Shapley-PC: Constraint-based Causal Structure Learning with a Shapley Inspired Framework

Fabrizio Russo and **Francesca Toni** {FABRIZIO, FT}@IMPERIAL.AC.UK
Imperial College London, UK

Editors: Biwei Huang and Mathias Drton

Abstract

Causal Structure Learning (CSL), also referred to as causal discovery, amounts to extracting causal relations among variables in data. CSL enables the estimation of causal effects from observational data alone, avoiding the need to perform real life experiments. Constraint-based CSL leverages conditional independence tests to perform causal discovery. We propose *Shapley-PC*, a novel method to improve constraint-based CSL algorithms by using Shapley values over the possible conditioning sets, to decide which variables are responsible for the observed conditional (in)dependences. We prove soundness, completeness and asymptotic consistency of Shapley-PC and run a simulation study showing that our proposed algorithm is superior to existing versions of PC.

Keywords: Causal Structure Learning, Causal Discovery, Graphical Models, Shapley Values

1. Introduction

Causal Structure Learning (CSL), also referred to as causal discovery, is the process of extracting causal relationships among variables in data, and represent them as graphs. Learning structural relations is important because of their causal interpretation. It corresponds to collecting and validating, with data, the assumptions necessary to perform causal inference, e.g. using causal graphical models (Peters et al., 2017) or Functional Causal Models (FCM) (Spirtes et al., 2000; Pearl, 2009). These models allow the estimation of causal effects, such as the impact of an action or treatment on an outcome. Causal effects are ideally discovered through real life experiments in the form of randomised control trials, but these can be expensive, time consuming or unethical, e.g. in establishing if smoking causes cancer, one would need some of the experiment's subjects to take up smoking. Thus, it is important to be able to use observational, as opposed to experimental, data to study causes and effects (Peters et al., 2017; Schölkopf et al., 2021).

CSL has been studied extensively in various settings and several methods have been proposed to address it (see e.g. (Glymour et al., 2019; Vowels et al., 2022; Zanga et al., 2022) for overviews). The literature includes three classes of methods: constraint-based, score-based and FCM-based methods. In this paper, we focus on constraint-based methods, and provide a novel CSL algorithm of this class.

Constraint-based methods use conditional independence tests and graphical rules based on d-separation (Pearl, 2009) to recover as much of the causal structure as possible, under different assumptions (Colombo and Maathuis, 2014). Under the assumption of causal sufficiency, i.e. that no latent common causes are present in the data, the PC[1] algorithm (Spirtes et al., 2000) recovers graphs encoding as much of the discoverable relations as possible (see §3). Depending on the assumptions, the output of constraint-based methods may be sound and complete (Spirtes et al., 2000)

[1] From its creators' names: **P**eter Spirtes and **C**lark Glymour.

and asymptotically consistent (Kalisch and Bühlmann, 2007; Harris and Drton, 2013). However, with a finite sample, errors can emerge from the several conditional independence tests performed.

The novel constraint-based method proposed in this paper improves the performance of PC, as well as other methods built on PC to mitigate its limitations on finite samples (Ramsey et al., 2006; Colombo and Maathuis, 2014; Ramsey, 2016), by using a novel perspective on CSL. Specifically, our method analyses the results of conditional independence tests using the game-theoretical concept of Shapley values (Shapley, 1953). Generally, Shapley values quantify the contribution of individual entities to an output created by a group of entities. They have been used in settings ranging from economics (Ichiishi, 1983) to machine learning (Lundberg and Lee, 2017; Frye et al., 2020; Heskes et al., 2020; Teneggi et al., 2023) and root cause analysis (Budhathoki et al., 2022), but, to the best of our knowledge, not for CSL. Overall, our contributions are as follows:

- We propose a novel decision rule that can be applied to constraint-based CSL algorithms to improve their robustness to errors in the independence tests (§4).
- We propose the *Shapley-PC algorithm*, integrating the novel decision rule within the PC-Stable algorithm (Colombo and Maathuis, 2014), proving that Shapley-PC preserves, in the sample limit, the soundness, completeness and consistency of the original PC-algorithm (§4).
- We provide an extensive evaluation of Shapley-PC, giving empirical evidence about the value-added of our decision rule when data distributions are "close-to-unfaithful" (Ramsey et al., 2006), and showing that it consistently outperforms PC-based predecessors while using the same information extracted from data (§5). Code is made available at `https://github.com/briziorusso/ShapleyPC`.

2. Preliminaries

Graph Notions. A graph $\mathcal{G} = (\mathbf{V}, E)$, is made up of a set of nodes $\mathbf{V} = \{X_1, \ldots, X_d\}$ and a set of edges $E \subseteq \mathbf{V} \times \mathbf{V}$. The nodes correspond to random variables, while the edges reflect the relationships between variables. A graph can be *directed* if it contains only directed edges ($\leftrightarrows$); *undirected* if it only has undirected edges ($-$) and *partially directed* if it has both. The *skeleton* $\mathcal{C}$ of a (partially) directed graph is the result of replacing all directed edges with undirected ones. A graph is *acyclic* if there is no directed path (collection of directed edges) that begins and ends with the same variable, in which case it is called a Directed Acyclic Graph (DAG). If an edge exists between two nodes, then these are adjacent. A graph is *complete* if all nodes are adjacent. The set of nodes adjacent to a node X_i, according to a graph $\mathcal{G}$, is denoted by $\text{adj}(\mathcal{G}, X_i)$. A node $X_j \in \text{adj}(\mathcal{G}, X_i)$ is called a parent of X_i if $X_j \rightarrow X_i$ and $\text{pa}(\mathcal{G}, X_i)$ is the set of parents of X_i. X_i is a descendant of X_j if there is a directed path from the latter to the former. A triple (X_i, X_j, X_k) is called an *Unshielded Triple (UT)* if X_i and X_k are not adjacent but each is adjacent to X_j, represented as $X_i - X_j - X_k$.

Each variable takes values from its own domain. Two variables X_i, X_j are *independent*, given a conditioning set $\mathbf{S} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$, if fixing the values of the variables in $\mathbf{S}$, X_i or X_j does not provide any additional information about X_j or X_i (resp.). In this case, we write $X_i \perp\!\!\!\perp X_j \mid \mathbf{S}$, call $\mathbf{S}$ a *separating set* for X_i, X_j and say that $\mathbf{S}$ *d-separates* X_i, X_j, by rendering them independent (see (Pearl, 2009, Def. 1.2.3) for a formal definition). A UT can be oriented as a *v-structure* $X_i \rightarrow X_j \leftarrow X_k$, where X_j is called a *collider*, by virtue of d-separation, as a collider is a variable that makes dependent other two variables that are independent otherwise.

Hence, if we observe $X_1 \perp\!\!\!\perp X_2 \mid \varnothing$ (denoted $X_1 \perp\!\!\!\perp X_2$ from now on) and $X_1 \not\perp\!\!\!\perp X_2 \mid \{X_3\}$ (denoted $X_1 \not\perp\!\!\!\perp X_2 \mid X_3$ from now on) we can infer that X_3 is a collider for X_1 and X_2, making $X_1 - X_3 - X_2$ a v-structure, i.e. $X_1 \to X_3 \leftarrow X_2$.

A DAG can be interpreted causally when nodes linked by directed edges are associated to causes and effects (Spirtes et al., 2000; Pearl, 2009). This allows manipulations that represent interventions (experiments) to estimate the causal effect of a variable upon another, without performing the actual experiments (Pearl, 2009). Causal *sufficiency* is the assumption that no latent common causes (confounders) are present in the data. Probabilistic measures are needed in practice to relate graphs to observational data.

Statistical Notions. A joint probability distribution P factorises according to a DAG $\mathcal{G}$ if $P(\mathbf{V}) = \prod_{i=1}^{d} P(X_i \mid \text{pa}(\mathcal{G}, X_i))$. P is said Markovian w.r.t. $\mathcal{G}$ if it respects the conditional independence relations entailed by $\mathcal{G}$ via d-separation. In turn, P is *faithful* to $\mathcal{G}$ if the opposite is true, i.e. DAG $\mathcal{G}$ reflects all conditional independences in P. Different DAGs can imply the same set of conditional independences, in which case they form a Markov Equivalence Class (MEC, (Richardson and Spirtes, 1999)). DAGs in a MEC present the same adjacencies and v-structures and are uniquely represented by a *Completed Partially* DAG (CPDAG) (Chickering, 2002). A CPDAG is a partially directed graph that has a directed edge if every DAG in the MEC has it, and an undirected edge if both directions appear in different DAGs in the MEC.

A *Conditional Independence Test (CIT)*, e.g. Fisher's Z (Fisher, 1970), HSIC (Gretton et al., 2007), KCI (Zhang et al., 2011), SCIT (Zhang et al., 2023) and ARECI (Chen et al., 2024) is a procedure whereby a test statistic measuring independence is constructed with a known asymptotic distribution under the null hypothesis $\mathcal{H}_0$ of independence. Calculating the test statistic on a given dataset allows to estimate the p-value (or observed significance level) of the test for that dataset, under $\mathcal{H}_0$. This is a measure of evidence against $\mathcal{H}_0$ (Casella and Berger, 2002). Under $\mathcal{H}_0$, p is uniformly distributed in the interval $[0, 1]$, which allows to set a significance level α that represents the pre-experiment Type I error rate (rejecting $\mathcal{H}_0$ when it is true), whose expected value is at most α (Hung et al., 1997). A CIT, denoted by $I(X_i, X_j \mid \mathbf{S})$, outputs an observed significance level p. If $I(X_i, X_j \mid \mathbf{S}) = p \geq \alpha$ then $X_i \perp\!\!\!\perp X_j \mid \mathbf{S}$. Instead, if $I(X_i, X_j \mid \mathbf{S}) = p < \alpha$ then we can reject $\mathcal{H}_0$ and declare the variables dependent: $X_i \not\perp\!\!\!\perp X_j \mid \mathbf{S}$. Under the alternative hypothesis of dependence, the distribution of p depends on the sample size and the true value of the test statistic. However, under any assumption, the distribution of p monotonically decreases and is markedly skewed towards 0 (Hung et al., 1997). This allows to compare p-values, with the highest p bearing the lowest likelihood of dependence (Hung et al., 1997; Ramsey, 2016; Raghu et al., 2018).

Shapley Values. Consider a team $\mathbf{N} = \{1, \ldots, n\}$ of players collaborating to achieve a collective value $v(\mathbf{N})$, where v is a value function that assigns a real number $v(\mathbf{S})$ to any coalition $\mathbf{S} \subseteq \mathbf{N}$. The Shapley value $\phi_v(i)$ (Shapley, 1953) quantifies the marginal contribution of a player $i \in \mathbf{N}$ when joining any possible coalition $\mathbf{S}$, averaged over all possible configurations of $\mathbf{S}$. This contribution is weighted according to the likelihood of each coalition's occurrence. Formally:

$$\phi_v(i) = \sum_{\mathbf{S} \subseteq \mathbf{N} \setminus \{i\}} \frac{|\mathbf{S}|!(n - |\mathbf{S}| - 1)!}{n!} [v(\mathbf{S} \cup \{i\}) - v(\mathbf{S})] \tag{1}$$

In our method (see §4), the "players" are the variables in our data, and $v(\mathbf{S})$ corresponds to a p-value (see Eq. 3). The Shapley value is widely recognised as a fair solution to credit attribution, as it satisfies four axioms that underlie its fairness definition: symmetry, efficiency, law of aggregation and the null player corollary.[2]

We note that the weighting factor in Eq. 1 ensures equal treatment of coalitions of the same size. This guarantees that the value assigned to each player (or variable, in our case) depends solely on their marginal contribution, regardless of the coalition they join.

3. PC-based Methods: State-of-the-art

The literature on CSL is broadly divided into three main approaches: constraint-based, score-based and Functional Causal Model (FCM)-based methods. In this work, we propose a novel method to improve constraint-based CSL, hence we focus on this category here. For overviews of all approaches we refer the reader to (Glymour et al., 2019; Vowels et al., 2022; Zanga et al., 2022).

Constraint-based CSL algorithms are based on CITs and graphical rules based on the d-separation criterion (Pearl, 2009). The PC-algorithm (Spirtes et al., 2000) operates under the assumptions of acyclicity, sufficiency, and faithfulness. It consists of three steps: 1) building a skeleton of the graph via adjacency search; 2) analysing UTs in the skeleton and orienting them as v-structures, and 3) orienting as many of the remaining undirected edges without creating new v-structures or cycles, using the propagation rules from (Meek, 1995). The algorithm is computationally efficient, especially for sparse graphs, and has been shown to be sound, complete (Spirtes et al., 2000) and consistent in the sample limit (Kalisch and Bühlmann, 2007; Harris and Drton, 2013). However, with finite samples, its results can vary depending on the variables' ordering.

To address this important limitation, PC-Stable (Colombo and Maathuis, 2014), renders the first step of PC order-independent by removing edges only after all tests with a given conditioning set size are performed. Tsagris (2019) instead uses one of the speed-up heuristics from (Spirtes et al., 2000, 5.4.2.4, Heuristic 3) which prioritises the strongest adjacencies when choosing the next test to perform, according to some probabilistic measure. Abellán et al. (2006) also propose to choose edges using a measure of strength, Bayesian in this case, (i) between groups of three adjacent variables with some inconsistent test or (ii) to study the removal of an edge by determining a minimum size cut sets between two nodes. For the skeleton step, we adopt the strategy from PC-Stable.

For the second step, Ramsey et al. (2006) break up the faithfulness assumption into adjacency-faithfulness and orientation-faithfulness. Assuming the former (i.e. that the edges are correctly identified) Conservative-PC (CPC) orients v-structures by checking that the latter assumption is satisfied in the data: for a UT $X_1 - X_3 - X_2$, X_3 is deemed a collider only if it is found in none of the separating sets for X_1, X_2. Majority-PC (MPC) (Colombo and Maathuis, 2014) relaxes the orientation-faithfulness check and orients v-structures if the potential collider appears in less than half of the separating sets of the other two nodes. PC-Max (Ramsey, 2016) selects the CIT with the maximum p-value and only orients the v-structure if the conditioning set for this test does not contain the collider under consideration. Tsagris (2019) propose some extra rules: checking for acyclicity (which we adopt in our algorithm too), checking for double colliders that violate orientation-faithfulness and checking for extra colliders created by Rule 1 of (Meek, 1995). Finally, ML4C (Dai et al., 2023) treats the v-structure orientation as a supervised learning problem: it trains a machine learning model on synthetic examples of v-structures and then predicts a binary label to decide upon UTs at test time.

[2] See (Shapley, 1953) for details on the axioms and e.g., (Young, 1985) in regards to the fairness definition.

Overall, CPC is conservative, MPC proposes a middle-ground rule, PC-Max is informed by the observed significance level of the tests, and ML4C introduces a black-box model for the estimation. Our proposed decision rule uses the same information as MPC, CPC and PC-Max, hence we select these as our baselines for comparison. Through our proposed rule, we analyse test results with Shapley values, lowering the dependence on single wrong tests from sample data, thus improving the discovery of v-structures, and the overall accuracy of the estimated causal graph.

Lifting the sufficiency assumption, Fast Causal Inference (FCI) (Spirtes et al., 2000; Colombo et al., 2012) outputs partial ancestral graphs to account for latent confounders. Variants like FCI-Max (Raghu et al., 2018), analogous to PC-Max (Ramsey, 2016), adapt collider identification for this generalised setting. When the acyclicity assumption is relaxed, the CCD algorithm (Richardson and Spirtes, 1999) recovers partially directed, cyclic graphs, and FCI has been shown to extend to cyclic settings as well (Mooij and Claassen, 2020). While we mention these algorithms for completeness, they fall outside the scope of this paper since we focus on acyclic and sufficient systems.

4. Shapley-PC

Shapley Decision Rule. We propose to orient v-structures based on the Shapley value of the variable under consideration to be a collider in a UT. For this, we define a principled decision rule based on game theory, that analyses the behaviour of the p-value of the independence tests between two variables, when adding a candidate collider to the conditioning set. Shapley values are very well suited for the task, in that they calculate the contribution of a player (a variable) upon joining a team (a conditioning set). Note that p-values here are treated as a measure of association between variables, akin to their interpretation and usage in (Tsamardinos et al., 2006).

Let $\mathcal{C}$ be a given skeleton, $X_i - X_j - X_k$ be a UT in $\mathcal{C}$ and

$$\mathbf{N}=\{\mathbf{S}|\mathbf{S} \subseteq \text{adj}(\mathcal{C}, X_i)\setminus\{X_j\} \vee \mathbf{S} \subseteq \text{adj}(\mathcal{C}, X_k)\setminus\{X_j\}\} \tag{2}$$

be the adjacency sets of X_i, X_k. Let n be the number of variables in adj$(\mathcal{C}, X_i) \cup$ adj$(\mathcal{C}, X_k)$. Then, we define the *Shapley Independence Value (SIV)* of X_j in the given UT as follows:

$$\phi_I(X_j, \{X_i, X_k\}) = \sum_{\mathbf{S}\in\mathbf{N}} w^n_{\mathbf{S}}[I(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) - I(X_i, X_k \mid \mathbf{S})] \tag{3}$$

where $w^n_{\mathbf{S}} = \frac{|\mathbf{S}|!(n-|\mathbf{S}|-1)!}{n!}$ is the weighting factor from Eq. 1. Note that this formulation is not guaranteed to respect some properties satisfied by Shapley values in general (Shapley, 1953), in particular the efficiency and symmetry axiom, but is guaranteed to satisfy other such properties, specifically the null player corollary, since it does not depend on the weighting. Although desirable in general, these properties are not fundamental in the context of this work as they are not conducive to identifying colliders (see Lemma 2).

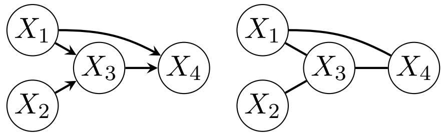
Applying Eq. 3, we recover the marginal contribution $\phi_I(X_j, \{X_i, X_k\})$ of a candidate collider X_j to the p-value of the independence test between the other two variables X_i, X_k in the UT, when it enters the conditioning set $\mathbf{S}$, regardless of the order in which it enters. Following (Hung et al., 1997; Ramsey, 2016; Raghu et al., 2018), the higher the p-value, the higher the likelihood of independence.

Thus, the lower $\phi_I(X_j, \{X_i, X_k\})$, the lower is the contribution of variable X_j to the independence of the common parents X_i, X_k, hence the maximum likelihood of it being a collider. This leads to our *decision rule*:

For any UT $X_i – X_j – X_k$, we declare X_j a collider if it has negative SIV $\phi_I(X_j, \{X_i, X_k\})$.

