
How PC-based Methods Err: Towards Better Reporting of Assumption Violations and Small Sample Errors

Sofia Faltenbacher^{1,*}

Jonas Wahl^{2,3,*}

Rebecca Herman¹

Jakob Runge¹

¹University of Potsdam, Potsdam, Germany

²German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

³Department of Philosophy, University of Bergen, Bergen, Norway

*Equal contribution

Abstract

Causal discovery methods based on the PC algorithm are proven to be sound if all structural assumptions are fulfilled and all conditional independence tests are correct. This idealized setting is rarely given in real data. In this work, we first showcase how local errors can lead to untrustworthy edge orientations far away from the error source, highlighting how consequential seemingly innocuous errors can become. Next, we introduce coherency scores to find assumption violations and small sample errors in the absence of a ground truth. These scores do not require statistical tests beyond those already executed by the causal discovery algorithm. Errors detected by our approach extend the set of errors that present themselves as orientation conflicts or ambiguities. We place our computationally cheap global error detection and quantification scores as a bridge between computationally expensive global answer-set-programming-based methods and less expensive local error detection methods. The scores are analyzed on simulated and real-world datasets.

1 INTRODUCTION

Causal discovery, also known as causal structure learning, is an active research field with new methods being published at almost every major machine learning conference; see Zanga et al. [2022], Assaad et al. [2022], Brouillard et al. [2025] for recent surveys on methods and applications. The field aims to provide tools for inferring qualitative cause-and-effect relationships from observational and experimental data, typically in the form of a graphical model, while elucidating the assumptions underlying such inference. One of the seminal causal discovery algorithms that inspired many others is the PC algorithm [Spirtes and Glymour, 1991].

The correctness of the PC algorithm and its descendants, such as FCI [Spirtes, 2001] and PCMCi [Runge, 2020], was proven for an idealized setting in the absence of finite sample errors. In this setting, it is shown that the output is maximally informative as long as the data generation process satisfies the assumptions of the algorithm [Spirtes et al., 2000]. Theoretical results on finite sample performance are much harder to prove and therefore rare; see Kalisch and Bühlmann [2007]. Assessments of finite sample performance employ any of a variety of performance metrics [Tsamardinos et al., 2006, Peters and Bühlmann, 2015, Henckel et al., 2024, Wahl and Runge, 2025], but these metrics usually require knowledge of the ground truth causal graph, limiting these evaluations to simulated data. In addition, it is often unclear how well real-world data conforms to the assumptions of causal discovery, leaving users wondering how much they can trust an algorithm’s results [Krich et al., 2020, Miersch et al., 2025].

As a consequence of statistical errors and violations of assumptions, PC-based methods may produce an output graph with conspicuous inconsistencies. An example of such an inconsistency is an orientation conflict, a consequence of the algorithm’s attempt to orient the same edge in different directions at different points in the procedure. These conflicts show that there is no output graph of the required type that matches the test results of all conditional independence tests executed by the algorithm; otherwise the method would have found it. The consequences of such inconsistencies are often left unaddressed but must be taken seriously as guarantees for downstream tasks like causal effect estimation no longer hold if the provided causal graph is not correct.

Related problems have recently been pointed out by researchers in the field. Jørgensen et al. [2025] discuss real-world examples where typically observed variables cannot be adequately modeled by an SCM. Miersch et al. [2025] have evaluated the robustness of PC-based time series causal discovery for flood drivers and have found that some key hydrological mechanisms are missed even for very large sample sizes. Gamella et al. [2025] have pointed out the

poor performance of several causal discovery algorithms on physical testbeds. We provide a more detailed discussion in Appendix C.1. These works underline the need for rigorous evaluation of causal discovery methods on real-world data, under assumption violations, and in the presence of statistical errors.

Currently, there are local checks for ambiguities [Ramsey et al., 2006, Colombo and Maathuis, 2014] and global but computationally expensive methods based on answer set programming (ASP) [Bromberg and Margaritis, 2009, Russo et al., 2024, Hyttinen et al., 2014] to address the inconsistency problem. Further, graph falsification approaches [Shipley, 2000, Eulig et al., 2025, Ramsey et al., 2024] as well as a method sanity check [Faller et al., 2024] consider adjacent topics. We differentiate our method from related work and point out synergies in a detailed comparison in Section 6. In essence, our method may be seen as a bridge between existing local refinements and global but computationally expensive methods. Specifically, our main contributions are as follows:

- We introduce a new, computationally cheap coherency score for PC-based methods. In essence, our score measures how well the conditional independence test results obtained during a method’s execution match the separations and connections implied by its output. Crucially, the score can be computed without access to a ground truth graph.
- We characterize the subset of erroneous output graphs of a PC-based method that are detectable without access to the ground truth and without running additional tests, see Figure 1.
- We show how computing our score on subsets of data helps to distinguish between assumption violations and statistical errors. We demonstrate error analysis with our scores on simulated and real-world data.
- We provide extensive experiments showcasing that our score serves as a heuristic proxy for the Structural Hamming Distance (SHD) to the unknown ground truth.

2 PRELIMINARIES AND NOTATION

In this work, $\mathbf{X} = (X^1, \dots, X^d)$ with $d \in \mathbb{N}$ denotes a set of nodes that represent random variables in a structural causal model (SCM) as defined by Pearl [2009], and $\mathcal{G}$ denotes a causal graph over $\mathbf{X}$. The causal graph implied by the underlying true SCM is denoted by $\mathcal{G}_{true}$. Causal graphs come with a notion of graphical *separation* such as d -separation for directed acyclic graphs (DAGs, e.g. Pearl [2009]), m -separation for maximal ancestral graphs (MAGs, e.g. [Zhang, 2008]), and σ -separation for cyclic graphs (e.g. [Bongers et al., 2021]). Unless indicated otherwise, statements in this work do not depend on the type of graph as

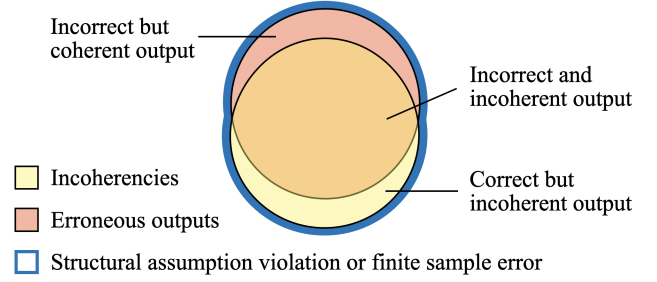


Figure 1: Resolving orientation conflicts and ambiguities always leads to incoherencies, see Corollary 1.1. Outputs without conflicts and ambiguities can still be incoherent, see Observation 3. Assumption violations or statistical errors that lead to erroneous outputs (orange intersection of red and yellow), see Example 3, are a proper subset of incoherencies that can lead to correct or incorrect graphs (yellow) and erroneous outputs that can be coherent or incoherent (red). See Example 9 for a correct but incoherent output and Example 4 for a coherent but incorrect output.

long as one consistently uses the corresponding criterion for separation and its counterpart: *connection*. We represent separation and connection statements with tuples consisting of two singular nodes and a (possibly empty) disjoint subset of nodes. The set of all such statements is denoted by

$$\mathcal{C}(\mathcal{G}) = \{(X, Y, \mathbf{S}) \mid X \neq Y \in \mathbf{X}, \mathbf{S} \subseteq \mathbf{X} \setminus \{X, Y\}\},$$

and given a causal graph $\mathcal{G}$, the relevant notion of separation specifies for each tuple $(X, Y, \mathbf{S})$ whether it is a *separation statement* $X \searrow_{\mathcal{G}} Y \mid \mathbf{S}$ (X and Y are separated given $\mathbf{S}$) or a *connection statement* $X \not\searrow_{\mathcal{G}} Y \mid \mathbf{S}$ (X and Y are connected given $\mathbf{S}$). We define the separation indicator function $\iota_{\mathcal{G}} : \mathcal{C}(\mathcal{G}) \rightarrow \{0, 1\}$ as $\iota_{\mathcal{G}}(X, Y, \mathbf{S}) = 1$ if $X \searrow_{\mathcal{G}} Y \mid \mathbf{S}$ and as $\iota_{\mathcal{G}}(X, Y, \mathbf{S}) = 0$ otherwise. The set of separations and connections are respectively denoted by

$$\mathcal{C}^{sep}(\mathcal{G}) = \{(X, Y, \mathbf{S}) \in \mathcal{C}(\mathcal{G}) \mid \iota_{\mathcal{G}}(X, Y, \mathbf{S}) = 1\},$$

$$\mathcal{C}^{con}(\mathcal{G}) = \{(X, Y, \mathbf{S}) \in \mathcal{C}(\mathcal{G}) \mid \iota_{\mathcal{G}}(X, Y, \mathbf{S}) = 0\}.$$

In a causal graph, an *unshielded triple* is a triple of nodes (X, Y, Z) such that the outer nodes are adjacent to the middle node but not to each other. An unshielded triple is a *collider* if the edges (X, Y) and (Y, Z) point towards Y (e.g. $X \rightarrow Y \leftarrow Z$), otherwise the triple is called a *non-collider*.

Two causal graphs $\mathcal{G}, \mathcal{H}$ in the same graph class are *Markov equivalent* if they share the same separations, i.e. $\mathcal{C}^{sep}(\mathcal{G}) = \mathcal{C}^{sep}(\mathcal{H})$ or, equivalently, the same connections. We denote their Markov equivalence class (MEC) by $[\mathcal{G}] = \{\mathcal{H} \mid \mathcal{H} \sim_M \mathcal{G}\}$. MECs of DAGs can be represented by Complete Partially Directed Acyclic Graphs (CPDAGs), those of MAGs by Partial Ancestral Graphs (PAGs).

A distribution $\mathcal{P}_{\mathbf{X}}$ over the variables $\mathbf{X}$ is called *Markovian* on a causal graph $\mathcal{G}$ or an equivalence class $[\mathcal{G}]$ if the

separation statements $X \bowtie_{\mathcal{G}} Y \mid \mathbf{S}$ imply the conditional dependencies $X \perp\!\!\!\perp_{\mathcal{P}_{\mathbf{X}}} Y \mid \mathbf{S}$ among the corresponding variables. It is called *faithful* on $\mathcal{G}$ if the conditional independencies $X \perp\!\!\!\perp_{\mathcal{P}_{\mathbf{X}}} Y \mid \mathbf{S}$ imply the separation statements for the corresponding nodes in $\mathcal{G}$.

$\mathcal{M}$ will be our symbol for a sound constraint-based causal discovery method. For example, $\mathcal{M}$ could be a stand-in for variants of the PC or FCI algorithm (see, e.g. Spirtes et al. [2000]), including the PC-stable algorithm (see, e.g. Colombo and Maathuis [2014]), or the PC-MCI algorithm family (see, e.g. Runge [2020]). We denote the output graph of a PC-based method $\mathcal{M}$ by $\mathcal{G}_{out}$.

Let $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{(d \times n)}$ be n samples drawn from the d -dimensional distribution $\mathcal{P}_{\mathbf{X}}$. Any constraint-based causal discovery method uses one or several statistical tests for conditional independence (CI), or dependence [Malinsky, 2024] for which all statements can be formulated analogously. Let T be a CI test and let α be the significance level. For brevity, we assume that the PC-based method uses only one type of CI test on all tested statements, but all results can be generalized in a straightforward way to algorithm variants using different types of tests.

