Are You Doing Better Than Random Guessing? A Call for Using Negative Controls When Evaluating Causal Discovery Algorithms

Anne Helby Petersen[1]

[1]Section of Biostatistics, Department of Public Health, University of Copenhagen, Copenhagen, Denmark

Abstract

New proposals for causal discovery algorithms are typically evaluated using simulations and a few selected real data examples with known data generating mechanisms. However, there does not exist a general guideline for how such evaluation studies should be designed, and therefore, comparing results across different studies can be difficult. In this article, we propose to use negative controls as a common evaluation baseline by posing the question: Are we doing better than random guessing? For the task of graph skeleton estimation, we derive exact distributional results under random guessing for the expected behavior of a range of typical causal discovery evaluation metrics, including precision and recall. We show that these metrics can achieve very favorable values under random guessing in certain scenarios, and hence warn against using them without also reporting negative control results, i.e., performance under random guessing. We also propose an exact test of overall skeleton fit, and showcase its use on a real data application. Finally, we propose a general pipeline for using negative controls beyond the skeleton estimation task, and apply it both in a simulated example and a real data application.

1 INTRODUCTION

Causal discovery algorithms seek to infer information about a causal data generating mechanism by analyzing empirical data it generated. The causal data generating mechanism is typically represented by a causal graph, for example an equivalence class of directed acyclic graphs (DAGs). A highly productive research community has published a plethora of new causal discovery algorithms within the last 30 years or so. Naturally, this fast growing battery of available algorithms requires some standards and guidelines for evaluating and benchmarking their performance. Because the result of a causal discovery algorithm is an estimated graph (or family of graphs), rather than one or more scalars, it is not entirely obvious how to use classic approaches for performance evaluation from neither machine learning nor statistics.

Nonetheless, machine learning classification metrics originally developed for evaluating prediction tasks are often used to evaluate causal discovery algorithms. Most commonly, precision and recall, or possibly their harmonic mean, the F1 score, are reported, although some studies also focus on other metrics, e.g., negative predictive value [Petersen et al., 2023b]. These metrics are computed from graph-level confusion matrices summarizing either agreement on placement of oriented edges (primarily used for DAG discovery evaluation), adjacencies (i.e., edge placement without considering orientation), and/or arrowheads among correctly placed adjacencies (conditional orientation). Typically, they are reported as averages over numerous simulations. Sometimes the results are stratified by graphical parameters (e.g., true graph density), data-related parameters (e.g., sample size), or simply reported as averages across several such settings.

Alternative metrics developed specifically for graphs also exist; the structural Hamming distance [Tsamardinos et al., 2006] is the most widely used example, probably due to its cheap computation and easy interpretation. A different metric focusing more on the causal implications of the graphs is the structural intervention distance (SID) [Peters and Bühlmann, 2015], although it is most naturally suited for DAG-DAG comparison, and hence not readily applicable for all discovery evaluation tasks. A more recent proposal is the adjustment identification distance [Henckel et al., 2024], which also focuses on differences in causal inference based on the graph, or the separation distance [Wahl and Runge, 2025], which counts agreement in separation statements. There are thus many different possible choices of metrics for evaluating causal discovery algorithms. However, no

general guidelines exist on how to then *interpret* the values these metrics take: What is a high or low number?

An often-used strategy for answering this question in experimental sciences is to conduct a controlled experiment. In such an experiment, the intervention of interest (here: a causal discovery algorithm) is compared to a control condition. When this control condition cannot have any influence on the outcome of interest, it is denoted a *negative control*. For example, say we want to study the impact of a fertilizer on plant growth. We plant 100 seeds in two plots with similar conditions, except that one (the treatment group) receives the fertilizer, while the other (the negative control) does not. After, say, 10 days, we measure the heights of the plants, and we use the average difference in heights as a measure of the effect of the fertilizer. By including the negative control we obtain a direct measure of the specific effect of the treatment.

Alternatively, we could have compared the fertilizer of interest with another active treatment, perhaps an alternative well-known fertilizer (denoted a *positive control*). This comparison gives less information about the specific treatment of interest. For example, we cannot falsify a hypothesis saying that neither fertilizer has any effect. In some scientific fields, for example human drug trials, using positive controls is the only viable option, as it would be utterly unethical to deny patients treatment (if one exists) in order to obtain a negative control group for evaluating a new proposed treatment. But of course, no such concerns are relevant for evaluating causal discovery algorithms. Nonetheless, the current standard practice is to report positively controlled experiments: A new candidate algorithm is typically compared to a selection of existing algorithms. Such a comparison in itself does not provide information about whether *either* of the considered algorithms work. Moreover, because causal discovery evaluations are very sensitive towards experimental settings concerning graph sparsity (as we will demonstrate below), it is not straight-forward to generalize findings from such a positively controlled experiment to infer what performance should be expected on just slightly different evaluation settings. This makes it very difficult to compare results across different evaluation studies with just marginally different designs.

We propose to use negative controls to obtain an interpretable benchmark for any causal discovery evaluation study: Namely, to investigate what values of the metrics of interest can be obtained using random guessing (a negative control), and report this alongside findings from positive controls (alternative algorithms). Others have reported results from a single random guess alongside causal discovery benchmarks [Lachapelle et al., 2020], but as we will argue, a more structured approach to including negative controls provides many benefits. We discuss negative controls in two different settings: First, we consider the task of skeleton estimation, that is estimating e.g., a DAG without taking orientation information into account. For this case, we derive exact distributional results for the expected behavior under random guessing (Section 3), and we use these results to compute expected negative control values for a range of often-used metrics (Section 4). We furthermore propose an exact test of overall skeleton fit (Section 5), and provide an example of its use on real data. Secondly, we consider more general metrics that are not only concerned with skeleton estimation, and propose a negative control pipeline for this case (Section 6). We provide two examples of its use, both in a simulation study and in a real data application.

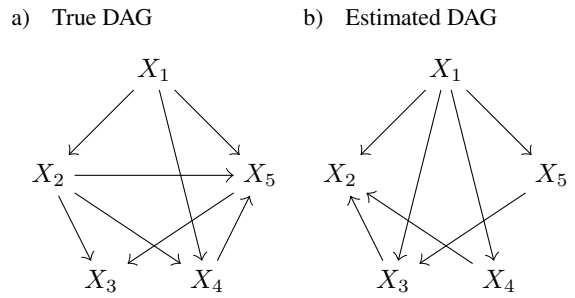

Figure 1: The True Underlying DAG (a) and an Estimated DAG (b) Obtained Using an Undisclosed Causal Discovery Procedure.

Table 1: Adjacency Confusion Matrix for the 5 Node DAG Example in Figure 1.

		Truth	
		Adjacency	Non-adjacency
Estimate	Adjacency	$tp = 6$	$fp = 1$
	Non-adjacency	$fn = 2$	$tn = 1$

But before we turn to these general results, we present an example case where well-known metrics such as adjacency precision and recall do perhaps not behave exactly as one would have expected.