We illustrate this rule with two examples: one in an idealised setting with perfect independence information and another under realistic conditions with potential inaccuracies.

Example 1 *For illustration, consider the DAG in the figure below (left) and the decision to orient the UT $X_1 - X_3 - X_2$ from the skeleton $\mathcal{C}$ on the right. Here, adj$(\mathcal{C}, X_1) = \{X_3, X_4\}$ and adj$(\mathcal{C}, X_2) = \{X_3\}$ so the following (correct) test results would be considered (e.g. for $\alpha = 0.05$):*



- $I(X_1, X_2) = 1 \geq \alpha$ *(thus* $X_1 \perp\!\!\!\perp X_2$*)*
- $I(X_1, X_2 \mid X_3) = 0 < \alpha$ *(thus* $X_1 \not\perp\!\!\!\perp X_2 \mid X_3$*)*
- $I(X_1, X_2 \mid X_4) = 0 < \alpha$ *(thus* $X_1 \not\perp\!\!\!\perp X_2 \mid X_4$*)*
- $I(X_1, X_2 \mid \{X_3, X_4\}) = 0 < \alpha$ *(so* $X_1 \not\perp\!\!\!\perp X_2 \mid \{X_3, X_4\}$*)*

Then our decision rule would quantify the contribution of X_3 to the independence of X_1, X_2: $\phi_I(X_3, \{X_1, X_2\}) = -0.5$. Here $n=2$, so $w^n_{\mathbf{S}} = 0.5$.

Thus, $\phi_I(X_3, \{X_1, X_2\}) = w^n_{\mathbf{S}}[I(X_1, X_2 \mid X_3) - I(X_1, X_2)] + w^n_{\mathbf{S}}[I(X_1, X_2 \mid \{X_3, X_4\}) - I(X_1, X_2 \mid X_4)] = 0.5(0-1) + 0.5(0-0) = -0.5$. We would therefore find that X_3 has negative contribution to the independence of X_1, X_2, and correctly identify it as a collider.

The correspondence between the value function in Eq. 1 and the one we employ in Eq. 3, comes from fixing X_i, X_k and only changing $\mathbf{S}$ to calculate SIVs for each potential collider X_j. This makes $I(\cdot)$ a function of $\mathbf{S}$ alone, like $v(\cdot)$ in the original formulation of Eq. 1.

With correct tests as in Example 1, the decision rules of all CPC, MPC, PC-Max and Shapley-PC correctly infer the v-structure from the marginal independence $X_1 \perp\!\!\!\perp X_2$ and the conditional dependencies between X_1 and X_2 given all subsets of other variables. However, our decision rule can also deal with more realistic settings, as illustrated next.

Example 2 *Consider the scenario where the true DAG is the same as in Example 1, but the following test results are obtained from data (again $\alpha = 0.05$):*

- $I(X_1, X_2) = 0.7 \geq \alpha$ *(thus* $X_1 \perp\!\!\!\perp X_2$*),*
- $I(X_1, X_2 \mid X_3) = 0.01 < \alpha$ *(thus* $X_1 \not\perp\!\!\!\perp X_2 \mid X_3$*),*
- $I(X_1, X_2 \mid X_4) = 0.1 \geq \alpha$ *(thus* $X_1 \perp\!\!\!\perp X_2 \mid X_4$*),*
- $I(X_1, X_2 \mid \{X_3, X_4\}) = 0.75 \geq \alpha$ *(thus* $X_1 \perp\!\!\!\perp X_2 \mid \{X_3, X_4\}$*).*

The last two tests (wrongly) render an independence.[3] *Here, the SIV for X_3 for the UT $X_1 - X_3 - X_2$ is $\phi_I(X_3, \{X_1, X_2\}) = -0.03$, and our decision rule is still able to correctly identify it as a collider and orient the v-structure. Instead, the decision rules employed by MPC and CPC do not orient it because of the inconsistency between $X_1 \not\perp\!\!\!\perp X_2 \mid \{X_3\}$ and $X_1 \perp\!\!\!\perp X_2 \mid \{X_3, X_4\}$, while PC-Max' decision rule does not orient the v-structure because the maximum p-value test contains X_3.*

If, instead, $I(X_1, X_2 \mid \{X_3, X_4\}) < 0.7$, PC-Max would also have identified the marginal independence $I(X_1, X_2) = 0.7$ as the maximum p-value, and correctly oriented the v-structure.

[3] Note that this scenario is not unlikely from data. $I(X_1, X_2 \mid X_4) = 0.1$ is just above α while $I(X_1, X_2 \mid \{X_3, X_4\}) = 0.75$ is entirely wrong: for increasing sizes of the conditioning sets the data sliced accordingly becomes thinner.

Example 2 *[continued] Suppose now that the same triple was not a v-structure but instead a chain $X_1 \rightarrow X_3 \rightarrow X_2$. With perfect information, we would observe $X_1 \not\perp\!\!\!\perp X_2$, $X_1 \perp\!\!\!\perp X_2 \mid X_3$, $X_1 \not\perp\!\!\!\perp X_2 \mid X_4$, and $X_1 \perp\!\!\!\perp X_2 \mid \{X_3, X_4\}$. Suppose that we obtain the following test results:*

- $I(X_1, X_2) = 0.8$ *(thus $X_1 \perp\!\!\!\perp X_2$),*
- $I(X_1, X_2 \mid X_3) = 0$ *(thus $X_1 \not\perp\!\!\!\perp X_2 \mid X_3$),*
- $I(X_1, X_2 \mid X_4) = 0$ *(thus $X_1 \not\perp\!\!\!\perp X_2 \mid X_4$),*
- $I(X_1, X_2 \mid \{X_3, X_4\}) = 0.7$ *(thus $X_1 \perp\!\!\!\perp X_2 \mid \{X_3, X_4\}$),*

The first two tests are incorrect, and Shapley-PC would calculate a negative contribution for X_3 $(\phi_I = -0.05)$, wrongly deeming it a collider. Similarly, PC-Max would pick up the wrong signal, as the highest p-value is $p = 0.8$ (a wrong test). In contrast, CPC and MPC would not orient the v-structure due to the inconsistency between the tests with $\mathbf{S} = \{X_3\}$ and $\{X_3, X_4\}$. This highlights that the conservativeness of CPC and MPC can be a desirable trait, preventing false positives when orienting v-structures. Their stricter decision rules, which require consistency across multiple conditioning sets, act as safeguards in cases where tests are unreliable. However, their reliance on the consistency across tests can hinder the discovery of plausible v-structures, while Shapley-PC enables a more nuanced assessment based on the tests' strength.

Having showcased how each decision rule has got merits and disadvantages, we next integrate our Shapley-based rule into the PC-Stable algorithm, analyse the theoretical guarantees of the resulting Shapley-PC algorithm, and test its empirical performance.

```
Algorithm 1 Shapley-PC
Input: $I(X_i, X_j \mid \mathbf{S})\ \forall\ X_i, X_j \in \mathbf{V}, \mathbf{S} \subseteq \mathbf{V} \setminus (X_i, X_j)$; $\alpha$
Step 1: Adjacency Search (Colombo and Maathuis, 2014)
$\mathcal{C} := \langle \mathbf{V}, E \rangle, E = \mathbf{V} \times \mathbf{V}$ ;            // Complete Graph over $\mathbf{V}$
for $X_i \in \mathcal{C}$ do
    for $X_j \in adj(\mathcal{C}, X_i)$ do
        for $\mathbf{S} \in \mathbf{N}$ do
            if $I(X_i, X_j \mid \mathbf{S}) \geq \alpha$ then
                $\mathcal{C} := \langle \mathbf{V}, E \setminus (X_i - X_j) \rangle$
return $\mathcal{C}$;                                            // Skeleton
Step 2: Orient v-structures (Our Decision Rule)
for $X_i - X_j - X_k \in \mathcal{C}$ do
    if $\phi_I(X_j, \{X_i, X_k\}) < 0$ then
        if $X_i - X_j - X_k$ not fully directed then
            if do not add a cycle or bi-directed edge then
                orient: $X_i \rightarrow X_j \leftarrow X_k$
return $\mathcal{C}$;                                  // Partially Oriented DAG
Step 3: Pattern Completion (Meek, 1995)
Apply Meek's rules to $\mathcal{C}$ until no more edges can be oriented
return CPDAG;                                          // MEC of the True DAG
```

The Shapley-PC algorithm. We now give our end-to-end CSL algorithm, integrating our novel Shapley-based orientation rule. Our proposed *Shapley-PC algorithm* employs our novel decision rule as sketched in Alg. 1. The first step is the adjacency search that outputs a skeleton $\mathcal{C}$, input of our decision rule in Step 2. Here, we start from a complete graph (line 1) and remove edges until no more independencies are found (lines 2-7).[4]

In Step 2, we calculate SIVs for all candidate colliders in UTs within the skeleton $X_i - X_j - X_k \in \mathcal{C}$ (line 8). While the number of tests in this step is the same as in CPC, MPC, and PC-Max, we obtain more granular information and analyse it using SIVs. Our decision rule (lines 9-12) is to declare X_j a collider if it has a negative contribution to the observed significance level for X_i, X_k. We apply two additional conditions: as in PC-Max, we avoid bi-directed edges by checking existing orientations; additionally, following (Tsagris, 2019), we check for acyclicity before making the orientation. If a bi-directed edge or a cycle is introduced, the UT is not oriented.

Finally, in Step 3 (line 14), groups of three and four adjacent variables are analysed and as many undirected edges as possible are oriented, using the rules from (Meek, 1995).

Compared to our reference versions of the PC algorithm in the literature, our proposed Shapley-PC also focuses on Step 2. Differently from CPC (Ramsey et al., 2006) and MPC (Colombo and Maathuis, 2014), we use a continuous characterisation of the degree of independence rather than a dichotomous (in)dependence relation. Additionally, we add checks for cycles and bi-directed edges that avoid creating invalid DAGs. PC-Max (Ramsey, 2016) also uses p-values to decide about colliders and checks for bi-directed edges. However, PC-Max rule is over-reliant on the test with maximum p-value which makes it more prone to mistakes than our proposed method.

Theoretical Guarantees. Having incorporated our SIVs into the PC algorithm, we now prove that Shapley-PC retains the theoretical guarantees of the original PC: soundness, completeness (Spirtes et al., 2000) and high-dimensional consistency (Kalisch and Bühlmann, 2007). In order to prove soundness and completeness of Shapley-PC, we need a quantitative representation of perfect independence information. We define the concept of perfect conditional independence test, or *perfect CIT*: a test that is able to extract perfect conditional independence information from data.

Definition 1 *For any $X_i, X_j \in \mathbf{V}$ and $\mathbf{S} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$, a* perfect *CIT is defined as:*

$$I_\infty(X_i, X_j \mid \mathbf{S}) = \begin{cases} 1 & \textit{if } X_i \perp\!\!\!\perp X_j \mid \mathbf{S} \\ 0 & \textit{otherwise} \end{cases}$$

We can then show the consistent behaviour of SIVs for evaluating if UTs should be oriented as v-structures (all proofs are in Appendix A).

Lemma 2 *Given a skeleton $\mathcal{C}$, a UT $X_i - X_j - X_k \in \mathcal{C}$, $X_i, X_j, X_k \in \mathbf{V}$, and a perfect CIT I_∞, the SIV of variable X_j $\phi_{I_\infty}(X_j, \{X_i, X_k\}) < 0$ if and only if X_j is a collider for X_i and X_k.*

Lemma 2 states that, given correct conditional independence information (i.e. a perfect CIT), our decision rule to identify colliders based on SIVs is correct. This allows us to prove that Shapley-PC algorithm is sound and complete when assuming faithfulness or infinite data (Ramsey et al., 2006).

[4] Note that, for lack of space, Step 1 is presented in a simplified version: in the full version (Colombo and Maathuis, 2014), the size of the conditioning set progressively increases, for efficiency.

Theorem 3 *Let $\mathbf{P}(\mathbf{V})$ be a joint distribution faithful to a DAG $\mathcal{G} = (\mathbf{V}, E)$, and assume access to perfect conditional independence information for all pairs $(X_i, X_j) \in \mathbf{V}$ given subsets $\mathbf{S} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$. Then the output of Shapley-PC is the CPDAG representing the MEC of $\mathcal{G}$.*

Beyond asymptotic correctness, the original PC algorithm has also been shown to exhibit high-dimensional consistency, in the sample limit with the number of variables growing at a slower rate than the sample, for sparse graphs and multivariate Gaussian distributions (Kalisch and Bühlmann, 2007) or Gaussian copulas (Harris and Drton, 2013). These results are contingent on PC only performing CITs between pairs of variables, with the size of the conditioning sets less or equal to the maximal graph degree, i.e. the maximum of the number of edges linking any node to the others. Our Shapley-PC does not alter these features, hence the consistency results are equally applicable.

By systematically evaluating the marginal effect of adding a variable to different conditioning sets, Shapley values provide a structured framework for assessing the role of a candidate collider in an UT. While our analysis of SIVs focuses on the asymptotic setting, where the weighting scheme of Shapley values does not influence results, the Shapley value framework defines the terms to aggregate and remains key to their theoretical foundation. This highlights the value of SIVs over simpler heuristics, such as empirical averages, and sets the stage for extending our results to finite-sample analyses where the weighting and axioms may become more significant.

Additional Properties. Shapley-PC improves the robustness of v-structure orientation in finite-sample settings by aggregating evidence across multiple CITs with intersecting conditioning sets. Unlike CPC and PC-Max, which rely on a single test and risk false dependencies, SIVs reduce the impact of individual errors, improving reliability. Our Shapley-based decision rule is also order-independent, akin to CPC (Ramsey et al., 2006) and MPC (Colombo and Maathuis, 2014).

Shapley-PC mitigates the reliance on a fixed significance threshold α in Step 2 of the algorithm by aggregating p-values across tests, removing the need for manual tuning (Colombo and Maathuis, 2014). However, as discussed in §2, uniformly distributed p-values under the null hypothesis can produce false signals, which could be mitigated by transforming them to probabilities (Claassen and Heskes, 2012), though this is out of scope for this paper.

In terms of computational overhead, Shapley-PC computes SIVs without resorting to sampling, as instead commonly done in the machine learning (Lundberg and Lee, 2017). SIVs remain feasible because they are only calculated for UTs, whose number depends on graph density: sparse graphs reduce computational costs. Additionally, the conditioning sets analysed by SIVs (Eq. 2) are a subset of the powerset used in standard Shapley values. As in CPC, MPC, and PC-Max, additional tests in Step 2 depend again on the graph density, but the majority of computation still lies in Step 1 (Ramsey et al., 2006). Table 2 empirically validates this claim.

Finally, Shapley-PC classifies nodes as colliders if their Shapley value is negative, a decision rule that is theoretically sound with infinite data. Alternative SIV-based rules could account for context-driven desiderata, such as favouring minimal SIVs, or below a threshold, in line with heuristics like majority voting (Colombo and Maathuis, 2014) or p-value maximisation (Ramsey, 2016).

5. Empirical Evaluation

We conduct a simulation study to compare Shapley-PC against existing versions of PC in the literature (see §2). For all methods, we use Fisher's Z (Fisher, 1970) as CIT and, in line with (Ramsey, 2016), we decrease the significance threshold for the independence tests for increasing number of nodes

($\alpha = 0.1, 0.05, 0.01$ for $|\mathbf{V}| = 10, 20, 50$, respectively). Details on baselines and implementation, including a comparison with KCI (Zhang et al., 2011) and χ^2(Pearson, 1900) tests are in Appendix B.

Data Generating Process (DGP). Given our theoretical guarantees for faithful and infinite data, in this section, we aim at probing our proposed Shapley-PC in scenarios where the distributions in the data are "close-to-unfaithful to the true graph" (Ramsey et al., 2006), which poses a considerable challenge to reliable causal discovery (Robins et al., 2003; Zhang and Spirtes, 2003). To this end, we adapt the strategy proposed in (Ramsey et al., 2006), and generate data with a proportion of weak links, likely to lead to violations of orientation-faithfulness as defined in (Ramsey et al., 2006), whereby inconsistent separating sets are retrieved from the independence tests.

The procedure is as follows. In each experiment, we first generate 10 random graphs for each combination of three parameters: graph type Erdös-Rényi (ER (Erdős and Rényi, 1959)) and Scale Free (SF (Barabási and Albert, 1999)), number of nodes $|\mathbf{V}| \in \{10, 20, 50\}$ and density $d = \{1, 2, 4\}$, with $|E| = |\mathbf{V}| \times d$. Graphs have a maximum degree of 10. Given the ground truth DAG, we simulate 4 different additive noise Structural Equation Models (SEMs) of the type $X_j = f_j(\text{pa}(\mathcal{G}, X_j)) + u_j$ for all $j \in [1, \ldots, |\mathbf{V}|]$ in topological order. In the SEMs, f_j is linear, with coefficients $\mathbf{W}$ initialised from a uniform distribution with coefficients $[-1.5, -0.5] \cup [0.5, 1.5]$ for 95% of the effects, and $[-0.001, 0.001]$ for the remaining 5%. Sampling from the range $[-0.001, 0.001]$ simulates the presence of weak edges. We then derive variables' values through the equation $\mathbf{X} = \mathbf{W}^T\mathbf{X} + u$ where the noise u is generated from Gaussian, Exponential, Gumbel and Uniform distributions. Finally, we vary the number of drawn samples (N) in function of the number of nodes: $N = s \times |\mathbf{V}|$, $s \in \{100, 500, 1000\}$ to check how data-hungry are the different algorithms.[5] More details on the DGPs are provided in Appendix B.4.1.

Evaluation Metrics. In line with (Ramsey et al., 2006; Ramsey, 2016), we analyse the ability to identify v-structures (colliders), which is the focus of our proposed algorithm, alongside the overall performance in recovering the causal arrows of the true graph. For the former, we summarise precision and recall in classifying correct UTs as v-structures, using F1 score (V-F1).[6] Also for the arrows we use F1, but calculated on the number of (in)correct arrowheads (AH-F1). As a reminder, precision is the number of correct classifications out of the estimated ones, while recall is out of the true ones. F1 is the harmonic mean of precision and recall. All the metrics are calculated on the output CPDAGs. Details about the metrics are in Appendix B.3, alongside breakdowns of F1 into precision and recall in §B.4.