Common CI tests include Fisher’s Z test for conditional independence in the linear Gaussian setting, Kernel-based methods [Zhang et al., 2011] or regression-based tests including the generalized covariance measure [Shah and Peters, 2020].

We define a conditional independence test indicator function $\iota_{(T,\alpha)} : \mathcal{C} \rightarrow \{0, 1\}$ for a CI test (T, α) as $\iota_{(T,\alpha)}(X, Y, \mathbf{S}) = 1$ if $X \perp\!\!\!\perp_{(T,\alpha)} Y \mid \mathbf{S}$ (i.e. the tuple tested independent), and $\iota_{(T,\alpha)}(X, Y, \mathbf{S}) = 0$ otherwise. By $\mathcal{T}(\mathcal{M}, T, \alpha) \subseteq \mathcal{C}(\mathcal{G})$ we denote the set of all statements that were tested during the execution of the algorithm $\mathcal{M}$ using test T at significance α . We simply write $\mathcal{T}$ if the referenced method, the CI test, and the threshold are fixed. The tuples that tested conditionally independent and those for which independence was rejected are denoted by

$$\begin{aligned} \mathcal{T}^{ind} &= \{(X, Y, \mathbf{S}) \in \mathcal{T}(\mathcal{M}, T, \alpha) \mid \iota_{(T,\alpha)}(X, Y, \mathbf{S}) = 1\}, \\ \mathcal{T}^{dep} &= \{(X, Y, \mathbf{S}) \in \mathcal{T}(\mathcal{M}, T, \alpha) \mid \iota_{(T,\alpha)}(X, Y, \mathbf{S}) = 0\}. \end{aligned}$$

3 SOURCES AND DETECTABILITY OF ERRORS

In this section, we discuss the sources and manifestations of errors and how they affect the output of a sound PC-based algorithm $\mathcal{M}$.

3.1 ERROR SOURCES

Soundness proofs of PC-based methods [Spirtes et al., 2000, Spirtes, 2001, Runge, 2020, Mooij and Claassen, 2020]

rely on a variety of structural assumptions and error-free CI tests, sometimes called *CI oracles*. If an assumption is violated or a CI test result is incorrect, e.g. due to finite sample randomness, soundness is no longer guaranteed. On finite data, CI testing is considered a difficult problem and requires additional assumptions as no CI test with power against any alternative exists [Shah and Peters, 2020].

Definition 1 (Erroneous Tuples). We call a tuple $(X, Y, \mathbf{S})$ an *erroneous tuple* if the CI test result for $(X, Y, \mathbf{S})$ of a method $\mathcal{M}$ with CI test (T, α) does not match the separation statement of $(X, Y, \mathbf{S})$ in the ground truth causal graph $\mathcal{G}_{true}$, i.e. if

$$\begin{aligned} X \bowtie_{\mathcal{G}_{true}} Y \mid \mathbf{S} \quad \text{but} \quad X \not\perp\!\!\!\perp_{(T,\alpha)} Y \mid \mathbf{S} \quad \text{or} \\ X \not\bowtie_{\mathcal{G}_{true}} Y \mid \mathbf{S} \quad \text{but} \quad X \perp\!\!\!\perp_{(T,\alpha)} Y \mid \mathbf{S}. \end{aligned}$$

We call a method $\mathcal{M}$ with CI test (T, α) *CI-deviant* on a dataset $\mathcal{D}$ if there is at least one erroneous tuple in $\mathcal{T}(\mathcal{M}, T, \alpha)$.

On real-world data with unknown ground truth, it is usually impossible to determine whether a specific triple is erroneous or not.

Example 1. For the PC algorithm [Spirtes et al., 2000], any of the following issues can lead to CI-deviance: faithfulness violations; incorrect CI test results due to an unlucky draw of samples, low power or the functional assumptions of the CI test not being fulfilled, e.g., using a linear CI test on variables with nonlinear relationships.

Assume that we generate data from an SCM with a known ground truth graph $\mathcal{G}_{true}$. We speak of a *graph-type mismatch* between $\mathcal{G}_{true}$ and a method $\mathcal{M}$ if $\mathcal{G}_{true}$ is not in the class of graphs considered by the method $\mathcal{M}$.

Example 2. An example of a graph-type mismatch is the application of the PC algorithm (which assumes data from a ground truth DAG) on a set of causally insufficient observed variables. As a minimal example, take two variables X and Y that are confounded by a hidden variable Z . No directed edge can represent that X and Y are confounded.

3.2 MANIFESTATIONS OF ERRORS

Before formally defining an erroneous output of a PC-based method, we recap the concepts of marked orientation conflicts and ambiguities in the output graph of a PC-based method and shed light onto the fact that orientation conflicts introduce global uncertainty in the output graph.

Orientation Conflicts An orientation conflict occurs if a method $\mathcal{M}$ tries to assign different orientations to the same edge at different stages of the algorithm. Conflicts can be approached in two alternative ways; either by marking conflicts explicitly in the graph, or by forcing the algorithm to

make a decision through a *conflict-resolution strategy*, for instance by ranking tested conditional independencies by some heuristic such as the p -value [Ramsey, 2016], or by allowing the algorithm to overwrite previous orientations. Clearly, these strategies have their own downsides. Overwriting makes the outcome depend on the variable order, while using p -values as a proxy for test reliability is at best a heuristic, as p -values are uniformly distributed under the null hypothesis [Ramsey et al., 2024]. In the presence of conflicts, even seemingly orientable edges may no longer be trustworthy, which we illustrate in Example 5 in the Appendix. Therefore, conflicts may not only reflect local, but also global unreliability of the orientations in the output graph. At the same time, the appearance of conflicts is valuable information, as conflicts indicate structural assumption violations or small sample errors, and when employing a conflict-resolution strategy, this information is no longer directly visible in the output graph. The coherency scores that we will introduce in Section 4 offer a middle ground and will be able to capture this information even after applying resolution strategies. At the same time, they can be used to evaluate different conflict-resolution strategies.

Ambiguities Some PC-based algorithms, including conservative PC [Ramsey et al., 2006] and PC with the majority rule [Colombo and Maathuis, 2014], may additionally mark some unshielded triples as *ambiguous*, indicating that the algorithm refuses to take a decision as to whether a triple is a collider or a non-collider after additional tests. Resolution strategies can also be applied to ambiguities.

Definition 2 (Erroneous Output). Consider a ground-truth graph $\mathcal{G}_{true}$ belonging to a pre-specified class of graphs. We call an output graph $\mathcal{G}_{out}$ of a method $\mathcal{M}$ an *erroneous output* if

1. it does not represent the Markov equivalence class of the ground truth graph $\mathcal{G}_{true}$ or
2. there are marked conflicts or ambiguities in $\mathcal{G}_{out}$.

Observation 1. A method $\mathcal{M}$ can be CI-deviant, but the output graph may still be Markov-equivalent to the unknown ground truth, i.e., not produce an erroneous output, see Examples 8 and 9 in the Appendix.

Observation 2. A graph-type mismatch does not necessarily lead to an erroneous output. For example, the output CPDAG of PC in Example 2 is $X - Y$, which corresponds to the PAG representing the Markov equivalence class of the ground truth MAG $X \leftrightarrow Y$, see also Example 7 in the Appendix.

Throughout this work, access to the ground truth $\mathcal{G}_{true}$ is not provided as we are interested in error detection in real-world scenarios. Therefore, we cannot directly test whether a result is CI-deviant or there was a graph-type mismatch. We now

introduce the concept of incoherencies, only looking at the list of CI tests and the output graph of a method.

Definition 3 (Incoherence). Let $\mathcal{M}$ be a PC-based method using a CI test (T, α) with output $\mathcal{G}_{out}$ without marked orientation conflicts or ambiguities. We call a tuple (X, Y, S) an *incoherent tuple* if the CI test result for (X, Y, S) of a method $\mathcal{M}$ does not match the separation statement of (X, Y, S) in the method’s output graph $\mathcal{G}_{out}$, i.e. if

$$\begin{aligned} X \bowtie_{\mathcal{G}_{out}} Y \mid S \quad \text{but} \quad X \not\perp\!\!\!\perp_{(T, \alpha)} Y \mid S \quad \text{or} \\ X \not\bowtie_{\mathcal{G}_{out}} Y \mid S \quad \text{but} \quad X \perp\!\!\!\perp_{(T, \alpha)} Y \mid S. \end{aligned}$$

We call the output of a PC-based method *incoherent* if there is at least one incoherent tuple. Colloquially, we will also refer to an incoherent tuple as an incoherency, while incoherence is a property of an output graph as a whole.

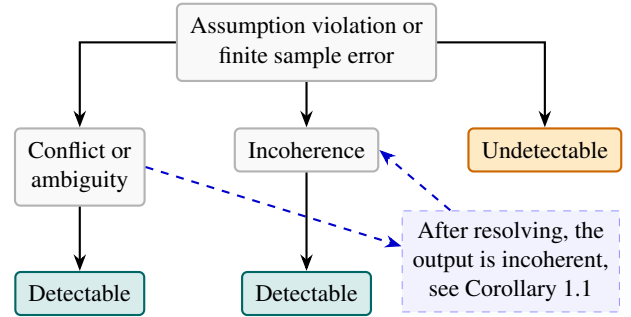


Figure 2: Incoherence is only defined for graphs without marked conflicts or ambiguities. Resolving all conflicts or ambiguities always leads to an incoherent result, see Corollary 1.1. A result can also be incoherent when there is no resolved conflict or ambiguity, see Observation 3. Proposition 1 will show that given only the information collected during one run of the algorithm, the set union of results with marked conflicts and ambiguities and results with incoherencies is exactly the set of detectable assumption violations and finite sample errors.

Testing for incoherence is a straightforward task: The results of the CI tests can be saved while performing the method, and separation queries can be answered for all common graph types that are outputs of PC-based algorithms such as CPDAGs and MAGs, after potentially resolving marked conflicts and ambiguities. For d-separation queries in CPDAGs and m-separation queries in MAGs, the computational complexity is $\mathcal{O}(n + m)$ where n is the number of nodes and m the number of edges, see, e.g. Perković et al. [2015].

The following lemma justifies the use of incoherence as a falsification tool. Colloquially speaking, in the absence of structural assumption violations and finite sample errors, a sound method does not show incoherence.

Lemma 1 (Incoherent and CI-deviant results). Assume that there is no graph-type mismatch between the data generation mechanism and the sound PC-based method $\mathcal{M}$. Then

the existence of an incoherent tuple implies the existence of an erroneous tuple, i.e. CI-deviance of $\mathcal{M}$.

All proofs can be found in Appendix A.

