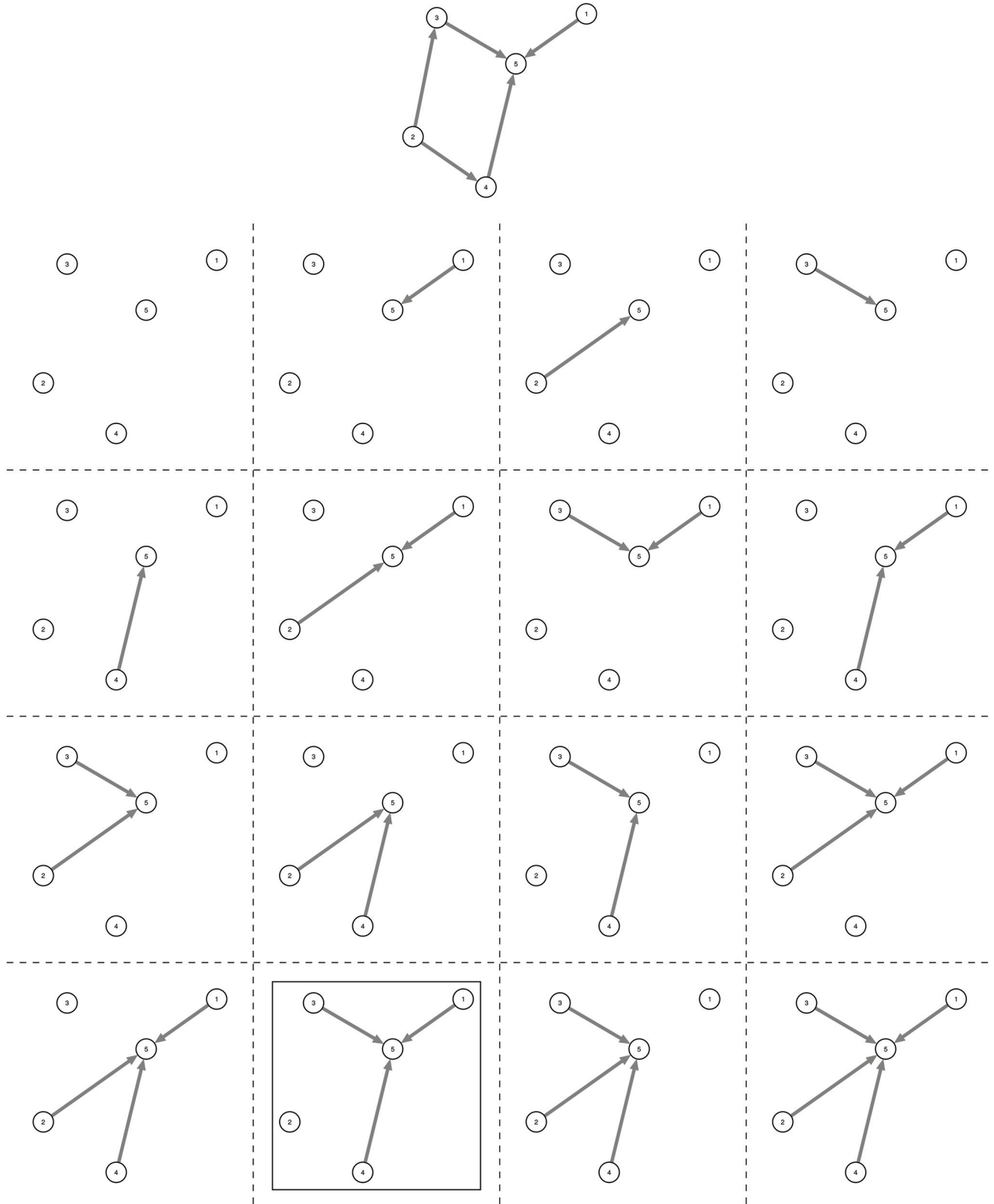
The data we used to illustrate the ICP-algorithm contains six unique environments: one environment that contains the observational data, and five environments in which we perturbed all nodes except node 5 (see the section on data simulation for a more detailed description). We only select these two environments because one of the main assumptions of the ICP-algorithm states that perturbations can take place at all nodes but the target node (Peters et al., 2016). We use the  $R$ -package *InvariantCausalPrediction* (Version .7-2, Meinshausen, 2018) to run the ICP and the HICP-algorithm.

Overall, the ICP-algorithm performs exceptionally well in this illustration. All edges that exist in the true causal graph (see Figure 2) are correctly estimated, and there are no edges that are incorrectly estimated or determined to be absent from the graph.

## Hidden Invariant Causal Prediction

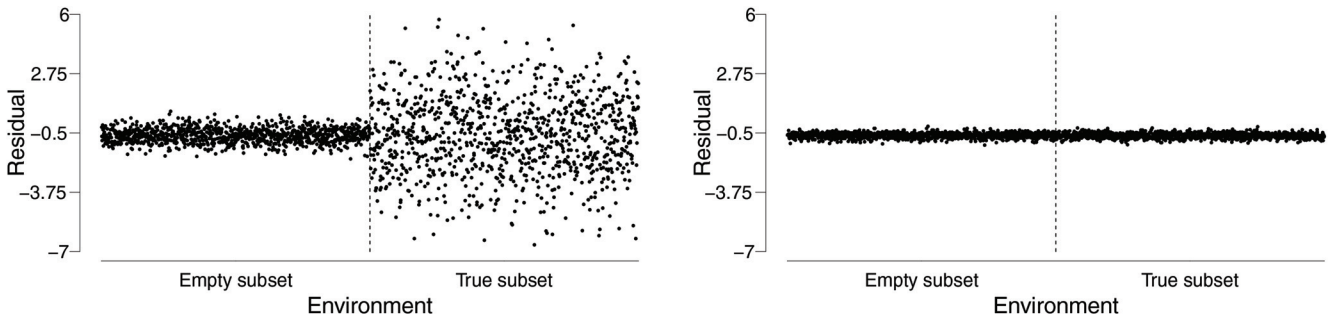
The HICP-algorithm (Peters et al., 2017) is similar to the ICP-algorithm discussed previously. The major difference between the two algorithms is that the HICP-algorithm controls for *hidden variables*, variables that are unobserved, but may affect the observed variables. Where the ICP-algorithm assumes that a target node’s direct causes and its residuals are independent, this assumption can no longer be satisfied when hidden variables exist. To illustrate, see Figure 8, after Peters et al. (2017). Here,  $Y$  denotes the target node,  $X$  its direct cause,  $H$  the hidden variable, and  $Z$  the instrumental variable. As seen in Figure 8, the

**Figure 6**  
*The True Causal Graph (Most Upper Panel) and All Possible Subsets That May Potentially Cause the Target Node 5*



*Note.* The set in the black box indicates the subset that captures the true causal relation with the target node.

**Figure 7**  
 Visualization of the Residual Distribution for the Empty Subset (Left Panel) and the True Subset (Right Panel)



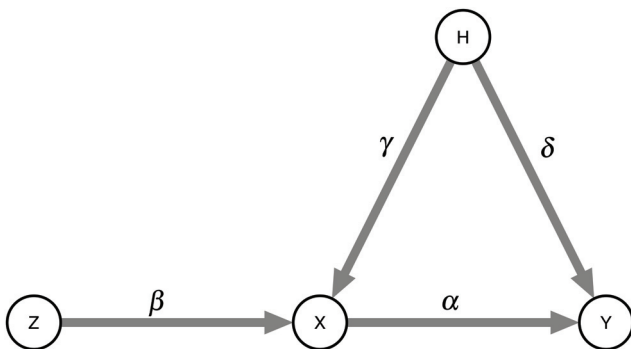
Note. The vertical dashed line indicates the partition of the residuals according to the different environments that we used for this illustration.

hidden variable affects both the target node and its direct cause. Thus,  $H$  is a confounding variable. If one were to use the ICP-algorithm, where hidden variables are not accounted for, the correlation between the target node  $Y$  and its direct cause  $X$  will be inflated due to the influence of the hidden variable  $H$ . It is impossible to infer the unique influence between  $X$  and  $Y$  in this illustration. Consider the setup in Figure 8. For explanatory purposes, the HICP-algorithm implements an instrumental variable  $Z$  to remove the effect of the hidden variable on  $X \rightarrow Y$ . This instrumental variable is assumed not to directly influence the target node  $Y$ . Here, the environmental variable  $\epsilon$  is used as the instrumental variable, as the division of the data into two separate environments does not directly influence the target node  $Y$ . By using the environmental variable  $\epsilon$  as the instrumental variable  $Z$ , the regression of the target node onto the remaining variables will be split for the different time points, and the difference between these time points is used to estimate the causal effect.

For explanatory purposes, we name the causal effect from  $X$  to  $Y$  to be  $\alpha$  (as shown in Figure 8 and as described by Peters et al., 2017). The variables  $X$  and  $Y$  are defined as follows:

$$\begin{aligned} X &= \beta Z + \gamma H + N_x \\ Y &= \alpha X + \delta H + N_y \end{aligned} \tag{9}$$

**Figure 8**  
 Illustration of the Hidden Invariant Causal Prediction (HICP)-Algorithm



Note. Figure is adapted from Peters et al. (2017).

which follows directly from Figure 8. The terms  $N_x$  and  $N_y$  denote the error terms (here we assume normally distributed variables with mean 0 and variance  $\sigma$ ). The estimate of the causal effect from  $X$  to  $Y$ ,  $\hat{\alpha}$  is defined as follows:

$$\hat{\alpha} = \frac{\text{cov}[X, Y]}{\text{var}[X]} = \alpha + \frac{\delta \gamma \text{var}[H]}{\text{var}[X]} \tag{10}$$

where  $\alpha$  denotes the causal effect from  $X$  to  $Y$ , and  $\frac{\delta \gamma \text{var}[H]}{\text{var}[X]}$  the bias term to account for the hidden variable. When there are no hidden variables,  $\delta$  equals 0, and the bias term will disappear as a result of this. When hidden variables exist but not accounted for, one ends up with a biased estimate for the causal effect from  $X$  to  $Y$ . This shows that, in this situation, the estimate for the causal effect is not representative of the true causal effect. The HICP-algorithm follows a two-step regression to estimate  $\alpha$ . It first regresses  $X$  on  $Z$  to estimate  $\hat{\beta}$ , where the estimate is denoted by  $\hat{\beta}$ . Then, this coefficient is used to estimate  $\alpha$ :

$$\hat{\alpha} = \frac{\text{cov}[\hat{\beta}Z, Y]}{\hat{\beta}^2 \text{var}[Z]} = \frac{\alpha \hat{\beta}^2 \text{var}[Z]}{\hat{\beta}^2 \text{var}[Z]} \tag{11}$$

When the sample size becomes large,  $\hat{\beta}$  and  $\beta$  will be arbitrarily close. Equation (11) shows that, in the limit, the estimate for  $\hat{\alpha}$  will be equal to the true causal effect. See Appendix C for a more detailed description. It is important to note that the HICP-algorithm assumes that the hidden variable  $H$  and the instrumental variable  $Z$  (the environmental variable  $\epsilon$ ) are independent of one another. The ICP-algorithm has to satisfy the assumption that the causes of a target node and its associated residuals are uncorrelated. In contrast, the HICP-algorithm frees up this assumption. Another difference between the HICP and the ICP-algorithm is that the HICP-algorithm does not create subsets of the set of nodes that remain after selecting a target node. To speed up computations, all variables are simultaneously tested to see if they are a cause of the target node.