Results. We report the results for 10 and 50 nodes graphs of different type and density for $s = 1000$ samples per node in Table 1. Results for $|\mathbf{V}| = 20$, $d = 1$ and $s \in \{100, 500\}$ are provided in Appendix B.4, since $|\mathbf{V}| = 20$ and $s \in \{100, 500\}$ corroborate the results in Table 1, while for $d = 1$, no significant variations across methods were observed.

From Table 1, we can see that Shapley-PC outperforms all other versions of PC for both ER and SF graphs of density $d = 2$ and $d = 4$. We conduct pairwise t-tests for difference in means and highlight the best results in bold if the best method is significantly different from the runner-up, with a significance threshold $\alpha = 0.05$. We additionally show the interval for the observed significance level. Details on the tests are in Appendix B.4.2.

[5] Compared to (Ramsey et al., 2006), we decreased the number of variations in nodes and densities to give space to the analysis of the effect of different types of graphs, noise distributions and sample sizes.

[6] We isolate the errors in orienting correctly identified UTs, in line with adjacency-faithfulness (Ramsey et al., 2006).

Table 1: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}| \in \{10, 50\}$. d is the number of edges per node in the true DAG. Bold if significantly different from the runner-up according to a t-test (see Appendix B.4.2 for details). Observed significance intervals: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

		Method	ER2	ER4	SF2	SF4
$\|\mathbf{V}\|$=10	AH-F1	PC-Stable	0.36±0.28	0.15±0.16	0.67±0.23	0.32±0.27
		CPC	0.42±0.26	0.15±0.17	0.74±0.13	0.34±0.23
		MPC	0.42±0.26	0.15±0.17	0.74±0.13	0.36±0.23
		PC-Max	0.5±0.22	0.07±0.12	0.73±0.2	0.39±0.26
		Shapley-PC	**0.63±0.16** **	**0.25±0.16** **	**0.82±0.08** **	**0.54±0.16** **
	V-F1	PC-Stable	0.46±0.36	0.22±0.32	0.81±0.29	0.49±0.41
		CPC	0.59±0.33	0.23±0.33	0.88±0.18	0.54±0.36
		MPC	0.58±0.34	0.22±0.33	0.88±0.18	0.59±0.36
		PC-Max	0.67±0.3	0.09±0.24	0.9±0.22	0.59±0.4
		Shapley-PC	**0.86±0.16** ***	0.36±0.42	**0.99±0.02** *	**0.84±0.23** **
$\|\mathbf{V}\|$=50	AH-F1	PC-Stable	0.25±0.3	0.04±0.09	0.63±0.37	0.25±0.35
		CPC	0.4±0.36	0.06±0.1	0.53±0.44	0.4±0.39
		MPC	0.35±0.34	0.04±0.09	0.51±0.44	0.36±0.39
		PC-Max	0.63±0.27	0.05±0.11	0.56±0.44	0.59±0.37
		Shapley-PC	**0.75±0.06** **	**0.19±0.15** ***	**0.9±0.04** ***	**0.83±0.07** ***
	V-F1	PC-Stable	0.31±0.37	0.08±0.17	0.67±0.4	0.28±0.4
		CPC	0.5±0.44	0.14±0.24	0.58±0.48	0.46±0.45
		MPC	0.42±0.41	0.08±0.18	0.57±0.49	0.42±0.45
		PC-Max	0.81±0.34	0.12±0.27	0.61±0.48	0.69±0.43
		Shapley-PC	**0.98±0.03** **	**0.48±0.36** ***	**1.0±0.0** ***	**0.99±0.03** ***

Interesting variations in performance can be observed across graphs' types, densities and sizes. Firstly, ER graphs are generally more challenging to retrieve than SF. Secondly, increasing density on ER graphs results to have higher impact on all algorithms than for SF graphs as evidenced by the bigger drop in performance from $d = 2$ to $d = 4$.

Thirdly, for the same density d, a larger number of nodes improves the results. This is because a density of $d = 4$ edges per node means 40 edges for a 10 nodes graph, which is very close to the maximum number of edges for the graph to remain acyclic $(|\mathbf{V}|(|\mathbf{V}| - 1))/2 = 45$. For a graph of 50 nodes, instead, having 200 edges is only about 15% of the way to the maximum number of edges. PC-based methods, generally, perform best on sparse graphs (Kalisch and Bühlmann, 2007), Shapley-PC improves performance on denser graphs. We conjecture that this is because of the increased number of tests necessary to analyse denser graphs and the ability of our method to prevent the judgment of orientation based on single wrong tests.

Besides the performance metrics in Table 1, we compare run times in Table 2. We can see that PC-Stable is the method that scales best with increasing number of nodes, while adding the SIVs calculation on top of the extra tests performed by CPC, MPC and PC-Max does not add considerable

Table 2: Runtime for the experiments in Table 1: median elapsed time in seconds for ER and SF graphs with nodes $|\mathbf{V}| \in \{10, 20, 50\}$.

	ER			**SF**		
$\|\mathbf{V}\|$	10	20	50	10	20	50
PC-Stable	0.1	0.6	4.7	0.1	0.6	2.4
CPC	0.1	0.7	5.5	0.1	0.8	4.3
MPC	0.1	0.7	5.4	0.1	0.8	4.4
PC-Max	0.1	0.7	5.5	0.1	0.8	4.5
Shapley-PC	0.1	0.7	6.3	0.1	0.8	5.2

time (less than 1s for $|\mathbf{V}| = 50$, $\sim 15\%$ higher than PC-Max). Interestingly, the extra testing is more expensive for SF graphs, as demonstrated by the bigger difference, compared to ER, between PC-Stable and all other methods. This is possibly due to the morphology of SF graphs, presenting hubs of highly connected nodes.

Pseudo-Real Data. In addition to the fully simulated data, we conduct experiments on datasets from the `bnlearn` repository. The datasets are sampled from Bayesian Networks with fixed conditional probability tables, provided by previous studies and stored in the repository. We use datasets generated from all three categories available: discrete, Gaussian and Conditional Linear Gaussian Bayesian Networks. Alarm and Insurance are fully discrete, Ecoli70 is fully continuous, while Mehra is mixed. We sample 50000 examples with 10 different seeds to measure performance and confidence intervals.

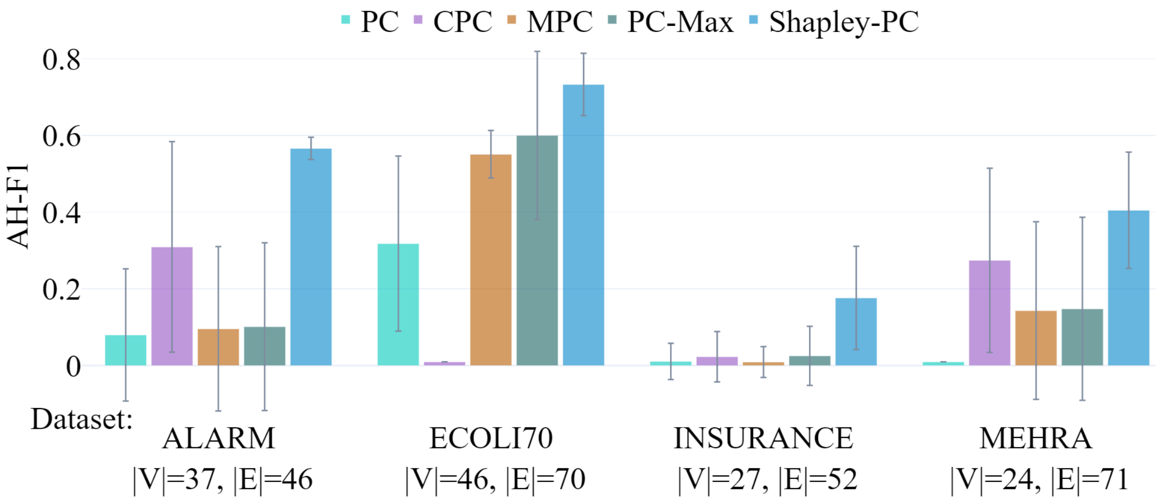


Figure 1: Mean and standard deviation ArrowHead F1 score for four datasets generated from pseudo-real Bayesian Networks from the `bnlearn` repository.

Results for these data are shown in Figure 1 where we report the average ArrowHead F1 scores and their standard deviations. Shapley-PC ranks 1^{st} on all four datasets. However, it is significantly different from all other methods (according to a t-test, $\alpha = 0.05$, see Appendix §B.5) for Alarm and Insurance, while on par ($p = 0.099$) with PC-Max on the Ecoli70 data and on par ($p = 0.166$) with CPC on the Mehra dataset. Details on the datasets, results for V-F1, and for additional datasets where no significant differences were observed, are in Appendix B.5.

6. Conclusion and Future Work

We proposed a decision rule for orienting v-structures in constraint-based CSL algorithms, based on Shapley values, an established concept from game theory (Shapley, 1953). We implemented our decision rule within the novel Shapley-PC algorithm and proved that it maintains the soundness, completeness and consistency guarantees of PC (Spirtes et al., 2000), that Shapley-PC is based on. We carried out an extensive evaluation of Shapley-PC, showing that it outperforms its PC-based predecessors in orienting v-structures and more generally recovering causal directions when the data contains weak links driving orientation unfaithfulness (Ramsey et al., 2006).

Our proposed decision rule takes as input a skeleton $\mathcal{C}$ to then analyse the strength of the associations between adjacent nodes and infer graph orientations. This procedure is directly transferable to constraint-based methods other than PC, possibly with less strict assumptions. One such algorithm is FCI (Spirtes et al., 2000) which lifts both the sufficiency (no latent confounders) and the acyclicity assumptions (Mooij and Claassen, 2020).

The applicability of our proposed method goes beyond constraint-based methods in that we can substitute p-values with any quantitative measure of association between variables. As shown in (Ramsey, 2016), in the context of the PC-Max algorithm, the scores underling score-based CSL methods such as GES (Chickering, 2002) or FGS (Ramsey et al., 2017) can be used in the same guise. Additionally, hybrid methods combine independence tests and scores to estimate causal graphs. An example of such algorithms is MMHC (Tsamardinos et al., 2006) that carries out a skeleton estimation before orienting edges using a score based on p-values. Our method could therefore also be easily extended to such methodologies.

Other directions for future work include the application of our decision rule to the skeleton estimation phase of constraint-based algorithms and to the Meek rules application. In fact, Meek rules can generate cyclic graphs (Tsagris, 2019) and, having to decide between arrows, one could use aggregated evidence in favour or against the orientation, from SIVs. It would also be interesting to study more/less conservative versions of our decision rule, to analyse the informativeness thereof in interactive discovery processes involving humans, and, in line with (Constantinou et al., 2023), to compare it to the other categories of CSL methods in the literature.

Acknowledgments

The authors would like to thank Anna Rapberger, Ruben Menke, Antonio Rago, Avinash Kori, Torgunn Ringsø, Peter Flach, Ben Glocker, and all the anonymous referees for the helpful feedback in the build up to this work. Russo was supported by UK Research and Innovation (grant number EP/S023356/1), in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence. Toni was partially funded by the ERC under the EU's Horizon 2020 research and innovation programme (grant agreement No. 101020934) and by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme. Any views or opinions expressed herein are solely those of the authors.

References

Joaquín Abellán, Manuel Gómez-Olmedo, and Serafín Moral. Some variations on the PC algorithm. In *Third European Workshop on Probabilistic Graphical Models, 12-15 September 2006, Prague, Czech Republic. Electronic Proceedings*, pages 1–8, 2006. URL http://www.utia.cas.cz/files/mtr/pgm06/41_paper.pdf.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999. URL http://www.doi.org/10.1126/science.286.5439.509.

Kailash Budhathoki, Lenon Minorics, Patrick Blöbaum, and Dominik Janzing. Causal structure-based root cause analysis of outliers. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2357–2369. PMLR, 2022. URL https://proceedings.mlr.press/v162/budhathoki22a.html.

G. Casella and R.L. Berger. *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002. ISBN 9780534243128. URL https://books.google.de/books?id=0x_vAAAAMAAJ.

Zhengming Chen, Jie Qiao, Feng Xie, Ruichu Cai, Zhifeng Hao, and Keli Zhang. Testing conditional independence between latent variables by independence residuals. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2024. URL https://doi.org/10.1109/TNNLS.2024.3368561.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *J. Mach. Learn. Res.*, 2:445–498, 2002. URL https://jmlr.org/papers/v2/chickering02a.html.

Tom Claassen and Tom Heskes. A Bayesian Approach to Constraint Based Causal Inference. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pages 207–216. AUAI Press, 2012. URL https://doi.org/10.48550/arXiv.1210.4866.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014. URL https://dl.acm.org/doi/10.5555/2627435.2750365.

Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40 (1):294–321, 2012. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/41713636.

Anthony C. Constantinou, Zhigao Guo, and Neville Kenneth Kitson. The impact of prior knowledge on causal structure learning. *Knowl. Inf. Syst.*, 65(8):3385–3434, 2023. URL https://doi.org/10.1007/s10115-023-01858-x.

Haoyue Dai, Rui Ding, Yuanyuan Jiang, Shi Han, and Dongmei Zhang. ML4C: seeing causality through latent vicinity. In *Proceedings of the 2023 SIAM International Conference on Data Mining, SDM 2023, Minneapolis-St. Paul Twin Cities, MN, USA, April 27-29, 2023*, pages 226–234. SIAM, 2023. URL https://doi.org/10.1137/1.9781611977653.ch26.

Paul Erdős and Alfréd Rényi. On random graphs I. *Publ. math. debrecen*, 6(290-297):18, 1959.

Ronald Aylmer Fisher. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pages 66–70. Springer, 1970.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d770c496aa3da6d2c3f2bd19e7b9d6b-Abstract.html.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Front. Genet., 04 June 2019 Sec. Statistical Genetics and Methodology*, 10:524, 2019. URL https://doi.org/10.3389/fgene.2019.00524.

Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, and Alexander J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 585–592. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/hash/d5cfead94f5350c12c322b5b664544c1-Abstract.html.

Naftali Harris and Mathias Drton. PC algorithm for nonparanormal graphical models. *J. Mach. Learn. Res.*, 14(1):3365–3383, 2013. URL https://dl.acm.org/doi/10.5555/2567709.2567770.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/32e54441e6382a7fbacbbbaf3c450059-Abstract.html.

H. M. James Hung, Robert T. O'Neill, Peter Bauer, and Karl Kohne. The behavior of the p-value when the alternative hypothesis is true. *Biometrics*, 53(1):11–22, 1997. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2533093.

T. Ichiishi. *Game Theory for Economic Analysis*. Economic Theory, Econometrics, and Mathematical Economics. Elsevier Science, 1983. ISBN 9780123701800. URL https://books.google.co.uk/books?id=zFm7AAAAIAAJ.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636, 2007. URL https://dl.acm.org/doi/10.5555/1314498.1314520.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Quebec, Canada, August 18-20, 1995*, pages 403–410. Morgan Kaufmann, 1995. URL https://doi.org/10.48550/arXiv.1302.4972.

Joris M. Mooij and Tom Claassen. Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020*, volume 124 of *Proceedings of Machine Learning Research*, pages 1159–1168. AUAI Press, 2020. URL http://proceedings.mlr.press/v124/m-mooij20a.html.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009. URL https://doi.org/10.1017/CBO9780511803161.

Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302):157–175, 1900.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017. URL https://mitpress.mit.edu/9780262037310/elements-of-causal-inference/.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Comput.*, 27(3):771–799, 2015. URL https://doi.org/10.1162/NECO_a_00708.

Vineet K. Raghu, Joseph D. Ramsey, Alison Morris, Dimitrios V. Manatakis, Peter Spirtes, Panos K. Chrysanthis, Clark Glymour, and Panayiotis V. Benos. Comparison of strategies for scalable causal discovery of latent variable models from mixed data. *Int. J. Data Sci. Anal.*, 6(1):33–45, 2018. URL https://doi.org/10.1007/s41060-018-0104-3.

Joseph Ramsey. Improving accuracy and scalability of the pc algorithm by maximizing p-value. *CoRR abs/1610.00378*, 2016. URL https://doi.org/10.48550/arXiv.1610.00378.

Joseph D. Ramsey, Jiji Zhang, and Peter Spirtes. Adjacency-faithfulness and conservative causal inference. In *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA, July 13-16, 2006*. AUAI Press, 2006. URL https://www.cmu.edu/dietrich/philosophy/docs/spirtes/uai06.pdf.

Joseph D. Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *Int. J. Data Sci. Anal.*, 3(2):121–129, 2017. URL https://doi.org/10.1007/s41060-016-0032-z.

Thomas Richardson and Peter Spirtes. Automated discovery of linear feedback models. In *Computation, Causation, and Discovery*. AAAI Press, 05 1999. ISBN 9780262315821. URL https://doi.org/10.7551/mitpress/2006.003.0010.

James Robins, Richard Scheines, Peter Spirtes, and Larry Wasserman. Uniform consistency in causal inference. *Biometrika*, 90, 09 2003. URL https://doi.org/10.1093/biomet/90.3.491.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proc. IEEE*, 109(5): 612–634, 2021. URL https://doi.org/10.1109/JPROC.2021.3058954.

Lloyd S Shapley. A value for n-person games (1953). *Contribution to the Theory of Games*, 1953. URL https://www.rand.org/content/dam/rand/pubs/papers/2021/P295.pdf.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000. ISBN 978-0-262-19440-2. URL https://www.cs.cmu.edu/afs/andrew/scs/cs/15-381/archive/OldFiles/lib/cvsub/.g/group/sdss/.g/group2/g/scottd/fullbook.pdf.

Jacopo Teneggi, Beepul Bharti, Yaniv Romano, and Jeremias Sulam. SHAP-XRT: the shapley value meets conditional independence testing. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=WFtTpQ47A7.

Michail Tsagris. Bayesian network learning with the PC algorithm: An improved and correct variation. *Appl. Artif. Intell.*, 33(2):101–123, 2019. URL https://doi.org/10.1080/08839514.2018.1526760.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.*, 65(1):31–78, 2006. URL https://doi.org/10.1007/s10994-006-6889-7.

Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. *ACM Comput. Surv.*, 55(4), November 2022. ISSN 0360-0300. URL https://doi.org/10.1145/3527154.

H. Peyton Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14(2):65–72, 1985. URL https://doi.org/10.1007/BF01769885.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: Theory and practice. *Int. J. Approx. Reason.*, 151:101–129, 2022. URL https://doi.org/10.1016/j.ijar.2022.09.004.

Hao Zhang, Yewei Xia, Kun Zhang, Shuigeng Zhou, and Jihong Guan. Conditional independence test based on residual similarity. *ACM Trans. Knowl. Discov. Data*, 17(8):117:1–117:18, 2023. URL https://doi.org/10.1145/3593810.

