Selecting Accurate Subgraphical Models from Possibly Inaccurate Graphical Models

Yi Han YIHAN2@ANDREW.CMU.EDU
5000 Forbes Avenue Pittsburgh, PA 15213

Joseph Ramsey JDRAMSEY@ANDREW.CMU.EDU
5000 Forbes Avenue Pittsburgh, PA 15213

Peter Spirtes PS7Z@ANDREW.CMU.EDU
5000 Forbes Avenue Pittsburgh, PA 15213

Editors: Biwei Huang and Mathias Drton

Abstract

Methods of statistically testing the accuracy of causal graphical models have traditionally been limited, with most focusing on parametric global assessments of the entire causal graph. However, whether or not a causal graphical model passes a statistical test, it is crucial for many practical applications to find which parts of the graph are accurately reconstructed and which are not. In this paper, we introduce the Vertex Checker, the only statistical test that we are aware of that takes as input a causal graphical model $\mathcal{G}$, a vertex X, and an alpha level, sample data, and a conditional independence test, and provides a non-parametric, asymptotically correct, statistical test of a local subgraph of X, is computationally feasible for dozens of variables, and is extendable to other kinds of causal graphical models. Through extensive simulations, we demonstrate the robustness of the Vertex Checker across various data types, causal graphs, and distributions both in terms of accuracy of graphical structure and of quantitative estimates of causal effects. Furthermore, we apply the Vertex Checker to the real-world Sachs dataset, showcasing its practical applicability in uncovering accurate substructures within causal graphs, even when the overall causal graphical model is rejected.

Keywords: causal discovery, subgraph evaluation

1. Introduction

Causal discovery algorithms (CDAs) take as input sample data and optional background knowledge, and aim to identify cause and effect relationships among observed (or even latent) variables under a variety of different assumptions. In causal discovery the cause and effect relationships are represented by a causal directed graph in which there is an edge from A to B if and only if A is a direct cause of B. The output of a CDA is a causal graphical model (CGM) that represents a causal directed graph or some set of causal directed graphs. In recent years, such algorithms have significantly influenced the way statistical tools are utilized in handling data. Classical CDAs include constraint-based methods such as Peter-Clark (PC) and Fast Causal Inference (FCI), score-based methods like Greedy Equivalence Search (GES), and functional-form approaches like Additive Noise Models (ANM) and Post-non-linear models (PNL) (Spirtes et al., 2000; Chickering, 2002; Zhang et al., 2009; Hoyer et al., 2009; Shimizu et al., 2006, 2011, 2009; Ramsey et al., 2017; Bühlmann et al., 2014). Algorithms such as these have been applied in widely different fields, for example, social networks, medicine, finance and etc (Ramsey et al., 2010; Glymour et al., 2019; Mooij et al., 2013; Moneta et al., 2013; Spirtes, 2010; Pearl, 2009; Schlussel, 2020; Peterson and Halpern, 2014; Weng

© 2025 Y. Han, J. Ramsey & P. Spirtes.

and Lee, 2019; Chickering and Heckerman, 1997). However, there is an inherent challenge to the reliability of causal discovery results. In practice, the estimated results may deviate significantly from reality due to reasons such as sub-optimal search parameters, violations of algorithmic assumptions, local optima, and small sample sizes. This raises an important question: when and to what extent should users trust the causal model outputs? Given these issues and the lack of known ground truth in modeling, it is important to have widely applicable evaluation tools for outcomes generated by different CDAs and search parameter choices.

A number of CDAs (e.g. Spirtes et al. (2000); Glymour et al. (2019); Heinze-Deml et al. (2018)) come with guarantees of correct graphical results in the large sample limit (large sample consistency) under a number of different assumptions. Hence in the large sample limit, where the assumptions of a CDA (including parametric assumptions) hold exactly, no further post-processing or further checks of correctness are needed or useful. However, on finite sample sizes, and where the assumptions may not hold exactly, it is not uncommon for CDAs to produce output that are not compatible with the population distribution (as determined by being rejected by a statistical test). In those cases, the combination of a CDA and a post-processor that rejects a CGM that is not compatible with the population distribution can be useful in determining when not to trust the output of a CDA. (Of course, being compatible with the population distribution as determined by a statistical test does not guarantee that the CDM is correct, since models that are overly complex may also be compatible with the population distribution.) To address this issue, a variety of statistical tests have been proposed (see section 2). However, the fact that a CGM is rejected (or fails to be rejected) as a whole does not mean that we should distrust (or trust) every part of it, and it is crucially important to identify which specific parts of the graph contribute to its overall performance.