When we select node 5 as our target node, the correct subset of direct causes is depicted in Figure 6. The causal coefficients  $\alpha$  are estimated as follows. Let  $X$  be a  $n \times (p - 1)$  matrix that contains the raw data (both observational and experimental data) for all variables but the target node:

This document is copyrighted by the American Psychological Association or one of its allied publishers. This article is intended solely for the personal use of the individual user and is not to be disseminated broadly.

$$\hat{\alpha} = (X'X)^{-1}X'y$$

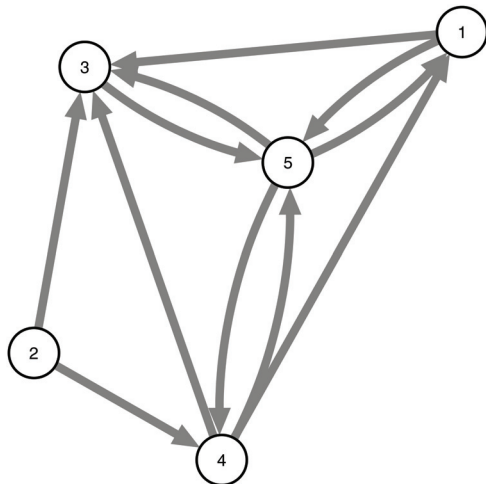
$$= \begin{pmatrix} -8.204 & 0.171 & 0.194 & 0.866 \\ 0.171 & -4.056 & -5.138 & -5.955 \\ 0.194 & -5.138 & -12.874 & -7.624 \\ 0.866 & -5.955 & -7.624 & -16.888 \end{pmatrix}^{-1} \begin{pmatrix} -7.124 \\ -10.961 \\ -20.295 \\ -23.772 \end{pmatrix} \tag{12}$$

$$\hat{\alpha} = \begin{pmatrix} 0.998 \\ 0.009 \\ 0.991 \\ 1.009 \end{pmatrix} \tag{13}$$

One can immediately see that nodes 1, 3, and 4 have a causal coefficient that is very close to 1 and that node 2 has an almost nonexistent causal coefficient. The causal coefficients in  $\alpha$  are then tested for significance. When we repeat the HICP-algorithm for each node in our illustration data (that does not contain any hidden variables), we end up with the causal graph depicted in Figure 9. It is noticeable that, next to the edges that are present in the true causal graph (shown in Figure 2), many spurious edges exist. Because the HICP-algorithm tests all variables in the data simultaneously, spurious edges can arise as a result of partialing out the effect of the hidden variables. With a large sample size (like the sample size of the illustration data), these spurious edges are easier deemed as significant causal relations.

We have added a detailed description of the entire estimation part as it occurs in the *R*-package for the interested reader in Appendix C. We use the *R*-package InvariantCausalPrediction (Version .7-2; Meinshausen, 2018) to run the HICP-algorithm. We programmed a wrapper function so that both the ICP and the HICP-algorithm are repeated for every variable in the data, and then combined into a single adjacency matrix, which is publicly available at <https://osf.io/n8gxx/>. Overall, the HICP-algorithm does not seem to perform too well in this illustration. All edges

**Figure 9**  
Illustration of the Hidden Invariant Causal Prediction (HICP)-Algorithm



that exist in the true causal graph (see Figure 2) are correctly estimated, but there are many edges (6) that are incorrectly estimated.

Table 1 gives an overview of the algorithms discussed in this section. In this overview, one can see which of the five algorithms is most suitable for a specific dataset. For example, all algorithms can be used to estimate a within-subjects causal graph with the exception of the TRANSWESD-algorithm, and the DR-FFL-algorithm is unsuitable for a between-subjects analysis. Table 1 also shows every algorithm's limitations. Based on the criteria that we set for possible application to psychological data, the ICP and the HICP-algorithms seem the most suitable and the most versatile.

**Data Simulation**

To study the accuracy of the algorithms that we described in the previous section, we simulate data according to a DAG, and apply each of the five algorithms to estimate a causal graph. In this section, we first discuss how we constructed the DAGs, after which we show how we simulated data based on these DAGs.

We start out with a  $p \times p$  adjacency matrix that consists of solely 0s. Then, we randomly select  $k$  cells and set them to 1s: a 0 indicates the absence of an edge, and a 1 the presence of an edge. The parameter  $k$  here denotes the number of edges that is dependent on the prespecified density  $d$  ( $d \cdot p^2 - p = k$ ). The diagonal of the matrix is always 0, as self-loops are not permitted at this point in time. All graphs were constructed such that the number of edges in the graph must equal  $k$ , (2) each node in the graph must have at least one connection, (3) all edge weights in the graph must equal 1, (4) all nodes in  $G$  must be connected in one component and (5), the graph cannot contain any cycles. This process may result in the following adjacency matrix:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

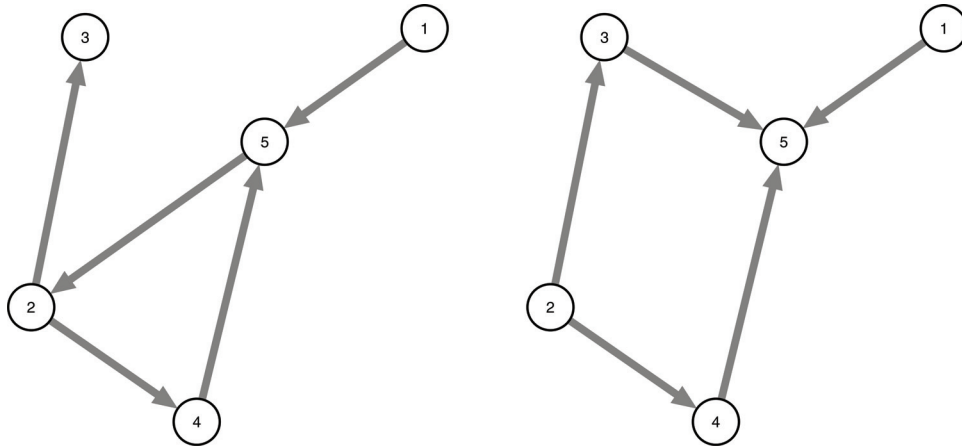
$$\rightarrow \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 10 shows the visualization of the graph after the initiation process (left panel) and after all criteria are satisfied (right panel). After the graph is finalized, its adjacency matrix is ordered so that all nonzero elements are in the lower-diagonal part of the matrix. This process does not alter the graph itself, solely its representation.

Based on the adjacency matrix of the simulated DAG that we call  $B$ , we simulate data in which some sort of perturbation has taken place. We create an  $n \times p$  matrix ( $X$ ) that we fill with numbers drawn from a normal distribution with a mean of 0 and a standard deviation of .5. We then select a node (called the target node), and we create  $n$  error terms called  $u$  drawn from a normal distribution with a mean of 0 and a standard deviation of .5. We then create observational data in a following manner:

This document is copyrighted by the American Psychological Association or one of its allied publishers. This article is intended solely for the personal use of the individual user and is not to be disseminated broadly.

**Figure 10**  
Visualization of the Directed Acyclic Graph (DAG) Simulation



Note. The left panel depicts a graph after the initiation process, and the right panel depicts a graph after it satisfies all conditions.

$$y = bX_b + u \quad (14)$$

where  $b$  contains the coefficient for the selected target node, and  $X_b$  the column in  $X$  corresponding with the target node. The next step is to create perturbation data. For each node independently, we select all data from our original matrix  $X$ , excluding the target node. This  $n \times (p - 1)$  matrix is then multiplied with  $p - 1$  values (called  $a$ ) drawn from a normal distribution with a predetermined mean ( $\bar{m}$ ) and standard deviation ( $SD$ ), creating a new matrix  $X_{per}$ . We then create experimental data similar to the previous step:

$$y_{per} = b(aX_{per}) + u \quad (15)$$

We repeat this process for all nodes in the graph. We also simulate data in which we added hidden variables, following Peters et al. (2017). The equations for the simulation of data with hidden variables are similar to the regular data simulation (shown in (14) and (15)) with the following addition. We create hidden variables by drawing  $n$  values from a normal distribution (with a mean of 1 and a standard deviation of 1). These  $n$  values are then multiplied by a parameter  $h$  that we have set to be 5. We then take the outer product of  $h$  and a vector of 1s of length  $p$ , and add this to our matrix  $X$ . This matrix is then used in a similar manner as described above.