Jiji Zhang and Peter Spirtes. Strong faithfulness and uniform consistency in causal inference. In *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, Acapulco, Mexico, August 7-10 2003*, pages 632–639. Morgan Kaufmann, 2003. URL https://dl.acm.org/doi/pdf/10.5555/2100584.2100661.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 804–813. AUAI Press, 2011. URL https://doi.org/10.48550/arXiv.1202.3775.

Appendix A. Proofs

Lemma 2 *Given a skeleton $\mathcal{C}$, a UT $X_i - X_j - X_k \in \mathcal{C}$, $X_i, X_j, X_k \in \mathbf{V}$, and a perfect CIT I_∞, the SIV of variable X_j $\phi_{I_\infty}(X_j, \{X_i, X_k\}) < 0$ if and only if X_j is a collider for X_i and X_k.*

Proof Assume a perfect CIT I_∞ (Def. 1). The marginal contribution of $X_j \in \mathbf{V}$ for a conditioning set $\mathbf{S} \in \mathbf{C}$ (Eq. 2) is: $\Delta_{X_j}(\mathbf{S}) = I_\infty(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) - I_\infty(X_i, X_k \mid \mathbf{S})$.

(⇒) Assume X_j is indeed a collider for X_i, X_k and consider the cases based on whether $\mathbf{S}$ blocks paths other than the one passing through X_j:
Case 1 ($\mathbf{S}$ blocks all other paths)
Conditioning on X_j opens a previously blocked path, inducing dependence:

$$I_\infty(X_i, X_k \mid \mathbf{S}) = 1, \quad I_\infty(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) = 0, \quad \Delta_{X_j}(\mathbf{S}) = -1.$$

Case 2 ($\mathbf{S}$ does not block all other paths)
Conditioning on X_j opens an additional path, not altering dependence:

$$I_\infty(X_i, X_k \mid \mathbf{S}) = I_\infty(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) = 0, \quad \Delta_{X_j}(\mathbf{S}) = 0.$$

Given the UT configuration, $X_i \notin \text{adj}(X_k)$, and adjacency faithfulness ensures that at least one $\mathbf{S}$ blocks all paths. Hence:

$$\phi_{I_\infty}(X_j, \{X_i, X_k\}) = \sum_{\mathbf{S} \in \mathbf{C}} w_{\mathbf{S}}^n \Delta_{X_j}(\mathbf{S}) < 0.$$

(⇐) Assume, instead, that X_j is not a collider for X_i, X_k, and again by case distinction:
Case 1 ($\mathbf{S}$ blocks all other paths)
Conditioning on X_j blocks the only open path, inducing independence:

$$I_\infty(X_i, X_k \mid \mathbf{S}) = 0, \quad I_\infty(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) = 1, \quad \Delta_{X_j}(\mathbf{S}) = 1.$$

Case 2 ($\mathbf{S}$ does not block all other paths)
Conditioning on X_j blocks one of the open paths, but does not alter dependence:

$$I_\infty(X_i, X_k \mid \mathbf{S}) = I_\infty(X_i, X_k \mid \mathbf{S} \cup \{X_j\}) = 0, \quad \Delta_{X_j}(\mathbf{S}) = 0.$$

By adjacency faithfulness:

$$\phi_{I_\infty}(X_j, \{X_i, X_k\}) = \sum_{\mathbf{S}\in\mathbf{C}} w^n_{\mathbf{S}} \Delta_{X_j}(\mathbf{S}) > 0.$$

Therefore, $\phi_{I_\infty}(X_j, \{X_i, X_k\}) < 0$ if and only if X_j is a collider for X_i and X_k. ■

Theorem 3 *Let $\mathbf{P}(\mathbf{V})$ be a joint distribution faithful to a DAG $\mathcal{G} = (\mathbf{V}, E)$, and assume access to perfect conditional independence information for all pairs $(X_i, X_j) \in \mathbf{V}$ given subsets $\mathbf{S} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$. Then the output of Shapley-PC is the CPDAG representing the MEC of $\mathcal{G}$.*

Proof The proof follows straightforwardly from Lemma 2 and two additional results in the literature. Assume faithfulness and perfect CITs, then:

- **Step 1**: The skeleton is guaranteed to be correct (Colombo and Maathuis, 2014, Thm. 2).
- **Step 2**: Given a correct skeleton, by Lemma 2, $\phi_I(X_j, \{X_i, X_k\}) < 0$ correctly identifies colliders in v-structures.
- **Step 3**: Given a PDAG, Meek's rules, are sound and complete (Meek, 1995, Thm. 2 and 3).

Thus, Shapley-PC outputs the correct CPDAG. ■

Appendix B. Details on Experiments

In this section we provide additional details for the experiments in §5 of the main text.

B.1. Baselines

We used the following four baselines with respective implementations (see §2 and §3 for context):

- PC-Stable[7] (Colombo and Maathuis, 2014) consists of three steps:
  1. building a skeleton of the graph via adjacency search: conditional independence tests are performed for each pair of variables in the data. For efficiency, the algorithm starts by performing marginal independence tests (empty conditioning set) and gradually increases the size of the conditioning set once all pairs of variables have been tested. If an independence is found for a pair of variables, the edge is removed after all variables have been tested for that conditioning set size. The separating set is stored for the pair of variables found independent.[8] This step outputs a skeleton $\mathcal{C}$.
  2. for each unshielded triple (UT) in the skeleton $X_i - X_j - X_k \in \mathcal{C}$ output of step 1, the UT is oriented as a v-structure $X_i \rightarrow X_j \leftarrow X_k$ if X_j is in the separating set for variables X_i, X_k.

[7] https://github.com/py-why/causal-learn

[8] This is the difference of PC-Stable with the original PC (Spirtes et al., 2000), that instead removes edges as soon as an independence is found, being then subject to the order in which the variables are tested.

3. all triangles (groups of three adjacent variables) and kites (group of four adjacent variables) are analysed with the rules for patterns (Meek, 1995). If certain configurations are obtained in step 2, further orientations are performed on the remaining undirected edges. The application of these rules returns a sound and complete CPDAG that represent the true DAG (Meek, 1995).

- Conservative-PC (CPC)[9] (Ramsey et al., 2006) is a modification of the PC(-Stable) algorithm. Step 1 can be the original or the stable version and 3 is the same, while the v-structure orientation rule is the main proposal. Ramsey et al. (2006) break up the faithfulness assumption into adjacency-faithfulness and orientation-faithfulness. Assuming the former (i.e. that the edges are correctly identified) CPC orients v-structures by checking that the latter assumption is satisfied in the data: for a UT $X_i - X_j - X_k$, X_j is deemed a collider only if it is found in none of the separating sets for X_i, X_k.

- Majority-PC (MPC)[10] (Colombo and Maathuis, 2014) relaxes the orientation-faithfulness check of CPC and orients v-structures if the potential collider appears in less than half of the separating sets of the other two nodes.

- PC-Max[11] (Ramsey, 2016) again only modifies Step 2 of PC(-Stable). It selects the CIT with the maximum p-value for a given UT and only orients it as a v-structure if the conditioning set for the selected CIT does not contain the variable under consideration.

B.2. Implementation

We provide an implementation of Shapley-PC based on the causal-learn python package.[12] Within causal-learn, we define a new PC function that accommodates our decision rule. The code is available at the following repository: https://github.com/briziorusso/ShapleyPC In the repository, we also made available the code to reproduce all experiments and we saved all the plots, presented herein and in the main text, in HTML format. Downloading and opening them in a browser allows the inspection of all the numbers behind the plots in an interactive way.

Hyperparameters For all the methods we used Fisher's Z test (Fisher, 1970), as implemented in causal-learn, with significance threshold $\alpha = 0.01$ for the bnlearn dataset and with decreasing α for increasing number of nodes in the fully synthetic simulations: we used $\alpha = 0.1, 0.05, 0.01$ for number of nodes $|\mathbf{V}| = 10, 20, 50$, respectively.

Computing infrastructure All experiments were ran on Intel(R) Xeon(R) w5-2455X CPU with 4600 max MHz and 128GB of RAM. We used python 3.10.12 on Ubuntu 22.04.

[9] our implementation, https://github.com/briziorusso/ShapleyPC, based on https://github.com/py-why/causal-learn

[10] our implementation, https://github.com/briziorusso/ShapleyPC, based on https://github.com/py-why/causal-learn

[11] https://github.com/py-why/causal-learn

[12] https://github.com/py-why/causal-learn

B.3. Evaluation Metrics

In line with (Ramsey et al., 2006; Ramsey, 2016), we analyse the ability to identify v-structures (colliders), which is the focus of our proposed algorithm, alongside the overall performance in recovering the causal arrows of the true graph. All the metrics are calculated on the output CPDAGs hence the (binary) adjacency matrices can have entries for both (X_i, X_j) and (X_j, X_i), in which case the edge is undirected.

For the accuracy in classifying v-structures, we use precision and recall in classifying correctly identified UTs and summarise it with the F1 Score. Specifically:

- V-Precision = V-TP/(V-TP + V-FP)
- V-Recall = V-TP/(V-TP + V-FN)
- V-F1 Score = 2 × $($V-P × V-R$)/($V-P + V-R$)$

where V-True Positive (V-TP) is the number of correctly estimated v-structures; V-False Positive (V-FP) is the number of UTs wrongly deemed as v-structures; V-False Negative (V-FN) is the number of v-structures not deemed as such.

Also to evaluate the accuracy in identifying causal directions in the true graph we use F1, but calculated on the number of (in)correct arrowheads (AH-F1) as follows:

- AH-Precision = AH-TP/(AH-TP + AH-FP)
- AH-Recall = AH-TP/(AH-TP + AH-FN)
- AH-F1 Score = 2 × $($AH-P × AH-R$)/($AH-P + AH-R$)$

where AH-True Positive (AH-TP) is the number of estimated edges with correct direction; False Positive (AH-FP) is the number of extra arrowheads; False Negative (AH-FN) is the number of missing arrowheads.

In addition to the metrics focusing on orientations and v-structures, we report two other commonly used metrics in CSL (see e.g. Constantinou et al. (2023)): Structural Hamming Distance (SHD) (Tsamardinos et al., 2006) and Structural Intervention Distance (SID) (Peters and Bühlmann, 2015).

SHD = E + M + R, where Extra (E) is the set of extra edges, Missing (M) are the ones missing from the skeleton of the estimated graph and Reversed (R) have incorrect direction.

SID quantifies the agreement to a causal graph in terms of interventional distributions. It aims at quantifying the incorrect causal inference estimations stemming out of a mistake in the causal graph estimation, akin to a downstream task error on a pre-processing step where the task is causal inference and the pre-processing step is finding the right graph to inform it. Both missing/extra edges and incorrect orientation will play a role in the incorrect causal inferences.

B.4. Synthetic Data

Here we provide detail for the simulation study presented in §5, in particular Table 1 and 2.

B.4.1. DGP DETAILS

In each experiment, we first generate 10 random graphs with maximum degree of 10 for each combination of three parameters:

- graph type: Erdös-Rényi (ER (Erdős and Rényi, 1959)) and Scale Free (SF (Barabási and Albert, 1999));
- number of nodes: $|\mathbf{V}| \in \{10, 20, 50\}$;
- density: $d = \{1, 2, 4\}$, with $|E| = |\mathbf{V}| \times d$.

Given the ground truth DAGs $\mathcal{G}$, we simulate Structural Equation Models (SEMs) belonging to the Additive Noise Model, formally:

$$X_j = f_j(\text{pa}(\mathcal{G}, X_j)) + u_j \quad \forall\, j \in [1, \ldots, |\mathbf{V}|] \tag{4}$$

where f_j is linear function with coefficients $\mathbf{W}$ and u_j are samples from a noise distribution.

The coefficients $\mathbf{W}$ are sampled from a uniform distribution with parameters $[-1.5, -0.5] \cup [0.5, 1.5]$ for 95% of the effects, and $[-0.001, 0.001]$ for the remaining 5%. Sampling effect magnitudes from the range $[-0.001, 0.001]$ simulates the presence of weak edges to induce violations of orientation-faithfulness (see §5 and (Ramsey et al., 2006)).

Finally, we sample $\mathbf{X} = \mathbf{W}^T\mathbf{X} + u$ where the noise u is generated from the following four distributions:

- Gaussian: $u \sim \mathcal{N}(0, 1)$
- Exponential: $u \sim E(1)$
- Gumbel: $u \sim G(0, 1)$
- Uniform: $u \sim U(-1, 1)$

We vary the number of drawn samples (N) in function of the number of nodes $N = s \times |\mathbf{V}|$, $s \in \{100, 500, 1000\}$ and refer to s as the proportional sample size. After sampling from the described DGPs we standardise the data using the standard scaler from `sklearn`.[13] Code to reproduce the simulated data is provided in our repository.

B.4.2. STATISTICAL TESTS

Here we provide details for the statistical tests used to measure the significance of the difference in the results presented in Table 1 in the main text. In Tables 6, 8, 7 and 9 we provide t-statistics and p-values for graphs ER and SF graphs of 10 and 50 nodes, respectively.

In each table we present pairwise comparisons of means, for V-F1 and AH-F1 scores presented in Table 1 of the main text. We use two-sample, unequal variance t-tests, with degrees of freedom of 39 (10 seeds and 4 noise distributions, minus 1).

B.4.3. ADDITIONAL RESULTS

Here we provide additional results that were not presented in the main text for space constraints. The results corroborate the ones presented in the main text.

[13] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Table 3: AH-F1 and V-F1 Scores ± std for ER1 and SF1 graphs of nodes $\vert\mathbf{V}\vert \in \{10, 50\}$. No significant differences according to a t-test, $\alpha = 0.05$.

	Method	**ER1**		**SF1**	
		$\vert\mathbf{V}\vert = 10$	$\vert\mathbf{V}\vert = 50$	$\vert\mathbf{V}\vert = 10$	$\vert\mathbf{V}\vert = 50$
AH-F1	PC-Stable	0.91±0.14	0.82±0.14	0.9±0.12	0.91±0.07
	CPC	0.97±0.08	0.87±0.11	0.99±0.03	1.0±0.01
	MPC	0.96±0.09	0.87±0.11	0.98±0.05	0.99±0.02
	PC-Max	0.96±0.09	0.87±0.11	0.95±0.08	0.96±0.04
	Shapley-PC	0.93±0.12	0.9±0.07	0.95±0.08	0.96±0.04
V-F1	PC-Stable	0.97±0.09	0.92±0.16	0.99±0.03	0.98±0.04
	CPC	1.0±0.0	0.93±0.13	1.0±0.0	1.0±0.0
	MPC	1.0±0.0	0.93±0.13	1.0±0.01	1.0±0.0
	PC-Max	1.0±0.0	0.97±0.08	1.0±0.0	1.0±0.0
	Shapley-PC	0.99±0.03	1.0±0.02	1.0±0.0	1.0±0.0

Sparsest Graphs (d=1) The results presented in Table 1 in the main text show AH-F1 and V-F1 for ER and SF graphs of density $d = \{2, 4\}$. Here we complete the picture and provide results for the sparsest graphs analysed: $d = 1$. We can see from Table 3, that all methods perform quite well of very sparse graphs. This result is in line with (Kalisch and Bühlmann, 2007). Given the limited opportunity for improvement, no significant differences between the various methods is observed.

Graphs of 20 Nodes The results presented in Table 1 in the main text show AH-F1 and V-F1 for ER and SF graphs of 10 anf 50 nodes. Here we complete the picture and provide results for graphs of 20 nodes. The results corroborate the ones presented in the main paper. From Table 4, we can see that Shapley-PC outperforms all other methods on ER4, SF2 and SF4. On ER2 it is not significantly different from PC-Max (according to a t-test, $\alpha = 0.05$), but better than all other methods.

Proportional Sample Size The results presented in Table 1 in the main text show AH-F1 and V-F1 for proportional sample size $s = 1000$. The proportional sample size is the number of samples per node in the dataset, with total number of samples $N = s * |V|$. Here we show the trends for $s \in \{100, 500, 1000\}$, in Fig. 2 and Fig. 3 for AH-F1 and V-F1, respectively. From the plots, we notice that the trends are mostly flat, demonstrating that none of the methods compared is very "data-hungry."

Noise Distributions Plots by noise distribution are provided as interactive plots in our repository (`https://github.com/briziorusso/ShapleyPC`), as they would not be easily displayed on A4 paper. No majour differences are observed across noise distributions.

B.4.4. ADDITIONAL METRICS

Precision and Recall The results presented in Table 1 in the main text show F1 scores for arrowheads and v-structures' classification. Here we provide a breakdown of the F1 scores into their components: Precision and Recall. In Table 10 we can see that Shapley-PC is significantly better

Table 4: AH-F1 and V-F1 Scores ± std for ER2, ER4, SF2, and SF4 graphs of 20 nodes. Bold if significantly different from the runner-up (according to a t-test, $\alpha = 0.05$).

	Method	ER2	ER4	SF2	SF4
AH-F1	PC-Stable	0.28±0.26	0.08±0.10	0.59±0.30	0.22±0.25
	CPC	0.37±0.25	0.11±0.11	0.52±0.35	0.25±0.27
	MPC	0.31±0.26	0.11±0.11	0.50±0.36	0.25±0.26
	PC-Max	**0.51±0.24**	0.08±0.11	0.63±0.32	0.37±0.28
	Shapley-PC	**0.56±0.19**	**0.17±0.12**	**0.81±0.04**	**0.61±0.09**
V-F1	PC-Stable	0.35±0.35	0.14±0.21	0.71±0.37	0.31±0.38
	CPC	0.52±0.36	0.23±0.26	0.63±0.43	0.36±0.40
	MPC	0.42±0.37	0.19±0.24	0.60±0.43	0.36±0.38
	PC-Max	**0.71±0.34**	0.22±0.32	0.77±0.39	0.54±0.41
	Shapley-PC	**0.82±0.27**	**0.42±0.33**	**0.99±0.02**	**0.97±0.05**

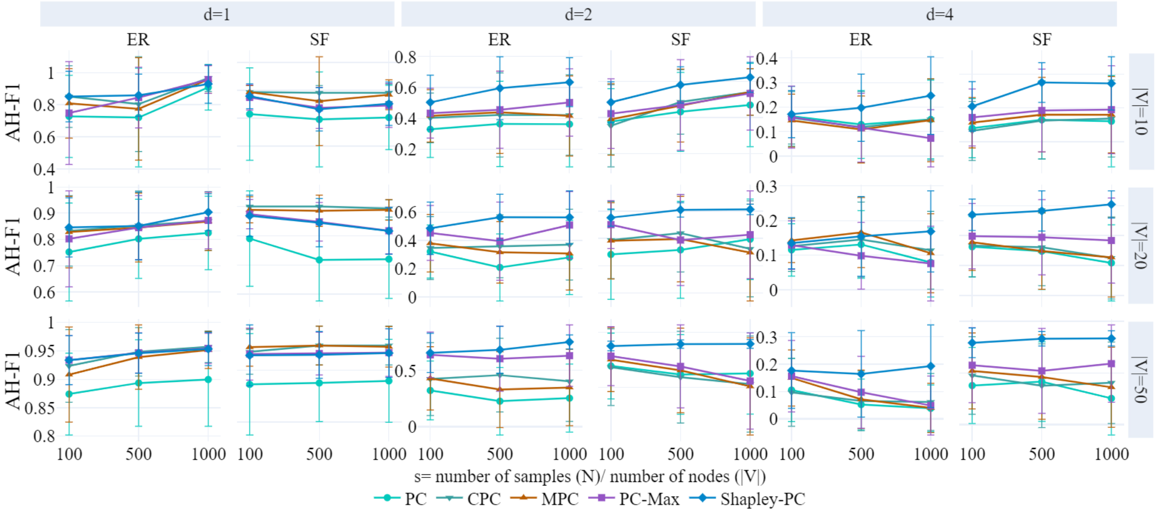


Figure 2: ArrowHead F1 scores by proportional sample size ($s \in \{100, 500, 1000\}$) for the fully synthetic data in §5.