In addition, in many causal discovery tasks, we are often only concerned with specific parts of an estimated CGM, such as the relationships between treatment variables and outcomes (e.g., in medicine). For example, identifying the parents of a disease variable (e.g., medications affecting the disease) is often sufficient (Ramsey et al., 2010; Mooij et al., 2013; Glymour et al., 2019). Instead of evaluating the entire CGM, focusing on relevant local structures is more appropriate and efficient. Local tests are valuable in these cases, ensuring accurate results where needed while avoiding unnecessary analysis on unrelated sections of the CGM. Computationally, many evaluation methods struggle with scalability and can be costly to execute (Glymour et al., 2019; Zhang and Hyvärinen, 2012). For instance, the Markov Checker (Ramsey et al., 2024), which is a recent advance in global graph tests, sometimes relies on the Kernel Conditional Independence (KCI) test (Zhang et al., 2011) for handling nonlinear models. As a result, its computational complexity increases quadratically with the number of nodes. Testing the local structures of a set of variables saves considerable computational time as compared to conducting a global test.

In this paper, we propose a novel test, the Vertex Checker, that (i) helps locate subgraphs of the output of a CDA which are both approximately qualitatively and quantitatively correct; (ii) can be applied both to CGMs that fail and CGMs that do not fail a global statistical test; and (iii) can be implemented as either a non-parametric or parametric test. The Vertex Checker tests a local subgraph (in this paper the Markov Blanket of a variable) of a CGM $\mathcal{G}$ containing a given variable X by testing a subset of the conditional independence relations entailed by $\mathcal{G}$. Although in this paper we apply the Vertex Checker to completed partially directed acyclic graphs (CPDAGs), the basic idea can easily be extended to multiple kinds of CGMs (e.g. DAGs with latent variables, mixed ancestral graph, partial ancestral graphs, undirected graphs (Spirtes et al., 2000; Zhang, 2008;

Richardson and Spirtes, 2002; Lauritzen, 1996), etc.) that entail different kinds of constraints. We demonstrate its effectiveness on both simulated and real world datasets.

The structure of this paper is as follows: In Section 2, we review the current literature. In Section 3, we introduce the Vertex Checker method, detailing its strengths and theoretical guarantees. In Section 4 we describe the theoretical properties of the Vertex Checker. In Section 5, we present the experimental results on both simulated and real-world datasets. In section 6 we describe the limitations of the Vertex Checker and future work, and Section 7 is the conclusion.

2. Literature Review

There are a vast array of model tests that have been proposed (Browne and Cudeck, 1992). For linear Gaussian models, the p-value of a Chi-squared test applied to a measure of the distance of the maximum likelihood estimate of the covariance matrix from the population covariance matrix is often used as a test to reject models with poor fit (Bollen, 1989; Hoyle, 1995). The Chi-squared test is a theoretically grounded method for rejecting incorrect models. However, it tests entire causal graphs for a given set of variables. Furthermore, in practice, its usefulness is very limited by its very strong parametric assumptions, and the fact that even small deviations from the true model (e.g. one or two incorrect edges in a linear Gaussian structural equation model) lead to rejection at moderate sample sizes (Bentler and Bonett, 1980; MacCallum et al., 1996; Barrett, 2007). In the social sciences, these problems with Chi-squared tests have led to the development of a number of other fit indices, including the Normed Fit Index (NFI) (Bentler and Bonett, 1980)) and the Comparative Fit Index (CFI) (Bentler, 1990)). Although there are simulation studies that have led to suggestions about what cutoff in scores to use to reject models (Hu and Bentler, 1999), they still make strong parametric assumptions and it is not clear how generally the suggested cutoffs apply. These are all global tests of the entire CGM. They could be applied to a given subgraph of a CGM (e.g. the Markov blanket of a variable X) by simply removing all of the variables not in the subgraph, but that would be ignoring much valuable information in the discarded part of the distribution about whether the subgraph is correct (by not paying attention to entailed conditional independence relations between X and variables not in the Markov blanket).