### Numerical Evaluation of Causal Inference Algorithms

In this simulation study we evaluated the performance of five methods of causal inference. We simulated six DAGs: three with  $p = 5$  nodes, and three with  $p = 10$  nodes. We varied the density of the graphs (the proportion of edges present in the graph)  $d \in \{.1, .25, .5\}$ . Figure 11 depicts all causal graphs that were created for this simulation study. We varied the number of participants  $n \in \{50, 100, 500, 1000; 5000\}$  and the mean of the perturbation distribution  $\bar{m} \in \{1, 5\}$ . These values correspond to a small and large perturbation effect. The standard deviation of the perturbation distribution varied  $SD \in \{.5, 5\}$ . These values correspond to an

effective and a noisy perturbation. For the DR-FFL and the TRANSWESD-algorithms, we varied  $\beta \in \{.5, 1, 1.64, 1.96, 2.58\}$ , and  $\gamma$  was set to 0. We simulated data with and without hidden variables to see how the addition of hidden variables affected the performance of the algorithms. We ran each simulation condition 100 times for each combination of parameters. We set the significance level for the PC, ICP, and HICP-algorithm to be .05. All simulated data, as well as the used R-code are publicly available at <https://osf.io/n8gxb/> and the [online supplemental materials](#).

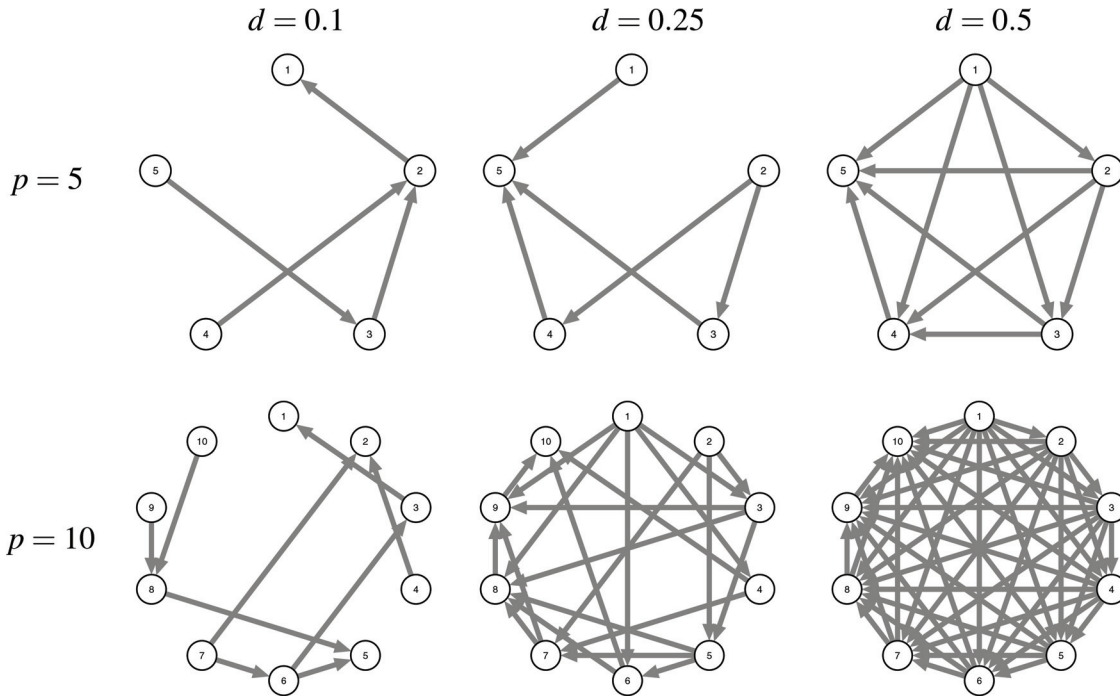
In the numerical evaluation we focus on *Matthew's correlation coefficient* (MCC; Powers, 2011). The MCC takes both true and false positives and negatives into account and gives a good overview of the overall performance of the different algorithms. The MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (16)$$

where  $TP$  represents the number of true positives,  $TN$  the number of true negatives,  $FP$  the number of false positives and  $FN$  the number of false negatives. The MCC can be interpreted the same way as a regular correlation coefficient (Matthews, 1975). The more positive the MCC, the better the correspondence between simulated and estimated edges. We also calculated other metrics (e.g., positive/negative predictive rate, false negative/positive rate), but we chose not to present these here. Results for all metrics can be found online at <https://osf.io/n8gxb/> and the [online supplemental materials](#).

Figure 12 shows the MCC for the different algorithms. For clarity of presentation, we only show results for  $p = 10$  nodes, with a graph density  $d = .25$ . All other results can be found online. Overall, the ICP and HICP-algorithms have the highest MCC. The MCC of the PC-algorithm is generally low. The PC-algorithm seems to benefit from a density that is not too high. The MCC increases when the graph density  $d$  increases from .1 to .25, but decreases again when  $d$  is increased to .5. Also, the size of the graph (reflected by  $p$ ) has an impact on the MCC: when  $p$  is increased from 5 to 10, the MCC decreases from on average .55 to

**Figure 11**  
Directed Acyclic Graphs (DAGs) That Were Used to Simulate Data

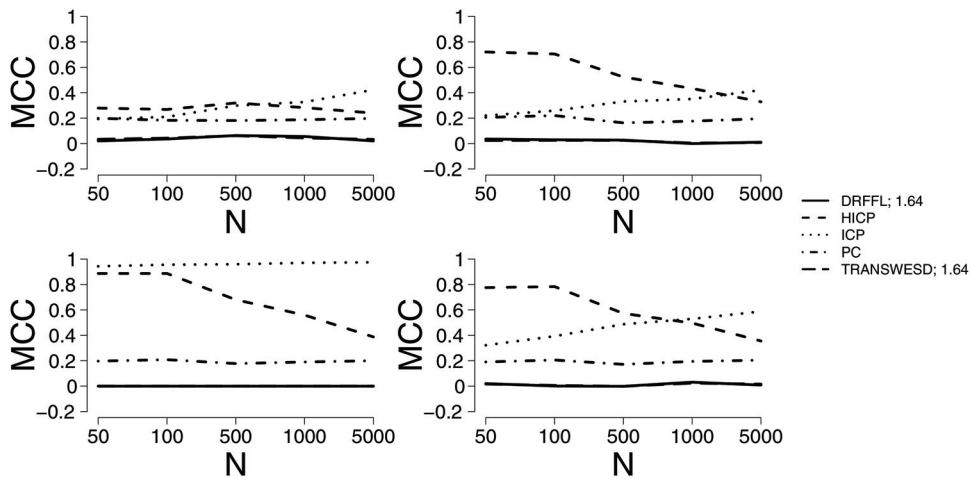


Note.  $p$  = number of nodes in the graph;  $d$  = the percentage of edges present in the graph.

.23. The PC-algorithm can have issues determining the direction of directed edges. In around 18.75% of the simulations, the PC-algorithm returned an undirected graph. It turns out that the MCC is almost always higher when we do not take the direction of the edges into account. This effect is especially present when the density of the

graphs is low. To illustrate, when  $d = .1$ , the average MCC increases from .47 to .87 when we do not take the direction of the edges into account, but when  $d = .5$ , the difference in average MCC is only .05 on average. These results indicate that the PC-algorithm can be useful in determining the pairs of variables between which a causal

**Figure 12**  
Matthew's Correlation Coefficient (MCC) for  $p = 10$  Nodes With a Network Density of  $d = 0.25$  When No Hidden Variables Were Simulated



Note. Top left =  $\bar{m} = 1, SD = 0.5$ , top right =  $\bar{m} = 1, SD = 5$ , bottom left =  $\bar{m} = 5, SD = 0.5$ , bottom right =  $\bar{m} = 5, SD = 5$ . DR-FFL= Down-Ranking of Feed-Forward Loops; HICP = Hidden Invariant Causal Prediction; ICP = Invariant Causal Prediction; PC = Peter and Clark; TRANSWESD = Transitive Reduction for Weighted Signed Digraphs.

relation exists, but that it may not be the appropriate algorithm to determine the direction of these causal relations.

In general, both the DR-FFL and the TRANSWESD-algorithms perform badly, as seen in Figure 12. Both the mean of the perturbation distribution ( $\bar{m}$ ) and its standard deviation ( $SD$ ) do not have an effect on their performances. When  $p = 5$ , both algorithms seem to perform a little better, but the difference in MCC is only about .1 between the two network sizes. The root cause of this seems to lie in the first step of the transitive reduction scheme that the two algorithms apply. Where on average each simulated graph had about 13.45 edges, both the DR-FFL and the TRANSWESD-algorithms returned on average 2.95. When there are only a few edges present in the graph, transitive reduction works sub optimally, as even fewer edges can be removed from the graph. Our findings indicate that transitive reduction may not be the best way to go when estimating a causal graph. We tried a different approach in the data simulation, but we found similar results. We also simulated data with an extremely large perturbation mean ( $\bar{m} = 100$ ), but the MCC hardly improved.

The ICP-algorithm seems to do a better job at correctly estimating the causal graphs in some cases. Overall, the ICP-algorithm works best at  $d = .25$ . The MCC when the network density is lower (or higher) is much lower. In general, when  $p = 5$ , the ICP-algorithm performs best (at  $d = .25$ ). In that situation, the mean and standard deviation of the perturbation distribution do not have an effect on the MCC. In contrast, when  $p = 10$ , the influence of these two parameters is much bigger. As shown in Figure 12, only when  $\bar{m} = 5$  and  $SD = .5$  is the MCC high, in all other cases it is mediocre at best. The ICP-algorithm can be conservative: of all the edges that it finds, the algorithm will only take the intersection as the (sub)set of causes of a target node. The results shown here suggest that the ICP-algorithm can estimate causal graphs pretty accurately when there are not too few or too many edges in the graph, and when the perturbation effect is strong and precise enough.

Lastly, the HICP-algorithm displays a mixed performance. When  $p = 5$ , we see a similar pattern as with the ICP-algorithm, but less extreme. With a graph density  $d = .25$ , the HICP-algorithm has a high MCC, but for the other two graph densities, the MCC is smaller. The declining effect that we observe in Figure 12 returns for other graph sizes and densities as well: when the sample size  $n$  increases, the MCC decreases. The cause of the decrease can be found in the number of false positives. Like we saw in the illustration in the previous section, the causal graph that is the result of the HICP-algorithm often contains spurious edges. This is likely due to the fact that the HICP-algorithm only investigates the entire set of remaining nodes in the graph to determine the causes of the target node, in contrast to the ICP-algorithm that investigates each possible subset separately. Because of this set-up, spurious edges can occur, which become significant more easily with a larger sample size. This indicates that the HICP-algorithm does not need a large sample to estimate a causal graph.