Code for all computations is available online at `https://github.com/annennenne/negcontrol-disco`.

2 PRECISION AND RECALL: A CAUTIONARY TALE

Consider the two DAGs in Figure 1. The left graph (a) is the true DAG, and the right graph (b) is an estimate produced by a causal discovery procedure. We compute their adjacency confusion matrix in order to evaluate the performance of the

Table 2: Generically Labelled Adjacency Confusion Matrix.

		Truth		
		Adjacency	Non-adjacency	Total
Est.	Adjacency	*TP*	*FP*	m_{est}
	Non-adjacency	*FN*	*TN*	-
	Total	m_{true}	-	m_{max}

Notes: *Entries marked with dashes are sums that will not be used for the derivations here. "Est." abbreviates estimate.*

discovery procedure (Table 1). This results in:

$$\text{precision} = \frac{tp}{tp+fp} = \frac{6}{7} \simeq 0.86 \qquad \text{and}$$
$$\text{recall} = \frac{tp}{tp+fn} = \frac{6}{8} = 0.75.$$

Are these numbers high or low? Although these values are not too far off from the performance of well-established causal discovery algorithms on simulated data (and much better than typical performance on "real" benchmarking datasets), we will argue that they are indeed as low as can be for this specific discovery task — because the "discovery algorithm" applied here was simply random guessing and hence had absolutely no information about the true data generating mechanism.

More specifically, we simulated 1000 random Erdős-Rényi type DAGs over 5 nodes each with 7 edges[1] and used these "random guesses" as estimates of the DAG in Figure 1 (a). This resulted in a median precision of 0.86 and a median recall of 0.75, i.e., numbers that exactly match the performance of the example just described. The DAG shown in Figure 1 (b) was one among many random draws that matches this median performance. Hence the large values of precision and recall cannot be attributed to a conveniently chosen random seed.

Is it then a curious artefact for very dense graphs? Or "small" graphs over e.g., 5 nodes? Neither is the case. As we will show in the following section, the phenomenon does not depend on the number of nodes, and depending on the choice of metrics, can occur also in modestly dense graphs.

3 DISTRIBUTIONAL RESULTS FOR ADJACENCY METRICS UNDER RANDOM GUESSING

Consider a DAG G over d nodes, and let m_{true} denote the number of edges in G. Let $\hat{G}$ be another DAG over the same d nodes used as an estimate of G, and let m_{est} be the number of edges in $\hat{G}$. Finally, let $m_{\text{max}} = \sum_{i=1}^{d-1} i = \frac{1}{2}(d-1)d$ denote the maximal number of possible edges in a DAG over d nodes (corresponding to a fully connected graph). We can describe the performance of $\hat{G}$ as an adjacency/skeleton estimator of G through a (generic) confusion matrix as seen in Table 2. Note that for a given causal discovery problem, namely estimating some given G, m_{max} and m_{true} can be considered fixed: m_{max} depends only on the number of nodes d, which does not change, and m_{true} is fixed given G. Moreover, for many causal discovery procedures, it further makes sense to consider m_{est} fixed — at least for a specific value of a tuning parameter (e.g., significance level for testing or penalty for a score) and a specific dataset — as we most often do not try to *estimate* the correct number of edges from data. Rather, algorithms are typically applied with a pre-specified value of the tuning parameter (e.g. significance level of $\alpha = 0.01$), which indirectly controls the resulting number of edges, m_{est} (see Sections 6 and 7 for considerations in cases where this latter assumption is not meaningful).

We now make the following important observation: If edges are placed uniformly in both G and $\hat{G}$ (corresponding to Erdős-Rényi type graphs), and we condition on the row and column sums of Table 2 (which is equivalent with conditioning on m_{true}, m_{est} and m_{max}), then by definition, the number of true positives will follow a hypergeometric distribution parameterized by m_{max}, m_{true} and m_{est}:

$$TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}} \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}}).$$

Note that this is an exact distributional result, not an asymptotic statement. This distributional result is well known from its use in the (one-sided) Fisher's exact test, and may also be motivated using a random urn experiment analogy, see Supplementary Materials A.

This observation gives rise to several useful applications: First, we can compute the expected value, median and uncertainty estimates (e.g., confidence interval) for the number of true positive adjacencies under random guessing. Secondly, since we are also conditioning on m_{max}, m_{true} and m_{est}, we can further compute expectations and draw statistical inference for any function of the confusion matrix, including precision, recall and F1. We provide formulas for these in Section 4. Thirdly, we can construct an exact statistical test of overall skeleton fit by considering how much the number of true positives in a given estimated graph diverts from its expected distribution under a null hypothesis of random edge placement. We propose such a test in Section 5.

4 EXPECTATIONS AND QUANTILES OF ADJACENCY METRICS UNDER RANDOM GUESSING

Since

$$TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}} \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}}),$$

[1]Note that this is not even the correct number of edges, although close to it, as the true DAG has 8 edges.

Table 3: Expected Values and Quantile Expressions Under Random Guessing for Five Commonly Used Adjacency Metrics.

Metric	Expected value	Quantile
Precision	$\frac{m_{\text{true}}}{m_{\text{max}}}$	$\frac{q_k}{m_{\text{est}}}$
Recall	$\frac{m_{\text{est}}}{m_{\text{max}}}$	$\frac{q_k}{m_{\text{true}}}$
F1	$\frac{2 \cdot m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}} \cdot m_{\text{est}} + m_{\text{max}} \cdot m_{\text{true}}}$	$\frac{2 \cdot q_k}{m_{\text{est}} + m_{\text{true}}}$
NPV	$1 - \frac{m_{\text{true}}}{m_{\text{max}}}$	$\frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + q_k}{m_{\text{max}} - m_{\text{est}}}$
Specificity	$1 - \frac{m_{\text{est}}}{m_{\text{max}}}$	$\frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + q_k}{m_{\text{max}} - m_{\text{true}}}$

Note: q_k *denotes the kth quantile from HyperGeom*$(m_{max}, m_{true}, m_{est})$.

by definition we have that

$$\mathrm{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) = \frac{m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}}}$$

and by considering the quantile function of $\text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, we can construct a confidence interval as e.g., the central 95% of the distribution, or find the expected median.

For fixed values of $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, Table 3 provides an overview of expected values and quantiles under random guessing for five metrics commonly used for evaluating adjacency placement for causal discovery algorithms, namely precision, recall, F1 score, negative predictive value (NPV) and specificity. As an example, we here showcase derivations for precision, and refer to Supplementary Materials B for derivations for the remaining four metrics.