Model fit is also often tested with regression residuals (Freedman, 2005; Hastie et al., 2009; Bollen, 1989; Shadish et al., 2002). It evaluates whether the conditional independencies implied by the model hold in the data. A common approach involves examining whether regression residuals are small and free of systematic patterns. However, this method has several limitations. It relies on parametric assumptions. For example, typical regression models often assume linearity, which may not hold for real data, leading to incorrect conclusions about the presence or absence of edges (Shimizu et al., 2006; Hoyer et al., 2008; Monti et al., 2020).

We know of no current method that is an asymptotically correct test of a subgraph of a given CGM, that is non-parametric, implicitly or explicitly tests a correct set of entailed conditional independence constraints entailed by a CGM, and is computationally feasible on dozens of variables.

3. Vertex Checker

The goal of the Vertex Checker is to provide a statistical test at a given level alpha of whether, under assumptions very widely assumed by CDAs, the null hypothesis that a local subgraph (defined below) of an output CGM that contains a particular vertex V is compatible with the population

distribution. If the local subgraph is true, the Vertex Checker rejects the local subgraph as being compatible with the population distribution at level alpha, then the probability of rejecting the local subgraph is less than or equal to alpha. However, a statistical test may also fail to reject a CGM that is not even approximately correct for several reasons. One kind of error in the output can occur due to the true causal model not being represented by the class of graphs that a particular CDA searches over (e.g. assumptions such as no latent confounders, no selection bias, no feedback, no inter-unit interference, i.i.d., etc.). If a CDA falsely assumes that there are no latent confounders, a CDA may output a CGM that contains no latent confounders that entails a proper subset of the constraints entailed by the true CGM (at the expense of some extra complexity) that nevertheless has an estimated distribution that is quite close to the sample distribution. The Vertex Checker is intended to provide information about cases where the output CGM is not close to the sample distribution due to problems such as insufficient sample size, sampling error, small parametric assumption violations, and suboptimal search parameter settings.

The Vertex Checker is a local version of a recently developed statistical test, the Markov Checker (Ramsey et al., 2024). However, in contrast to the Vertex Checker, the Markov Checker evaluates only entire graphs. This fact entails that the Vertex Checker is generally much faster than the Markov Checker, provides useful information about parts of a causal graph that may be approximately correct even when the Markov Checker rejects a CGM. In addition, in contrast to the Markov Checker, we provide tests on both non-linear and non-Gaussian models, and how failure to reject is related to the accuracy of both *qualitative* and *quantitative* estimates of causal effects.

The Vertex Checker has two main parts: first, for a given local subgraph of a CGM it identifies a set of conditional independence relations entailed by the local subgraph and second it tests whether the set of conditional independence relations are compatible with the population data. In order to explain these two steps, we employ standard causal graph terminology, some of which is explained in the Appendix A, and makes the following widely made assumptions about the relationship between CGMs and statistical independence explained below.

The local Markov condition is a relationship between a probability distribution P and a DAG $\mathcal{G}$. P satisfies the local Markov condition for $\mathcal{G}$ if each variable is independent of the set of its non-parents and non-descendants conditional on its parents. The local Markov condition is equivalent to the following global Markov condition (Pearl, 1988). **D-separation** is a graphical relationship between three disjoint sets of variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ (see the Appendix A.1 for the exact definition.)

Definition 1 *(Lauritzen, 1996; Koller and Friedman, 2009; Pearl, 1988)[GMC] Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a directed acyclic graph, with $\mathbf{V}$ the set of random variables and $\mathbf{E}$ the directed edges. DAG $\mathcal{G}$ and probability distribution $P(\mathbf{V})$ satisfies the **Global Markov Condition** if for any disjoint subsets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, if $\mathbf{X}$ and $\mathbf{Y}$ are* d-separated *by $\mathbf{Z}$, then $\mathbf{X}$ is conditionally independent of $\mathbf{Y}$ conditional on $\mathbf{Z}$ in P.*

The relationship between a causal DAG for a population and the probability distribution in that population is given by the following commonly made Causal Markov Assumption (CMA).