Figure 13 gives a more detailed insight into the strengths and weaknesses of each algorithm. To increase readability of the causal graphs, we chose to display the results for  $p = 5$  and  $d = .25$ . These graphs are publicly available for all simulation conditions. The number of false positives (red edges) for the DR-FFL, TRANSWESD, and HICP-algorithm immediately stand out. The HICP-algorithm stands out from the DR-FFL and the

TRANSWESD-algorithm because it also has a high number of true positives, indicated by the thickness and saturation of the blue edges. The conservativeness of the ICP-algorithm is less visible, but present nonetheless. Where the HICP and the PC-algorithm correctly identified the present edges (as shown in the true graph) in either 99–100% of the simulations, the ICP-algorithm's rate lies around 90–96%, which is still very high. The PC-algorithm's struggle with determining the direction of the edges is also depicted in Figure 13. The direction of two edges here ( $e_{42}$  and  $e_{32}$ ) are just as often correctly as incorrectly identified.

We also ran the simulation study using data that contained hidden variables. The results are very similar to the results using data without hidden variables. See Appendix D for the results using data with hidden variables.

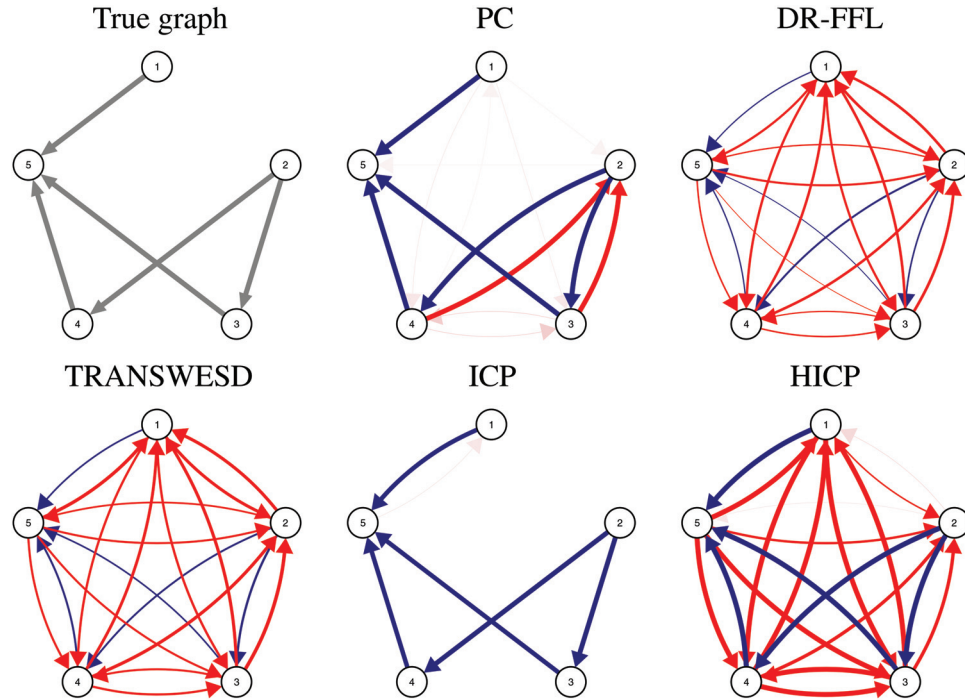
### Application to Empirical Perturbation Data

The dataset used here is a dataset collected by Hoekstra et al. (2018) and comprises 30 participants who completed a questionnaire that measures their attitude toward meat consumption. The questionnaire consists of 11 items that focused on the cognitive (six items) and affect (five items) side of one's attitude toward meat consumption. Responses were measured on a 6-point Likert scale ranging from 1 (*completely disagree*) to 6 (*completely agree*). The experiment started with the questionnaire serving as a baseline measure, after which a hypothetical scenario was presented intended to perturb a participant's response to a single item. For example, if a participant responded to the item "animals are inferior to humans" with "completely agree," this participant would get the hypothetical scenario "imagine animals are not inferior to humans. How does this influence your attitude toward the consumption of meat?" Participants then completed the same questionnaire. This process was repeated for every item in the questionnaire. This resulted in 12 questionnaires for each participant. All participants ( $N = 30$ ) were included in the analyses. We removed four missing measurements of three participants as one item was missing from these measurements. Participants had a mean age of 23.63 years ( $SD = 9.01$  years) and 70% identified themselves as female. Table 2 gives an overview of the items, including the means, standard deviation and the mean change between the baseline and the perturbation environment. All raw data, including the questionnaire and hypothetical scenarios, are published online (Hoekstra et al., 2018).

Similar to the simulation, we only used the observational data to run the PC-algorithm. In the illustration of the DR-FFL-algorithm, we averaged the data across the sample to create a between-subjects graph. We arbitrarily set  $\beta$  to be 1.3 for both the DR-FFL and the TRANSWESD-algorithm and  $\gamma$  to be 1. For the ICP and HICP-algorithms, we followed a similar approach to that used in Kossakowski et al. (2021). We compare every combination of environments (one observational and 11 perturbation environments, resulting in 66 pairs of environments) and combine these results into a single causal graph. The thicker and more saturated an edge between two nodes, the more often this causal relation is found and the more confident we can be in the true existence of that causal relation. For the PC, the ICP and the HICP-algorithms, we set  $\alpha = .01$ . Similar to Meinshausen et al. (2016), combining the separate graphs from the different combinations of interventions may result

**Figure 13**

Visualization of the Number of True Positives and False Positives for  $p = 5$ ,  $d = 0.25$ ,  $n = 5,000$ ,  $\bar{m} = 5SD = 0.5$ , and  $\beta = 0.5$  Without the Addition of Hidden Variables



Note. Blue edges indicate true positives, and red edges indicate false positives. The saturation and thickness of the edge represents how often that edge was (in)correctly estimated. Upper left = true graph; upper middle = Peter and Clark (PC); upper right = Down-Ranking of Feed-Forward Loops (DR-FFL); lower left = Transitive Reduction for Weighted Signed Digraphs (TRANSWESD); lower middle = Invariant Causal Prediction (ICP); lower right = HICP. See the online article for the color version of this figure.

in cycles. These cycles do not invalidate the causal interpretation only if the parameters in the cycle are not too large (i.e., the product of the parameters in the cycle is smaller than 1; see Rothenhäusler et al., 2015, for further details).

Figure 14 shows the results for each algorithm. We standardized the layout to improve visual inspection. The results follow the simulation results with a small sample size. The PC-algorithm only

found two edges, but could not determine their causal direction due to a lack of information that is needed for this; more edges are needed to determine the causal direction. The DR-FFL-algorithm removed four edges from the perturbation graph that is initially formed (using  $\beta = 1.3$ ), while the TRANSWESD-algorithm did not remove any edges (using  $\beta = 1.3$  and  $\gamma = 1$ ). The ICP-algorithm returned many edges due to the low number of participants.

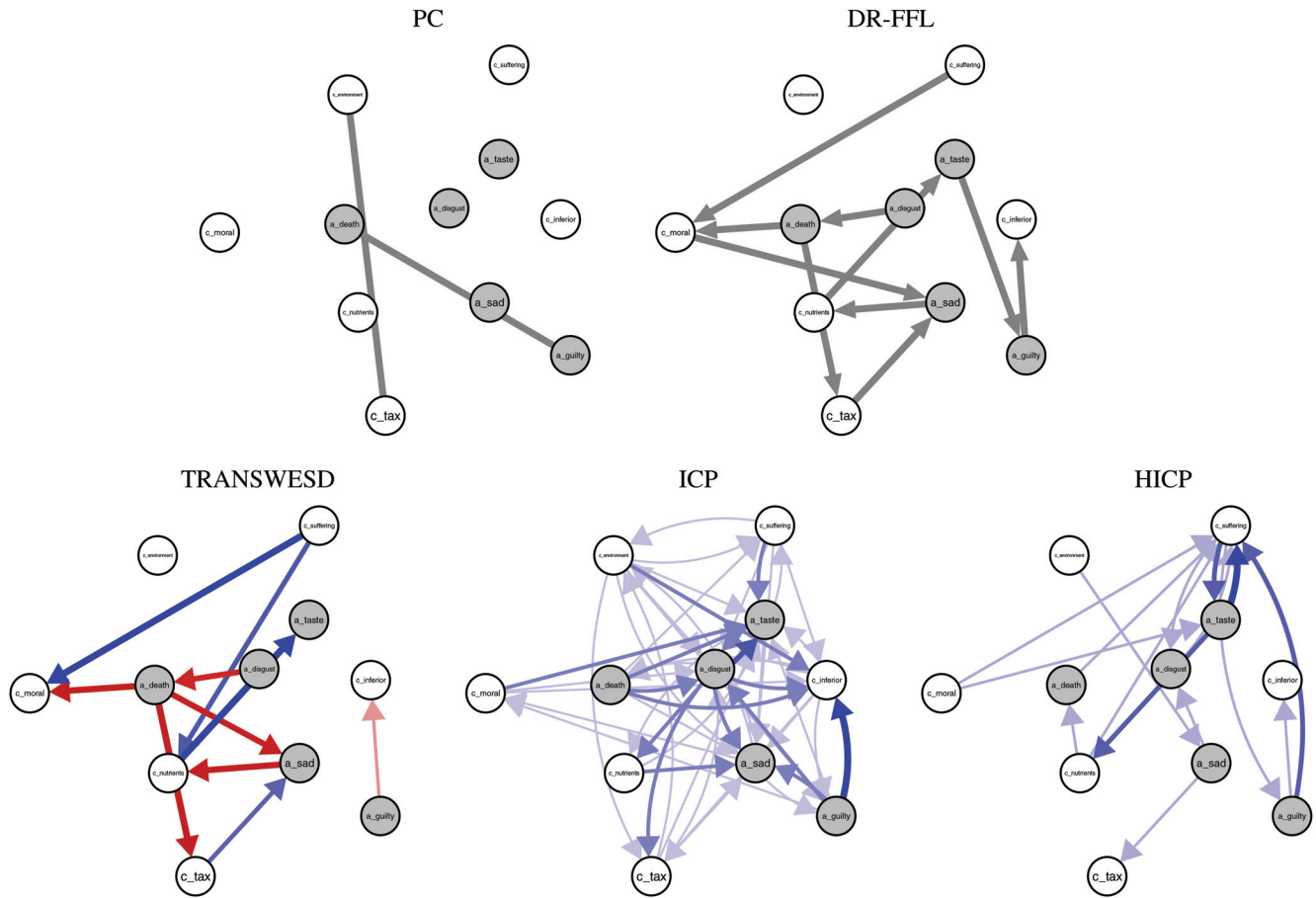
**Table 2**

Items of the Attitude Toward Meat Consumption Questionnaire With Their Assigned Item Label, Means (SD) Across Measurements, and  $M$  (SD) Changes Between the Baseline Measurement and the Perturbation Measurement