Expectation and quantiles for adjacency precision We first express precision as a function of TP, m_{max}, m_{true} and m_{est}:

$$\text{prec} = \frac{TP}{TP + FP} = \frac{TP}{TP + m_{\text{est}} - TP} = \frac{TP}{m_{\text{est}}}$$

Since this is a linear function of TP, we can straightforwardly compute the expectation:

$$\begin{aligned}\mathrm{E}(\text{prec} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) &= \frac{1}{m_{\text{est}}} \mathrm{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) \\ &= \frac{m_{\text{true}}}{m_{\text{max}}}\end{aligned}$$

The linearity also makes it easy to obtain e.g., an exact 95% confidence interval under the null hypothesis of random guessing by simply applying the same transformation to the appropriate quantiles of $\text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$. For example, an exact 95% confidence interval for precision under the null is given by

$$\left(\frac{1}{m_{\text{est}}} q_{(0.025, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}, \frac{1}{m_{\text{est}}} q_{(0.975, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})} \right)$$

where $q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}$ is used to denote the kth quantile of the probability mass function of $\text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$. Similarly, we obtain the median precision by simply computing

$$\text{median}(\text{prec}) = \frac{1}{m_{\text{est}}} q_{(0.5, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}.$$

General remarks concerning Table 3 A notable feature of Table 3 is that, conditional on $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, the expected precision is simply the density of the true DAG G, and the expected recall is the density of the estimated DAG $\hat{G}$. Furthermore, the expected values of NPV and specificity are given as 1 minus the expectations of precision and recall, respectively, and hence they do not provide additional information. However, without random guessing, this is of course not generally the case, so they are still useful to compute in order to provide a nuanced and multifaceted evaluation of a given causal discovery procedure.

Moreover, we note that under random guessing, the expected precision does not depend on the number of edges in the estimated graph (m_{est}), only on the number of edges in the true graph (m_{true}) and the maximal possible number of edges (m_{max}). But recall increases linearly as a function of the number of estimated edges. Hence, if we are using random guessing, a "free lunch" in optimizing precision and recall is achievable simply by estimating a very large number of edges, even including the trivial fully connected graph. This can also be seen from the expected value of the F1 score under random guessing, which increases monotonically with the number of estimated edges: It is always better to just add another edge.

Note that all statistical inference based on Table 3 relies on a single distributional result. Hence, while we may construct confidence intervals for e.g., both precision and recall to aid our own interpretation of a specific estimation problem, there is no additional statistical information gained: We might as well just draw inference directly on the number of true positives.

We will now consider two small example applications of the results from Table 3. First, we revisit the example from Section 2 and compute the median, expected value, and a 95% confidence interval for precision and recall for this case. Next, we provide an overview of how the expected F1 score varies as a function of m_{est} and m_{true} under random guessing.

4.1 EXAMPLE: EXPECTED ADJACENCY PRECISION AND RECALL FOR A DENSE 5 NODE DAG

Consider the problem from Section 2 regarding estimating a DAG skeleton over 5 nodes. Such a DAG can have at most $m_{\text{max}} = \frac{1}{2}(5-1)5 = 10$ edges. Assume that the true DAG

has $m_{\text{true}} = 8$ edges, while a randomly drawn graph over the same 5 nodes has $m_{\text{est}} = 7$ edges. What performance can we then expect from this random guessing procedure? With reference to Table 3, we find

$$\mathrm{E}(\text{prec} \mid m_{\text{max}} = 10, m_{\text{true}} = 8, m_{\text{est}} = 7) = \frac{8}{10} = 0.80$$

with a 95% confidence interval of

$$\left(\frac{q_{(0.025,10,8,7)}}{7}, \frac{q_{(0.975,10,8,7)}}{7}\right) = \left(\frac{5}{7}, \frac{7}{7}\right) \simeq (0.71, 1.00).$$

Hence for this DAG estimation task, it will not be highly unusual to obtain adjacency precisions as high as 1.00 under random guessing, and thus adjacency precision is not very useful for assessing performance. We can also compute the median precision:

$$\text{median}(\text{prec}) = \frac{q_{(0.5,10,8,7)}}{7} = \frac{6}{7} \simeq 0.86$$

This is the same value as found in the simulations presented in Section 2.

For adjacency recall we find

$$\mathrm{E}(\text{recall} \mid m_{\text{max}} = 10, m_{\text{true}} = 8, m_{\text{est}} = 7) = \frac{7}{10} = 0.70$$

and we compute a 95% confidence interval as

$$\left(\frac{q_{0.025,10,8,7}}{8}, \frac{q_{0.975,10,8,7}}{8}\right) = \left(\frac{5}{8}, \frac{7}{8}\right) \simeq (0.63, 0.88).$$

One is not included in this confidence interval and hence adjacency recall does have some discriminatory power for this DAG estimation task. We compute the median:

$$\text{median}(\text{recall}) = \frac{q_{(0.5,10,8,7)}}{8} = \frac{6}{8} = 0.75$$

Once again, this matches our simulation-based findings from Section 2.

4.2 EXAMPLE: ADJACENCY F1 SCORES FOR A 5 NODE DAG WITH VARYING DENSITY

Figure 2 provides an overview of obtained F1 scores under random guessing across all possible combinations of estimated number of edges (horizontal axis) and true number of edges (marked in color) for 5 node DAGs. We see that it is quite possible to obtain a large F1 score by random guessing if the true DAG is not very sparse, and especially, if the estimate is also not very sparse. But we stress that neither has to be overly or unrealistically dense either: For a true graph that has just 5 edges — i.e., only one edge more than the sparsest graph that is connected — a randomly drawn DAG with 5 edges will result in an expected F1 of 0.5, and placing all 10 possible edges results in an F1 score of 0.66. If we instead consider a more dense graph, e.g., a true DAG with 8 edges, we are back in the scenario already considered above, and we see that we can find a peak F1 score of 0.89 by placing all edges.

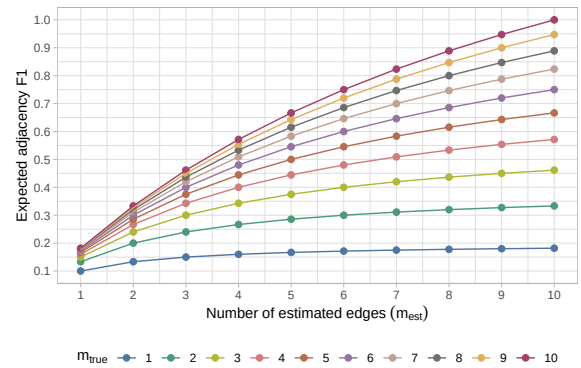

Figure 2: Expected Adjacency F1 Scores Under Random Guessing for Estimating 5 Node DAGs.

Table 4: Adjacency Confusion Matrix Replicated From Petersen et al. [2023a].