Observation 3. *If the output of a PC-based algorithm has no conflicts and ambiguities, the result can still be incoherent.*

Example 3. We look at the following ground truth structural causal model:

$$\begin{aligned} X &= \eta_X, & Z &= X + \eta_Z, \\ W &= Z + \eta_W, & Y &= 2X - 2W + \eta_Y; \end{aligned}$$

with independent, exogenous noise variables $\eta_X, \eta_Z, \eta_W, \eta_Y \sim \mathcal{N}(0, 1)$. The corresponding true causal graph is $\mathcal{G}_{true}$ in Figure 3 on the left. The effects along the two directed paths from X to Y negate each other, violating faithfulness. Assuming no other sources of errors, the classical PC algorithm finds the conditional independencies $\mathcal{T}^{ind} = \{(X, Y, \emptyset), (Z, Y, \{X, W\}), (W, X, \{Z\})\}$; all other tuples are tested dependent. The output graph is depicted in Figure 3 on the right. No orientation conflicts arise, but X and Y are conditionally independent while being d-connected in $\mathcal{G}_{out}$, so that $(X, Y, \emptyset)$ is an incoherent tuple. The criterion to distinguish d-separation and d-connection in CPDAGs can, for example, be found in Perković et al. [2015].

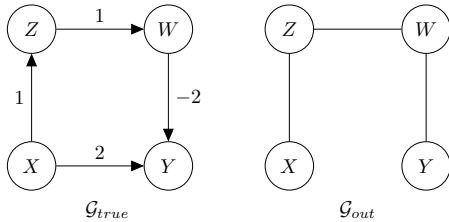


Figure 3: Faithfulness violation leading to an incoherent erroneous output.

However, checking for incoherencies is not sufficient to find all erroneous outputs as the next example shows.

Observation 4. *If the output of a PC-based algorithm has no conflicts, ambiguities or incoherencies, the result can still be erroneous.*

Example 4. We look at the following ground truth structural causal model:

$$X = \eta_X, \quad Y = 0.2 \cdot X + \eta_Y, \quad Z = 0.2 \cdot Y + \eta_Z,$$

with independent, exogenous noise variables $\eta_X, \eta_Y, \eta_Z \sim \mathcal{N}(0, 1)$. The corresponding true causal graph is $\mathcal{G}_{true}$ in Figure 4. The total causal effect of X on Z is 0.04. Assuming no other sources of errors, and choosing the classical

PC algorithm as the method $\mathcal{M}$, the nodes X and Z will be tested independent in most simulations even for 10000 samples, see Section D in the Appendix. Note that the tuple $(X, Z, \{Y\})$ is not tested anymore by $\mathcal{M}$, as the edge between X and Z has already been removed. Therefore $\mathcal{T}^{ind} = \{(X, Z, \emptyset)\}$. There are no orientation conflicts, and the method has no internal incoherencies. We have no chance to detect that, in fact, there was an incorrect CI test, and that the output is erroneous without further tests or expert knowledge. A PC-based method $\mathcal{M}$ that also tests $(X, Z, \{Y\})$ (and finds independence), such as parallelized versions, majority- or conservative-PC would detect incoherence in this particular case. Examples where additional CI tests do not help to detect incoherence are discussed in Examples 6 and 7 in the Appendix.

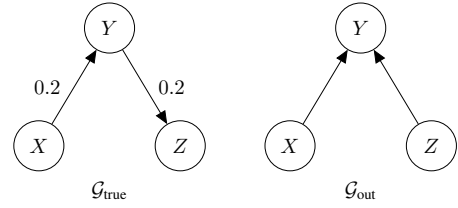


Figure 4: Comparison of the true causal graph and an undetectable erroneous graph structure.

Observation 4 highlights that without access to the ground truth, expert knowledge or additional tests, we cannot in general detect errors of PC-based methods. In other words: There can be output graphs that are coherent but still wrong – and there is no chance to detect it with only the list of CI tests and the output graph of a method.

We now distinguish three manifestations of CI-deviances or graph-type mismatches on the distribution level (D1-D3) and on the graph level (G1-G3), leading to our main results on detectability of errors without access to a ground truth. Again, we assume that we ran a method $\mathcal{M}$ on the set of variables $\mathbf{X}$, which found the independencies $\mathcal{T}^{ind}$, the dependencies $\mathcal{T}^{dep}$ and produced the output graph $\mathcal{G}_{out}$. The following three cases are mutually exclusive.

Case-D1: There is no distribution $\mathcal{P}_{\mathbf{X}}$ that can represent the observed conditional independence test results. More formally, there is no joint distribution $\mathcal{P}_{\mathbf{X}}$ on $\mathbf{X}$ such that $(X, Y, \mathbf{S}) \in \mathcal{T}^{ind}$ implies $X \perp\!\!\!\perp_{\mathcal{P}_{\mathbf{X}}} Y \mid \mathbf{S}$ and $(X, Y, \mathbf{S}) \in \mathcal{T}^{dep}$ implies $X \not\perp\!\!\!\perp_{\mathcal{P}_{\mathbf{X}}} Y \mid \mathbf{S}$. For instance, if the measured test results violate basic logical properties of conditional independence relations such as the *semi-graphoid axioms* [Pearl, 2009], then no distribution $\mathcal{P}_{\mathbf{X}}$ can accommodate them.

Case-D2: There is a distribution $\mathcal{P}_{\mathbf{X}}$ that can represent the conditional independence test results, but no such distribution is Markovian and faithful to $\mathcal{G}_{out}$. Figure

11 shows an example of this situation as the result of a causal sufficiency violation.

Case-D3: There is a distribution $\mathcal{P}_X$ that can represent the conditional independence test results and is Markovian and faithful to $\mathcal{G}_{out}$. In this case, we cannot detect that the output is erroneous given the performed CI test results and the output graph, see Example 4 as well as Examples 6 and 7 in the Appendix.

The output of a PC-based method belongs to one of the following three categories.

Case-G1: Orientation conflicts or ambiguities are marked in $\mathcal{G}_{out}$. Conflicts or ambiguities are only marked if there was an assumption violation or an incorrect CI test result, as the algorithm is proven to be sound otherwise.

Case-G2: No conflicts or ambiguities are marked in $\mathcal{G}_{out}$, but the results of the PC-based method are incoherent. An example of this case was discussed in Example 3.

Case-G3: No conflicts or ambiguities are marked in $\mathcal{G}_{out}$ and the results of the PC-based method are coherent. This case was illustrated in Example 4, see also Examples 6 and 7 in the Appendix.

The following result characterizes the subsets of undetectable erroneous outputs. All proofs of the following results can be found in Section A in the Appendix.

Proposition 1 (Characterization of Undetectable Erroneous Outputs). *Erroneous outputs of a method $\mathcal{M}$ have no marked conflicts and ambiguities and are coherent with the CI test results $\mathcal{T}^{ind} \cup \mathcal{T}^{dep}$ (Case-G3) if and only if there is a distribution that can represent the conditional independencies $\mathcal{T}^{ind}$ and is Markovian and faithful to $\mathcal{G}_{out}$ (Case-D3).*

Corollary 1.1 (Resolved Conflicts and Ambiguities Imply Incoherencies). *Resolving conflicts or ambiguities always leads to incoherencies.*

We have therefore shown that given the list of CI tests of a method $\mathcal{M}$ and its output graph, checking for incoherence—either after resolving or in addition to flagging conflicts and ambiguities—is sufficient to detect all CI-deviances and graph-type mismatches that are theoretically detectable given the information at hand. The results are summarized and visualized in Figure 1 and Figure 2.

4 COMPUTING COHERENCE

We quantify the overall coherency of a method by introducing with a coherency score.

Definition 4 (Coherency Score). Let $\mathcal{M}$ be a PC-based method using a test T with threshold α . Let $w : \mathcal{T} \rightarrow [0, \infty)$

be a non-trivial weight function. The *coherency score* is defined as

$$sc(w, \mathcal{M}) = 1 - \frac{\sum_{(X,Y,S) \in \mathcal{T}} w(X, Y, S) \text{inc}(X, Y, S)}{\sum_{(X,Y,S) \in \mathcal{T}} w(X, Y, S)},$$

where $\text{inc}(X, Y, S) = |\iota_{\mathcal{G}}(X, Y, S) - \iota_{(T, \alpha)}(X, Y, S)|$. Choosing $w(X, Y, S) = 1$ yields the *total coherency score*.

If the total coherency score is less than one, an incoherency is detected. Given the previous results, this allows us to find all CI-deviances or graph-type mismatches detectable without further information, i.e. only given the list of CI test results performed by $\mathcal{M}$ and its output graph $\mathcal{G}_{out}$. To our best knowledge, we are the first to provide a method to do so.

Remark 1 (Weight Function Choices). For different settings, users might find some incoherencies more acceptable than others. For example, if one knows that effect sizes are smaller than one, users might want to use the path-length-adjusted score to account for the fact that the longer the path is, the more likely it is for the first and the last node in a chain to be false positively tested as independent. By allowing for weight function choices, such adjustments to our score are possible.

Different weight functions, i.e., scores other than the total coherency score, are discussed in Appendix B. A test for when a score may be considered too low given a particular structure is provided in Appendix F.

5 EXPERIMENTS

All details on the implementations as well as run-times for the experiments can be found in Appendix D. The code can be found here: https://github.com/this-is-sofia/UAI2026_paper_supplementary_code.

Falsification and error analysis We demonstrate that for real-world and simulated datasets, our score detects more assumption violations and incorrect CI tests than those that present themselves as orientation conflicts or ambiguities. The auto-mpg data set [Quinlan, 1993] containing technical specs of cars is used to illustrate the PC algorithm as it yields no orientation-conflicts using the default version (Fisher’s Z test, threshold 0.05), for example in the documentation of the popular causal inference Python library *dowhy*. We show that there are incoherencies (total coherency score 0.917) that do not show as orientation conflicts. This implies that when using PC on the auto-mpg dataset, there is a CI-deviance or a graph-type mismatch, i.e., no DAG exists that can represent the CI tests performed by the method. All details including the output graph can be found in Appendix D. For results on the cellular signaling dataset [Sachs et al., 2005], also see Appendix D.

Simulational studies overview Across 10,400 graphs with varying numbers of nodes and different edge densities, we demonstrate that our score serves as a proxy for the structural Hamming distance (SHD), the Parent-AID, the Ancestor-AID and the OSet-AID [Henckel et al., 2024] between the output graph and the ground truth. We generated data sets from 10,400 random DAGs with random edge weights, 1000 samples each and varying numbers of nodes and edge densities. The results are summarized in Figure 5 and Table 1. We further discuss limitations, details on the experimental setup and runtime in Appendix D.



Figure 5: The total coherency scores computed without access to the ground truth serve as a heuristic proxy for the SHD to the ground truth graph.

Table 1: The Negative Correlation Between Each Metric and the Coherency Score is Significant.

Metric	Correlation	P-Value
SHD	-0.5092	$< 2.2 \times 10^{-308}$
Parent-AID	-0.5570	$< 2.2 \times 10^{-308}$
Ancestor-AID	-0.4951	$< 2.2 \times 10^{-308}$
OSet-AID	-0.5042	$< 2.2 \times 10^{-308}$

Assumption violations Leveraging the insight that finite sample errors decrease as sample size increases, whereas assumption violations yield scores below one even with oracle CI tests, we propose a heuristic evaluation scheme to distinguish between the two in real-world data. If all assumptions for the soundness of a method $\mathcal{M}$ are fulfilled, the score reaches one as the sample size increases. For models with detectable assumption violations, i.e., the subset of assumption violations that manifest themselves as incoherencies given oracle CI statements, the score remains at a score smaller than one in theory and simulations. We illustrate the observation in Figure 6 where we show detectable assumption violations (moderate confounding, strong confounding) and an undetectable assumption violation (dense confounding). All details on the models, further simulations, and explanations can be found in Appendix D. Although it is a difficult problem to distinguish a low from a high score or a

plateau from a slow convergence, the observation motivates heuristics to distinguish between assumption violations and statistical errors using our scores on increasing subsets of a given dataset described in the Appendix F.