Item	Item label	$M$ (SD)	Mean perturbation effect (SD)
Eating is morally wrong	c_moral	3.73 (1.95)	-1.27 (3.73)
Meat contains important nutrients for your body	c_nutrients	3.66 (1.73)	0.43 (1.87)
The production of meat is harmful for the environment	c_environment	3.66 (1.83)	2.50 (1.76)
Animals are inferior to people	c_inferior	3.05 (1.89)	-0.83 (1.32)
By consuming meat you contribute to animal suffering	c_suffering	3.68 (1.79)	1.63 (1.88)
There should be a tax on meat	c_tax	3.65 (1.87)	1.70 (2.39)
I like the taste of meat	a_taste	3.65 (2.02)	3.90 (1.56)
Meat reminds me of death and suffering of animals	a_death	3.29 (1.88)	-1.50 (1.66)
If I had to stop eating meat I would feel sad	a_sad	3.86 (1.90)	-1.83 (2.79)
If I eat meat I feel guilty	a_guilty	3.38 (1.95)	-2.50 (2.32)
If I eat meat I feel disgust	a_disgust	3.12 (1.98)	-0.03 (1.3)

Note. The mean perturbation effect was measured by taking the mean of the difference between the baseline measurement and the measurement in which the specific item was perturbed.

**Figure 14**  
Results for Every Algorithm



*Note.* White nodes signify cognitive items, whereas gray items signify affective items regarding participants' attitude towards meat consumption. DR-FFL= Down-Ranking of Feed-Forward Loops; HICP = Hidden Invariant Causal Prediction; ICP = Invariant Causal Prediction; PC = Peter and Clark; TRANSWESD = Transitive Reduction for Weighted Signed Digraphs. See the online article for the color version of this figure.

The graph returned by the HICP-algorithm is very promising and interpretable. The edge *a\_taste* → *c\_suffering* is often found. This means that, when a participant's taste in meat changes, a change in their opinion on the contribution to animal suffering when eating meat changes as well. The reverse relation *c\_suffering* → *a\_taste* was found as well, but to a lesser extent. The results for the HICP-algorithm also appear to be quite stable. The graph does not change when changing  $\alpha$  from .05 to .01. All in all, this empirical example suggests that especially the HICP-algorithm is suitable for psychological data of this size.

## Discussion

The present study compared five different algorithms that are used for causal inference. We provided a simulation study in which we showed how well each algorithm is able to estimate the causal graph under which the data were simulated. We simulated data from causal graphs with different properties to assess the effect of the number of nodes and the density of the graph on the estimation of the graph itself. The results that we showed did not

present us with a clear winner: only under specific circumstances did each algorithm perform well. The exception to this are the DR-FFL and TRANSWESD-algorithms that never showed a good performance, contrary to earlier studies on these algorithms (Pinna et al., 2013). We also applied the various algorithms to empirical data that measured participants' attitude toward meat consumption. The HICP-algorithm seemed to perform best for this dataset, and provided a clear and interpretable graph.

Every algorithm that is discussed here had its own advantages and disadvantages. The PC-algorithm only uses observational data to estimate the causal relations between variables. Although this implies that less data are needed to run the algorithm, we did see that the MCC of the PC-algorithm is never higher than .8. Furthermore, we found that the PC-algorithm often has the location of a causal relation correctly identified, but not its direction. This may suggest that the PC-algorithm is useful to find the location of the causal relations, but that some other algorithm is needed to identify the direction of that relation.

It is quite noticeable that the DR-FFL and the TRANSWESD-algorithms do not perform as well as the ICP and HICP-algorithm.

In nearly 75% of all simulations 10% or less of the number of possible edges survived the first step of these algorithms. This is why the MCC for both the DR-FFL and the TRANSWESD is so low across simulation conditions.

We performed a similar simulation study where we only select 1 node (instead of  $p - 1$ ) that was perturbed and used to create our experimental data. However, results from this design are similar to the results presented here. We also varied the  $\beta$ ; and  $\gamma$ ; values to obtain better results, but there was no clear picture for the settings in our simulations. It is possible that a different data simulation design might give different results, but this remains to be investigated.

The ICP-algorithm has proven to be able to correctly identify edges that are present in the true graphs in our simulations. It also often makes the correct decision when edges are absent in the true graphs. However, the ICP-algorithm will only perform well when specific conditions are satisfied with respect to the data. The high MCC was obtained in this study when the mean of the perturbation distribution was strong, and its associated standard deviation small. In all other conditions, the MCC was mediocre and in some cases poor. The ICP-algorithm has a low false positive rate at the cost of a relatively low true positive rate. A conservative attitude is not necessarily a disadvantage of an algorithm, but when it is applied to empirical data, the resulting graph may be sparser than one would hope for when the sample size is large (see Kossakowski et al., 2021, for an example), or more dense when the sample size is small. Also, the ICP-algorithm investigates every possible subset of the variables that remain after selecting a target variable. This step leads to computational issues when graphs with a large number of nodes are studied. When a graph has  $p = 5$  nodes, the number of subsets per target variable is 16. When a graph has  $p = 10$  nodes, the number of subset grows to 516, and for  $p = 15$  nodes, the number of subsets equals 16,384. In the future we hope to develop an adaptation for the ICP-algorithm where a subset selection is made in such a way that the computation time decreases while maintaining similar specificity and sensitivity values.

Lastly, the HICP-algorithm outperforms the other algorithms in terms of the MCC in many simulation conditions. However, as is clearly shown in Figure 13, many edges are incorrectly seen as present (false positives; red edges) next to the correctly identified edges (true positives; blue edges). This phenomenon most likely occurs because not every possible subset is investigated separately, as is the case in the ICP-algorithm. This approach was implemented because of computational issues.

Both the ICP and the HICP-algorithm have a nodewise approach where each variable in the graph is the target node in turn. This implies that a reciprocal relation between two variables ( $X \rightarrow Y$  and  $Y \rightarrow X$ ) is possible, as each variable is the target node during the analyses. When one wants to estimate reciprocal causal relations, more environments are needed to accurately estimate these relations. We chose to exclude reciprocal causal relations from our study due to the fact that the other algorithms that are discussed are not able to estimate these. In future research, reciprocal causal relations could be investigated by adding them to the graph in a simulation study.

Next to the disadvantages that we discussed previously, we made some arbitrary decisions for the algorithms in this study. The PC, ICP, and HICP-algorithm all require a significance level that we set to be .05. The DR-FFL and the TRANSWESD-algorithms have one or two threshold parameters that need to be set before the analysis. We chose to use different values to evaluate the effect of these parameters. Results of our simulation study

showed that the value of these thresholds impact the MCC of the algorithm: the higher the threshold, the fewer edges are returned after the first step of these two algorithms and the fewer edges can be reduced from the graph. Choosing a value for these threshold parameters is not trivial. Ideally, one would want to set these parameters in such a way that the false and true positive rate are balanced. It remains that, with the DR-FFL and the TRANSWESD-algorithm, setting the threshold parameter(s) is no trivial matter and confounds the results tremendously. Future research could look into the possibility of using maximum likelihood estimation to obtain a reasonable threshold parameter based on the data.

We simulated data without and with the addition of hidden variables. Although they are present, it is hard to find differences between the results for data without, and data with hidden variables. Due to the high number of simulation conditions that we already have, we chose not to add any by varying the strength of the hidden variable. It is possible that a hidden variable with a stronger effect may result in larger differences in specificity and sensitivity. In a future extension of this study, one can vary the hidden variable to investigate the effect of a hidden variable on the results.

As every algorithm has its own advantages and disadvantages, a possible combination of two or more algorithm may be the solution. For instance, one can use the PC-algorithm to determine the skeleton of a causal graph, and use that input for the ICP-algorithm. In this combination, the number of subsets decreases substantially. Another option would be to copy the subset design of the ICP-algorithm, and use in with the HICP-algorithm. Investigating multiple subsets may result in a lower number of false positives and a more accurate depiction of the true causal graph.

The set of algorithms discussed in this study is not complete. Other algorithms exist in the literature that are potentially interesting to further explore. However, these algorithms do not combine observational and perturbation data but only use observational data. Algorithms that use observational data include a directional dependence model using copulas (Sungur, 2005), a linear causal acyclic model (Shimizu et al., 2006), or a directional dependence analysis with possible confounding variables (Wiedermann & Sebastian, 2019), and general cyclic linear models (Drton et al., 2019). One could even extend the existing simulation study to include algorithms that estimate causal relations between latent variables (e.g., Shimizu et al., 2009), or estimate nonlinear causal relations (e.g., Heinze-Deml et al., 2018) instead of just linear causal relations, or use those nonlinearity algorithms to resolve causal relations ( Mooij et al., 2011). The HICP-algorithm is a suitable option when variables and their residuals are correlated, which is an indication of hidden or confounding variables.

Measurement error will often, if not always be present in psychological research. In our simulation study we added measurement error to reflect this. Because the basic elements in our study were regressions, naturally, with increasing amounts of measurement error, the power will decrease. In our study, we found measurement error, but no systematic errors like spurious edges that appear in every simulation result. It is possible to add a SEM (latent variable) model to model the measurement error. This would entail several indicator variables per construct; thus, modeling the measurement error more explicitly, and then creating the network between the latent variables of the SEM model. A different option would be the model presented by Zhang et al. (2018) that can also manage measurement error.

To our knowledge, this is the first study that compared different algorithms for causal inference based on experimental data. Based on the simulation results, we gain more insight into the accuracy of each algorithm, and how suited they are for empirical (psychological) data. The ICP and HICP-algorithm are the top candidates to be used in psychological research. As hidden variables are a common problem in psychological research, a possible combination of the ICP and HICP-algorithm may be the best plan of attack to estimate causal relations between psychological variables. Results from the empirical application demonstrated that the ICP and the HICP-algorithms have the most potential to be suitable for psychological data.