Definition 2 *(Spirtes et al., 2000; Pearl, 2009)[CMA] Let $\mathcal{G} = (V, E)$ be a causal DAG for a population with distribution $P(\mathbf{V})$ where every direct cause of any pair of vertices in $\mathbf{V}$ is also in $\mathbf{V}$ (i.e. $\mathbf{V}$ is causally sufficient). It satisfies the **Causal Markov Assumption** if for any variable $X \in \mathbf{V}$, X is conditionally independent of the set of its non-parents and non-descendants $\mathbf{ND}(X)$*

given its parents $\mathbf{Pa}(X)$ *in* $\mathcal{G}$*. Formally,*

$$X \perp (\mathbf{ND}(X) \setminus \mathbf{Pa}(\mathbf{X})) \mid \mathbf{Pa}(X),$$

where $\perp$ *denotes conditional independence.*

We use the subgraph of a causal DAG containing variables in a Markov blanket of a vertex X as the local graph, and to determine which set of conditional independencies to test. The Markov blanket of a vertex X (denoted $\mathbf{MB}(X)$) in a DAG $\mathcal{G}$ consists of the parents, children, and parents of children of X. Given a DAG that satisfies the CMA, it is entailed that X is conditionally independent of all non-members of its Markov blanket conditional on its Markov blanket (Pearl, 1988).

The CDAs used in this paper to generate output that is input to the Vertex Checker all output Markov equivalence classes of DAGs represented by CPDAGs. A **Markov equivalence class** of DAGs is a set of DAGs that all contain the same variables and the same d-separation relations. A CPDAG $\mathcal{G}$ represents a Markov equivalence class $\mathcal{M}$ of DAGs iff X and Y are adjacent in $\mathcal{G}$ iff X and Y are adjacent in every DAG in $\mathcal{M}$. An edge between X and Y is oriented as $X \to Y$ in $\mathcal{G}$ iff $X \to Y$ in every DAG in $\mathcal{M}$, and as $X - Y$ otherwise. Since every DAG represented by a CPDAG shares the same set of d-separation relations, they all entail the same set of conditional independence relations; if any of the DAGs represented by a CPDAG satisfies the global Markov condition, then they all do (Spirtes et al., 2000; Chickering, 2002). The Vertex Checker takes either a DAG or a CPDAG as input. When given a CPDAG $\mathcal{C}$ as input, it first turns $\mathcal{C}$ into some arbitrary member $\mathcal{G}$ of the Markov equivalence class represented by $\mathcal{C}$. This step is used to select which conditional independencies are tested by the Markov Checker. However, when evaluating the performance of the Markov Checker, the original CPDAG, not the DAG it is transformed into, is used.

Under the null, the distributions of independent p-values with continuous CDF are distributed uniformly, as shown in Theorem 3 (Ramsey et al., 2024). This principle forms the foundation of the Vertex Checker, which evaluates whether the conditional independence relationships in an estimated graph align with the expected uniform distribution of p-values, thereby assessing the correctness of the local graph structure. The detailed procedure is summarized below into Algorithm 1.

Theorem 3 *(Ramsey et al., 2024)[Uniformity Check] If* $\mathcal{P}$ *is a set of valid p-values of statistical tests of the form* $X \perp Y \mid \mathbf{Z}$ *obtained from independent samples, where each* $P(X, Y \mid \mathbf{Z})$ *has a continuous cumulative distribution function, then* $\mathcal{P}$ *is uniformly distributed, i.e.,* $\mathcal{P} \sim U(0, 1)$*.*

The uniformity test has several desirable properties. It is a non-parametric test and it is a test of an entire set of conditional independence relations, not just each one individually. If a large set of conditional independence relations were tested individually, then some of them are bound to fail due to sampling error, even if the whole set passes the test. An alternative would be to perform a Bonferroni adjustment; however, this is generally considered to be too conservative (Moran, 2003).