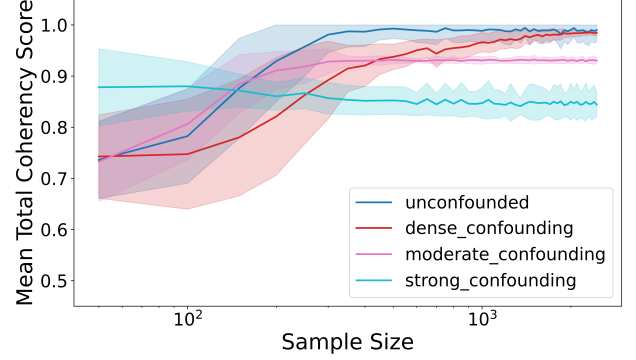


Figure 6: The mean total coherency score for a causal structure without assumption violations (dark blue) vs. for the same structure with different degrees of hidden confounding (red, pink, light blue). The scores reach a plateau at a score smaller across sample sizes for a wide range of assumption violation models.

Method selection We provide an example comparing the FCI to the PC algorithm in the presence of hidden confounding, highlighting the potential of coherency scores for method selection. Consider the following SCM with unobserved variables:

$$\begin{aligned}
 U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & U_3 &= \eta_{U_3}, & U_4 &= \eta_{U_4} \\
 X &= U_1 + U_2 + \eta_X, & Y &= U_2 + U_3 + \eta_Y, \\
 Z &= U_3 + U_4 + \eta_Z, & V &= U_4 + U_1 + \eta_V.
 \end{aligned}$$

where $\eta_X, \eta_Y, \eta_Z, \eta_V, \eta_{U_1}, \eta_{U_2}, \eta_{U_3}, \eta_{U_4} \sim \mathcal{N}(0, 1)$ are independent exogenous noise variables and U_1, U_2, U_3, U_4 unobserved. While oracle-FCI has a total coherency score of 1, oracle-PC with any possible orientation conflict resolution strategy yields a total coherency score of $\frac{15}{16}$, see Figure 11 Appendix D. This allows us to favor oracle-FCI over oracle-PC. Details on the oracle version of the algorithms and the same example with simulated data and CI tests instead of oracle, i.e. true, CI statements can also be found in Appendix D.

6 DISTINCTION FROM AND SYNERGIES WITH RELATED WORK

The large number of broadly related methods, which differ subtly but significantly in terms of objectives, prerequisites, and practicability, requires a detailed comparison. Table 2 provides an overview, all papers mentioned in the table as well as other related works are discussed in detail in this section.

Table 2: A Comparison of Method Refinements and Method Falsification Approaches Applicable Without Access to Expert Knowledge or the Ground Truth Graph. Abbreviations: ACIT = Additional Conditional Independence Tests; Ref. = Refinement; Fals.= Falsification. ACITs Are the Main Driver of Computational Complexity. (*): Including ACITs Possible but not Neccessary.

Paper	Scope	Approach	ACIT
Ram06	Method ref.	local	yes
Col14	Method ref.	local	yes
Bro09	Method ref.	global	yes
Rus24	Method ref.	global	yes
Hyt14	Method ref.	global	yes
Fal24	Method fals.	global	yes
FaWa26 (this)	Method fals.	global	no(*)

6.1 REFINEMENTS OF CONSTRAINT-BASED ALGORITHMS

In the following, we will discuss the strengths and limits of existing tools to increase robustness and consistency in the context of causal discovery and causal structure learning.

Conservative- and majority-PC To remedy some of the pitfalls of the PC algorithm in practice, e.g. the order dependence in the presence of orientation conflicts, the variants conservative-PC [Ramsey et al., 2006] and majority-PC [Colombo and Maathuis, 2014] have been introduced to increase robustness and trust in the result. Both methods essentially add additional CI tests to find ambiguous triples, i.e. unshielded triples that may or may not be oriented as colliders depending on which CI test results one trusts. We have shown that given this new extended list of CI tests compared to the classical PC algorithms, incoherent results are a real superset of results with ambiguous triples; see Observation 3 and Example 3.

Argumentative causal discovery There have been advances in making constraint-based causal discovery algorithms more consistent globally using argumentation frameworks, i.e., logical frameworks to test whether and how severely statements and deductions of statements contradict each other. Argumentation frameworks in causal discovery are used to identify CI statements that should be disregarded to find a set of CI tests that is consistent in itself and/or matches a graph of the type required by the algorithm. Two noteworthy approaches in this area are:

- Bromberg and Margaritis [2009] identify tests that are most contradictory with the knowledge base of CI tests and further CI tests that can be derived from them via the (semi-)graphoid axioms.

- Russo et al. [2024] identify tests that are most contradictory with the prerequisites of the PC algorithm, in particular a DAG as ground truth and a match between the d-separations and the list of CI tests.

Both methods provide a more elaborate inconsistency resolution than the conflict-resolution strategies implemented in common causal discovery packages such as keep first, keep last or sorting by p-value [Ramsey, 2016]. However, there are also several limitations:

1. They both require further assumptions or heuristics. E.g. both methods need an ordering of CI tests by their trustworthiness which both implement by sorting by p-value which also both papers point out as a heuristic without formal guarantees.
2. Both are infeasible for large numbers of variables due to high computational expenses, Russo et al. [2024] explicitly mention 10 variables as a limit.
3. We argue that reporting how many CI test results both methods had to disregard in order to find a consistent subset of all performed tests is valuable information which should be reported. This is where we see a synergy when combined with our approach, shedding light on the fact that errors might not only imply that some tests are incorrect due to an unlucky draw of samples. It can also be due to an assumption violation in which case the method is not fitted for the dataset at hand. The rate of tests that were disregarded to obtain a consistent result is an important information to judge how much we trust the output graph.

We also note that for Bromberg and Margaritis [2009] a consistent set of CI tests does not imply the existence of a matching graph of the type required by the method $\mathcal{M}$, see Case-D2, illustrated by an example in Figure 11.

Optimization via ASP solvers Hyttinen et al. [2014] approach the inconsistency problem in a straightforward way. They take a list of independence and dependence statements as an input, which they choose to be all possible tests on a given dataset. Then, they solve a minimization problem to find the representative graph that minimizes the sum of the weights of the given conditional independence and dependence constraints that are not implied by the representative graph. They achieve high accuracy and theoretical guarantees. On the downside, this approach explodes in complexity: they report that for seven variables the method takes up to half an hour and for eight variables many instances run for several hours. Like our method, this is a global approach to the inconsistency problem with a different goal: They find the most consistent graph while we report incoherencies as a sanity check of output of a method. Using SGS [Spirtes et al., 2000] as the PC-based method $\mathcal{M}$ and their weight function which can be computed categorical or linear Gaussian variables as our weight function, coherency scores

can be seen a measure of trust in their results on a single instance. However, our coherency scores can also be used as a sanity check for methods with more scalable runtimes. The score itself only takes less than a second to be computed on an instance for eight variables.

6.2 TESTING GRAPHS AGAINST A DATA SET

There are several tools available to check how well an (expert) given graph or quantitative expert knowledge [Grünbaum et al., 2023, Eulig et al., 2025, Ramsey et al., 2024, Shipley, 2000]. This is different from our setting in which we evaluate the coherence of a method, i.e. whether a method contradicts itself indicating assumption violations or statistical errors to sanity check a method, not a graph. However, some graph falsification methods can be applied to graphs that are the output of a causal discovery algorithm and we further discuss them in Appendix C.3.

6.3 METHOD SANITY CHECK WITHOUT GROUND TRUTH OR EXPERT KNOWLEDGE

Faller et al. [2024] check whether the causal graphs learned when a causal discovery algorithm is applied to different subsets of the variables are compatible and introduce different notions of compatibility. Their method is applicable to a wide range of methods including score-based approaches. Their score also serves as a heuristic proxy for the SHD to the unknown ground truth and they also remark that the output graph might be Markov-equivalent to the ground truth despite the output on a subset being incompatible, which, however, still reduces trust in the result. This can be seen as a reflection of the theoretical limits to the evaluation of causal discovery methods without access to the ground truth.

While our sanity check does not rely on any additional CI tests or causal discovery method runs, Faller et al. [2024] perform causal discovery with (super-)exponential worst-case runtime on several subsets of the variables requiring additional CI tests when the investigated method is constraint-based. At the same time, their score is applicable to score-based methods, which our score does not cover, see also Appendix C.2.

7 DISCUSSION AND OUTLOOK

We have shown that testing for incoherencies is an exhaustive error detection approach given only the information collected in one run of a PC-based algorithm and without access to the ground truth. We highlight the potential for error analysis. Similar to many criteria for statistical model selection, such as AIC, our scores can be used to select hyperparameters that maximize the respective score. We encourage causal discovery users to report our computationally cheap

coherency scores to detect and quantify detectable errors whenever using PC-based methods on real-world data, and to investigate individual incoherencies in detail whenever feasible.

Limitations While a coherency score smaller than one implies that there was a CI-deviance or a graph-type mismatch, the resulting output graph might still correctly represent the Markov equivalence class of the unknown ground-truth graph. On the other hand, a perfect score does not imply that the output graph is correct, as some CI-deviances and graph-type mismatches are undetectable. We therefore stress that the score can only serve as a heuristic proxy.

In some cases, it may be hard to judge when a score should be considered too low, and the output should be discarded. This issue can be addressed, but not resolved, by the discussed weight function choices and reference values. This is not a flaw of our method in particular, but a general limitation of error detection and quantification in this setting.

Acknowledgements

J.R., J.W., and R.H. received funding from the European Research Council (ERC) Starting Grant CausalEarth (Grant Agreement No. 948112). J.W. was supported by the European Regional Development Fund (ERDF) and the German Federal State of Saarland within the scope of (To)CERTAIN (Project ID EFRE-AuF-0000942), and by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project MAC-MERLin (Grant Agreement No. 01IW24007).

References

- Charles K Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73: 767–819, 2022.
- Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M. Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5): 2885–2915, 2021. Publisher: Institute of Mathematical Statistics.
- Facundo Bromberg and Dimitris Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research*, 10(2), 2009.
- Philippe Brouillard, Chandler Squires, Jonas Wahl, Karen Sachs, Alexandre Drouin, Dhanya Sridhar, et al. The landscape of causal discovery data: Grounding causal discovery in real-world applications. In *Causal Learning and Reasoning*, pages 834–873. PMLR, 2025.