## References

- Drton, M., & Richardson, T. (2004). Iterative conditional fitting for Gaussian Ancestral Graph models. *Proceedings of the 20th conference on uncertainty in artificial intelligence* (pp. 130–137). AUAI Press.
- Drton, M., Fox, C., & Wang, Y. S. (2019). Computation of maximum likelihood estimates in cyclic structural equation models. *The Annals of Statistics*, 47(2), 663–690. <https://doi.org/10.1214/17-AOS1602>
- Glymour, C., & Scheines, R. (1986). Causal modeling with the TETRAD program. *Synthese*, 68, 37–63.
- Granger, C. W. J. (1980). Testing for causality. *Journal of Economic Dynamics and Control*, 2, 329–352. [https://doi.org/10.1016/0165-1889\(80\)90069-X](https://doi.org/10.1016/0165-1889(80)90069-X)
- Hamaker, E. L., Kuiper, R. M., & Grasman, R. P. P. (2015). A critique of the cross-lagged panel model. *Psychological Methods*, 20(1), 102–116. <https://doi.org/10.1037/a0038889>
- Heinze-Deml, C., Peters, J., & Meinshausen, N. (2018). Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), Article 20170016. <https://doi.org/10.1515/jci-2017-0016>
- Hoekstra, R. H. A., Kossakowski, J. J., & van der Maas, H. L. J. (2018). Psychological perturbation data on attitudes towards the consumption of meat. *Journal of Open Psychology Data*, 6, 1–3. <https://doi.org/10.5334/jopd.37>
- Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, 81(396), 945–960. <https://doi.org/10.1080/01621459.1986.10478354>
- Kalisch, M., & Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8, 613–636.
- Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., & Bühlmann, P. (2012). Causal inference using graphical models with the R package *pcalg*. *Journal of Statistical Software*, 47(11), 1–26. <https://doi.org/10.18637/jss.v047.i11>
- Klamt, S., Flassig, R. J., & Sundmacher, K. (2010). TRANSWESD: Inferring cellular networks with transitive reduction. *Bioinformatics*, 26(17), 2160–2168. <https://doi.org/10.1093/bioinformatics/btq342>
- Kossakowski, J., van Oudheusden, L. B., McNally, R. J., Waldorp, L. J., Riemann, B. C., & van der Maas, H. L. J. (2021). *Introducing the causal graph approach to psychopathology: An illustration in patients with obsessive-compulsive disorder* [Manuscript in preparation]. <https://doi.org/10.31234/osf.io/ed2v5>
- Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta - Protein Structure*, 405(2), 442–451. [https://doi.org/10.1016/0005-2795\(75\)90109-9](https://doi.org/10.1016/0005-2795(75)90109-9)
- Meinshausen, N. (2018). *InvariantCausalPrediction: Invariant causal prediction* [Computer software manual]. <https://cran.r-project.org/package=InvariantCausalPrediction>
- Meinshausen, N., Hauser, A., Mooij, J. M., Peters, J., Versteeg, P., & Bühlmann, P. (2016). Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences of the United States of America*, 113(27), 7361–7368. <https://doi.org/10.1073/pnas.1510493113>
- Mooij, J. M., Janzing, D., Heskes, T., & Schölkopf, B. (2011). On causal discovery with cyclic additive noise models. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (pp. 539–647). MIT Press.
- Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., & Schölkopf, B. (2016). Distinguishing cause from effect using observational data: Methods and benchmarks. *Journal of Machine Learning Research*, 17, 1–102.
- Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press.
- Pearl, J. (2009). *Causality: Models, reasoning, and inference* (2nd ed.). Cambridge University Press.
- Pearl, J., & Mackenzie, D. (2018). *The book of why: The new science of cause and effect*. Basic Books.
- Pearl, J., & Verma, T. S. (1991). A theory of inferred causation. In J. Allen, R. Fikes, & E. Sandewall (Eds.), *Knowledge representation and reasoning: Proceedings of the second international conference* (pp. 441–452). Morgan Kaufmann Publishers.
- Peters, J., Bühlmann, P., & Meinshausen, N. (2016). Causal inference using invariant prediction: Identification and confidence intervals. *Royal Statistical Society*, 78, 947–1012. <https://doi.org/10.1111/rssb.12167>
- Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of causal inference: Foundations and learning algorithms*. MIT Press.
- Pinna, A., Heise, S., Flassig, R. J., de la Fuente, A., & Klamt, S. (2013). Reconstruction of large-scale regulatory networks based on perturbation graphs and transitive reduction: Improved methods and their evaluation. *BMC Systems Biology*, 7(1), 73. <https://doi.org/10.1186/1752-0509-7-73>
- Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2, 37–63.
- Rice, J. J., Tu, Y., & Stolovitzky, G. (2005). Reconstructing biological networks using conditional correlation analysis. *Bioinformatics*, 21(6), 765–773. <https://doi.org/10.1093/bioinformatics/bti064>
- Rothenhäusler, D., Heinze, C., Peters, J., & Meinshausen, N. (2015). Backshift: Learning causal cyclic graphs from unknown shift interventions. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* 28 (pp. 1513–1521). Curran Associates, Inc.
- Shimizu, S., Hoyer, P. O., & Hyvärinen, A. (2009). Estimation of linear non-Gaussian acyclic models for latent factors. *Neurocomputing*, 72(7–9), 2024–2027. <https://doi.org/10.1016/j.neucom.2008.11.018>
- Shimizu, S., Hoyer, P. O., Hyvärinen, A., & Kerminen, A. (2006). A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7, 2003–2030.
- Spirites, P., Glymour, C., & Scheines, R. (2000). Causation, prediction, and search. In T. M. Press (Ed.), *Computation, causation, and discovery* (2nd ed.). MIT Press. <https://doi.org/10.1002/sim.1415>
- Sungur, E. A. (2005). A note on directional dependence in regression setting. *Communications in Statistics - Theory and Methods*, 34(9–10), 1957–1965. <https://doi.org/10.1080/03610920500201228>
- Usami, S., Murayama, K., & Hamaker, E. L. (2019). A unified framework of longitudinal models to examine reciprocal relations. *Psychological Methods*, 24(5), 637–657. <https://doi.org/10.1037/met0000210>
- Wiedermann, W., & Sebastian, J. (2019). Direction dependence analysis in the presence of confounders: Applications to linear mediation models using observational data. *Multivariate Behavioral Research*, 55(4), 495–515. <https://doi.org/10.1080/00273171.2018.1528542>
- Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research*, 20, 557–585.

Zhang, K., Gong, M., Ramsey, J., Batmanghelich, K., Spirtes, P., & Glymour, C. (2018). Causal discovery with linear non-gaussian models under measurement error: Structural identifiability results. In A. Globerson & R. Silva (Eds.), *Uncertainty in artificial intelligence: Proceedings of the thirty-fourth conference* (pp. 1063–1073). AUAI Press.

Zyphur, M. J., Allison, P. D., Tay, L., Voelkle, M. C., Preacher, K. J., Zhang, Z., Hamaker, E. L., Shamsollahi, A., Pierides, D. C., Koval, P., & Diener, E. (2019). From data to causes I: Building a General Cross-Lagged Panel Model (GCLM). *Organizational Research Methods*, 23(4), 651–687. <https://doi.org/10.1177/1094428119847278>

## Appendix A

### D-Separation in a DAG

Earlier we described the four different causal structures that can exist in a causal graph (see Figure 1). By determining the conditional (in)dependencies between sets of variables, the PC-algorithm estimates a *directed acyclic graph* (DAG). The notion of conditional independencies can be extended to the idea of *directed (d) separation*. D-separation generalizes a separation in a graph between two variables. Consider the graph in Figure 1, where we have a path from variable  $X$  to variable  $Y$  (Figure 1, left panel). Variables  $X$  and  $Y$  are d-separated in graph  $G$  (denoted by  $X \perp\!\!\!\perp_G Y \mid Z$ ) given a variable  $Z$  if  $Z$  blocks the path from any node in  $X$  to any node in  $Y$ . D-separation is relatively easy in the case of a chain or a common cause structure (Figure 1, three left panels). With these specific structure,  $X$  and  $Y$  are d-separated given  $Z$  when  $Z$  is observed. When  $Z$  is observed, it “blocks” the path from  $X$  to  $Y$ . The reverse is true for the collider structure (Figure 1, right panel):  $X$  and  $Y$  are d-separated as long as  $Z$ , or any of its descendants are not conditioned on. For a disjoint set of random variables  $X$ ,  $Y$ , and  $Z$  with joint probability distribution  $\mathbb{P}$ , we note that  $X$  is conditionally independent of  $Y$  given  $Z$  by  $X \perp\!\!\!\perp_{\mathbb{P}} Y \mid Z$ . The notion of d-separation is used in the following assumption:

*Assumption 1: We assume that for disjoint sets of variables  $X$ ,  $Y$ , and  $Z$  the causal Markov condition is satisfied, which specifies that*

$$X \perp\!\!\!\perp_G Y \mid Z \Rightarrow X \perp\!\!\!\perp_{\mathbb{P}} Y \mid Z \quad (17)$$

This assumption guarantees that, when we find that two variables are d-separated, in the graph  $G$  these two variables are conditionally independent given a third variable in the probability distribution  $P$ . While the Markov assumption allows to move from the graph to the distribution, the reverse is not guaranteed. To be able to infer from conditional independencies obtained from the probability distribution that certain causal relations are valid, we also require that any conditional independence in the probability distribution implies a d-separation in the graph. This is entailed in the following assumption:

*Assumption 2: We assume that for disjoint sets of variables  $X$ ,  $Y$ , and  $Z$  the causal faithfulness condition is satisfied, which specifies that*

$$X \perp\!\!\!\perp_{\mathbb{P}} Y \mid Z \Rightarrow X \perp\!\!\!\perp_G Y \mid Z \quad (18)$$

This assumption ensures that, when two variables are conditionally independent given a third in the probability distribution  $P$ , they are also d-separated in the graph  $G$  given that third variable. The Markov and faithfulness assumptions allow for consistent inference for a causal graph. See Pearl (2009) or Peters et al. (2017) for more details.