		Experts	
		Adjacency	Non-adjacency
TPC	Adjacency	10	20
	Non-adjacency	20	181

5 A TEST OF OVERALL SKELETON FIT

We can also use the distributional results presented above to construct an exact test of overall skeleton fit. More specifically, for an estimated DAG $\hat{G}$ with m_{est} edges, we test the null hypothesis

$$H_0 : \hat{G} \text{ was obtained by randomly placing } m_{\text{est}} \text{ edges.}$$

This is done by comparing the observed number of true positives, tp_{obs} with the appropriate hypergeometric distribution. Formally, if we let $X \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, a one-sided p-value for H_0 is computed as

$$P(X \geq tp_{\text{obs}})$$

i.e., the probability of getting at least as many true positives as the observed number if edges were in fact randomly placed. Note that since the test is exact (and based on a discrete probability distribution), it will be conservative.

5.1 APPLICATION: TEMPORAL PC ON THE METROPOLIT COHORT DATASET

We will reanalyze data from Petersen et al. [2023a]. In that study, the temporal PC algorithm (TPC) [Petersen et al., 2021] was used on a cohort data set of $n = 3145$ Danish men to identify possible causes of depression and heart disease, as well as their interplay. Two experts were also asked to construct a model for the data based on existing studies and subject-field knowledge, and their DAG was compared to the output of TPC. For the comparison here,

we assume that the expert model is correct and wish to evaluate if TPC performs better than a negative control at estimating the expert model. The DAGs have 22 nodes, and hence $m_{\text{max}} = 231$ possible edges.

Table 4 shows the adjacency confusion matrix comparing the expert and TPC models. Note that the two models did not disagree on edge orientation among shared adjacencies (although one shared adjacency was left unoriented by TPC). Hence in this case, the adjacency performance comparison summarizes all edge-wise comparisons of the two outputs. Note also that TPC was set to find the same number of edges as the experts did, i.e. $m_{\text{true}} = m_{\text{est}} = 30$, and hence the symmetry in the confusion matrix is by design.

We conduct an overall test of skeleton fit by comparing the obtained number of true positives, $tp_{\text{obs}} = 10$, with $\text{HyperGeom}(231, 30, 30)$ and we find $p = 0.002$. Hence, we reject H_0 and conclude that TPC performs significantly better than random guessing in this application.

6 SIMULATION-BASED NEGATIVE CONTROLS FOR MORE GENERAL METRICS

Although the results provided above cover some of the most commonly reported metrics for causal discovery evaluation, other interesting metrics cannot be expressed as functions of the adjacency confusion matrix, and hence are out the scope of the results presented thus far.

One example is conditional orientation metrics (also sometimes referred to as *arrowhead* metrics) (see e.g., Andrews et al. [2019]). These metrics describe correct orientations among correctly placed edges. We conjecture that simple exact distributional results under random guessing do not exist for this classification task. The main issue is that consecutive edge placement steps are *not* independent when the goal is to output e.g., a valid DAG: If we have already placed oriented edges such that $X \to Y \to Z$, it is no longer possible to have an edge pointing from Z to X, as this would introduce a cycle and the graph would then no longer be a valid DAG. Thus, describing expected behavior under random guessing when also taking edge orientations into account is more complicated.

However, we can easily use simulation to obtain an empirical estimate of the distribution of a given metric under random edge placement — oriented or not. Let b be the number of repetitions in the simulation study, and let f denote some metric of interest.We propose the following procedure:

1. **Standard simulation study:** Conduct the simulation study as usual: Simulate b "true" DAGs $G^1_{\text{true}}, ..., G^b_{\text{true}}$, generate appropriate data for each, and use the causal discovery algorithm of interest to obtain estimated graphs $\hat{G}^1_{\text{algo}}, ..., \hat{G}^b_{\text{algo}}$. For each $i \in \{1, ..., b\}$, compare the true and estimated graphs by computing the metric of interest, $f(G^i_{\text{true}}, \hat{G}^i_{\text{algo}})$, and for each estimated graph $\hat{G}^i_{\text{algo}}$, count the number edges, m^i_{est}.
2. **Negative control simulation:** For each $i \in \{1, ..., b\}$, draw a negative control random DAG $\hat{G}^i_{\text{NC}}$ with number of edges sampled randomly from $\{m^1_{\text{est}}, ..., m^b_{\text{est}}\}$ (with replacement).
3. **Negative control evaluation:** Compare each negative control with the corresponding true graph by computing the metric of interest, $f(G^i_{\text{true}}, \hat{G}^i_{\text{NC}})$. Report the mean as the expected performance under random guessing, and use the empirical quantiles to construct e.g., a 95% confidence interval.
4. **Comparison:** Finally, compare the metrics obtained under random guessing with the metrics obtained for the evaluated algorithm. In order to draw statistical inference, consider pairwise comparisons and conduct a one-sided statistical test. Compute the p-value as

$$p = \frac{1}{b} \sum_{i=1}^{b} \mathbf{1} \left(f(G^i_{\text{true}}, \hat{G}^i_{\text{algo}}) \leq f(G^i_{\text{true}}, \hat{G}^i_{\text{NC}}) \right)$$

for metrics where small values are favorable (otherwise reverse the inequality).

Note that it is important for obtaining valid statistical inference that it is conducted on the pairwise comparisons of performance, as inference e.g., based on whether or not confidence intervals overlap is highly conservative [Knol et al., 2011].

If the evaluated algorithm does not estimate a DAG, we suggest that Step 2 is altered to match the output of the evaluated algorithm. For example, if the algorithm only aims to learn the Markov equivalence class of the data generating DAG, as represented by a CPDAG, we would simply use negative control CPDAGs in Step 2 by first drawing DAGs and subsequently finding their encompassing CPDAGs.

We have here focused on the case of a simulation study where many different ground truth graphs are simulated in Step 1, but in Section 6.2 we also provide an example of how to adapt the procedure to be suited for evaluation of a real data application where there is only a single ground truth.

6.1 EXAMPLE: SIMULATION STUDY EVALUATING THE PC ALGORITHM

We showcase the proposed simulation-based negative control procedure by evaluating the performance of the PC algorithm [Spirtes and Glymour, 1991]. We construct a small toy simulation study considering the task of learning 10-node DAGs (or more specifically, CPDAGs corresponding to their

Table 5: Comparisons of PC Algorithm and Negative Controls.