- David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.
- Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3921–3962, 2014.
- Elias Eulig, Atalanti A Mastakouri, Patrick Blöbaum, Michaela Hardt, and Dominik Janzing. Toward falsifying causal graphs using a permutation-based test. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26778–26786, 2025.
- Philipp M Faller, Leena C Vankadara, Atalanti A Mastakouri, Francesco Locatello, and Dominik Janzing. Self-compatibility: Evaluating causal discovery without ground truth. In *International Conference on Artificial Intelligence and Statistics*, pages 4132–4140. PMLR, 2024.
- Juan L Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for AI methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025.
- Daniel Grünbaum, Maïke L Stern, and Elmar W Lang. Quantitative probing: Validating causal models with quantitative domain knowledge. *Journal of Causal Inference*, 11(1):20220060, 2023.
- Leonard Henckel, Theo Würtzen, and Sebastian Weichwald. Adjustment identification distance: A gadget for causal structure learning. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.
- Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. Constraint-based causal discovery: conflict resolution with answer set programming. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 340–349, 2014.
- Frederik Hytting Jørgensen, Luigi Gresele, and Sebastian Weichwald. What is causal about causal models and representations? *arXiv preprint arXiv:2501.19335*, 2025.
- Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(3), 2007.
- Christopher Krich, Jakob Runge, Diego G Miralles, Mirco Migliavacca, Oscar Perez-Priego, Tarek El-Madany, Arnaud Carrara, and Miguel D Mahecha. Estimating causal networks in biosphere–atmosphere interaction with the PCMCi approach. *Biogeosciences*, 17(4):1033–1061, 2020.
- Daniel Malinsky. A cautious approach to constraint-based causal model selection. *arXiv preprint arXiv:2404.18232*, 2024.
- Peter Miersch, Wiebke Günther, Jakob Runge, and Jakob Zscheischler. Evaluating the robustness of PCMCi+ for causal discovery of flood drivers. *Artificial Intelligence for the Earth Systems*, 4(4):240114, 2025.
- Joris M Mooij and Tom Claassen. Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *Conference on Uncertainty in Artificial Intelligence*, pages 1159–1168. Pmlr, 2020.
- Vivian Nastl and Moritz Hardt. Do causal predictors generalize better to new domains? *Advances in Neural Information Processing Systems*, 37:31202–31315, 2024.
- Judea Pearl. *Causality*. Cambridge University Press, Cambridge, 2nd edition, 2009.
- Emilija Perković, Johannes Textor, Markus Kalisch, and Marloes H Maathuis. A complete generalized adjustment criterion. In *Uncertainty in Artificial Intelligence- Proceedings of the Thirty-First Conference (2015)*, pages 682–691. AUAI Press, 2015.
- Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3):771–799, 2015.
- Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.
- Anne Helby Petersen. Are you doing better than random guessing? A call for using negative controls when evaluating causal discovery algorithms. In *Conference on Uncertainty in Artificial Intelligence*, pages 3465–3479. PMLR, 2025.
- R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. DOI: <https://doi.org/10.24432/C5859H>.
- Joseph Ramsey. Improving accuracy and scalability of the PC algorithm by maximizing p-value. *arXiv preprint arXiv:1610.00378*, 2016.
- Joseph Ramsey, Peter Spirtes, and Jiji Zhang. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI’06, pages 401–408. AUAI Press, 2006.
- Joseph D Ramsey, Bryan Andrews, and Peter Spirtes. Choosing dag models using Markov and minimal edge count in the absence of ground truth. *arXiv preprint arXiv:2409.20187*, 2024.
- Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series

- datasets. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1388–1397. PMLR, 2020.
- Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D. Mahecha, Jordi Muñoz-Marí, Egbert H. van Nes, Jonas Peters, Rick Quax, Markus Reichstein, Marten Scheffer, Bernhard Schölkopf, Peter Spirtes, George Sugihara, Jie Sun, Kun Zhang, and Jakob Zscheischler. Inferring causation from time series in Earth system sciences. *Nature Communications*, 10(1): 2553, June 2019. Number: 1.
- Fabrizio Russo, Anna Rapberger, and Francesca Toni. Argumentative causal discovery. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, pages 938–949, 2024.
- Karen Sachs, Omar Perez, Dana Pe’er, Douglas A. Luffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.
- Rajen D. Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514 – 1538, 2020.
- Bill Shipley. A new inferential test for path models based on directed acyclic graphs. *Structural Equation Modeling*, 7(2):206–218, 2000.
- Peter Spirtes. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 278–285. PMLR, 2001.
- 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 Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Boston, 2000.
- Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1): 31–78, 2006.
- Jonas Wahl and Jakob Runge. Separation-based distance measures for causal graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 3412–3420. PMLR, 2025.
- Marcel Wienöbst, Leonard Henckel, and Sebastian Weichwald. Embracing discrete search: A reasonable approach to causal structure learning. *arXiv preprint arXiv:2510.04970*, 2025.
- Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: theory and practice. *International Journal of Approximate Reasoning*, 151:101–129, 2022.
- Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9:1437–1474, 2008.
- Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 804–813. AUAI Press, 2011.

How PC-based Methods Err: Towards Better Reporting of Assumption Violations and Small Sample Errors (Supplementary Material)

Sofia Faltenbacher^{1,*}

Jonas Wahl^{2,3,*}

Rebecca Herman¹

Jakob Runge¹

¹University of Potsdam, Potsdam, Germany

²German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

³Department of Philosophy, University of Bergen, Bergen, Norway

*Equal contribution

A PROOFS

Proof of Lemma 1

Proof. By the soundness of PC-based methods, the true causal graph $\mathcal{G}_{true}$ corresponding to the ground truth data generating mechanism is recovered if all assumptions are fulfilled and all CI tests are correct. The existence of an incoherent tuple implies there is no graph $\mathcal{G}$ of the type required by the method such that all CI test results match the separation and connection statements implied by $\mathcal{G}$. Otherwise, it would be the output by the soundness of the method. Therefore, if the graph type required by the method matches the graph type of $\mathcal{G}_{true}$, at least one tuple must be erroneous, i.e. at least one CI test result does not match the separation or connection statement implied by $\mathcal{G}_{true}$. $\square$

Proof of Proposition 1

Proof. D1 to D3 and G1 to G3 each partition the set of CI-deviances. We prove that $D3 = G3$.

$\subseteq$: D3 is ensured to be a subset of G3 by the soundness of the PC-like algorithm: If given the CI test results, there is a distribution that is Markovian and faithful to G_{out} , then the equivalence class of the graph will correctly be found by the PC-like algorithm, in particular there will be no conflicts, ambiguities or incoherencies.

$\supseteq$: We prove that G3 is a subset of D3 by showing that $D1 \cap G3 = \emptyset$ and $D2 \cap G3 = \emptyset$.

$D1 \cap G3 = \emptyset$: D1 states that no distribution can represent the CI test results $\mathcal{T}^{ind} \cup \mathcal{T}^{dep}$. The equivalence class represented by $\mathcal{G}_{out}$ represents a set of separation statements $\mathcal{C}_{out}^{sep}$ such that a corresponding set of conditional independence statements can be represented by a family of distributions. As the CI test results cannot be represented by a distribution, this one-to-one correspondence must be violated for at least one tuple $(X, Y, \mathbf{S})$, i.e.

$$\iota_{(T,\alpha)}(X, Y, \mathbf{S}) \neq \iota_{\mathcal{G}}(X, Y, \mathbf{S}).$$

$D2 \cap G3 = \emptyset$: D2 states that the distribution that represents the conditional independencies $\mathcal{T}^{ind}$ is not Markovian and faithful to any graph in the equivalence class represented by $\mathcal{G}_{out}$. By definition, that means for at least one tuple $(X, Y, \mathbf{S})$,

$$\iota_{(T,\alpha)}(X, Y, \mathbf{S}) \neq \iota_{\mathcal{G}}(X, Y, \mathbf{S}).$$

Together, $D3 = G3$. $\square$

Proof of Corollary 1.1

Proof. Proposition 1 shows that if there are marked conflicts or ambiguities, there is no graph of the graph type assumed by the method $\mathcal{M}$ that can represent the results of the CI test. Therefore, resolving all conflicts and ambiguities leads to incoherent results, i.e., a mismatch between CI test results and separation statements. $\square$

B SCORES FOR DIFFERENT WEIGHT FUNCTIONS

We explicitly formulate the scores for different weight functions used in the paper and introduce additional interesting weight functions.

- **(Standard) Independence-Connection-Mismatch.** The weight function

$$w(X, Y, \mathbf{S}) = \begin{cases} 0 & \text{if } (X, Y, \mathbf{S}) \in \mathcal{T}^{dep} \\ 1 & \text{else,} \end{cases}$$

yields the Independence-Connection-Mismatch:

$$r_{ICM}(w, \mathcal{M}) = 1 - \frac{\sum_{(X, Y, \mathbf{S}) \in \mathcal{T}^{ind}} (1 - \iota_{\mathcal{G}}(X, Y, \mathbf{S}))}{|\mathcal{T}^{ind}|}.$$

- **(Standard) Dependence-Separation-Mismatch.** The weight function

$$w(X, Y, \mathbf{S}) = \begin{cases} 0 & \text{if } (X, Y, \mathbf{S}) \in \mathcal{C}^{con} \\ 1 & \text{else,} \end{cases}$$

yields the Dependence-Separation-Mismatch:

$$r_{DSM}(w, \mathcal{M}) = 1 - \frac{\sum_{(X, Y, \mathbf{S}) \in \mathcal{C}^{sep}} (1 - \iota_{(T, \alpha)}(X, Y, \mathbf{S}))}{|\mathcal{C}^{sep}|}.$$

- **Conditioning-Set-Size-Adjusted Total Coherency Score.** The weight function

$$w(X, Y, \mathbf{S}) = \exp(-|\mathbf{S}|)$$

yields the Conditioning-Set-Size-Adjusted Total Coherency Score..

- **Conditioning-Set-Size-Adjusted Independence-Connection-Mismatch.** The weight function

$$w(X, Y, \mathbf{S}) = \begin{cases} 0 & \text{if } (X, Y, \mathbf{S}) \in \mathcal{T}^{dep} \\ \exp(-|\mathbf{S}|) & \text{else,} \end{cases}$$

yields the Conditioning-Set-Size-Adjusted Independence-Connection-Mismatch.

- **Conditioning-Set-Size-Adjusted Dependence-Separation-Mismatch.** The weight function

$$w(X, Y, \mathbf{S}) = \begin{cases} 0 & \text{if } (X, Y, \mathbf{S}) \in \mathcal{C}^{con} \\ \exp(-|\mathbf{S}|) & \text{else,} \end{cases}$$

yields the Conditioning-Set-Size-Adjusted Dependence-Separation-Mismatch.

- **Path Length Adjusted Coherency Score.** We can compute a path length adjusted faithfulness coherency score by choosing

$$w(X, Y, \mathbf{S}) = \exp(-d(X, Y)),$$

where $d(X, Y)$ denotes the length of the shortest collider free path between X and Y . This accounts for the fact that we want to place less weight on incoherencies of tuples that are connected by a long path on which small direct effects could lead to a very low total effect. It can be combined with the Independence-Connection-Mismatch or the Dependence-Separation-Mismatch analogously to the previously presented combined scores.

C MINI-REVIEW

C.1 RECENT CRITISISM OF CAUSAL DISCOVERY

We discuss papers questioning the performance of causal discovery on real data as well as works on the accuracy of score and constraint based causal discovery algorithms.