(Appendices continue)

Appendix B

Theory of Transitive Reduction

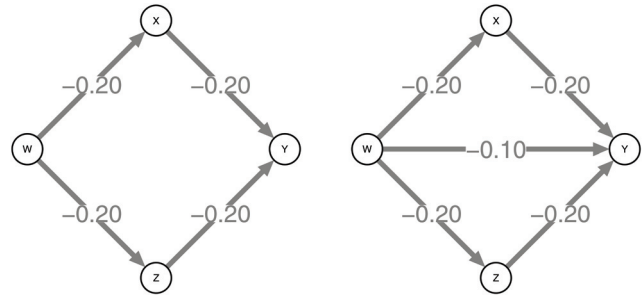
Both the DR-FFL and the TRANSWESD-algorithm use transitive reduction to estimate a causal graph. Both algorithms first draw up a perturbation graph in which causal relations between variables exist that exceed a prespecified threshold. Often (but not always), the correlation between two variables is used. The idea here is that, when a correlation between two variables is nonzero, then there must be either a direct or an indirect relation between these two variables. Transitive reduction aims to remove direct effects where there should not be one, by considering alternative paths between two variables.

To illustrate transitive reduction, we have set up two examples, visualized in Figure B1. Here, we consider the causal relation between variables  $W$  and  $Y$ . Wright (1921) showed that the correlation between  $W$  and  $Y$  is sum of the product of the path coefficients, denoted by  $\beta_{ij}$ . In the first example, shown in the left panel of Figure B1, two paths exist from  $W$  to  $Y$ :  $W \rightarrow X \rightarrow Y$  and  $W \rightarrow Z \rightarrow Y$ . The correlation  $\rho_{WY}$  then becomes  $(-.20) + (-.20)(-.20) = .08$ . The following criterion is used to remove a direct effect from the perturbation graph:

$$\min \left\{ \left| \rho_{WX_1}^{\{\emptyset, s\}} \right|, \dots, \left| \rho_{X_k Y}^{\{\emptyset, s\}} \right| \right\} > \left| \rho_{WY}^{\{\emptyset, s\}} \right| \tag{19}$$

where  $\{\emptyset, s\}$  denote the observational environment in which no perturbations have taken place ( $\emptyset$ ), and the experimental environment in which perturbations have taken place on variable  $s$ . The variables  $X_1 \dots X_k$  denote the variables that lie on the path from  $W$  to  $Y$ . In other words, (19) states that if the smallest absolute path coefficient is larger than the direct effect between two variables, then the direct effect is to be removed from the perturbation graph. In our illustration, the correlation between  $W$  and  $Y$  ( $\rho_{WY} = .08$ ) is smaller than the smallest path coefficient on either path (all path coefficients are  $.20$ ) and there

**Figure B1**  
Two Examples of Perturbation Graphs One for Which Transitive Reduction is Appropriate (Left Panel) and One for Which it is Not (Right Panel)



should not be a direct effect from  $W$  to  $Y$ . Therefore, the left panel of Figure B1 shows that transitive reduction is able to come to the right conclusion. However, the criterion in (19) is *necessary*, but it turns that that it is not sufficient to find the true causal graph. This is shown with the example in the right panel of Figure B1. Here, there is a direct effect from  $W$  to  $Y$ . Now  $\rho_{WY}$  becomes  $-.02$ , which is still smaller than the smallest path coefficient. Here, transitive reduction would erroneously remove the direct effect from  $W$  to  $Y$ . This shows that transitive reduction may not reach the correct causal graph, especially when the path coefficients are small. Specifically, the criterion in (19) will not work when the sum of the direct effect and  $\rho_{ij}$  is smaller than the smallest absolute path coefficient on any path between  $i$  and  $j$ .

(Appendices continue)

This document is copyrighted by the American Psychological Association or one of its allied publishers. This article is intended solely for the personal use of the individual user and is not to be disseminated broadly.

## Appendix C

## Formal Description of the HICP-Algorithm

The HICP-algorithm controls for hidden variables by using an instrumental variable  $Z$ . This instrumental variable cannot directly influence the target node  $Y$ , as shown in Figure 8. By using the instrumental variable  $Z$ , the regression of the target node onto the remaining variables will be split for the different time points, and the difference between these time points is used to estimate the causal effect. The causal effect from  $X$  to  $Y$ , denoted by  $\hat{\alpha}$ , is calculated as follows:

$$\hat{\alpha} = \frac{\text{cov}[X, Y]}{\text{var}[X]} = \alpha + \frac{\delta\gamma\text{var}[H]}{\text{var}[X]} \quad (20)$$

$$\begin{aligned} \hat{\alpha} &= \frac{\text{cov}[X, Y]}{\text{var}[X]} \\ &= \frac{\text{cov}[X, \alpha\beta Z + (\alpha\gamma + \delta)H]}{\text{var}[X]} \\ &= \frac{\alpha\beta\text{cov}[X, Z] + (\alpha\gamma + \delta)\text{cov}[X, H]}{\text{var}[X]} \\ &= \frac{\alpha\beta\text{cov}[\beta Z + \gamma H, Z] + (\alpha\gamma + \delta)\text{cov}[\beta Z + \gamma H, H]}{\text{var}[X]} \\ &= \frac{\alpha\beta^2\text{var}[Z] + \gamma(\alpha\gamma + \delta)\text{var}[H]}{\hat{\beta}\text{var}[Z] + \gamma^2\text{var}[H] + \text{var}[N_x]} \\ &= \frac{\alpha\beta^2 + \gamma^2\alpha + \delta\gamma}{\beta^2 + \gamma^2 + \sigma_x^2} \\ &= \frac{\alpha(\beta^2 + \gamma^2) + \delta\gamma}{(\beta^2 + \gamma^2)} \\ &= \alpha + \frac{\delta\gamma}{\text{var}[X]} \end{aligned} \quad (21)$$

where  $\alpha$ ,  $\beta$ ,  $\delta$ , and  $\gamma$  represent relations between  $X$ ,  $Y$ ,  $H$ , and  $Z$ . See Figure 8 for a visual representation. We can rewrite Equation (21) as follows:

$$\begin{aligned} \hat{\alpha} &= \frac{\text{cov}[X, Y]}{\text{var}[X]} \\ &= \frac{\text{cov}[X, \alpha X] + \text{cov}[X, \delta H]}{\text{var}[X]} \\ &= \frac{\alpha\text{var}[X] + \delta\gamma\text{var}[H]}{\text{var}[X]} \\ &= \alpha + \frac{\delta\gamma\text{var}[H]}{\text{var}[X]} \end{aligned} \quad (22)$$

where the term  $\frac{\delta\gamma\text{var}[H]}{\text{var}[X]}$  will be 0 when there are no hidden variables. To estimate  $\hat{\alpha}$ , the HICP-algorithm uses a two-step procedure. It first estimates  $\hat{\beta}$  (the effect from the instrumental variable  $Z$  to variable  $X$ ), after which  $\hat{\beta}$  is used to estimate  $\hat{\alpha}$ :

$$\begin{aligned} \hat{\alpha} &= \frac{\text{cov}[\hat{\beta}Z, Y]}{\hat{\beta}^2\text{var}[Z]} \\ &= \frac{\hat{\beta}\text{cov}[Z, \hat{\beta}Z + e]}{\hat{\beta}^2\text{var}[Z]} \\ &= \frac{\hat{\beta}^2\text{var}[Z]}{\hat{\beta}^2\text{var}[Z]} \end{aligned} \quad (23)$$

Note that the equations here are used after the target node is selected. The computational steps that are taken to estimate all the causal effects are described below for a target node. We programmed a function that repeats these steps for every variable in the data. The HICP-algorithm uses the instrumental variable to divide the data into two subsets. The first subset contains data from the first environment (often that part of the data in which no perturbation has taken place). The second subset consists of all the remaining data. We can rewrite Equation 21 to make it computationally appropriate:

$$\begin{aligned} \hat{\alpha} &= (X'X)^{-1}X'y \\ &= \left[ \begin{array}{cc} X'_1X_1 & X'_2X_2 \end{array} \right]^{-1} \cdot \left[ \begin{array}{c} X'_1Y_1 \\ X'_2Y_2 \end{array} \right] \\ &= \frac{\text{cov}[X_1, Y] - \text{cov}[X_2, Y]}{\text{var}[X_1] - \text{var}[X_2]} \end{aligned} \quad (24)$$

where  $X_1$  and  $X_2$  represent the predictor variables for the two environments, and  $Y_1$  and  $Y_2$  denote the scores on the target node for the two environments. The parameters  $n_1$  and  $n_2$  denote the number of participants that exist in the two environments. The result of Equation (24) is a  $p \times 1$  matrix that holds all the regression coefficients from every remaining node to the target node. After calculating  $\hat{\alpha}$  we proceed with the calculation of  $Z$ -values for all participants per environment:

$$Z_{i,\varepsilon} = -\tau \cdot \sum_{p=1}^p \tau \left[ \frac{X'_1Y_1}{n_1} - \frac{X'_2Y_2}{n_2} \right] + Y_{i,\varepsilon}\tau \quad (25)$$

where  $\tau = \left[ \frac{X'_1X_1}{n_1} - \frac{X'_2X_2}{n_2} \right]^{-1} \cdot X_{i,\varepsilon}$ . The parameter  $\tau$  is created for each participant  $i$  and environment  $\varepsilon$  individually. The matrix  $X_{i,\varepsilon}$  is a  $1 \times p$  vector that holds the observational data for participant  $i$  and  $p$  variables. Two separate  $n \times p$  matrices emerge from this equation: one for the first environment, and one for the second environment. The next step includes the calculation of  $\sigma$ :

(Appendices continue)

$$\sigma = \sqrt{\text{diag} \left( \frac{s^2(Z_1)}{n_1} + \frac{s^2(Z_2)}{n_2} \right)} \quad (26)$$

where  $s^2(Z_1)$  and  $s^2(Z_2)$  denote the covariance matrix of the Z-values that we calculated previously for environments 1 and 2, and  $n_1$  and  $n_2$  the number of participants in the first and second environment. The term *diag* here indicates that we only take the diagonal of the result of  $\frac{s^2(Z_1)}{n_1} + \frac{s^2(Z_2)}{n_2}$ . In the last step we calculate the  $p$ -values associated with  $\hat{\alpha}$ . These are calculated in the following manner:

$$p = \max \begin{cases} 2K \cdot 1 - t \left( |\hat{\beta}| / \max(10^{-10}, \sigma) \right) \\ 1 \end{cases} \quad (27)$$

where  $K$  is the number of environments (2 in this study). The parameter  $t()$  denotes the critical value in a  $t$ -distribution for a value of  $\frac{|\hat{\beta}|}{\max(10^{-10}, \sigma)}$ , with degrees of freedom  $n - 1$  (the total sample size). To estimate the maximal effect for each variable, we first determine the  $Z$ -value:

$$Z = \text{qnorm} [\max(0.5, 1 - \alpha/(2K))] \sigma \quad (28)$$

which is then used in combination with  $\hat{\alpha}$  to calculate the maximal effect:

$$\theta = (\hat{\beta}) \cdot \max(0, |\hat{\beta}| - Z) \quad (29)$$

The maximal effects for insignificant variables is set to be 0 due to the max term that exists in  $\theta$ .