(according to a t-test, $\alpha = 0.05$) than all other methods, for both precision and recall wrt ArrowHead classifications. Results are analogous for v-structure classifications, shown in Table 11.

SHD and SID In addition to the results that focus on the v-structures and arrowhead orientations, as presented in main text and the additional results in this section, we also present results using SID (Peters and Bühlmann, 2015) in Table 12 and SHD (Tsamardinos et al., 2006) in Table 13. Both metrics measure error, hence the lower the better. Since we calculate all metrics on CPDAGs, SID

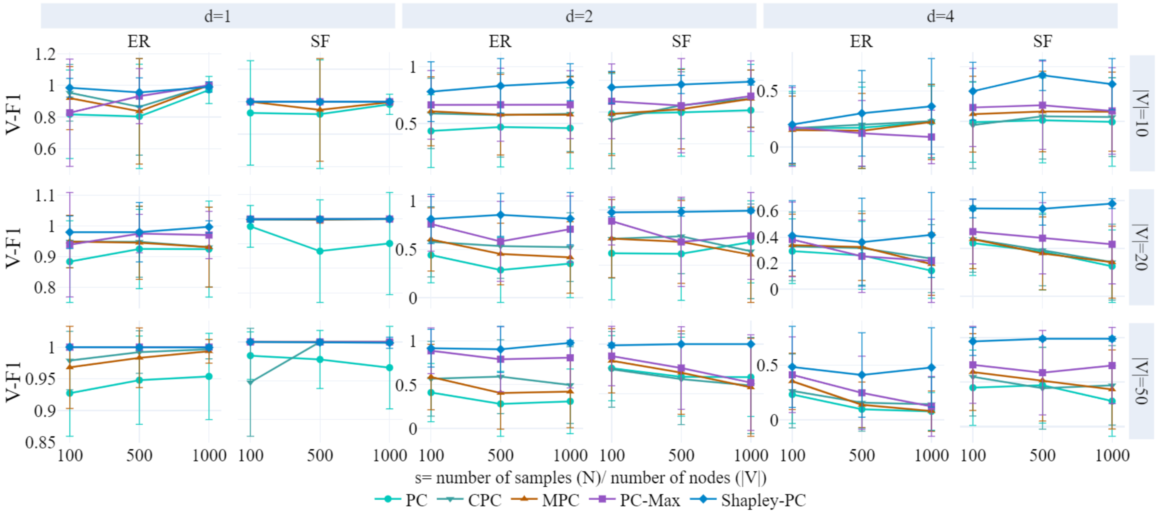


Figure 3: V-structure F1 scores by proportional sample size ($s \in \{100, 500, 1000\}$) for the fully synthetic data in §5.

estimates a best and worst scenario (SID-Low and High, respectively) depending on the orientation of the undirected edges in the output CPDAG.

From Table 12, we can see that Shapley-PC is significantly better than all other methods for the best case scenario (SID-Low) on 10 nodes graphs. For the worst case scenario (SID-High) all methods are on par for ER4, and Shapley-PC and PC-Max are on par, and better than all others for SF4. Shapley-PC is significantly better than all other methods for the remaining types of 10 nodes graphs. For 50 nodes graphs, which are sparser (see results discussion in the main text §5), Shapley-PC is better than PC, CPC and MPC, but not significantly better than PC-Max.

Comparison of Shapley-PC with our baselines based SHD are shown in Table 13. We can see that PC-Stable is significantly worse than all other methods for ER1, ER2 and SF2, while no significant differences are observed for the remaining three graph types.

Overall, Shapley-PC is never worse than any other baseline, based on both SID and SHD.

We remark that SHD and SID are more general graphical metrics, that do not take into account that there can be errors in skeleton and orientations, and that these can be isolated one from the other. With ArrowHead F1, we measure the orientation capabilities of the different methods, that with these metrics are confounded by errors in the skeleton.

B.5. Pseudo-Real Data

B.5.1. DATASETS DETAILS

For the experiments on pseudo-real data, we used ten datasets from the `bnlearn` repository which is widely used for research in CSL. The datasets are sampled from Bayesian Networks (BN) with fixed conditional probability tables stored in the repository. The BNs used in our experiments are from all three categories in the repository: Discrete, Gaussian and Conditional Linear Gaussian. The

Table 5: Details of the Bayesian Network from bnlearn used to generate the pseudo-real datasets in §5. "Cat" and "Cont" are counts of the categorical and continuous variables in the produced datasets, respectively. $|N|$ and $|E|$ are the number of nodes and edges in the graph and d = $|E|/|N|$ their proportion.

Dataset Name	Type	Cat	Cont	$\|N\|$	$\|E\|$	d
ALARM	Discrete	37	0	37	46	1.24
CHILD	Discrete	20	0	20	25	1.25
HEPAR2	Discrete	70	0	70	123	1.76
INSURANCE	Discrete	27	0	27	52	1.93
ARTH150	Gaussian	0	107	107	150	1.40
ECOLI70	Gaussian	0	46	46	70	1.52
MAGIC-IRRI	Gaussian	0	64	64	102	1.59
MAGIC-NIAB	Gaussian	0	44	44	66	1.50
MEHRA	Linear Gaussian	8	16	24	71	2.96
SANGIOVESE	Linear Gaussian	1	14	15	55	3.67

number of nodes vary from 15 to 107 and the number of edges from 25 to 150. Details on the number of nodes, edges and density of the DAGs underlying these data are reported in Table 5, together with links to a more detailed description from the bnlearn repository.

Having downloaded all the .bif or .rda files from the repository, we load the Bayesian network and the associated conditional probability tables and sample 50000 observations, with 10 different seeds. We encoded the labels using the label encoder from sklearn for categorical variables and applied standard scaling from sklearn for the continuous ones. The BNs, together with the code to reproduce the dataset, is provided in our repository (https://github.com/briziorusso/ShapleyPC/datasets).

B.5.2. STATISTICAL TESTS

Here we present details of the statistical tests used to measure the significance of the difference in the results presented in Fig. 1 in the main text. In Tables 14 and 15 we provide t-statistics and p-values for the Alarm, Insurance, Ecoli70 and Mehra datasets. In each table we present pairwise comparisons of means (shown in brackets together with standard deviations), for the AH-F1 and V-F1 presented in Fig. 1 of the main text.

Additional Metrics In Fig. 1 in the main text, we show the results on four of the ten dataset detailed in Table 5, according to ArrowHead F1 Score. In this section we report additional metrics, in line with the experiments on synthetic data. In particular, we visualise V-F1 (Fig. 4), SHD (Fig. 5) and SID (Fig. 6). Precision and Recall are left out because they show very similar trends to AH-F1 and V-F1 presented herein, but are provided in our repository as interactive plots.

In Fig. 4, we report V-F1 scores for the same set of datasets as in the main text. For AH-F1 (Fig. 1 in the main text), Shapley-PC is significantly better than all other methods on Alarm and Insurance. For V-F1, Shapley-PC is better than all other methods on Alarm, Insurance and Mehra. For Ecoli70, we are on par with PC-Max, and better than all others. According to SHD (Fig. 5) and SID (Fig. 6), no significant differences are observed across datasets and methods.

B.5.3. ADDITIONAL DATASETS

The results presented in Fig. 1 show AH-F1 for four datasets out of the ten analysed. We show results for the remaining six datasets (Arth150, Child, Hepar2, Magic-irri, Magic-niab and Sangiovese) in Fig. 7 (AH-F1) and Fig. 8 (V-F1). Out of these six datasets, Shapley-PC results to be significantly better than all other methods according to AH-F1 on Arth150 and Sangiovese. According to V-F1, Shapley-PC is better than all others on Sangiovese, and on par with CPC, improving on all other methods, on Arth150. No significant differences are observed on the remaining four datasets, apart from PC-Stable being worse than all other methods on the Magic-irri and Magic-niab datasets.

Table 6: Two-sample, unequal variance t-tests for difference in means for ER graphs with $|\mathbf{V}|$ = 10. Significance levels: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.', 0.1 ' ' 1. DoF: n_a = n_b = 39.

Type	**Metric**	**Methods**	**Means±Std**	**t**	**p-value**
ER2	**V-F1**	PC-S vs CPC	0.46 ± 0.4 vs 0.59 ± 0.3	-1.64	0.104
		PC-S vs MPC	0.46 ± 0.4 vs 0.58 ± 0.3	-1.54	0.128
		PC-S vs PC-M	0.46 ± 0.4 vs 0.67 ± 0.3	-2.80	0.006**
		PC-S vs SPC	0.46 ± 0.4 vs 0.86 ± 0.2	-6.53	0.000***
		CPC vs MPC	0.59 ± 0.3 vs 0.58 ± 0.3	0.11	0.916
		CPC vs PC-M	0.59 ± 0.3 vs 0.67 ± 0.3	-1.12	0.265
		CPC vs SPC	0.59 ± 0.3 vs 0.86 ± 0.2	-4.73	0.000***
		MPC vs PC-M	0.58 ± 0.3 vs 0.67 ± 0.3	-1.23	0.222
		MPC vs SPC	0.58 ± 0.3 vs 0.86 ± 0.2	-4.85	0.000***
		PC-M vs SPC	0.67 ± 0.3 vs 0.86 ± 0.2	-3.66	0.001***
	AH-F1	PC-S vs CPC	0.36 ± 0.3 vs 0.42 ± 0.3	-0.98	0.331
		PC-S vs MPC	0.36 ± 0.3 vs 0.42 ± 0.3	-0.89	0.377
		PC-S vs PC-M	0.36 ± 0.3 vs 0.50 ± 0.2	-2.52	0.014*
		PC-S vs SPC	0.36 ± 0.3 vs 0.63 ± 0.2	-5.39	0.000***
		CPC vs MPC	0.42 ± 0.3 vs 0.42 ± 0.3	0.09	0.926
		CPC vs PC-M	0.42 ± 0.3 vs 0.50 ± 0.2	-1.52	0.133
		CPC vs SPC	0.42 ± 0.3 vs 0.63 ± 0.2	-4.42	0.000***
		MPC vs PC-M	0.42 ± 0.3 vs 0.50 ± 0.2	-1.62	0.109
		MPC vs SPC	0.42 ± 0.3 vs 0.63 ± 0.2	-4.54	0.000***
		PC-M vs SPC	0.50 ± 0.2 vs 0.63 ± 0.2	-3.08	0.003**
ER4	**V-F1**	PC-S vs CPC	0.23 ± 0.3 vs 0.23 ± 0.3	-0.08	0.940
		PC-S vs MPC	0.23 ± 0.3 vs 0.22 ± 0.3	0.03	0.973
		PC-S vs PC-M	0.23 ± 0.3 vs 0.09 ± 0.2	2.15	0.035*
		PC-S vs SPC	0.23 ± 0.3 vs 0.36 ± 0.4	-1.64	0.106
		CPC vs MPC	0.23 ± 0.3 vs 0.22 ± 0.3	0.11	0.915
		CPC vs PC-M	0.23 ± 0.3 vs 0.09 ± 0.2	2.19	0.032*
		CPC vs SPC	0.23 ± 0.3 vs 0.36 ± 0.4	-1.55	0.125
		MPC vs PC-M	0.22 ± 0.3 vs 0.09 ± 0.2	2.05	0.044*
		MPC vs SPC	0.22 ± 0.3 vs 0.36 ± 0.4	-1.64	0.106
		PC-M vs SPC	0.09 ± 0.2 vs 0.36 ± 0.4	-3.54	0.001***
	AH-F1	PC-S vs CPC	0.15 ± 0.2 vs 0.15 ± 0.2	0.01	0.989
		PC-S vs MPC	0.15 ± 0.2 vs 0.15 ± 0.2	0.09	0.930
		PC-S vs PC-M	0.15 ± 0.2 vs 0.07 ± 0.1	2.45	0.017*
		PC-S vs SPC	0.15 ± 0.2 vs 0.25 ± 0.2	-2.73	0.008**
		CPC vs MPC	0.15 ± 0.2 vs 0.15 ± 0.2	0.07	0.943
		CPC vs PC-M	0.15 ± 0.2 vs 0.07 ± 0.1	2.38	0.020*
		CPC vs SPC	0.15 ± 0.2 vs 0.25 ± 0.2	-2.70	0.009**
		MPC vs PC-M	0.15 ± 0.2 vs 0.07 ± 0.1	2.26	0.027*
		MPC vs SPC	0.15 ± 0.2 vs 0.25 ± 0.2	-2.74	0.008**
		PC-M vs SPC	0.07 ± 0.1 vs 0.25 ± 0.2	-5.59	0.000***

Table 7: Two-sample, unequal variance t-tests for difference in means for SF graphs with $|\mathbf{V}| = 10$. Significance levels: 0 '***', 0.001 '**', 0.01 '*', 0.05 '.', 0.1 ' ' 1. DoF: $n_a = n_b = 39$.

Type	**Metric**	**Methods**	**Means±Std**	**t**	**p-value**
SF2	**V-F1**	PC-S vs CPC	0.81 ± 0.3 vs 0.88 ± 0.2	−1.36	0.179
		PC-S vs MPC	0.81 ± 0.3 vs 0.88 ± 0.2	−1.35	0.183
		PC-S vs PC-M	0.81 ± 0.3 vs 0.90 ± 0.2	−1.55	0.125
		PC-S vs SPC	0.81 ± 0.3 vs 0.99 ± 0.0	−3.93	0.000***
		CPC vs MPC	0.88 ± 0.2 vs 0.88 ± 0.2	0.01	0.989
		CPC vs PC-M	0.88 ± 0.2 vs 0.90 ± 0.2	−0.36	0.721
		CPC vs SPC	0.88 ± 0.2 vs 0.99 ± 0.0	−3.73	0.001***
		MPC vs PC-M	0.88 ± 0.2 vs 0.90 ± 0.2	−0.37	0.712
		MPC vs SPC	0.88 ± 0.2 vs 0.99 ± 0.0	−3.72	0.001***
		PC-M vs SPC	0.90 ± 0.2 vs 0.99 ± 0.0	−2.56	0.014*
	AH-F1	PC-S vs CPC	0.67 ± 0.2 vs 0.74 ± 0.1	−1.72	0.091
		PC-S vs MPC	0.67 ± 0.2 vs 0.74 ± 0.1	−1.73	0.090
		PC-S vs PC-M	0.67 ± 0.2 vs 0.73 ± 0.2	−1.32	0.191
		PC-S vs SPC	0.67 ± 0.2 vs 0.82 ± 0.1	−3.95	0.000***
		CPC vs MPC	0.74 ± 0.1 vs 0.74 ± 0.1	−0.01	0.992
		CPC vs PC-M	0.74 ± 0.1 vs 0.73 ± 0.2	0.20	0.840
		CPC vs SPC	0.74 ± 0.1 vs 0.82 ± 0.1	−3.36	0.001**
		MPC vs PC-M	0.74 ± 0.1 vs 0.73 ± 0.2	0.21	0.834
		MPC vs SPC	0.74 ± 0.1 vs 0.82 ± 0.1	−3.34	0.001**
		PC-M vs SPC	0.73 ± 0.2 vs 0.82 ± 0.1	−2.57	0.013*
SF4	**V-F1**	PC-S vs CPC	0.49 ± 0.4 vs 0.54 ± 0.4	−0.52	0.606
		PC-S vs MPC	0.49 ± 0.4 vs 0.59 ± 0.4	−1.09	0.280
		PC-S vs PC-M	0.49 ± 0.4 vs 0.59 ± 0.4	−1.12	0.266
		PC-S vs SPC	0.49 ± 0.4 vs 0.84 ± 0.2	−4.67	0.000***
		CPC vs MPC	0.54 ± 0.4 vs 0.59 ± 0.4	−0.61	0.546
		CPC vs PC-M	0.54 ± 0.4 vs 0.59 ± 0.4	−0.66	0.511
		CPC vs SPC	0.54 ± 0.4 vs 0.84 ± 0.2	−4.43	0.000***
		MPC vs PC-M	0.59 ± 0.4 vs 0.59 ± 0.4	−0.08	0.936
		MPC vs SPC	0.59 ± 0.4 vs 0.84 ± 0.2	−3.70	0.000***
		PC-M vs SPC	0.59 ± 0.4 vs 0.84 ± 0.2	−3.38	0.001**
	AH-F1	PC-S vs CPC	0.32 ± 0.3 vs 0.34 ± 0.2	−0.29	0.774
		PC-S vs MPC	0.32 ± 0.3 vs 0.36 ± 0.2	−0.69	0.493
		PC-S vs PC-M	0.32 ± 0.3 vs 0.39 ± 0.3	−1.16	0.252
		PC-S vs SPC	0.32 ± 0.3 vs 0.54 ± 0.2	−4.52	0.000***
		CPC vs MPC	0.34 ± 0.2 vs 0.36 ± 0.2	−0.44	0.662
		CPC vs PC-M	0.34 ± 0.2 vs 0.39 ± 0.3	−0.96	0.341
		CPC vs SPC	0.34 ± 0.2 vs 0.54 ± 0.2	−4.74	0.000***
		MPC vs PC-M	0.36 ± 0.2 vs 0.39 ± 0.3	−0.55	0.583
		MPC vs SPC	0.36 ± 0.2 vs 0.54 ± 0.2	−4.25	0.000***
		PC-M vs SPC	0.39 ± 0.3 vs 0.54 ± 0.2	−3.23	0.002**

Table 8: Two-sample, unequal variance t-tests for difference in means for ER graphs with $|\mathbf{V}| = 50$. Significance levels: 0 ’***’, 0.001 ’**’, 0.01 ’*’, 0.05 ’.’, 0.1 ’ ’ 1. DoF: $n_a = n_b = 39$.