SCM realism Jørgensen et al. [2025] developed a framework to investigate whether the assumption that measured variables follow a causal model can be falsified. They point out that this issue may particularly arise for representations, i.e., a function (e.g., aggregation) of the underlying variables, as illustrated by the classical total cholesterol example, see Jørgensen et al. [2025], Example 6.4. This can be seen as a criticism of the assumption that measured variables between which we want to explore relationships follow the logic of SCMs which is a core assumption of many causal discovery methods. The work criticizes what could be described as SCM realism, i.e. assuming observed variables can be adequately modeled by an SCM.

Weak performance on real-world data There have been improvements in setting up simulational studies to evaluate causal discovery methods with a focus on benchmarking against random guessing, e.g. by Petersen [2025]. However, the question of how well the algorithms perform on semi-synthetic and real-world data remains widely open. Nastl and Hardt [2024] tested PC, FCI, and Invariant Causal Prediction (ICP, [Peters et al., 2016]) on real-data and showed that on their selection of 16 real-world data sets, the causal parents found by these methods do not improve out-of-domain prediction compared to using all features as predictors. In favor of the methods, one can point out that Nastl and Hardt [2024] tested them on 16 particular datasets for each of which it must be evaluated how well the data conform to the methods' assumptions and if the methods are configured accordingly. Nevertheless, the work indicates that there is a large set of applications for which the methods cannot be applied straightforwardly in their default implementations. Or, as they put it: "Across 16 datasets, we were unable to find a single example where causal predictors generalize better to new domains than a standard machine learning model trained on all available features."

Weak performance on semi-synthetic data Further, Gamella et al. [2025] introduced the Causal Chambers which generate sensor data from a light and a wind tunnel. In the Chambers, variables can be intervened upon, and the ground truth is known as the underlying physics are well understood. In their case studies, they found that GES [Chickering, 2003], a popular score-based causal discovery method, "struggle[s] with the nonlinear effects of the polarizer angles [...] and the weak effects of the additional light-emitting diodes". For time series data from the wind tunnel, they found that "PCMCI+ [Runge et al., 2019] exhibits low recall and performs similar to random guessing". For PCMCI+, Gamella et al. [2025] drop edges where conflicts arose. On the one hand, this changes the results drastically and might partially explain the weak performance. On the other, this showcases that many PC variants have not implemented appropriate tools for handling orientation conflicts. The case studies of Gamella et al. [2025] only investigate particular configurations of specific algorithms and are not an exhaustive evaluation of causal discovery. Nevertheless, their work illustrates weaknesses in current causal discovery evaluation and hopefully leads to more extensive benchmarking on semi-synthetic data in the future.

C.2 CONSTRAINT-BASED VS. SCORE BASED CAUSAL DISCOVERY

Recent work, e.g. Wienöbst et al. [2025] suggests that score-based methods outperform PC-based methods in discovering DAGs from synthetic data, particularly for linear Gaussian models. The devil's advocate might ask: Why even care to introduce a new evaluation method tailored to error reporting for PC-based causal discovery algorithms? The reason why we aim for better error reporting for PC-based causal discovery methods is twofold.

Potential for relaxation of assumptions Synthetic DAG data in the linear Gaussian setting is not the end of the road. Both sets of methods still have to be evaluated more systematically on synthetic, semi-synthetic and real-world data for different set-ups and with varying relaxed assumptions. Better evaluation methods enable the way forward. PC-based methods promise conceptually easier inclusion of hidden common causes, nonlinear relationships and cycles than score-based methods and methods for all these settings are available and used.

Easy error reporting to and by practitioners needed Profanely, PC-based methods are, due to their tailoring to different set-ups like stationary time series data or the inclusion of hidden common causes and cycles, widely used by practitioners [Brouillard et al., 2025] which is in itself a reason for exhaustive error reporting. The presence of a detectable error introduces uncertainty in the entire result that should be reported to and by practitioners.

C.3 TESTING OUTPUT GRAPHS

Ramsey et al. [2024] suggest using their proposed Markov checker for hyperparameter tuning of causal discovery methods. Our analysis of error propagation complement their work as our described detectability limits also apply to their setting. Their work also has a conceptual similarity to ours as they test whether the Markov property is fulfilled for a range of graph types and separation criteria while their theoretical focus and use cases differ from ours.

Eulig et al. [2025] mention that their method can be used to falsify output graphs of causal discovery methods with the strong limitation that only CI tests that were neither used to construct the output graph of the algorithm, nor implied via semi-graphoid axioms can be the input for their method.

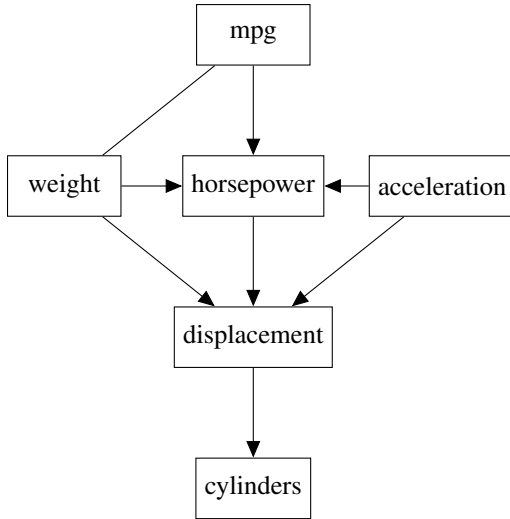
D EXPERIMENTAL RESULTS

For the experiments, we use a parallelized version of PC implemented in the Python package *causy* as the method $\mathcal{M}$. The conditional independence test of choice is Fisher’s Z and the threshold α is set to 0.05. Computing the score theoretically takes $\mathcal{O}(n + m) \cdot |\mathcal{T}|$ where n is the number of nodes, m is the number of edges and $|\mathcal{T}|$ is the number of CI tests performed by the PC-like method. Note that we only check separation statements for all tuples tested and do not do further statistical tests. We provide computational times on a MacBook Pro with an Apple M2 Pro Chip and 32 GB memory.

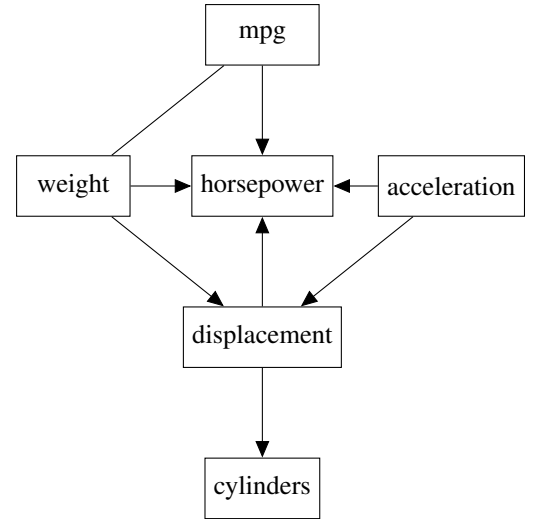
Detecting Incoherencies in the Auto MPG data set. Computing all scores on the auto-mpg data set with six features (variables) and 392 instances (samples) on the computer specified in the introduction of this section took 0.07 seconds.

The variables *mpg* and *displacement* show an ambiguity, in particular, they are tested independent given the three sets $\{\{acceleration, weight\}, \{horsepower, weight\}, \{cylinders, weight\}\}$. Depending on which conditioning set is tested first, *horsepower* is or is not in the conditioning set for which *mpg* and *displacement* are tested independent. As *mpg* – *horsepower* – *displacement* is an unshielded triple in the skeleton, it changes the output graph which one is tested first. This is an ambiguity which is labeled by methods like conservative-PC or resolved by rules like the majority rule. In our case, it can be recovered due to using the parallelized version which does more CI tests than necessary in the classical PC algorithm together with our scores.

We resolved the ambiguity in both ways possible, the output graphs are shown in Figure 7. Table 3 shows the score results.



(a) Graph if we lay more emphasis on the test that says that horsepower *is* in the conditioning set yielding that mpg and displacement are independent and *do not* orient the unshielded triple as a collider.



(b) Graph if we lay more emphasis on the test that says that horsepower is *not* in the conditioning set yielding that mpg and displacement are independent and *do* orient the unshielded triple as a collider.

Figure 7: Comparison of output graphs of PC on auto mpg data after ambiguity resolution.

Detecting Incoherencies in the Sachs data set. The data set contains 11 phosphoproteins and phospholipids in individual cells in each perturbation data set. Sachs et al. [2005] learn a Bayesian network and compare it to expert knowledge. For our configuration, the parallelized version of PC with ParCorr as the CI test returned a graph with orientation conflicts, implying a CI-deviance or a graph-type mismatch. Resolving them naively (strategy: keep first) leads to the a relatively high coherency score of 0.98. Further analysis would be necessary to interpret this result.

Simulation studies: score by SHD, Parent-AID, Ancestor-AID and OSet-AID to ground truth. We generated random DAGs with 4 to 9 nodes and $\text{round}(\#nodes/2)$ to $(\#nodes \cdot (\#nodes - 1))/2$ edges. For each number of nodes

	Resolve as collider	Resolve as non-collider
Total Coherency Score	0.9174	0.9174
Dependence-Separation-Mismatch (DSM)	0.9504	0.9669
Independence-Connection-Mismatch (ICM)	0.9669	0.9504
Conditioning-Set-Size-Adjusted	0.8895	0.9004
Conditioning Set Size Adjusted DSM	0.9185	0.9358
Conditioning Set Size Adjusted ICM	0.9709	0.9646
Number of Orientation Conflicts	0	0

Table 3: Coherency scores for PC on auto mpg data with the two possible ambiguity resolutions.

and edges, we drew 100 random DAGs and generated 1000 samples each. We randomly drew edge weights between 0.8 and 1.2. We then applied parallel PC and computed the score for each instance. For the method and the computer specified in the introduction of this section, it took 86469 seconds (i.e. 24 hours or 1442 minutes) to compute, where the majority of the computational time is allocated to running a parallel PC for 104000 graphs and only 497 seconds (i.e. 8 minutes) for computing all scores. Computing the total coherency score for one graph took between $8e - 05$ seconds (4 nodes, 2 edges) and 0.5 seconds (9 nodes, 36 edges).

We plotted the mean total coherency scores over 100 instances for each setting in Figure 5 by the structural hamming distance (SHD) to the ground truth. We note that we compute the SHD between two CPDAGs: The output of (parallel) PC and the CPDAG of the ground truth DAG from which we sampled. Also, we used the version of SDH for CPDAGs which counts edge differences once, not twice if the edge type differs as it is, for example, also implemented in the Python package *causal-learn*. Our results suggest that the total coherency score on average serves as a proxy for the SHD to the ground truth when the ground truth is not available. Similar to Faller et al. [2024], we point out that this can only serve as a heuristic and we do not provide theoretical guarantees — indeed, we proved one would need further information to derive them. Also, as discussed in the main paper, there can be undetectable CI-deviances leading to a score of one even though the SHD to the ground truth is nonzero as well as detectable CI-deviances leading to a correct output graph, i.e. an SHD of zero.

We also note that the parallel implementation of PC does more tests than necessary due to the parallelizes execution of CI tests with different conditioning sets and conditioning set sizes. The parallelizes version therefore finds more CI-deviances than a classical implementation of PC would.