*(Appendices continue)*

## Appendix D

## Numerical Evaluation of Causal Inference Algorithms With Hidden Variables

We also ran the simulation study using data that contained hidden variables. Figure D1 shows the MCC for the five algorithms that we investigated. The results are very similar to the results using data without hidden variables. For the PC-algorithm, we again see that the graph density  $d$  influences the MCC, where it reaches the highest numbers when  $d = .25$ . This effect only appears when the graph size  $p = 5$ . With  $p = 10$ , the MCC is generally low (average MCC = .09) at  $d = .5$  and will only increase to mediocre (average MCC = .56) values when  $d = .1$ .

The picture we painted earlier for the DR-FFL and the TRANSWESD-algorithms does not improve when hidden variables are included. When the graph size  $p = 5$ , the average MCC lies around .15, whereas when  $p = 10$ , the average MCC is around .03. The sample size  $n$  does not seem to influence the performance of both algorithms. On the other hand, the threshold parameter  $\beta$  has a big impact. The lower  $\beta$ , the higher the MCC is. To illustrate, when  $\beta = .5$ , the average MCC is .50, whereas when  $\beta = 2.58$ , the average MCC is close to zero. The threshold parameter  $\beta$  influences how many edges are retained after the first step in both the DR-FFL and the TRANSWESD-algorithm. The higher the threshold, the lower the number of edges that are present in the perturbation graph and the lower the MCC.

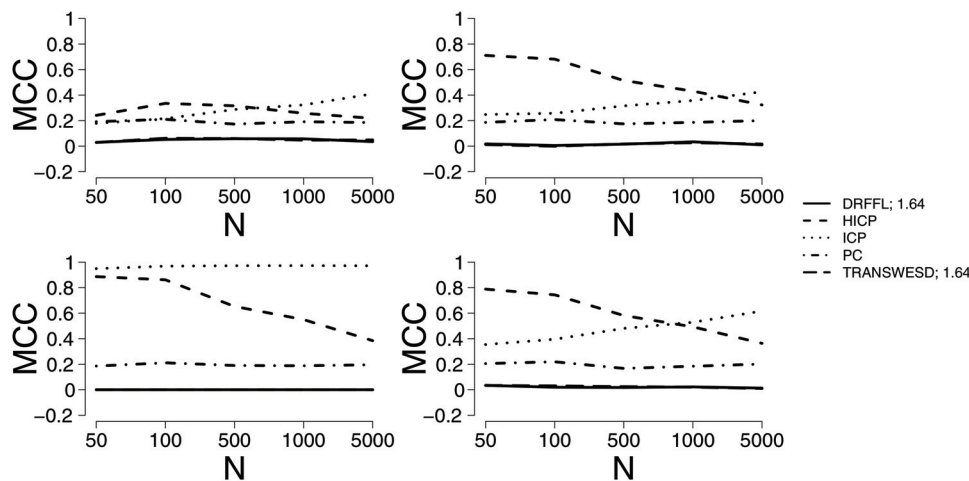
The ICP-algorithm has the best performance when there is a medium number of edges in the graph ( $d = .25$ ). As we saw before, we observe high MCC values with the smaller graphs ( $p = 5$ ). When  $p$  is increased to 10, the ICP-algorithm becomes

more conservative, resulting in a lower MCC. Only when the mean of the perturbation distribution ( $\bar{m}$ ) is high and the standard deviation small can the ICP-algorithm accurately estimate causal graphs. This indicates that the ICP-algorithm needs a strong and effective perturbation to correctly identify causal relations.

The mixed performance that we saw earlier with respect to the HICP-algorithm is also present when we add hidden variables to the data. This means that the HICP-algorithm can accurately estimate causal graphs with a small sample size. When the sample size increases, the accuracy decreases. This effect is present in almost every simulation condition. The only exception is when the graph density is low ( $d = .1$ ). In that case, the MCC increases when the sample size increases.

Figure D2 paints a similar picture that we saw in the previous section. The lack of accuracy of the DR-FFL and the TRANSWESD-algorithm is clearly visible, as are the spurious edges that are estimated by the HICP with a large sample size. Even though hidden variables are added to these data, the ICP-algorithm shows the highest number of true positives, combined with the lowest number of false positives for this simulation condition. Lastly, the PC-algorithm can have issues with determining the direction of an edge. This problem emerges independent of the presence of hidden variables, as we have seen

**Figure D1**  
Matthew's Correlation Coefficient (MCC) for  $p = 10$  Nodes With a Network Density of  $d = 0.25$   
With the Addition of Hidden Variables

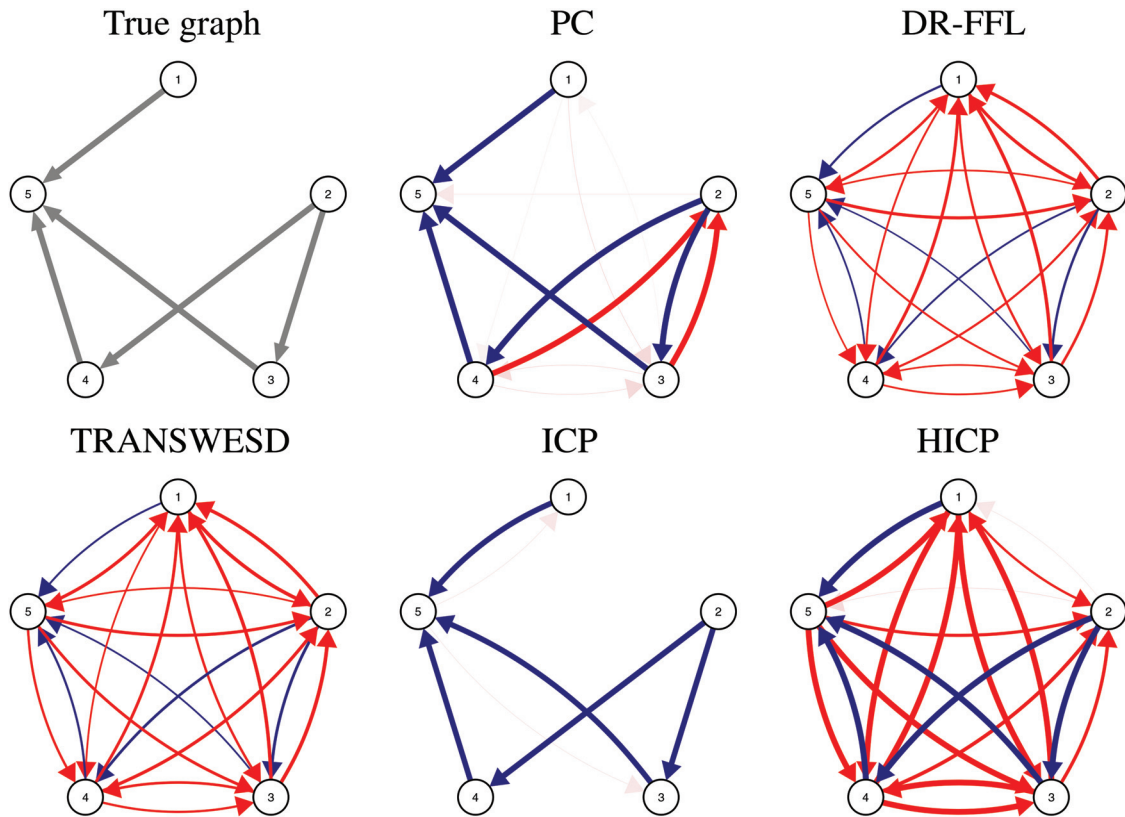


Note. Top left =  $\bar{m} = 1$ ,  $SD = 0.5$ , top right =  $\bar{m} = 1$ ,  $SD = 5$ , bottom left =  $\bar{m} = 5$ ,  $SD = 0.5$ , bottom right =  $\bar{m} = 5$ ,  $SD = 5$ . DR-FFL= Down-Ranking of Feed-Forward Loops; HICP = Hidden Invariant Causal Prediction; ICP = Invariant Causal Prediction; PC = Peter and Clark; TRANSWESD = Transitive Reduction for Weighted Signed Digraphs.

(Appendices continue)

**Figure D2**

Visualization of the Number of True Positives and False Positives for  $p = 5$ ,  $d = 0.25$ ,  $n = 5,000$ ,  $\bar{m} = 5$ ,  $SD = 0.5$ , and  $\beta = 0.5$  With the Addition of Hidden Variables



*Note.* Blue edges indicate true positives, and red edges indicate false negatives. The saturation and thickness of the edge represents how often that edge was (in)correctly estimated. Upper left = true graph; upper middle = Peter and Clark (PC); upper right = Down-Ranking of Feed-Forward Loops (DR-FFL); lower left = Transitive Reduction for Weighted Signed Digraphs (TRANSWESD); lower middle = Invariant Causal Prediction (ICP); lower right = Hidden Invariant Causal Prediction (HICP). See the online article for the color version of this figure.

this in the previous section as well. All in all, these results may suggest that the ICP-algorithm is the saver option when one wants to estimate a causal graph.

Received August 27, 2019

Revision received September 6, 2020

Accepted December 4, 2020 ■