	PC		Negative control		
	Mean	CI	Mean	CI	p
Dense case ($m_{\text{true}} = 30$)					
SHD	27.33	$(21, 33)$	31.23	$(26, 36)$	0.202
Adjacency precision	0.85	$(0.65, 1.00)$	0.66	$(0.42, 0.87)$	0.122
Adjacency recall	0.38	$(0.27, 0.50)$	0.29	$(0.17, 0.43)$	0.245
Orientation precision	0.65	$(0, 1)$	0.50	$(0, 1)$	0.360
Orientation recall	0.40	$(0.00, 0.78)$	0.37	$(0.00, 0.78)$	0.464
Proportion recovered v-structures	0.05	$(0.0, 0.2)$	0.02	$(0.00, 0.14)$	0.563
SID (lower bound)	67.73	$(46, 83)$	74.23	$(56, 85)$	0.317
SID (upper bound)	79.48	$(61, 90)$	79.10	$(63, 88)$	0.557
Sparse case ($m_{\text{true}} = 15$)					
SHD	10.1	$(4, 15)$	21.30	$(17, 25)$	0.002
Adjacency precision	0.9	$(0.73, 1.00)$	0.33	$(0.091, 0.571)$	0.000
Adjacency recall	0.7	$(0.47, 0.87)$	0.25	$(0.067, 0.467)$	0.001
Orientation precision	0.9	$(0.5, 1.0)$	0.52	$(0, 1)$	0.273
Orientation recall	0.5	$(0.00, 0.91)$	0.36	$(0, 1)$	0.316
Proportion recovered v-structures	0.3	$(0.0, 0.8)$	0.01	$(0.00, 0.14)$	0.106
SID (lower bound)	29.3	$(7, 55)$	51.01	$(29, 74)$	0.072
SID (upper bound)	51.5	$(22, 81)$	58.43	$(36, 81)$	0.350

Notes: CI denotes a 95% confidence interval based on the empirical distribution. The p-values corresponds to one-sided tests.

Markov equivalence classes) from linear Gaussian data generated according to the DAGs. This is a scenario where the PC algorithm is sound and complete in the large sample limit, so in principle we should expect good performance. However, it is well-known that PC struggles on finite data when the true data generating mechanism is dense, because the algorithm is biased towards sparse graphs (see e.g., [Petersen et al., 2023b]). We therefore consider a moderate sample size of $n = 400$ observations, and two settings for how dense the true DAGs are: A dense case with $m_{\text{true}} = 30$ edges and a sparse case with $m_{\text{true}} = 15$. We expect that PC performs better than negative controls in the sparse case, but not in the dense case.

To evaluate PC, we consider five different metrics that make use of orientation information (and are hence beyond the scope of Sections 3-5): Structural Hamming distance, orientation precision, orientation recall, proportion recovered v-structures, and structural intervention distance (SID) lower and upper bounds[2]. For comparability with the previous sections, we also report two adjacency metrics considered above; namely adjacency precision and recall. We provide definitions of the metrics and additional details about the simulation study in Supplementary Materials C.

Table 5 presents the results. In the dense case, we find that none of the metrics have sufficient discriminatory power to distinguish between PC and the negative control (testing at e.g., a 5% significance level). For this case, PC estimated graphs with numbers of edges ranging from 7 to 19 with a mean of 13.3, which is severely biased towards sparsity, as expected.

For the sparse case, the number of edges estimated by PC ranges from 7 to 16 with a mean of 11.5, i.e., a better match with m_{true}. We find that some metrics show significant differences between PC and the negative control, while others do not: SHD, and adjacency precision and recall are significantly better for PC than the negative controls (at a 5% level), while the others are not. While the mean values for PC and negative controls are generally quite far apart, the 95% confidence intervals reveal very broad distributions for orientation precision, orientation recall, proportion recovered v-structures and both SID bounds. Hence, there is a lot of variability in these metrics — both for PC and negative controls — and they are thus perhaps not very useful for evaluating this case. Another takeaway from this application is that reporting only mean values of metrics is not advisable; ideally, some description of the distribution (e.g., confidence intervals) should be included.

Overall, we find that including negative controls can thus also be used to provide insights into the level of informativeness of a specific metric in scenarios where we have an established consensus of whether a certain causal discovery procedure "works well" or not.

[2]The SID can only be reported as bounds as we are comparing a true DAG with an estimated CPDAG, see details in Supplementary Materials C.

6.2 APPLICATION: STRUCTURAL HAMMING DISTANCES ON THE SACHS DATA

Table 6: Structural Hamming Distances for the Sachs Data.

	Observed		Negative control	
Algorithm	SHD	m_{est}	Mean SHD	p
PC	23	24	31.54	0.001
NOTEARS	22	16	27.10	0.050
LiNGAM	30	33	34.43	0.083
GES	30	30	34.24	0.114
BOSS	35	32	35.24	0.510

In this application, we consider the Sachs dataset [Sachs et al., 2005], which is often used to evaluate causal discovery algorithms. The ground truth DAG for the Sachs dataset has 11 nodes and $m_{\text{true}} = 20$ edges[3].

We evaluate the performance of five causal discovery procedures, namely PC [Spirtes and Glymour, 1991], GES [Chickering, 2002], LiNGAM [Shimizu et al., 2006], NOTEARS [Zheng et al., 2018] and BOSS [Andrews et al., 2023]. We apply each of these algorithms to the Sachs dataset and compute their SHD and estimated number of edges, m_{est}. Based on m_{est}, we simulate 1000 negative controls separately for each algorithm, and report the mean SHD over the negative controls. For algorithms that return a DAG (LiNGAM and NOTEARS), we simulate negative control DAGs, and for algorithms that return a CPDAG (PC, GES, and BOSS), we simulate negative control CPDAGs. In all cases we compare to the ground truth DAG by computing a one-sided p-value testing how often the discovery algortihm performs at least as well as the negative control (according to SHD). Additional details about the application are provided in Supplementary Materials D.

Table 6 summarizes the results. We see that while the smallest SHD value is obtained by NOTEARS, PC produces the smallest p-value ($p = 0.001$) and hence is the furthest removed from random guessing, seconded by NOTEARS ($p = 0.050$) and LiNGAM ($p = 0.083$). In the other end of the spectrum, we find that GES, and especially BOSS, are not significantly different from random guessing testing at e.g., a 10% significance level.

This application illustrates that it is not very meaningful to compare and interpret differences in SHD without taking into account the number of edges placed. A lower SHD does not necessarily mean that an algorithm is further removed from random guessing; it may just reflect a more preferable level of sparsity in the estimated graph, which can be an artefact of the chosen metric. For example for SHD on the Sachs dataset, we can obtain SHD $= m_{\text{true}} = 20$ — i.e., a value that outperforms all considered algorithms — simply by "estimating" the empty graph. Hence, a more meaningful ranking may come about by considering the size of the negative control p-values.

[3]Note that there also exists an alternative version with only 17 edges.

7 DISCUSSION

The results presented here were developed with the aim of evaluating algorithms in an artificial "lab" setting where we have access to a known ground truth. Subsequent to such evaluations, the algorithms should of course also be tested in practice in real data applications without a known ground truth graph. How to asses performance in such scenarios is fundamentally difficult, as causal discovery is a unsupervised problem. One approach for validating causal discovery on real world data is to compare graphs (or resulting effect estimates) found by causal discovery algorithms with expert-made graphs based on theory or existing literature [Petersen et al., 2023a, Gururaghavendran and Murray, 2024]. However, such comparisons are of course only feasible when a comprehensive body of knowledge about the considered variables already exists, and even in this case, it is highly time consuming to construct expert graphs. A more broadly applicable evaluation strategy has been proposed by Eulig et al. [2024]. They compare conditional independence fit of a candidate DAG (given by experts or estimated using causal discovery) with a baseline obtained by randomly permuting nodes. By doing so, they construct a statistical test of fit with a similar interpretation to the test proposed in Section 5, although focusing on conditional independencies implied by the DAG rather than edge presence directly. However, as the authors note, such a test is not readily applicable for validating causal discovery fit, as most algorithms use conditional independence information for estimating the graph, and hence a subsequent test based on the same information would result in overfitting. This could however be resolved by applying data splitting if the sample size renders such an approach feasible.