Type	Metric	Methods	Means±Std	t	p-value
ER2	V-F1	PC-S vs CPC	0.31 ± 0.4 vs 0.50 ± 0.4	−2.05	0.043*
		PC-S vs MPC	0.31 ± 0.4 vs 0.42 ± 0.4	−1.26	0.210
		PC-S vs PC-M	0.31 ± 0.4 vs 0.81 ± 0.3	−6.24	0.000***
		PC-S vs SPC	0.31 ± 0.4 vs 0.98 ± 0.0	−11.45	0.000***
		CPC vs MPC	0.50 ± 0.4 vs 0.42 ± 0.4	0.79	0.429
		CPC vs PC-M	0.50 ± 0.4 vs 0.81 ± 0.3	−3.53	0.001***
		CPC vs SPC	0.50 ± 0.4 vs 0.98 ± 0.0	−6.95	0.000***
		MPC vs PC-M	0.42 ± 0.4 vs 0.81 ± 0.3	−4.56	0.000***
		MPC vs SPC	0.42 ± 0.4 vs 0.98 ± 0.0	−8.56	0.000***
		PC-M vs SPC	0.81 ± 0.3 vs 0.98 ± 0.0	−3.13	0.003**
	AH-F1	PC-S vs CPC	0.25 ± 0.3 vs 0.40 ± 0.4	−2.02	0.047*
		PC-S vs MPC	0.25 ± 0.3 vs 0.35 ± 0.3	−1.32	0.191
		PC-S vs PC-M	0.25 ± 0.3 vs 0.63 ± 0.3	−5.86	0.000***
		PC-S vs SPC	0.25 ± 0.3 vs 0.75 ± 0.1	−10.22	0.000***
		CPC vs MPC	0.40 ± 0.4 vs 0.35 ± 0.3	0.70	0.485
		CPC vs PC-M	0.40 ± 0.4 vs 0.63 ± 0.3	−3.21	0.002**
		CPC vs SPC	0.40 ± 0.4 vs 0.75 ± 0.1	−6.12	0.000***
		MPC vs PC-M	0.35 ± 0.3 vs 0.63 ± 0.3	−4.11	0.000***
		MPC vs SPC	0.35 ± 0.3 vs 0.75 ± 0.1	−7.42	0.000***
		PC-M vs SPC	0.63 ± 0.3 vs 0.75 ± 0.1	−2.75	0.009**
ER4	V-F1	PC-S vs CPC	0.08 ± 0.2 vs 0.14 ± 0.2	−1.38	0.172
		PC-S vs MPC	0.08 ± 0.2 vs 0.08 ± 0.2	−0.06	0.954
		PC-S vs PC-M	0.08 ± 0.2 vs 0.12 ± 0.3	−0.88	0.380
		PC-S vs SPC	0.08 ± 0.2 vs 0.48 ± 0.4	−6.32	0.000***
		CPC vs MPC	0.14 ± 0.2 vs 0.08 ± 0.2	1.30	0.198
		CPC vs PC-M	0.14 ± 0.2 vs 0.12 ± 0.3	0.35	0.731
		CPC vs SPC	0.14 ± 0.2 vs 0.48 ± 0.4	−4.86	0.000***
		MPC vs PC-M	0.08 ± 0.2 vs 0.12 ± 0.3	−0.82	0.414
		MPC vs SPC	0.08 ± 0.2 vs 0.48 ± 0.4	−6.21	0.000***
		PC-M vs SPC	0.12 ± 0.3 vs 0.48 ± 0.4	−4.96	0.000***
	AH-F1	PC-S vs CPC	0.04 ± 0.1 vs 0.06 ± 0.1	−1.10	0.275
		PC-S vs MPC	0.04 ± 0.1 vs 0.04 ± 0.1	−0.08	0.933
		PC-S vs PC-M	0.04 ± 0.1 vs 0.05 ± 0.1	−0.48	0.635
		PC-S vs SPC	0.04 ± 0.1 vs 0.19 ± 0.2	−5.63	0.000***
		CPC vs MPC	0.06 ± 0.1 vs 0.04 ± 0.1	1.00	0.320
		CPC vs PC-M	0.06 ± 0.1 vs 0.05 ± 0.1	0.55	0.584
		CPC vs SPC	0.06 ± 0.1 vs 0.19 ± 0.2	−4.52	0.000***
		MPC vs PC-M	0.04 ± 0.1 vs 0.05 ± 0.1	−0.39	0.695
		MPC vs SPC	0.04 ± 0.1 vs 0.19 ± 0.2	−5.50	0.000***
		PC-M vs SPC	0.05 ± 0.1 vs 0.19 ± 0.2	−4.91	0.000***

Table 9: Two-sample, unequal variance t-tests for difference in means for SF graphs with $|\mathbf{V}| = 50$. Significance levels: 0 ’***’, 0.001 ’**’, 0.01 ’*’, 0.05 ’.’, 0.1 ’ ’ 1. DoF: $n_a = n_b = 39$.

Type	Metric	Methods	Means±Std	t	p-value
SF2	V-F1	PC-S vs CPC	0.67 ± 0.4 vs 0.58 ± 0.5	0.85	0.397
		PC-S vs MPC	0.67 ± 0.4 vs 0.57 ± 0.5	1.03	0.307
		PC-S vs PC-M	0.67 ± 0.4 vs 0.61 ± 0.5	0.56	0.579
		PC-S vs SPC	0.67 ± 0.4 vs 1.00 ± 0.0	−5.30	0.000***
		CPC vs MPC	0.58 ± 0.5 vs 0.57 ± 0.5	0.17	0.864
		CPC vs PC-M	0.58 ± 0.5 vs 0.61 ± 0.5	−0.27	0.787
		CPC vs SPC	0.58 ± 0.5 vs 1.00 ± 0.0	−5.44	0.000***
		MPC vs PC-M	0.57 ± 0.5 vs 0.61 ± 0.5	−0.44	0.661
		MPC vs SPC	0.57 ± 0.5 vs 1.00 ± 0.0	−5.58	0.000***
		PC-M vs SPC	0.61 ± 0.5 vs 1.00 ± 0.0	−5.08	0.000***
	AH-F1	PC-S vs CPC	0.63 ± 0.4 vs 0.53 ± 0.4	1.07	0.290
		PC-S vs MPC	0.63 ± 0.4 vs 0.51 ± 0.4	1.29	0.201
		PC-S vs PC-M	0.63 ± 0.4 vs 0.56 ± 0.4	0.75	0.454
		PC-S vs SPC	0.63 ± 0.4 vs 0.90 ± 0.0	−4.58	0.000***
		CPC vs MPC	0.53 ± 0.4 vs 0.51 ± 0.4	0.21	0.832
		CPC vs PC-M	0.53 ± 0.4 vs 0.56 ± 0.4	−0.29	0.773
		CPC vs SPC	0.53 ± 0.4 vs 0.90 ± 0.0	−5.26	0.000***
		MPC vs PC-M	0.51 ± 0.4 vs 0.56 ± 0.4	−0.50	0.617
		MPC vs SPC	0.51 ± 0.4 vs 0.90 ± 0.0	−5.50	0.000***
		PC-M vs SPC	0.56 ± 0.4 vs 0.90 ± 0.0	−4.85	0.000***
SF4	V-F1	PC-S vs CPC	0.28 ± 0.4 vs 0.46 ± 0.4	−1.87	0.065.
		PC-S vs MPC	0.28 ± 0.4 vs 0.42 ± 0.4	−1.39	0.167
		PC-S vs PC-M	0.28 ± 0.4 vs 0.69 ± 0.4	−4.34	0.000***
		PC-S vs SPC	0.28 ± 0.4 vs 0.99 ± 0.0	−11.16	0.000***
		CPC vs MPC	0.46 ± 0.4 vs 0.42 ± 0.4	0.46	0.649
		CPC vs PC-M	0.46 ± 0.4 vs 0.69 ± 0.4	−2.28	0.025*
		CPC vs SPC	0.46 ± 0.4 vs 0.99 ± 0.0	−7.40	0.000***
		MPC vs PC-M	0.42 ± 0.4 vs 0.69 ± 0.4	−2.75	0.007**
		MPC vs SPC	0.42 ± 0.4 vs 0.99 ± 0.0	−8.07	0.000***
		PC-M vs SPC	0.69 ± 0.4 vs 0.99 ± 0.0	−4.42	0.000***
	AH-F1	PC-S vs CPC	0.25 ± 0.4 vs 0.40 ± 0.4	−1.80	0.076.
		PC-S vs MPC	0.25 ± 0.4 vs 0.36 ± 0.4	−1.29	0.201
		PC-S vs PC-M	0.25 ± 0.4 vs 0.59 ± 0.4	−4.12	0.000***
		PC-S vs SPC	0.25 ± 0.4 vs 0.83 ± 0.1	−10.12	0.000***
		CPC vs MPC	0.40 ± 0.4 vs 0.36 ± 0.4	0.50	0.619
		CPC vs PC-M	0.40 ± 0.4 vs 0.59 ± 0.4	−2.15	0.034*
		CPC vs SPC	0.40 ± 0.4 vs 0.83 ± 0.1	−6.77	0.000***
		MPC vs PC-M	0.36 ± 0.4 vs 0.59 ± 0.4	−2.68	0.009**
		MPC vs SPC	0.36 ± 0.4 vs 0.83 ± 0.1	−7.57	0.000***
		PC-M vs SPC	0.59 ± 0.4 vs 0.83 ± 0.1	−4.04	0.000***

Table 10: ArrowHead Precision and Recall for ERd and SFd graphs of 10 and 50 nodes. d is the number of edges per node in the true DAG. Bold if significantly different from the runner-up (according to a t-test, $\alpha = 0.05$).

		Method	ER2	ER4	SF2	SF4
$\lvert\mathbf{V}\rvert = 10$	**Precision**	PC-Stable	0.4±0.3	0.16±0.18	0.73±0.25	0.42±0.35
		CPC	0.46±0.29	0.16±0.19	0.82±0.14	0.44±0.29
		MPC	0.45±0.28	0.16±0.19	0.82±0.14	0.47±0.29
		PC-Max	0.55±0.24	0.08±0.13	0.80±0.23	0.50±0.33
		Shapley-PC	**0.70±0.18**	**0.29±0.20**	**0.91±0.11**	**0.71±0.20**
	Recall	PC-Stable	0.34±0.26	0.14±0.15	0.62±0.22	0.26±0.22
		CPC	0.39±0.25	0.14±0.16	0.68±0.13	0.28±0.19
		MPC	0.39±0.25	0.14±0.16	0.68±0.13	0.30±0.19
		PC-Max	0.47±0.21	0.07±0.11	0.68±0.19	0.32±0.22
		Shapley-PC	**0.59±0.16**	**0.22±0.14**	**0.76±0.08**	**0.44±0.14**
$\lvert\mathbf{V}\rvert = 50$	**Precision**	PC-Stable	0.29±0.35	0.06±0.13	0.65±0.38	0.27±0.38
		CPC	0.46±0.41	0.11±0.18	0.56±0.46	0.44±0.42
		MPC	0.40±0.39	0.06±0.14	0.53±0.46	0.39±0.42
		PC-Max	0.72±0.31	0.08±0.18	0.58±0.46	0.63±0.40
		Shapley-PC	**0.86±0.05**	**0.33±0.25**	**0.93±0.04**	**0.89±0.07**
	Recall	PC-Stable	0.22±0.27	0.03±0.06	0.61±0.36	0.24±0.33
		CPC	0.36±0.32	0.04±0.07	0.50±0.42	0.38±0.37
		MPC	0.31±0.30	0.03±0.07	0.49±0.43	0.34±0.36
		PC-Max	0.56±0.25	0.03±0.08	0.54±0.42	0.56±0.36
		Shapley-PC	**0.67±0.07**	**0.14±0.11**	**0.86±0.05**	**0.79±0.08**

Table 11: V-structure Precision and Recall for ERd and SFd graphs of 10 and 50 nodes. d is the number of edges per node in the true DAG. Bold if significantly different from the runner-up (according to a t-test, α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
$\lvert\mathbf{V}\rvert$ = 10	**Precision**	PC-Stable	0.46±0.37	0.35±0.41	0.88±0.27	0.56±0.45
		CPC	0.60±0.36	0.32±0.41	0.96±0.09	0.65±0.42
		MPC	0.60±0.37	0.31±0.40	0.95±0.09	0.68±0.40
		PC-Max	**0.72±0.33**	0.19±0.39	0.92±0.22	0.65±0.42
		Shapley-PC	**0.85±0.21**	0.55±0.47	**1.0±0.01**	**0.86±0.25**
	Recall	PC-Stable	0.48±0.37	0.27±0.39	0.77±0.30	0.47±0.40
		CPC	0.60±0.35	0.25±0.36	0.84±0.22	0.50±0.36
		MPC	0.60±0.35	0.24±0.37	0.85±0.22	0.56±0.37
		PC-Max	0.67±0.32	0.09±0.24	0.88±0.23	0.56±0.39
		Shapley-PC	**0.91±0.14**	**0.35±0.42**	**0.99±0.04**	**0.84±0.24**
$\lvert\mathbf{V}\rvert$ = 50	**Precision**	PC-Stable	0.32±0.38	0.07±0.16	0.70±0.41	0.31±0.42
		CPC	0.50±0.44	0.12±0.21	0.59±0.49	0.47±0.45
		MPC	0.42±0.41	0.07±0.16	0.57±0.50	0.42±0.45
		PC-Max	0.80±0.34	0.12±0.26	0.62±0.49	0.69±0.43
		Shapley-PC	**0.97±0.04**	**0.44±0.34**	**1.0±0.0**	**0.98±0.03**
	Recall	PC-Stable	0.30±0.36	0.09±0.20	0.64±0.39	0.27±0.38
		CPC	0.50±0.44	0.18±0.30	0.58±0.48	0.46±0.44
		MPC	0.42±0.42	0.10±0.23	0.56±0.49	0.41±0.44
		PC-Max	0.82±0.35	0.13±0.29	0.61±0.48	0.68±0.43
		Shapley-PC	**0.99±0.02**	**0.53±0.39**	**1.0±0.0**	**0.99±0.02**

Table 12: Structural Interventional Distance (SID, the lower the better) for ERd and SFd graphs of 10 and 50 nodes. d is the number of edges per node in the true DAG. Bold if significantly different from the runner-up (according to a t-test, α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
|V| = 10	**SID-Low**	PC-Stable	42±17	72±7	14±10	37±14
		CPC	39±14	70±6	13±10	42±12
		MPC	40±13	70±6	13±10	42±12
		PC-Max	35±11	71±9	11±7	36±14
		Shapley-PC	**29±13**	**65±10**	**7±4**	**28±12**
	SID-High	PC-Stable	59±14	78±6	27±12	57±14
		CPC	57±11	76±5	26±12	60±14
		MPC	57±10	76±5	26±11	57±15
		PC-Max	56±8	79±6	22±8	**50±13**
		Shapley-PC	**51±11**	77±6	**20±7**	**48±13**
|V| = 50	**SID-Low**	PC-Stable	792±196	1906±155	189±84	394±172
		CPC	559±168	1904±189	128±59	308±116
		MPC	644±156	1880±157	**119±57**	315±107
		PC-Max	**494±127**	**1682±197**	**113±57**	**216±109**
		Shapley-PC	**471±140**	**1569±172**	**93±43**	**173±85**
	SID-High	PC-Stable	1073±254	2110±139	263±96	484±189
		CPC	900±244	2188±110	221±74	426±155
		MPC	953±229	2136±120	**199±64**	419±143
		PC-Max	**803±251**	**2012±191**	**186±75**	**311±147**
		Shapley-PC	**792±268**	**1962±172**	**170±54**	**272±125**

Nodes	ER1	ER2	ER4	SF1	SF2	SF4
10	SHD	SHD	SHD	SHD	SHD	SHD
PC-Stable	1.3±1.8	13.0±4.2	35.4±3.6	2.0±1.9	6.5±2.4	18.5±3.4
CPC	0.9±1.3	12.8±4.1	34.9±3.1	1.6±1.5	6.2±2.4	20.4±3.4
MPC	0.9±1.3	13.1±4.2	34.8±3.0	1.6±1.6	6.2±2.3	20.1±3.4
PC-Max	0.9±1.3	12.0±3.1	36.7±2.5	1.7±1.6	5.9±2.3	**18.3±3.6**
Shapley-PC	1.1±1.6	10.7±4.1	**34.4±3.5**	1.6±1.6	**4.9±1.8**	**17.1±3.2**
50						
PC-Stable	6.9±4.1	53.3±8.6	188.0±10.8	5.5±2.5	17.5±5.9	34.6±13.0
CPC	**4.8±2.3**	42.5±9.9	180.5±8.7	4.5±1.8	**14.6±4.0**	31.7±11.4
MPC	**4.9±2.5**	45.4±8.4	183.3±12.6	4.6±1.8	**14.6±4.5**	32.6±11.2
PC-Max	**4.8±2.4**	**39.5±7.6**	178.1±8.5	4.7±1.9	**13.5±4.0**	28.7±13.1
Shapley-PC	**4.9±2.4**	**38.5±8.1**	171.7±11.0	4.7±2.0	**13.8±4.7**	26.7±11.9

Table 13: Structural Hamming Distance (SHD, the lower the better) for ERd and SFd graphs of 10 and 50 nodes. d is the number of edges per node in the true DAG. Bold if significantly different from the runner-up (according to a t-test, α = 0.05).

Table 14: Two-sample, unequal variance t-tests for difference in means for Alarm and Insurance Data. Significance levels: 0 ’***’, 0.001 ’**’, 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1. DoF: n_a = n_b = 9.