For computing the Parent-AID, the Ancestor-AID and the OSet-AID, we used the Python package *gadjid* [Henckel et al., 2024]. We disregarded graphs that included cycles after conflict resolution. Plots of each metric and the scores can be found in Figure 8, Figure 9 and Figure 10.

We computed the Pearson correlation and the corresponding p-values with *scipy*. It returned zero as the real p value is smaller than what the 64 bit floats that *scipy* uses can handle, i.e. $< 2.2 \times 10^{-308}$.

Assumption violations For the assumption violation plot in Figure 6, we drew from the following data-generating SCMs and applied the PC algorithm in the previously describes configuration. For each sample size, we used 100 draws of samples from the SCM and computed the mean score depicted interpolated by a piecewise linear function in the figure, as well as its standard deviation depicted as a transparent environment. All noise terms are standard Gaussian and independent.

1. Unconfounded Model, no hidden variables:

$$X = \eta_X, \quad Y = \eta_Y, \quad Z = X + Y + \eta_Z, \quad V = Z + \eta_V, \quad W = Z + \eta_W.$$

2. Moderate confounding with hidden variables $\{U_1, U_2\}$:

$$\begin{aligned} U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & X &= 0.5 \cdot U_1 + \eta_X, & Y &= 0.5 \cdot U_1 + \eta_Y, \\ Z &= X + Y + \eta_Z, & V &= Z + 0.5 \cdot U_2 + \eta_V, & W &= Z + 0.5 \cdot U_2 + \eta_W. \end{aligned}$$

3. Strong confounding with hidden variables $\{U_1, U_2\}$:

$$\begin{aligned} U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & X &= U_1 + \eta_X, & Y &= U_2 + \eta_Y, \\ Z &= X + Y + \eta_Z, & V &= Z + U_2 + \eta_V, & W &= Z + U_1 + \eta_W. \end{aligned}$$

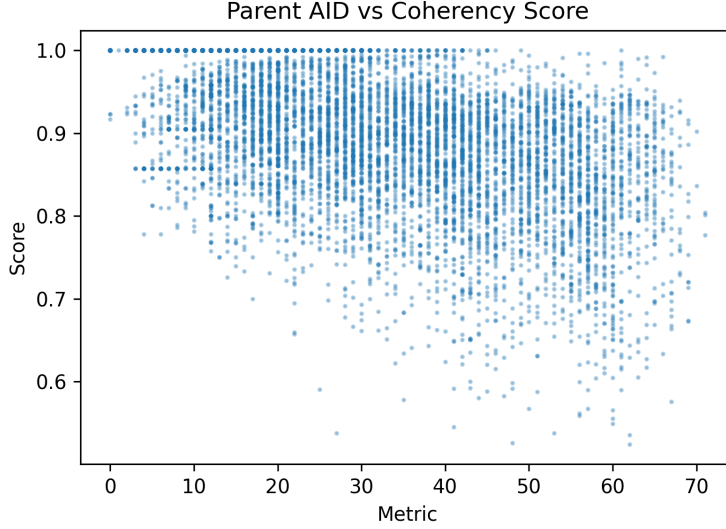


Figure 8: Parent-AID and scores for 10,400 graphs.

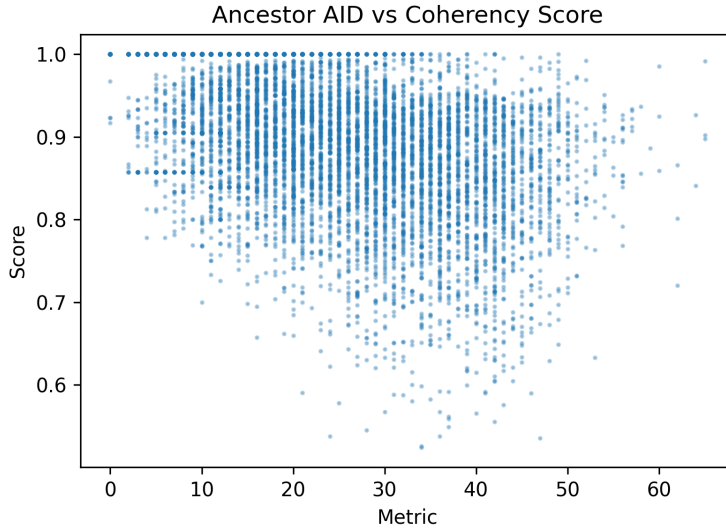


Figure 9: Ancestor-AID and scores for 10,400 graphs.

4. Dense confounding with hidden variables $\{U_1, U_2, U_3, U_4\}$:

$$\begin{aligned}
 U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & U_3 &= \eta_{U_3}, & U_4 &= \eta_{U_4}, \\
 X &= U_1 + U_3 + \eta_X, & Y &= U_2 + U_4 + \eta_Y, & Z &= X + Y + \eta_Z, & V &= Z + U_2 + U_3 + \eta_V, \\
 W &= Z + U_1 + U_4 + \eta_W.
 \end{aligned}$$

Looking at Figure 6, we can see that the coherency score is high for the dense confounding model. This is a natural edge case that can be understood by looking at the extreme case: For a model in which all pairs of nodes are confounded, the output graph of the PC algorithm will always be a fully connected graph which matches with the conditional independence results, i.e. $\mathcal{T}^{ind} = \{\}$. Therefore, the coherency score can detect less CI-deviances for densely confounded models.

The same observation of constant low scores across sample sizes can be made for other detectable assumption violations such as the faithfulness violations induced by the data-generating SCM in Example 3, see Table 4 for mean total coherency scores and the standard deviation across sample sizes (100 repetitions per sample size).

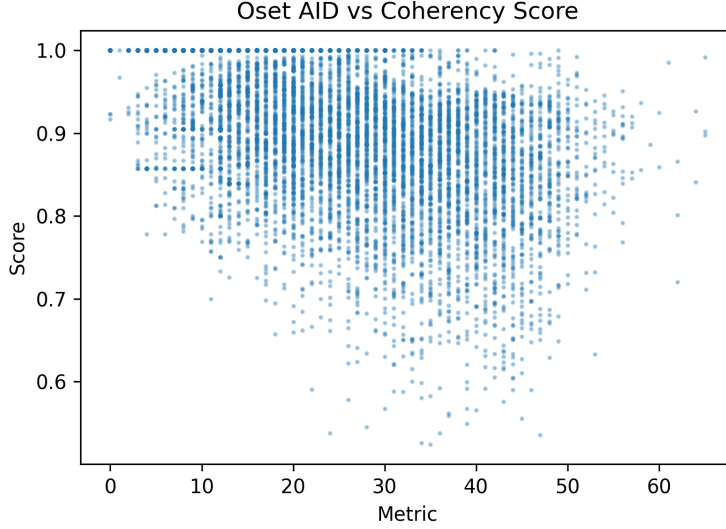


Figure 10: OSet-AID and scores for 10,400 graphs.

Table 4: Faithfulness Violation from Example 3

	50	100	1000	10000
Total	0.9 (0.056)	0.879 (0.038)	0.876 (0.03)	0.884 (0.041)

Method Selection Consider the following SCM:

$$\begin{aligned}
 U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & U_3 &= \eta_{U_3}, & U_4 &= \eta_{U_4} \\
 X &= U_1 + U_2 + \eta_X, & Y &= U_2 + U_3 + \eta_Y, & Z &= U_3 + U_4 + \eta_Z, & V &= U_4 + U_1 + \eta_V.
 \end{aligned}$$

where $\eta_X, \eta_Y, \eta_Z, \eta_V, \eta_{U_1}, \eta_{U_2}, \eta_{U_3}, \eta_{U_4} \sim \mathbb{N}(0, 1)$ are independent exogenous noise variables and U_1, U_2, U_3, U_4 unobserved, see Figure 11. We run the algorithm on the observed variables and note that there is a structural assumption violation, in particular, a causal insufficiency. This is the example discussed in the main paper. On simulated data, we find that the score is almost constant between sample sizes and detects the incoherent results, see Table 5 for mean scores and standard deviations. It illustrates that the method selection criterion illustrated with oracle-PC on oracle CI statements provided in the main paper translates to versions of PC on simulated data in a natural way. For an undetectable causal insufficiency, see Example 7.

Table 5: Causal Insufficiency

Sample Size	50	100	1000	10000
Total	0.891 (0.084)	0.846 (0.044)	0.843 (0.033)	0.849 (0.039)
Independence-Connection-Mismatch	0.896 (0.08)	0.846 (0.044)	0.843 (0.033)	0.849 (0.039)
Dependence-Separation-Mismatch	0.995 (0.037)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Orientation Conflicts	1.27	3.14	3.58	3.44

Details on Example 4 Consider the following SCM for a constant $c \in \mathbb{R}$:

$$X = \eta_X, \quad Y = cX + \eta_Y, \quad Z = cY + \eta_Z,$$

where $\eta_X, \eta_Y, \eta_Z \sim \mathbb{N}(0, 1)$ are independent exogenous noise variables. This is the model for a simple mediated effect from X to Z . We now simulate data from this SCM for different effect strengths c , run the PC algorithm and compute the scores for the results. We do so for different sample sizes and average over 100 draws for each sample size. The resulting graphs

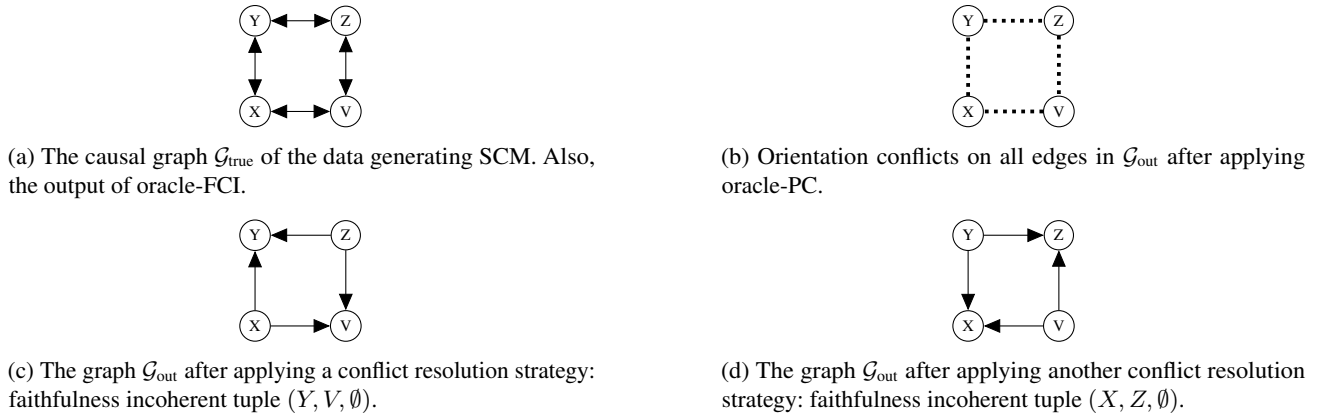


Figure 11: Comparison of outputs of PC on causally insufficient data with/without conflict resolution.

are shown in Figure 12. The average scores and standard deviations are provided in Table 6 for $c = 0.2$ (most simulations yield the graph in Figure 12 (d) as the output graph), in Table 7 for $c = 1$ (most simulations yield the graph in Figure 12 (b) as the output graph) and Table 8 for $c = 10$ (simulations with smaller samples often yield the graph in Figure 12 (c) as the output, with increasing sample size (b) becomes predominant).