The distributional results for adjacency metrics presented in Section 3 are conditional on three quantities: The maximal number of edges in the DAG (m_{max}), the number of edges in the true DAG (m_{true}) and the number of edges in the estimated DAG (m_{est}). Clearly, conditioning on the first two is completely uncontroversial, but conditioning on m_{est} may be debated. Our motivation for doing so is as follows: Many causal discovery algorithms — e.g., PC [Spirtes and Glymour, 1991], FCI [Spirtes et al., 2000], GES [Chickering, 2002], and GRaSP [Lam et al., 2022] — require choosing a tuning parameter, which will in practice directly control the number of outputted edges (e.g., test significance level in constraint-based algorithms or score penalties in score-based algorithms). Although we do not generally have a characterization of the relationship between the tuning parameter values and the resulting value of m_{est}, the relationship is generally deterministic in a single causal

discovery application. The way causal discovery algorithms are mostly applied, the tuning parameter is *not* chosen in a data-driven manner, but rather set at somewhat arbitrary "standard" values. Alternatively, in some instances, it may be chosen based on external background knowledge [Petersen et al., 2023a]. In either case, the number of edges in the estimated graph is *de facto* chosen a priori, in which case we can meaningfully condition on it.

Some work has been proposed for data-driven tuning of causal discovery algorithms [Biza et al., 2020]. If such methods are applied, m_{est} will generally be estimated from data. In this case, we lose the distributional results for the adjacency metrics presented in Section 3, as the number of true positives will no longer be hypergeometrically distributed. But we can still use the proposed simulation-based pipeline from Section 6 to obtain a negative control for such algorithms.

Another reason for preferring to condition on m_{est} is related to interpretability. By only considering one value of m_{est}, we compare our causal discovery procedure with a well-specified negative control condition, namely placing the same number of edges at random. This argument also has implications for how we ought to compare two different causal discovery algorithms; to increase interpretability, we advise to either tune the algorithms to estimate the same number of edges and then compare their outputs, or follow the strategy from Section 6.2 and compare negative control p-values. Otherwise, we may be comparing sparse outputs with dense ones without accounting for the different difficulties in estimating sparse and dense graphs, and as we have seen above, such a comparison may not be meaningful, and will definitely be difficult to interpret. Tuning algorithms to produce equally dense outputs has an additional benefit by removing the (difficult to interpret) tuning parameters from the evaluation equation altogether, and replacing them with the simpler notion of outputted graph density.

However, we want to stress that it can be problematic to consider only a single density in a simulation-based evaluation study if the algorithm being evaluated is able to thereby learn the (unique) intended density [Petersen et al., 2023b]. This evaluation design flaw has been present for several supervised discovery algorithms [Li et al., 2020, Xu and Xu, 2021, Yu et al., 2019], and could harm transportability greatly. We propose that a range of densities should therefore always be considered when conducting simulation-based evaluation studies of discovery algorithms that may learn the density directly from training data. But to ease interpretation, the results should ideally be presented stratified according to density.

We have considered a range of different metrics that may be used to evaluate causal discovery methods, but the list is clearly not exhaustive. We have focused on edgewise and structural evaluations, as these are most commonly reported [Gentzel et al., 2019], but we advice that all evaluation studies should include a critical consideration of what metrics are relevant for the specific intended use case. For example, if an intended usecase is mostly focused on causal information flow, and not whether effects are direct or indirect, it is natural to consider a metric that explicitly counts preserved ancestral information [Bang et al., 2024]. Or, if one is interested in using a causal discovery estimate for subsequent effect estimation and inference, one should include metrics on the intervention distribution, possibly targeting a specific causal estimand of interest [Gentzel et al., 2019].

The work presented here has focused on Erdős-Rényi type graphs. This assumption is important for the distributional results in Sections 3 - 5, as the hypergeometric distribution requires random draws. Non-central versions of the hypergeometric distribution allows for biased draws of edges, but we do not believe this is very useful for describing causal graphs: It would allow certain edges to be more likely to be present than others, but would still not consider graph properties beyond singular edges and hence not be appropriate for describing for example graphs that exhibit clustering. However, if a specific evaluation study wants to target such graphs, the simulation-based method proposed in Section 6 can straightforwardly be applied, simply by simulating random graphs from the intended target graph type, see e.g., [Albieri and Didelez, 2014].

As mentioned in Section 5, the exact distributional results for adjacency metrics will by definition result in conservative statistical inference, i.e. conservative control of type I error in statistical tests and overly wide confidence intervals. Due to the discrete nature of the hypergeometric distribution, this is especially pronounced when m_{max} is small, i.e. when there are only few nodes. However, we argue that the considered null hypothesis is very crude — assuming completely random replacement of m_{est} edges — and hence we do not consider conservative inference to be very problematic. Informally, we would ideally like to perform markedly better than random guessing, not just borderline significantly so!

In conclusion, we believe the results and examples provided here showcase that we need to acknowledge that causal discovery is not just another machine learning problem. Estimating a high-dimensional object such as a graph is difficult, and evaluating how well one did is equally challenging. If we do not take into account the most fundamental property of the graphs we simulate for evaluation — their densities — we are not producing useful results that will be likely to generalize to new data with other graph densities. We believe that the use of negative controls will be a useful next step in the direction of more transparent and interpretable evaluations. We of course all hope to do better than random guessing, so let us make it easy to see when we do - and when we do not.

Acknowledgements

The author thanks Vanessa Didelez for her insightful feedback on this work.

References

Vanna Albieri and Vanessa Didelez. Comparison of statistical methods for finding network motifs. *Statistical Applications in Genetics and Molecular Biology*, 13(4):403–422, 2014. doi: doi:10.1515/sagmb-2013-0017. URL https://doi.org/10.1515/sagmb-2013-0017.

Bryan Andrews, Joseph Ramsey, and Gregory F Cooper. Learning high-dimensional directed acyclic graphs with mixed data-types. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*, pages 4–21. PMLR, 2019.

Bryan Andrews, Joseph Ramsey, Ruben Sanchez Romero, Jazmin Camchong, and Erich Kummerfeld. Fast scalable and accurate discovery of dags using the best order score search and grow shrink trees. *Advances in neural information processing systems*, 36:63945–63956, 2023.

Christine W Bang, Janine Witte, Ronja Foraita, and Vanessa Didelez. Improving finite sample performance of causal discovery by exploiting temporal structure. *arXiv preprint arXiv:2406.19503*, 2024.