Note that there are two distinct kinds of tests employed in the Vertex Checker: there are statistical tests of conditional independence, and there is $U(\mathcal{P})$, the result of a statistical test of the uniformity of the p-values generated by the statistical tests of conditional independence. The statistical test of the uniformity of the p-values that we employ is non-parametric, in this case a Kolmogorov-Smirnoff test (Kolmogorov, 1933; Smirnov, 1948). The statistical tests of conditional independence that are employed can either be parametric (e.g. a chi-squared test of linear Gaussian models) or non-parametric (e.g. KCI). In the latter case the algorithm as a whole does not make parametric assumptions. In addition, the tests of conditional independence may only be asymptotically correct, in

Algorithm 1 Vertex Checker

Input: Estimated DAG or CPDAG $\mathcal{G}_{es} = (\mathbf{V},\mathbf{E})$, random variable X, data set D, conditional independence test CI, significance level α

Output: Pass or Fail, p-value of uniformity test

If $\mathcal{G}_{es}$ is a CPDAG, transform it into an arbitrary DAG in the Markov equivalence class represented by $\mathcal{G}_{es}$.

Initialize: $\mathcal{P} = \emptyset$

foreach $Y \in \mathbf{V} \backslash \mathbf{MB}(X)$ **do**

  Test $Y \perp X \mid \mathbf{MB}(X)$

  Record p-values from conditional independence tests and append to $\mathcal{P}$

end

Hypothesis: The set of variables in the Markov blanket of X is correct, implying p-values in $\mathcal{P}$ follow a uniform distribution between 0 and 1

Test uniformity of list $\mathcal{P}$ with Kolmogorov-Smirnov test or similar tests, obtaining the p-value $U(\mathcal{P})$

if $U(\mathcal{P}) > \alpha$ **then**

  Return Pass and $U(\mathcal{P})$

end

else

  Return Fail and $U(\mathcal{P})$

end

which case the correctness of the Vertex Checker may also suffer if the sample size is too small. For those reasons, we have performed simulations using both parametric and non-parametric statistical tests of conditional independence.

The Causal Markov Assumption entails that $\mathbf{V} \backslash \mathbf{MB}(X) \perp X \mid \mathbf{MB}(X)$. This in turn entails that for every $Y \in \mathbf{V} \backslash \mathbf{MB}(X), Y \perp X \mid \mathbf{MB}(X)$. The converse (known as the composition axiom of conditional independence) is not always true. However, for many families of distributions (e.g. Gaussian, multinomial, and any distribution in which the conditional independencies are algebraic constraints on the parameters), the conditions under which the converse is false are of Lebesgue measure 0 in the parameters (Lauritzen, 1996; Spirtes et al., 2000). We choose to test every $Y \in \mathbf{V} \backslash \mathbf{MB}(X), Y \perp X \mid \mathbf{MB}(X)$ for two reasons: first, it is much easier to test in terms of sample size needed for useful power, and it provides a list of p-values needed for the Vertex Checker, rather than a single p-value. However, one way that the Vertex Checker can fail is if composition is false.

4. Theoretical Results

In this section, we show that under reasonable assumptions, the Vertex Checker provides a correct p-value in the large sample limit.

Assumption 1

1. *The samples $\{\mathbf{X}_i\}_{i=1}^N$ are i.i.d. with no missing values.*
2. *The measured variables are causally sufficient, and the true causal graph is a DAG without selection bias.*

3. $(\mathcal{G}, P)$ *satisfies CMA.*

4. *For a DAG* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *if for every* $Y \in \mathbf{V} \backslash \mathbf{MB}(X)$, $Y \perp X \mid \mathbf{MB}(X)$ *then* $\mathbf{V} \backslash \mathbf{MB}(X) \perp X \mid \mathbf{MB}(X)$.