Dataset	Metric	Methods	Means±Std	t	p-value
Alarm	V-F1	CPC vs MPC	0.53 ± 0.4 vs 0.24 ± 0.4	1.75	0.098.
		CPC vs PC-S	0.53 ± 0.4 vs 0.18 ± 0.3	2.37	0.030*
		CPC vs PC-M	0.53 ± 0.4 vs 0.24 ± 0.4	1.73	0.101
		CPC vs SPC	0.53 ± 0.4 vs 0.85 ± 0.1	−2.67	0.024*
		MPC vs PC-S	0.24 ± 0.4 vs 0.18 ± 0.3	0.39	0.700
		MPC vs PC-M	0.24 ± 0.4 vs 0.24 ± 0.4	−0.01	0.991
		MPC vs SPC	0.24 ± 0.4 vs 0.85 ± 0.1	−5.04	0.001***
		PC-S vs PC-M	0.18 ± 0.3 vs 0.24 ± 0.4	−0.40	0.692
		PC-S vs SPC	0.18 ± 0.3 vs 0.85 ± 0.1	−7.19	0.000***
		PC-M vs SPC	0.24 ± 0.4 vs 0.85 ± 0.1	−4.97	0.001***
	AH-F1	CPC vs MPC	0.37 ± 0.3 vs 0.16 ± 0.3	1.78	0.091.
		CPC vs PC-S	0.37 ± 0.3 vs 0.13 ± 0.2	2.33	0.032*
		CPC vs PC-M	0.37 ± 0.3 vs 0.16 ± 0.3	1.80	0.089.
		CPC vs SPC	0.37 ± 0.3 vs 0.57 ± 0.0	−2.38	0.041*
		MPC vs PC-S	0.16 ± 0.3 vs 0.13 ± 0.2	0.33	0.743
		MPC vs PC-M	0.16 ± 0.3 vs 0.16 ± 0.3	0.01	0.994
		MPC vs SPC	0.16 ± 0.3 vs 0.57 ± 0.0	−4.81	0.001***
		PC-S vs PC-M	0.13 ± 0.2 vs 0.16 ± 0.3	−0.33	0.749
		PC-S vs SPC	0.13 ± 0.2 vs 0.57 ± 0.0	−6.64	0.000***
		PC-M vs SPC	0.16 ± 0.3 vs 0.57 ± 0.0	−4.84	0.001***
Insurance	V-F1	CPC vs MPC	0.05 ± 0.1 vs 0.02 ± 0.1	0.73	0.474
		CPC vs PC-S	0.05 ± 0.1 vs 0.03 ± 0.1	0.31	0.757
		CPC vs PC-M	0.05 ± 0.1 vs 0.07 ± 0.2	−0.35	0.732
		CPC vs SPC	0.05 ± 0.1 vs 0.42 ± 0.2	−4.54	0.001***
		MPC vs PC-S	0.02 ± 0.1 vs 0.03 ± 0.1	−0.34	0.736
		MPC vs PC-M	0.02 ± 0.1 vs 0.07 ± 0.2	−0.89	0.391
		MPC vs SPC	0.02 ± 0.1 vs 0.42 ± 0.2	−5.17	0.000***
		PC-S vs PC-M	0.03 ± 0.1 vs 0.07 ± 0.2	−0.59	0.567
		PC-S vs SPC	0.03 ± 0.1 vs 0.42 ± 0.2	−4.71	0.000***
		PC-M vs SPC	0.07 ± 0.2 vs 0.42 ± 0.2	−3.83	0.001**
	AH-F1	CPC vs MPC	0.04 ± 0.1 vs 0.02 ± 0.1	0.72	0.484
		CPC vs PC-S	0.04 ± 0.1 vs 0.02 ± 0.1	0.59	0.563
		CPC vs PC-M	0.04 ± 0.1 vs 0.04 ± 0.1	−0.10	0.920
		CPC vs SPC	0.04 ± 0.1 vs 0.21 ± 0.1	−3.78	0.002**
		MPC vs PC-S	0.02 ± 0.1 vs 0.02 ± 0.1	−0.11	0.911
		MPC vs PC-M	0.02 ± 0.1 vs 0.04 ± 0.1	−0.75	0.465
		MPC vs SPC	0.02 ± 0.1 vs 0.21 ± 0.1	−4.76	0.000***
		PC-S vs PC-M	0.02 ± 0.1 vs 0.04 ± 0.1	−0.64	0.532
		PC-S vs SPC	0.02 ± 0.1 vs 0.21 ± 0.1	−4.54	0.000***
		PC-M vs SPC	0.04 ± 0.1 vs 0.21 ± 0.1	−3.47	0.003**

Table 15: Two-sample, unequal variance t-tests for difference in means for Ecoli70 and Mehra Data. Significance levels: 0 '***', 0.001 '**', 0.01 '*' 0.05 '.' 0.1 ' ' 1. DoF: n_a = n_b = 9.

Dataset	Metric	Methods	Means±Std	t	p-value
Ecoli70	V-F1	CPC vs MPC	0.01 ± 0.0 vs 0.60 ± 0.1	−19.57	0.000***
		CPC vs PC-S	0.01 ± 0.0 vs 0.29 ± 0.2	−3.93	0.003**
		CPC vs PC-M	0.01 ± 0.0 vs 0.72 ± 0.3	−7.88	0.000***
		CPC vs SPC	0.01 ± 0.0 vs 0.89 ± 0.1	−23.53	0.000***
		MPC vs PC-S	0.60 ± 0.1 vs 0.29 ± 0.2	3.88	0.002**
		MPC vs PC-M	0.60 ± 0.1 vs 0.72 ± 0.3	−1.33	0.210
		MPC vs SPC	0.60 ± 0.1 vs 0.89 ± 0.1	−6.19	0.000***
		PC-S vs PC-M	0.29 ± 0.2 vs 0.72 ± 0.3	−3.72	0.002**
		PC-S vs SPC	0.29 ± 0.2 vs 0.89 ± 0.1	−7.39	0.000***
		PC-M vs SPC	0.72 ± 0.3 vs 0.89 ± 0.1	−1.74	0.108
	AH-F1	CPC vs MPC	0.01 ± 0.0 vs 0.55 ± 0.1	−27.65	0.000***
		CPC vs PC-S	0.01 ± 0.0 vs 0.32 ± 0.2	−4.27	0.002**
		CPC vs PC-M	0.01 ± 0.0 vs 0.60 ± 0.2	−8.52	0.000***
		CPC vs SPC	0.01 ± 0.0 vs 0.73 ± 0.1	−28.17	0.000***
		MPC vs PC-S	0.55 ± 0.1 vs 0.32 ± 0.2	3.12	0.011*
		MPC vs PC-M	0.55 ± 0.1 vs 0.60 ± 0.2	−0.68	0.510
		MPC vs SPC	0.55 ± 0.1 vs 0.73 ± 0.1	−5.64	0.000***
		PC-S vs PC-M	0.32 ± 0.2 vs 0.60 ± 0.2	−2.82	0.011*
		PC-S vs SPC	0.32 ± 0.2 vs 0.73 ± 0.1	−5.42	0.000***
		PC-M vs SPC	0.60 ± 0.2 vs 0.73 ± 0.1	−1.80	0.099.
Mehra	V-F1	CPC vs MPC	0.45 ± 0.4 vs 0.26 ± 0.4	1.04	0.311
		CPC vs PC-S	0.45 ± 0.4 vs 0.01 ± 0.0	3.49	0.007**
		CPC vs PC-M	0.45 ± 0.4 vs 0.28 ± 0.5	0.91	0.377
		CPC vs SPC	0.45 ± 0.4 vs 0.76 ± 0.3	−1.98	0.065.
		MPC vs PC-S	0.26 ± 0.4 vs 0.01 ± 0.0	1.88	0.092.
		MPC vs PC-M	0.26 ± 0.4 vs 0.28 ± 0.5	−0.10	0.925
		MPC vs SPC	0.26 ± 0.4 vs 0.76 ± 0.3	−3.10	0.007**
		PC-S vs PC-M	0.01 ± 0.0 vs 0.28 ± 0.5	−1.89	0.091.
		PC-S vs SPC	0.01 ± 0.0 vs 0.76 ± 0.3	−8.30	0.000***
		PC-M vs SPC	0.28 ± 0.5 vs 0.76 ± 0.3	−2.85	0.012*
	AH-F1	CPC vs MPC	0.27 ± 0.2 vs 0.14 ± 0.2	1.24	0.230
		CPC vs PC-S	0.27 ± 0.2 vs 0.01 ± 0.0	3.49	0.007**
		CPC vs PC-M	0.27 ± 0.2 vs 0.15 ± 0.2	1.18	0.253
		CPC vs SPC	0.27 ± 0.2 vs 0.41 ± 0.2	−1.45	0.166
		MPC vs PC-S	0.14 ± 0.2 vs 0.01 ± 0.0	1.82	0.101.
		MPC vs PC-M	0.14 ± 0.2 vs 0.15 ± 0.2	−0.04	0.965
		MPC vs SPC	0.14 ± 0.2 vs 0.41 ± 0.2	−2.99	0.009**
		PC-S vs PC-M	0.01 ± 0.0 vs 0.15 ± 0.2	−1.83	0.100.
		PC-S vs SPC	0.01 ± 0.0 vs 0.41 ± 0.2	−8.24	0.000***
		PC-M vs SPC	0.15 ± 0.2 vs 0.41 ± 0.2	−2.87	0.011*

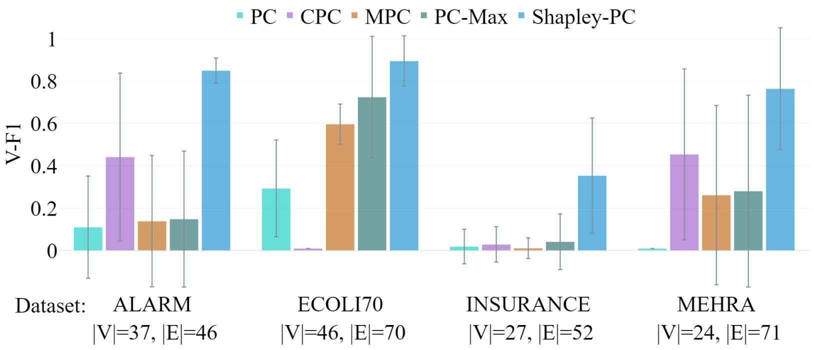


Figure 4: V-structure F1 for the datasets in Fig. 1 in the main text.

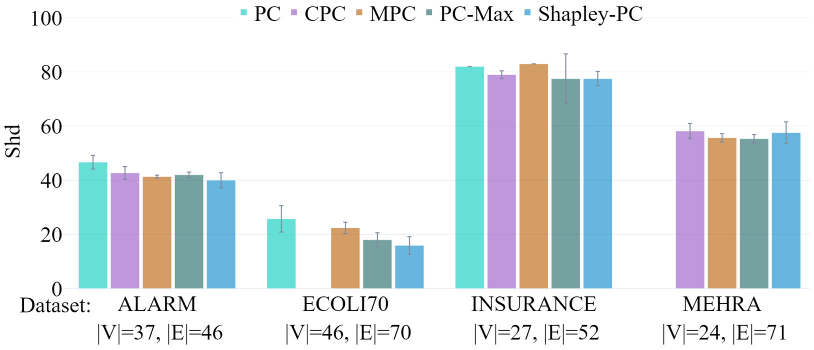


Figure 5: SHD, the lower the better, for the datasets in Fig. 1 in the main text.

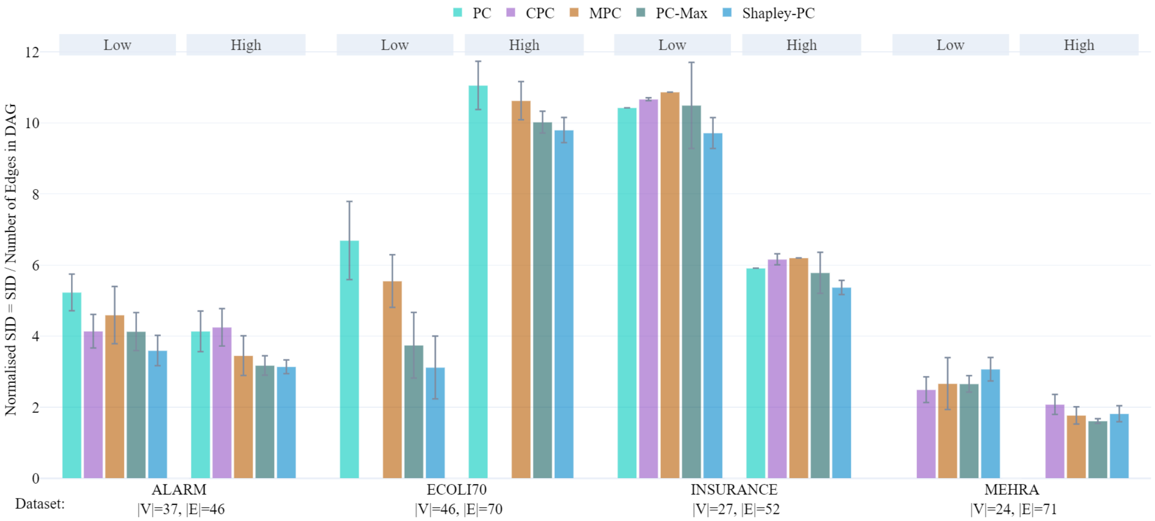


Figure 6: Normalised SID, the lower the better, for the datasets in Fig. 1 in the main text.

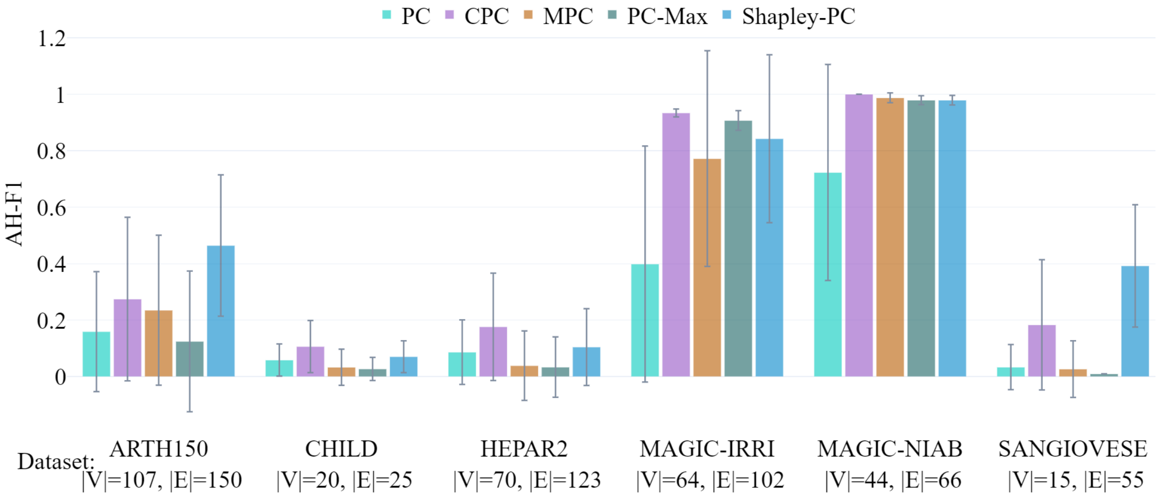


Figure 7: ArrowHead F1 for additional datasets in the `bnlearn` repository.

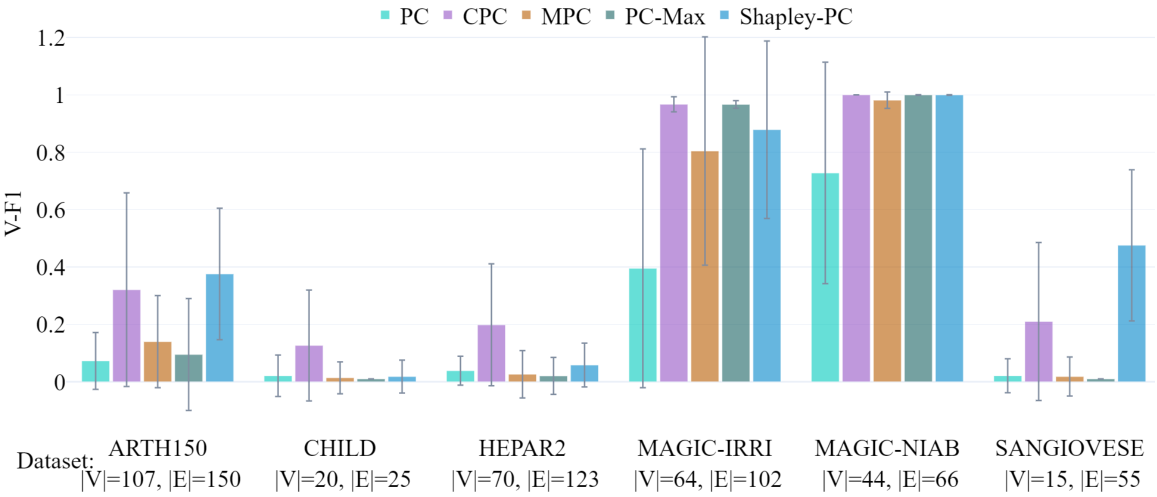


Figure 8: V-structure F1 for additional datasets in the `bnlearn` repository.

B.6. CIT Comparison

Here we provide a comparison of performance and runtime when changing the CIT used to establish independence.

B.6.1. SYNTHETIC DATA

Comparison between Fisher Z and KCI Tests. Tables 18 and 19 compare the performance of Shapley-PC and other PC-based methods using Fisher Z and KCI tests across various noise distributions.

- **Gaussian Noise (Table 18):** Fisher Z, aligned with the data's assumptions, achieves higher F1 scores. For example, under Gaussian noise with $d = 2$ graphs (ER2 or SF2), Shapley-PC with Fisher Z achieves the highest AH-F1 and V-F1 scores with statistical significance.

- **Non-Gaussian Noise (Table 19):** While KCI's nonparametric nature should theoretically perform better, the results are mixed. KCI does not consistently outperform Fisher Z, and variances remain comparable due to the limited sample size ($s = N/|V| = 100$). This trend holds across individual noise types (Tables 20, 21, and 22).

Influence of Noise Distribution and Graph Structure. Under Gaussian noise, Fisher Z achieves better performance, especially with Shapley-PC (Table 18). Non-Gaussian distributions challenge Fisher Z's assumptions, narrowing performance gaps (Tables 19 and 17).

Performance trends from the main text persist:

- **Graph Type (ER vs. SF):** SF graphs consistently achieve higher F1 scores due to their hub structure, which simplifies identifying dependencies (Tables 18 and 19).

- **Edge Density ($d = 2$ vs. $d = 4$):** Sparser graphs ($d = 2$) are easier to learn due to smaller and more reliable conditioning sets, yielding higher F1 scores compared to denser graphs ($d = 4$). Shapley-PC maintains a leading position in both cases.