Konstantina Biza, Ioannis Tsamardinos, and Sofia Triantafillou. Tuning causal discovery algorithms. In *International Conference on Probabilistic Graphical Models*, pages 17–28. PMLR, 2020.

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

Elias Eulig, Atalanti A Mastakouri, Patrick Blöbaum, Michaela Hardt, and Dominik Janzing. Toward falsifying causal graphs using a permutation-based test. *arXiv preprint arXiv:2305.09565v2*, 2024.

Amanda Gentzel, Dan Garant, and David Jensen. The case for evaluating causal models using interventional measures and empirical data. *Advances in Neural Information Processing Systems*, 32, 2019.

Rajesh Gururaghavendran and Eleanor J Murray. Can algorithms replace expert knowledge for causal inference? a case study on novice use of causal discovery. *American Journal of Epidemiology*, page kwae338, 2024.

Leonard Henckel, Theo Würtzen, and Sebastian Weichwald. Adjustment identification distance: A gadjid for causal structure learning. In *Uncertainty in Artificial Intelligence*, pages 1569–1598. PMLR, 2024.

Mirjam J Knol, Wiebe R Pestman, and Diederick E Grobbee. The (mis) use of overlap of confidence intervals to assess effect modification. *European journal of epidemiology*, 26:253–254, 2011.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020.

Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Uncertainty in Artificial Intelligence*, pages 1052–1062. PMLR, 2022.

Hebi Li, Qi Xiao, and Jin Tian. Supervised whole dag causal discovery. *arXiv preprint arXiv:2006.04697*, 2020.

Jonas Peters. *SID: Structural Intervention Distance*, 2023. URL https://CRAN.R-project.org/package=SID. R package version 1.1.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3):771–799, 2015.

Anne H Petersen, Merete Osler, and Claus T Ekstrøm. Data-driven model building for life-course epidemiology. *American Journal of Epidemiology*, 190(9):1898–1907, 2021.

Anne Helby Petersen. *causalDisco: Tools for Causal Discovery on Observational Data*, 2022. URL https://cran.r-project.org/web/packages/causalDisco/index.html. R package version 0.9.1.

Anne Helby Petersen, Claus Thorn Ekstrøm, Peter Spirtes, and Merete Osler. Constructing causal life-course models: Comparative study of data-driven and theory-driven approaches. *American Journal of Epidemiology*, 192(11):1917–1927, 2023a.

Anne Helby Petersen, Joseph Ramsey, Claus Thorn Ekstrøm, and Peter Spirtes. Causal discovery for observational sciences using supervised machine learning. *Journal of Data Science*, 21(2), 2023b.

Joseph D Ramsey, Kun Zhang, Madelyn Glymour, Ruben Sanchez Romero, Biwei Huang, Imme Ebert-Uphoff, Savini Samarasinghe, Elizabeth A Barnes, and Clark Glymour. Tetrad—a toolbox for causal discovery. In *8th international workshop on climate informatics*, pages 1–4, 2018.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

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

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

Jonas Wahl and Jakob Runge. Separation-based distance measures for causal graphs. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Chuanyu Xu and Wei Xu. Causal structure learning with one-dimensional convolutional neural networks. *IEEE Access*, 9:162147–162155, 2021.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International conference on machine learning*, pages 7154–7163. PMLR, 2019.

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

Are You Doing Better Than Random Guessing? A Call for Using Negative Controls When Evaluating Causal Discovery Algorithms (Supplementary Material)

Anne Helby Petersen[1]

[1]Section of Biostatistics, Department of Public Health, University of Copenhagen, Copenhagen, Denmark

A RANDOM URN EXPERIMENT MOTIVATION FOR DISTRIBUTIONAL RESULT

Consider Table 2 and assume that all edges both in G and $\hat{G}$ were placed uniformly at random. Then, given the number of true (m_{true}), estimated (m_{est}) and maximum total (m_{max}) edges, the number of true positives can be seen as the result of a simple random urn experiment with two colors of balls, say, blue and white: White balls correspond to adjacencies included in G, and blue balls are adjacencies not in G. A random causal discovery procedure will then metaphorically draw "balls" (i.e., edges) randomly without replacement, and some will be true positives (white), while others will be false positives (blue). Since the number of white balls (m_{true}), the number of draws (m_{est}) and the total number of balls (m_{max}) are all known a priori, the number of drawn white balls (true positive adjacencies) will by definition follow a hypergeometric distribution: $TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}} \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$.

B COMPUTATIONS FOR TABLE 3

Below, we let $q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}$ be the kth quantile from $\text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$. We also note, and use repeatedly below, that $TP \mid (m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) \sim \text{HyperGeom}(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, and hence $\text{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) = \frac{m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}}}$.

We will compute the expectation and quantiles for each of these four adjacency metrics: Recall, F1, negative predictive value (NPV), and specificity.

Recall: We write recall as a function of TP, m_{max}, m_{true} and m_{est}:

$$\begin{aligned}\text{recall} &= \frac{TP}{TP + FN} \\ &= \frac{TP}{TP + m_{\text{true}} - TP} \\ &= \frac{TP}{m_{\text{true}}}.\end{aligned}$$

Since this is a linear function of TP, the conditional expectation given (m_{max}, m_{true}, m_{est}) is

$$\begin{aligned}\text{E}(\text{recall} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) &= \frac{1}{m_{\text{true}}} \text{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) \\ &= \frac{1}{m_{\text{true}}} m_{\text{est}} \frac{m_{\text{true}}}{m_{\text{max}}} \\ &= \frac{m_{\text{est}}}{m_{\text{max}}}\end{aligned}$$

and the kth quantile of the recall distribution, conditional on $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, is given by

$$\frac{1}{m_{\text{true}}} q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}.$$

F1: We write the F1 score as a function of $TP, m_{\text{max}}, m_{\text{true}}$ and m_{est}:

$$\begin{aligned}
\text{F1} &= \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \\
&= \frac{2 \cdot TP}{2 \cdot TP + m_{\text{est}} - TP + m_{\text{true}} - TP} \\
&= \frac{2 \cdot TP}{m_{\text{est}} + m_{\text{true}}}.
\end{aligned}$$

Since this is a linear function of TP, the conditional expectation given $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$ is

$$\begin{aligned}
\text{E}(\text{F1} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) &= \frac{2 \cdot \text{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}{m_{\text{est}} + m_{\text{true}}} \\
&= \frac{2 \cdot \frac{m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}}}}{m_{\text{est}} + m_{\text{true}}} \\
&= \frac{2 \cdot m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}} \cdot (m_{\text{est}} + m_{\text{true}})}
\end{aligned}$$

and the kth quantile of the F1 distribution, conditional on $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, is given by

$$\frac{2 \cdot q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}}{m_{\text{est}} + m_{\text{true}}}.$$

NPV: We write the negative predictive value (NPV) as a function of $TP, m_{\text{max}}, m_{\text{true}}$ and m_{est}:

$$\begin{aligned}
\text{NPV} &= \frac{TN}{TN + FN} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP}{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP + FN} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP}{m_{\text{max}} - m_{\text{est}}}.
\end{aligned}$$