Theorem 4 (Uniformity of p-values) *Let the null hypothesis H_0 be that the set of variables in the Markov Blanket of X in the input to the Markov Checker is correct. Given Assumption 1, under H_0, if the data sets for the statistical tests performed by the Vertex Checker are independent of each other, the p-values in $\mathcal{P}$ in the Vertex Checker for variable X are uniformly distributed on the interval $[0, 1]$ in the large sample limit.*

Theorem 5 (Correctness of p-values) *Let the null hypothesis H_0 be that the set of variables in the Markov Blanket of X in the input to the Markov Checker is correct. Given Assumption 1, under H_0, if the data sets for the statistical tests performed by the Vertex Checker are independent of each other, the probability of falsely rejecting H_0 is less than or equal to $U(\mathcal{P})$.*

Theorem 4 follows directly from Theorem 3 and the **Causal Markov Assumption**. According to CMA, each vertex X is independent of all other variables conditioned on its Markov Blanket, **MB**(X). According to Theorem 3, if the p-values are obtained from independent samples, they should follow a uniform distribution. Theorem 5 then follows directly from Theorem 4 and Assumption 1. Theorem 5 does not indicate that the Vertex Checker is useful by itself, since it is always possible to find causal graphs that satisfy the Markov condition, e.g. any complete graph. That is why we only use the Vertex Checker in order to test the output of a CDA, which itself imposes simplicity constraints on its output, and is already avoiding inputs which are overly complex.

There are alternative sets of conditional independence relations other than the ones entailed by the Markov Blanket of X that could be used to test a local subgraph of X, e.g. the Local Markov Condition. Testing the Markov Blanket of X however, is more local than testing the local Markov condition, since the a test of the local Markov condition may fail because of an error in a feature of the DAG that is an arbitrary distance from X, e.g. mistaking a descendant of X for a non-descendant of X. A disadvantage of using the Markov Blanket to select conditional independencies to test is that this does not test the structure within the Markov Blanket, only the membership in the Markov Blanket, and the set of Markov Blanket conditions for each vertex does not entail the Global Markov Condition.

In the large sample limit, where the assumptions of a consistent causal search algorithm (including parameteric assumptions) hold exactly, no further post-processing (as in the Vertex Checker) is needed. However, on finite sample sizes, where the assumptions may not hold exactly, it is not uncommon for causal search algorithms to produce output that is not Markov to the population distribution. In those cases, the combination of a causal search algorithm and a post-processor which rejects parts of the output can be more useful than either part (the causal search algorithm and the Vertex Checker) is alone. However, whether that is the case for a particular data set depends on how likely the CDA is to make errors that could be corrected by the Vertex Checker, and how likely the Vertex Checker is to make errors. For that reason, we have done extensive simulation tests.

4.1. Implementation of the Vertex Checker

The implementation of Algorithm 1 can be challenging due to the difficulty of selecting appropriate conditional independence (CI) tests, and conducting those tests in such a way that the data providing

the p-values are independent of each other (as required by the theorem). CI testing is inherently more complex than testing for unconditional independence (Bergsma, 2004; Lauritzen, 1996). We discuss different tests of conditional independence under different assumptions in the Appendix E.

Another concern in implementing Algorithm 1 is that collecting p-values from conditional independence tests involving shared variables may introduce dependency among the p-values. This could violate the independence assumption of the uniformity test. While this issue can be theoretically addressed by limiting the number of overlapping samples when calculating p-values (Bickel, 2004; Meinshausen and Bühlmann, 2010), it has been shown empirically (Ramsey et al., 2024) that such worries are unnecessary, and the results are not significantly affected by such dependencies.

5. Experimental Results

In this section, we demonstrate the effectiveness of the Vertex Checker on both simulated and real-world datasets. We divide the simulations into graphs rejected (globally fail) or accepted (globally pass) by the Markov Checker. For graphs that fail globally, we highlight the presence of vertices with good approximations to the parent sets. Conversely, for graphs that pass globally, we show that not all components are reliable, exposing local areas that may fail despite strong global performance. This granularity is important for downstream tasks on graph-based models and understanding their local structures.

For the real-world application, we apply the Vertex Checker to the Sachs dataset (Sachs et al., 2005). Although the algorithms used to estimate the causal graph fail the Markov Checker, strong local performance is observed for several individual vertices, consistent with the domain-specific knowledge presented in the dataset.

5.0.1. EVALUATION METRICS

The Vertex Checker tests whether the vertices in the Markov blanket are approximately correct in the DAG (or CPDAG). To quantify this, we use two key metrics, the F1 score and IDA score (Saito and Rehmsmeier, 2015; Maathuis et al., 2009). Because there are various features of the Markov blanket which can be useful, we apply the F1 score and and IDA score to a variety of such features.