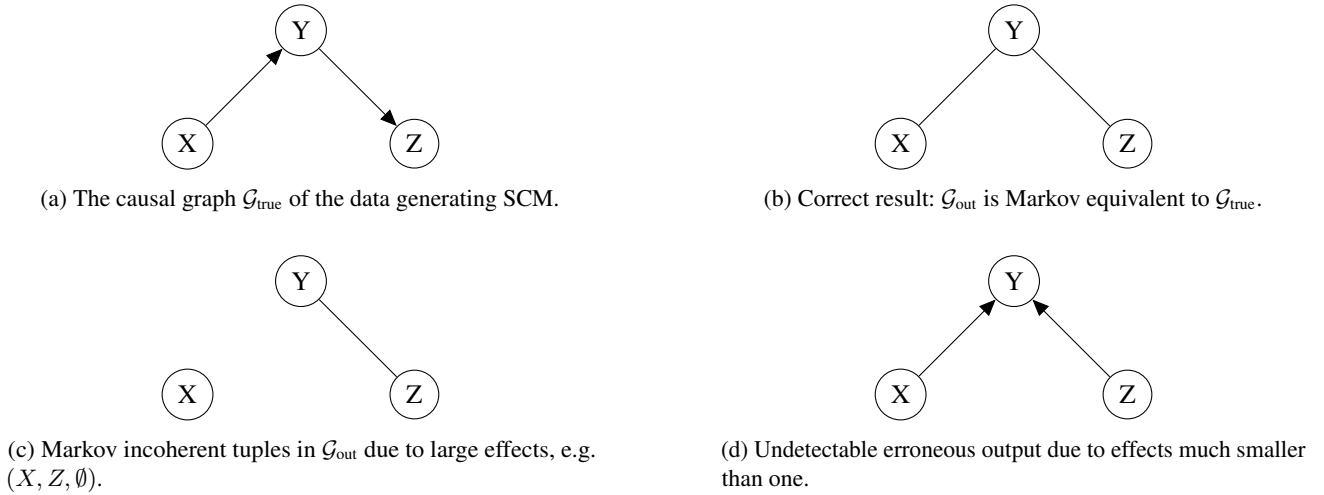


Figure 12: Three output graphs after performing PC on data generated by the SCM in (a) with different effect sizes.

Table 6: Mediated Effect with $c = 0.2$

	50	100	1000	10000
Total	0.997 (0.033)	0.993 (0.047)	1.0 (0.0)	1.0 (0.0)
Independence-Connection-Mismatch	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Dependence-Separation-Mismatch	0.997 (0.033)	0.993 (0.047)	1.0 (0.0)	1.0 (0.0)
Orientation Conflicts	0.0	0.0	0.0	0.0

E ADDITIONAL EXAMPLES

Example 5. Let $\mathcal{M}$ be the PC algorithm with marked orientation conflicts. We discuss the output graph in Figure 13. We assume that, based on its CI test results, the algorithm has found two colliders $W_1 \rightarrow X_3 \leftarrow W_2$ and $W_3 \rightarrow X_6 \leftarrow W_4$, all other unshielded triples have been classified as non-colliders. According to PC's orientation rules applied to the first collider,

Table 7: Mediated Effect with $c = 1$

	50	100	1000	10000
Total	0.99 (0.057)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Independence-Connection-Mismatch	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Dependence-Separation-Mismatch	0.99 (0.057)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Orientation Conflicts	0.0	0.0	0.0	0.0

Table 8: Mediated Effect with $c = 10$

	50	100	1000	10000
Total	0.703 (0.104)	0.74 (0.138)	0.973 (0.09)	1.0 (0.0)
Independence-Connection-Mismatch	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Dependence-Separation-Mismatch	0.703 (0.104)	0.74 (0.138)	0.973 (0.09)	1.0 (0.0)
Orientation Conflicts	0.0	0.0	0.0	0.0

all edges between X_3 and X_6 have to be directed to the right. On the other hand, applying the same rules to the second collider directs these edges to the left, resulting in a conflict which is flagged at one or all of the dotted edges (depending on the implementation). The underlying reason of the conflict is that a DAG with the depicted skeleton for which the set of unshielded colliders is exactly $\{W_1 \rightarrow X_3 \leftarrow W_2, W_3 \rightarrow X_6 \leftarrow W_4\}$, does not exist. A possible source of this error may be that the collider $W_1 \rightarrow X_3 \leftarrow W_2$ was misclassified and is actually a non-collider. The seemingly unaffected orientations $\{X_3 \rightarrow X_2 \rightarrow X_1\}$ are derived from this collider and might thus be incorrect as well.

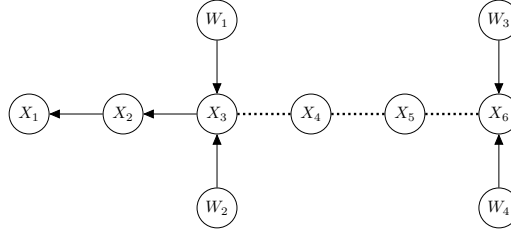


Figure 13: Without conflict resolution, the dotted edges are marked as orientation conflicts, while also the seemingly orientable edges are no longer reliable.

Example 6. We consider the following ground truth structural causal model:

$$X = \eta_X, \quad Z = X + \eta_Z, \quad Y = 2X - 2Z + \eta_Y,$$

with independent, exogenous noise variables $\eta_X, \eta_Z, \eta_Y \sim \mathcal{N}(0, 1)$. The corresponding true causal graph is $\mathcal{G}_{true}$ in Figure 14. Note that there is a perfect counteractive mechanism: the direct positive effect of X on Y plus the mediated negative effect of X on Y via Z cancel out, i.e. a faithfulness violation. Assuming no other sources of errors, and choosing the classical PC algorithm as the PC-like method $\mathcal{M}$, this leads to testing X and Y to be unconditionally independent and all other tuples to be dependent, i.e. $\mathcal{T}^{ind} = (X, Y, \emptyset)$. There is a distribution capturing these independencies and dependencies which is also Markovian and faithful to $\mathcal{G}^{out}$ (D3), see Figure 14. There are no orientation conflicts and the method is *not* incoherent (G3). In this case, we have no chance to detect that in fact there was a faithfulness violation and the output graph $\mathcal{G}_{out}$ is erroneous given the list of CI tests and the output graph.

Example 7. We want to stress that there can also be undetectable causal insufficiencies. Consider the following SCM:

$$\begin{aligned} U_1 &= \eta_{U_1}, & U_2 &= \eta_{U_2}, & U_3 &= \eta_{U_3}, \\ X &= U_1 + U_2 + \eta_X, & Y &= U_2 + U_3 + \eta_Y, & Z &= U_3 + U_1 + \eta_Z. \end{aligned}$$

where $\eta_X, \eta_Y, \eta_Z, \eta_{U_1}, \eta_{U_2}, \eta_{U_3} \sim \mathcal{N}(0, 1)$ are independent exogenous noise variables and U_1, U_2, U_3 unobserved. The resulting graphs are shown in Figure 15.

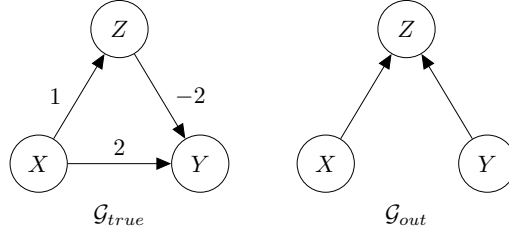


Figure 14: Faithfulness violation leading to a coherent but erroneous output.

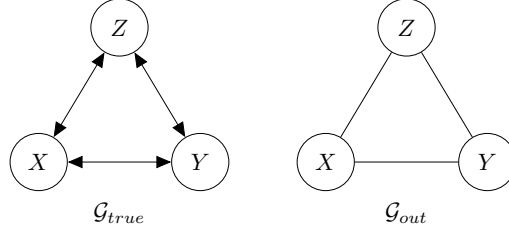


Figure 15: Causal insufficiency leading to a coherent result.

Example 8. As a simple example, take majority-PC as the method $\mathcal{M}$ with one incorrect CI test result leading to an ambiguity and an ambiguity resolution (majority rule) that ignores the one incorrect CI test result. The resulting graph will then be Markov-equivalent to the ground truth causal graph despite one incorrect CI test result.

Example 9. Consider the classical PC algorithm as the method $\mathcal{M}$ on data generated by the following SCM:

$$X = \eta_X, \quad Z_1 = X + \eta_{Z_1}, \quad Z_2 = Z_1 + \eta_{Z_2}, \quad Y = Z_2 + \eta_Y.$$

Now, consider the following CI test results: $\mathcal{T}^{ind} = \{(X, Z_2, \{Z_1\}), (Y, Z_1, \{Z_2\}), (X, Y, \{Z_1, Z_2\})\}$, where the two independencies with conditioning set size one have been missed due to a small sample errors or CI test assumption violations, in particular $(X, Y, \{Z_1\})$ and $(X, Y, \{Z_2\})$. The PC algorithm returns the correct output CPDAG, i.e. the undirected chain $X - Z_1 - Z_2 - Y$, but the result is CI-deviant. In fact, it is even incoherent, i.e., detectably CI-deviant as $(X, Y, \{Z_1\})$ and $(X, Y, \{Z_2\})$ are tested dependent but d-separated in the output graph.

F FURTHER HEURISTICS AND TESTS

Increasing subsets For all assumption violations that lead to incoherencies, the theory implies that for the sample size $n \rightarrow \infty$, the score converges at a score smaller than one. Our simulational studies show that for many settings, this convergence is visible even for moderate sample sizes. These observations motivate the following procedure:

1. Compute the score of the method $\mathcal{M}$ on the whole dataset $\mathcal{D}$ as well as for increasing subsets of $\mathcal{D}$.
2. Analyze whether the scores for increasing subsets stagnate on a low score across sample sizes or if the score increases with increasing sample size.
3. A stagnating low score might indicate an assumption violation.

We stress that this is a heuristic for error analysis and not a formal statistical test.

Negative control A heuristic test to generate a reference value at significance β . This test can be used to develop a reference value for a particular setting.

- 1. Compute score on dataset:** Execute PC-based algorithm $\mathcal{M}$ with CI test (T, α) on the a dataset $\mathcal{D}$ with sample size n . Compute the desired coherency score $sc(\cdot, \mathcal{M})$. Record the resulting output graph $\hat{\mathcal{G}}$ with edge density p .
- 2. Generate simulated data for comparison:** Generate K (e.g. $K = 1000$) Erdős-Rényi random graphs over the nodes with expected edge density p . Generate data from these graphs according to the algorithm's assumptions. E.g., if the algorithm does not allow for hidden confounding, do not discard variables.

3. Evaluate: Run $\mathcal{M}$ on each simulation and compute score $sc(\cdot, \mathcal{M})$ in each run. Reject the null if the score on the original dataset is worse than $(1 - \beta) \cdot 100$ percent of the simulations.