Runtime and Complexity. KCI's computational complexity is substantially higher than Fisher Z, making it impractical to use large sample sizes in our experiments. For example, the runtime for a single independence test is approximately 0.1 seconds for Fisher Z compared to 200 seconds for KCI (Table 16). Consequently, we used $N/|V| = 100$ as a practical compromise. While the trends observed are consistent with theoretical expectations, this small sample size introduces higher variance, particularly in KCI results, and limits the ability to fully realise its potential advantages. These caveats should be considered when interpreting the comparison.

B.6.2. BNLEARN DATA

Using χ^2 for Discrete Bayesian networks (BNs). In this section, we report results using the χ^2 test, which is better suited for discrete data. Since our synthetic data is all continuous, we analyse potential differences deriving from usage of a different CIT from the one used in the main paper, using the discrete BNs in the `bnlearn` repository. Results are presented in Fig. 9 and 10. By comparing these results to those obtained with the Fisher Z test (Fig. 1), we gain valuable insights into how the choice of CI test affects performance.

- **Alarm:** All methods achieve near-optimal performance, with no significant differences across methods.
- **Child:** Here, all PC variants outperform PC-Stable, reflecting the ability of Shapley-PC and other Step 2-enhanced methods to improve performance even when the CI test is well-suited to the data.
- **Insurance:** Shapley-PC and PC-Max perform significantly better than all other methods. The advantage of these methods in scenarios with more complex dependencies demonstrates their ability to leverage the additional information provided by p-values.
- **Hepar2:** PC-Stable and MPC perform significantly worse than all other methods, while Shapley-PC remains competitive. This highlights Shapley-PC's adaptability to varying scenarios where CPC and MPC rules might instead fai.

Comparing these results to those obtained with the Fisher Z test reveals crucial differences. Fisher Z, designed for continuous, linear-Gaussian data, is not well-suited for discrete data. For the **Alarm** and **Insurance** datasets, this mismatch leads to a significant performance degradation for all methods except Shapley-PC, which demonstrates a remarkable degree of robustness even under suboptimal conditions.

This resilience is less pronounced for the **Child** and **Hepar2** datasets, where the performance of all methods declines when using Fisher Z. These results highlight the importance of aligning the CI test with the data type to achieve optimal performance. However, even in these scenarios, Shapley-PC maintains a competitive edge, demonstrating its adaptability and reliability across diverse datasets and CI tests.

Overall, these findings underscore the flexibility of Shapley-PC in handling both appropriate and inappropriate CI tests, and its strong performance when paired with tests suited to the data type, such as χ^2 for discrete BNs.

	KCI		**Fisher Z**	
$|\mathbf{V}|$	ER	SF	ER	SF
PC-Stable	181.7	96.5	0.086	0.082
CPC	223.3	143.2	0.1	0.104
MPC	218.1	143.9	0.102	0.104
PC-Max	220.4	143.8	0.105	0.109
Shapley-PC	215.9	141.1	0.103	0.106

Table 16: Runtime comparison between KCI and Fisher Z: median elapsed time in seconds for ER and SF graphs with nodes $|\mathbf{V}| = \{10\}$.

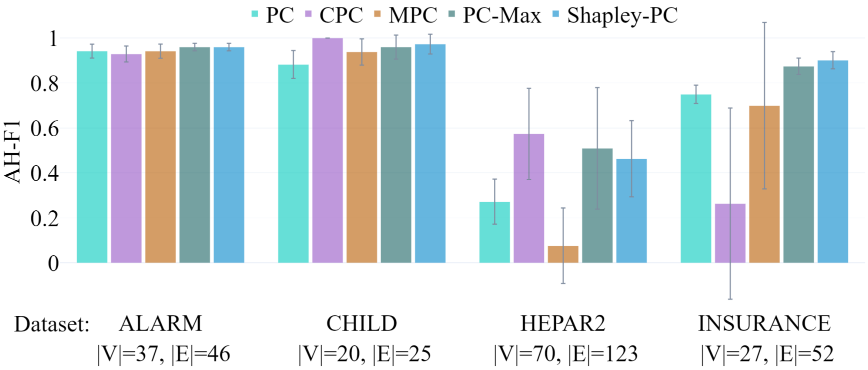


Figure 9: ArrowHead F1 using χ^2 for the discrete BNs of the `bnlearn` repository.

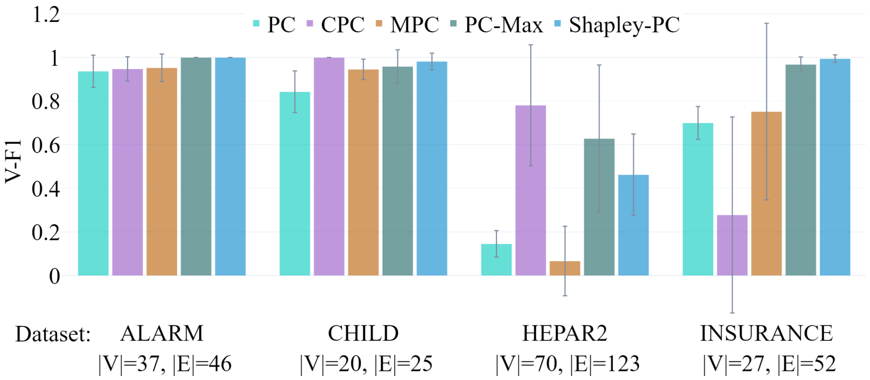


Figure 10: V-Structure F1 using χ^2 for the discrete BNs of the `bnlearn` repository.

Table 17: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|= 10$. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and all considered noise distributions. Bold if significantly different from the runner-up according to a t-test (α = 0.05)

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.33±0.18	0.16±0.12	0.58±0.18	0.28±0.17
		CPC	0.40±0.16	0.16±0.11	0.55±0.23	0.26±0.17
		MPC	0.42±0.17	0.15±0.11	0.59±0.20	0.31±0.15
		PC-Max	**0.43±0.16**	0.16±0.13	**0.62±0.19**	**0.34±0.17**
		Shapley-PC	**0.50±0.18**	0.17±0.10	**0.68±0.11**	**0.41±0.15**
	V-F1	PC-Stable	0.43±0.32	0.17±0.33	0.79±0.25	0.49±0.37
		CPC	0.59±0.31	0.17±0.31	0.75±0.30	0.47±0.40
		MPC	0.61±0.31	0.15±0.30	0.79±0.26	0.57±0.35
		PC-Max	**0.66±0.30**	0.18±0.35	0.87±0.23	0.63±0.36
		Shapley-PC	**0.78±0.26**	0.20±0.34	**0.95±0.10**	**0.78±0.27**
KCI	**AH-F1**	PC-Stable	0.37±0.16	0.12±0.10	0.54±0.15	0.21±0.16
		CPC	0.38±0.14	0.13±0.10	0.58±0.10	0.22±0.13
		MPC	0.38±0.16	0.13±0.10	0.56±0.15	**0.23±0.14**
		PC-Max	0.41±0.15	0.13±0.11	0.58±0.15	**0.31±0.14**
		Shapley-PC	0.45±0.19	0.14±0.11	**0.62±0.10**	**0.33±0.15**
	V-F1	PC-Stable	0.58±0.34	0.18±0.30	0.77±0.23	0.40±0.41
		CPC	0.60±0.34	0.15±0.30	0.83±0.20	0.43±0.38
		MPC	0.60±0.35	0.16±0.31	0.82±0.25	0.44±0.38
		PC-Max	0.64±0.32	0.15±0.33	**0.84±0.20**	**0.62±0.43**
		Shapley-PC	0.71±0.34	0.17±0.33	**0.91±0.17**	**0.68±0.41**

Table 18: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|$= 10. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and the noise is Gaussian. Bold if significantly different from the runner-up according to a t-test (α = 0.05)

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.31±0.23	0.20±0.09	0.64±0.10	0.26±0.12
		CPC	**0.41±0.18**	0.15±0.06	0.54±0.23	0.23±0.15
		MPC	**0.43±0.21**	0.15±0.07	0.57±0.14	0.28±0.13
		PC-Max	**0.50±0.07**	0.20±0.12	**0.66±0.09**	**0.28±0.20**
		Shapley-PC	**0.52±0.08**	0.18±0.10	**0.70±0.08**	**0.40±0.10**
	V-F1	PC-Stable	0.39±0.30	0.22±0.37	0.86±0.16	0.50±0.33
		CPC	**0.60±0.33**	0.23±0.39	0.70±0.33	0.31±0.30
		MPC	**0.62±0.33**	0.22±0.37	0.74±0.21	0.50±0.28
		PC-Max	**0.79±0.17**	0.30±0.48	**0.92±0.10**	0.51±0.41
		Shapley-PC	**0.83±0.11**	0.23±0.39	**0.97±0.05**	**0.74±0.29**
KCI	**AH-F1**	PC-Stable	0.27±0.14	0.09±0.11	0.53±0.15	0.24±0.16
		CPC	0.32±0.17	0.10±0.10	0.57±0.09	0.27±0.15
		MPC	0.27±0.19	0.10±0.14	0.57±0.11	0.30±0.15
		PC-Max	0.32±0.17	0.11±0.09	0.60±0.10	0.30±0.14
		Shapley-PC	0.38±0.16	0.11±0.08	0.61±0.09	0.33±0.16
	V-F1	PC-Stable	0.38±0.37	0.11±0.32	0.81±0.24	0.52±0.38
		CPC	0.46±0.43	0.15±0.30	0.86±0.15	0.49±0.38
		MPC	0.35±0.42	0.09±0.30	0.85±0.15	0.55±0.36
		PC-Max	0.46±0.44	0.10±0.32	0.94±0.09	0.64±0.46
		Shapley-PC	0.53±0.38	0.11±0.30	0.95±0.09	0.70±0.39

Table 19: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|$= 10. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and the noise is Non-Gaussian. Bold if significantly different from the runner-up according to a t-test (α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.33±0.17	0.15±0.13	0.55±0.20	0.29±0.19
		CPC	**0.40±0.16**	0.16±0.12	0.56±0.23	0.28±0.18
		MPC	**0.41±0.16**	0.14±0.12	**0.59±0.21**	0.33±0.16
		PC-Max	**0.41±0.18**	0.15±0.13	**0.61±0.22**	**0.37±0.15**
		Shapley-PC	**0.50±0.20**	0.17±0.10	**0.68±0.12**	**0.41±0.16**
	V-F1	PC-Stable	0.45±0.33	0.15±0.32	0.77±0.27	0.49±0.39
		CPC	0.59±0.31	0.15±0.28	0.77±0.30	0.52±0.42
		MPC	0.60±0.30	0.13±0.28	0.80±0.28	0.59±0.37
		PC-Max	**0.62±0.33**	0.14±0.29	0.85±0.26	**0.67±0.34**
		Shapley-PC	**0.76±0.30**	0.19±0.33	0.95±0.11	**0.79±0.26**
KCI	**AH-F1**	PC-Stable	0.41±0.15	0.13±0.09	0.55±0.15	0.20±0.16
		CPC	0.40±0.12	0.14±0.10	0.58±0.16	0.20±0.12
		MPC	0.42±0.13	0.14±0.10	0.56±0.20	0.21±0.14
		PC-Max	0.43±0.14	0.13±0.11	0.57±0.20	**0.31±0.13**
		Shapley-PC	0.47±0.19	0.15±0.11	0.62±0.15	**0.33±0.15**
	V-F1	PC-Stable	0.64±0.31	0.21±0.36	0.75±0.23	0.36±0.41
		CPC	0.64±0.30	0.16±0.30	0.82±0.21	0.41±0.38
		MPC	0.68±0.29	0.18±0.31	0.80±0.28	0.40±0.38
		PC-Max	0.70±0.25	0.16±0.34	0.81±0.26	**0.62±0.43**
		Shapley-PC	0.77±0.30	0.19±0.34	0.90±0.19	**0.67±0.42**

Table 20: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|$= 10. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and the noise is Exponential. Bold if significantly different from the runner-up according to a t-test (α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.34±0.16	0.16±0.14	0.57±0.10	0.28±0.21
		CPC	0.38±0.18	0.15±0.12	**0.58±0.24**	0.28±0.18
		MPC	0.41±0.15	0.14±0.11	**0.64±0.15**	**0.32±0.15**
		PC-Max	0.37±0.23	0.10±0.12	**0.68±0.11**	**0.43±0.10**
		Shapley-PC	0.44±0.23	0.17±0.12	**0.71±0.09**	**0.44±0.14**
	V-F1	PC-Stable	0.46±0.37	0.17±0.36	0.80±0.14	0.44±0.41
		CPC	0.54±0.36	0.18±0.30	**0.80±0.30**	0.42±0.41
		MPC	0.62±0.31	0.13±0.28	**0.88±0.17**	**0.49±0.39**
		PC-Max	0.58±0.42	0.07±0.21	**0.94±0.08**	**0.77±0.29**
		Shapley-PC	0.69±0.39	0.12±0.25	**0.98±0.03**	**0.83±0.18**
KCI	**AH-F1**	PC-Stable	**0.44±0.22**	0.14±0.11	0.54±0.17	0.19±0.19
		CPC	0.40±0.15	0.16±0.10	0.59±0.23	0.20±0.11
		MPC	**0.43±0.14**	0.18±0.09	0.58±0.24	0.18±0.15
		PC-Max	**0.49±0.14**	0.14±0.13	0.60±0.22	**0.34±0.14**
		Shapley-PC	**0.53±0.13**	0.15±0.12	0.62±0.24	**0.37±0.18**
	V-F1	PC-Stable	**0.69±0.36**	0.21±0.35	0.69±0.29	0.29±0.41
		CPC	**0.57±0.36**	0.20±0.32	0.80±0.30	0.36±0.32
		MPC	0.63±0.30	0.25±0.33	0.80±0.34	0.29±0.33
		PC-Max	**0.80±0.19**	0.27±0.44	0.83±0.30	**0.69±0.38**
		Shapley-PC	**0.88±0.16**	0.28±0.41	0.86±0.31	**0.71±0.39**

Table 21: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|$= 10. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and the noise is Gumbel. Bold if significantly different from the runner-up according to a t-test (α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.33±0.21	0.16±0.13	0.58±0.26	0.28±0.15
		CPC	**0.43±0.12**	0.19±0.12	0.59±0.24	0.24±0.20
		MPC	**0.43±0.15**	0.16±0.13	0.63±0.25	0.30±0.16
		PC-Max	**0.47±0.12**	0.18±0.11	0.54±0.31	0.31±0.20
		Shapley-PC	**0.53±0.17**	0.15±0.09	0.69±0.13	0.40±0.16
	V-F1	PC-Stable	0.45±0.33	0.17±0.36	0.78±0.34	0.47±0.33
		CPC	0.67±0.20	0.07±0.21	0.80±0.30	**0.49±0.44**
		MPC	0.64±0.24	0.07±0.21	0.82±0.31	**0.53±0.36**
		PC-Max	**0.75±0.14**	0.07±0.21	0.74±0.41	**0.55±0.40**
		Shapley-PC	**0.84±0.15**	0.07±0.21	0.95±0.13	**0.80±0.14**
KCI	**AH-F1**	PC-Stable	0.39±0.13	0.14±0.08	0.52±0.16	0.22±0.18
		CPC	0.42±0.11	0.15±0.10	0.57±0.13	0.21±0.16
		MPC	0.45±0.12	0.10±0.08	0.52±0.23	0.25±0.16
		PC-Max	0.44±0.14	0.15±0.12	0.54±0.23	0.32±0.12
		Shapley-PC	0.42±0.27	0.17±0.13	0.65±0.10	0.32±0.14
	V-F1	PC-Stable	0.67±0.27	0.21±0.30	0.69±0.20	0.42±0.46
		CPC	0.75±0.20	0.20±0.12	**0.77±0.13**	**0.43±0.40**
		MPC	0.82±0.20	0.12±0.12	**0.70±0.31**	**0.47±0.42**
		PC-Max	0.75±0.18	0.05±0.16	**0.72±0.31**	**0.59±0.43**
		Shapley-PC	0.69±0.39	0.15±0.34	**0.91±0.11**	**0.66±0.46**

Table 22: ArrowHead (AH) and V-structure (V) F1 Scores ± std for ERd and SFd graphs of nodes $|\mathbf{V}|$= 10. d is the number of edges per node in the true DAG. The proportional sample size is $N/|V|$ = s = 100 and the noise is Uniform. Bold if significantly different from the runner-up according to a t-test (α = 0.05).

		Method	**ER2**	**ER4**	**SF2**	**SF4**
Fisher Z	**AH-F1**	PC-Stable	0.34±0.15	0.13±0.13	0.52±0.22	0.30±0.23
		CPC	**0.40±0.18**	0.13±0.12	0.49±0.22	0.31±0.18
		MPC	**0.40±0.18**	0.13±0.12	0.52±0.23	0.37±0.16
		PC-Max	**0.40±0.18**	0.16±0.14	0.60±0.16	0.36±0.14
		Shapley-PC	**0.53±0.20**	0.18±0.08	0.63±0.14	0.39±0.20
	V-F1	PC-Stable	0.42±0.33	0.12±0.25	0.72±0.32	0.55±0.44
		CPC	**0.54±0.37**	0.19±0.34	0.70±0.31	0.65±0.41
		MPC	**0.54±0.37**	0.19±0.34	0.71±0.32	0.74±0.35
		PC-Max	**0.53±0.34**	0.28±0.39	0.86±0.16	0.68±0.32
		Shapley-PC	**0.76±0.32**	0.38±0.43	0.92±0.15	0.73±0.40
KCI	**AH-F1**	PC-Stable	0.40±0.11	0.11±0.09	0.58±0.11	0.19±0.12
		CPC	0.39±0.12	0.12±0.10	0.58±0.12	0.19±0.10
		MPC	0.37±0.12	0.16±0.11	0.59±0.14	**0.21±0.11**
		PC-Max	0.37±0.12	0.11±0.09	0.57±0.13	**0.28±0.15**
		Shapley-PC	0.47±0.14	0.14±0.09	0.60±0.10	**0.31±0.13**
	V-F1	PC-Stable	0.57±0.32	0.20±0.42	0.89±0.13	0.38±0.40
		CPC	0.61±0.33	0.17±0.36	0.88±0.13	0.44±0.44
		MPC	0.60±0.32	0.17±0.36	0.89±0.11	0.43±0.40
		PC-Max	0.55±0.31	0.17±0.36	0.87±0.16	0.57±0.50
		Shapley-PC	0.74±0.31	0.13±0.28	0.94±0.09	0.65±0.46