F1 score (Saito and Rehmsmeier, 2015; van Rijsbergen, 1979): The F1 score combines precision and recall to provide a balanced measure of performance, with details in Appendix B. In our experiments, we calculate F1 score for various features of the Markov blanket of a given vertex: (1) *Parents*: identifying which vertices serve as parents of X, (2) *Children*: identifying which vertices are children of X, (3) *Adjacency*: identifying vertices that are adjacent to X, and (4) *Orientation*: identifying vertices that are adjacent to X with directed edges.

IDA score: The IDA score builds upon the IDA (Intervention Calculus When the DAG is Absent) (Maathuis et al., 2009; Nandy et al., 2017), which estimates linear coefficients between variable pairs in a linear Gaussian CPDAG. Since CPDAGs contain undirected edges with unknown orientations, a single estimate of a linear coefficient isn't always possible. IDA addresses this by orienting each undirected edge in all compatible directions, estimating the coefficient for each orientation, and using the min and max of these estimates to provide an interval. We calculate the distance from this interval to the true coefficient, where the distance is zero if the true value lies within the interval, or the minimum distance to its endpoints otherwise. Additionally, the IDA score computes the absolute difference between the true coefficients and the interval estimates for the co-

efficients of a vertex's parent set. This allows us to evaluate whether the Vertex Checker improves estimation accuracy. More details can be found in Appendix C.

5.1. Simulation Results

Algorithms	Graph Types	Variable Size	Sample Size	Evaluation Metrics
PC	Linear Gaussian	20, 50	500, 1500	F1 score: Adjacency
	Linear Exponential			F1 score: Orientation
FGES	Nonlinear			F1 score: Parents
BOSS	Discrete			F1 score: Children
	Mixed Type			IDA score

Table 1: Simulation settings for evaluating the PC, FGES, and BOSS algorithms under various graph types, variable sizes, and sample sizes using multiple evaluation metrics.

To demonstrate the robustness of the Vertex Checker, as shown in Table 1, we conduct experiments on a variety of graph types, algorithms, variable and sample sizes. Due to space limitations, we provide a subset of the results here, with additional details available in the Appendix G, as well as the simulation details (see the Appendix F.1, A.2). We are not testing the algorithms that provide the graphs that are input to the Vertex Checker; we are testing how well the Vertex Checker is able to use these outputs to select vertices which have relatively accurate local structures (e.g. parents), or relatively accurate estimates of the structural coefficients of related edges.

In our experiments, we evaluate the causal model on the same data used for training. This raises a potential concern that such evaluation might lead to overly optimistic performance estimates (Hastie et al., 2009; Bishop and Nasrabadi, 2006). To address this issue, we conducted additional experiments using a train-test split, where the model is trained on one portion of the data and evaluated on an unseen portion. This approach highlights the robust performance of the Vertex Checker when applied to new data. Part of the results are presented in Appendix G.2.

5.1.1. VERTEX CHECKER ON MARKOV-REJECTED GRAPHS

By simulating and testing Vertex Checker on graphs rejected by the Markov Checker ($M-rejected$), we notice that oftentimes around 45% of the vertices actually pass the Vertex Checker as shown in Figure 4, indicating their good local Markov structures (V_{pr}: variable passing rate). This further demonstrates the necessity of Vertex Checker in uncovering the local properties of graphs.

To evaluate the effectiveness of the Vertex Checker, we separate the vertices into two groups: those that *PASS* the test and those that *FAIL*. For each group, we compute the average Vertex Checker p-values and evaluation metrics (as defined earlier). We then calculate the correlation between the p-values and the evaluation metrics. A significant positive correlation (or negative in the case of the IDA score) would indicate that the Vertex Checker effectively distinguishes vertices with better local structures. The detailed procedure is summarized in Appendix D.

Table 2 presents the Pearson correlations between the Vertex Checker p-values and the F1 score on Adjacency and Parents, with the first number in brackets representing the correlations and the second the p-values, under linear Gaussian distribution. Across different settings and models, the

correlations are consistently significantly positive. This suggests that vertices with higher Vertex Checker p-values are significantly more likely to have accurate adjacent edges and parent sets in the linear Gaussian cases.