Since this is a linear function of TP, the conditional expectation given $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$ is

$$\begin{aligned}
\text{E}(\text{NPV} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) &= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + \text{E}(TP \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}{m_{\text{max}} - m_{\text{est}}} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + \frac{m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}}}}{m_{\text{max}} - m_{\text{est}}} \\
&= 1 - \frac{m_{\text{true}}}{m_{\text{max}}} \\
&= 1 - \text{E}(\text{precision} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}})
\end{aligned}$$

and the kth quantile of the NPV distribution, conditional on $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, is given by

$$\frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}}{m_{\text{max}} - m_{\text{est}}}.$$

Specificity: We write specificity as a function of $TP, m_{\text{max}}, m_{\text{true}}$ and m_{est}:

$$\begin{aligned}
\text{specificity} &= \frac{TN}{TN + FP} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP}{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP + FP} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + TP}{m_{\text{max}} - m_{\text{true}}}.
\end{aligned}$$

Since this is a linear function of *TP*, the conditional expectation given $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$ is

$$\begin{aligned}
\mathrm{E}(\text{specificity} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}}) &= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + \mathrm{E}(\mathit{TP} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}{m_{\text{max}} - m_{\text{true}}} \\
&= \frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + \frac{m_{\text{est}} \cdot m_{\text{true}}}{m_{\text{max}}}}{m_{\text{max}} - m_{\text{true}}} \\
&= 1 - \frac{m_{\text{est}}}{m_{\text{max}}} \\
&= 1 - \mathrm{E}(\text{recall} \mid m_{\text{max}}, m_{\text{true}}, m_{\text{est}})
\end{aligned}$$

and the kth quantile of the specificity distribution, conditional on $(m_{\text{max}}, m_{\text{true}}, m_{\text{est}})$, is given by

$$\frac{m_{\text{max}} - m_{\text{est}} - m_{\text{true}} + q_{(k, m_{\text{max}}, m_{\text{true}}, m_{\text{est}})}}{m_{\text{max}} - m_{\text{true}}}.$$

C DETAILS ABOUT PC ALGORITHM SIMULATION STUDY

Let G be the true graph and $\hat{G}$ be an estimate. We consider the following metrics:

Structural Hamming distance: The structural Hamming distance (SHD) counts the number of edge reversals, removals and additions needed in order to transform $\hat{G}$ into G [Tsamardinos et al., 2006].

Adjacency precision and recall: Precision and recall computed from the adjacency/skeleton confusion matrix, as described in Section 2.

Orientation precision and recall: Precision and recall computed from a conditional orientation confusion matrix. The conditional orientation confusion matrix is contructed as follows: For all edges that are both in G and $\hat{G}$, each edge endpoint is classified as:

- True positive if there is an arrowhead both in G and $\hat{G}$.
- True negative if there is a tail both in G and $\hat{G}$.
- False positive if there is an arrowhead in $\hat{G}$, but a tail in G.
- False negative if there is a tail in $\hat{G}$, but an arrowhead in G.

Proportion recovered v-structures: v-structures are node triples with the structure $A \to B \leftarrow C$ where A and C are non-adjacent. This metric counts how many such structures are correctly recovered by $\hat{G}$, divided by the total number of v-structures in G. If there are no v-structures in G, the value is set to 1 (interpreted as all structures being recovered).

Structural intervention distance: The structural intervention distance (SID) counts the number of node pairs (X_i, X_j) for which $\hat{G}$ does not provide a valid adjustment set for the total causal effect of X_i on X_j (assuming G is the truth) [Peters and Bühlmann, 2015]. This property is only well-defined for DAG-DAG comparisons: If $\hat{G}$ is a CPDAG, it may have undirected edges and hence the total causal effects may not be identifed. In this case, one instead computes the SID for each DAG in the equivalence class specified by the CPDAG and reports the minimum and maximum values as bounds. For computing SIDs, we use the `SID` R package [Peters, 2023] with default settings[1].

We now provide a step-by-step example of how the negative controls are computed and used, following the template from Section 6. We focus on a single metric (SHD) and setting (dense). We proceed as follows:

1. Standard simulation study: We draw 1000 random Erdős-Rényi type DAGs over 10 nodes, each with $m_{\text{true}} = 30$ edges. From each DAG, we simulate 400 independent Gaussian observations with randomly drawn regression parameters and error variances[2]. For each dataset, we apply the PC algorithm using a test for vanishing partial correlations with significance level $\alpha = 0.05$.

(a) We compare each of the 1000 true DAGs with the corresponding estimated CPDAG provided by the PC algorithm by computing their SHDs. We find a mean SHD of 27.33 with a 95% confidence interval of $(21; 33)$ (based on the empirical quantiles).

[1] Note that some cases result in warnings due to the estimated graph not being a proper CPDAG, or having large connected components. In both cases, SID is computed based on local expansions of the graphs.

[2] We use default options for regression parameters and error variances from the `simGausFromDAG()` function in the `causalDisco` R package [Petersen, 2022].

(b) We store the true DAGs as well as the distribution of the estimated number of edges across the 1000 applications of the PC algorithm. The number of estimated edges from PC range from 7 to 19 with a mean of 13.3.

2. **Negative control simulation:** We draw 1000 random CPDAGs over 10 nodes with number of edges independently sampled from the m_{est} distribution from Step 1 (b).

3. **Negative control evaluation:** For each of the 1000 negative controls, we compare with a true DAG from Step 1 by computing the SHD. We find a mean SHD of 30.23 (95% CI: $26; 36$).

4. **Comparison:** We conduct pairwise comparisons of SHD values obtained by PC and negative controls. We find

$$p = \frac{1}{1000} \sum_{i=1}^{1000} \mathbf{1} \left(\text{SHD}(G^i_{\text{true}}, \hat{G}^i_{\text{PC}}) \geq \text{SHD}(G^i_{\text{true}}, \hat{G}^i_{\text{NC}}) \right) = 0.202$$

and hence conclude that PC is not significantly different from the negative controls.

D DETAILS ABOUT SACHS DATA APPLICATION

For PC, LiNGAM, GES and BOSS, we apply the algorithms in Tetrad version 7.6.7 [Ramsey et al., 2018] with default settings. For NOTEARS, we report SHD and number of edges as provided in [Zheng et al., 2018].

The negative control DAGs are simulated as Erdős-Rényi type DAGs. The negative control CPDAGs are constructed by first simulating an Erdős-Rényi type DAG, and secondly constructing the CPDAG corresponding to its Markov equivalence class.

We use the 20 edge "truth" version of the Sachs dataset because that is also what was used to evaluate the NOTEARS algorithm in [Zheng et al., 2018]. We obtain both the ground truth graph and the Sachs dataset from this repository: `https://github.com/cmu-phil/example-causal-datasets`.