In Table 3, we present the results for linear non-Gaussian, nonlinear, and discrete data types under various algorithms. For simplicity, we fix the size of the data, with the number of variables equal 50 and the number of samples equal 1500. We can see that the results continue to show significant positive correlations, indicating that the Vertex Checker is robust and effective across a wide variety of data types and scenarios.

In practice, we often care more about the the accuracy of quantitative predictions. To further demonstrate the effectiveness of the Vertex Checker, we evaluated its ability in distinguishing vertices with more accurate estimated edge coefficients under linear models.

Using the IDA score, we recorded the average distance between the estimated parent coefficients of vertices and the ground truth values. The overall results, shown in Figure 1 and Figure 2, demonstrate that vertices passing the Vertex Checker consistently have lower distances from the true coefficients compared to those that fail. We further summarize the correlation between the IDA score and Vertex Checker p-values in Table 7, which indicates that vertices with higher p-values are more likely to have accurate estimations on parent coefficients. The results provide additional support for the reliability and utility of the Vertex Checker in identifying accurate local structures.

5.1.2. VERTEX CHECKER ON MARKOV-ACCEPTED GRAPHS

Analogously to the case where an output CPDAG is globally rejected by the Markov Checker, we aim to demonstrate that having a globally well-performing output CPDAG graph is far from sufficient for reliable conclusions about the local graphs around each vertex. As illustrated in Figure 3, oftentimes vertices fail local checks even when the graph as a whole passes.

We show that the Vertex Checker remains highly informative by identifying the subset of vertices with problematic local structures, as shown in Table 4. This capability is crucial for providing a more granular view of the graph's structure, allowing us to pinpoint specific vertices that deviate from expected patterns.

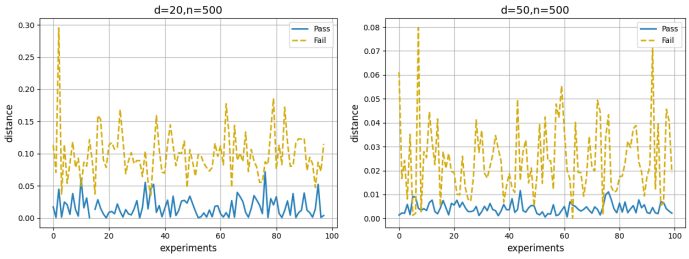

Figure 1: IDA score results for PC algorithm with linear exponential distribution (500 samples)

5.2. Empirical Results

To demonstrate the applicability of the Vertex Checker, we applied it to the well-known Sachs dataset (Sachs et al., 2005), a benchmark in systems biology frequently used for causal discovery and inference. The dataset consists of continuous measurements of 11 biochemical signals in human

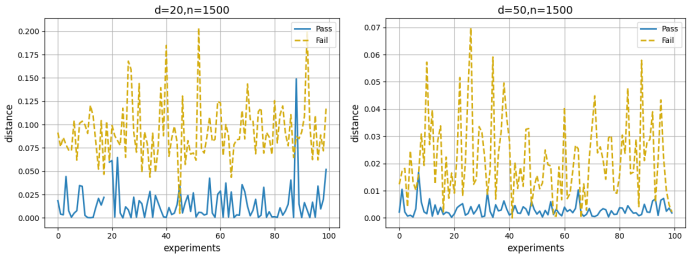

Figure 2: IDA score results for PC algorithm with linear exponential distribution (1500 samples)

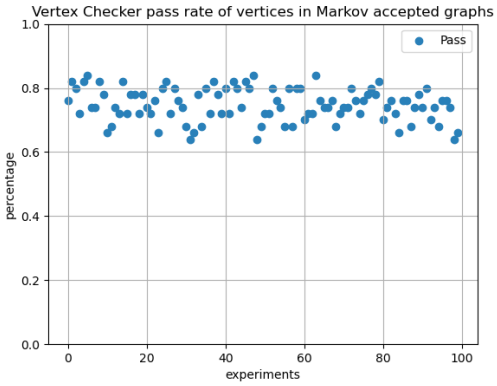

Figure 3: V_{pr} on M-accepted graphs

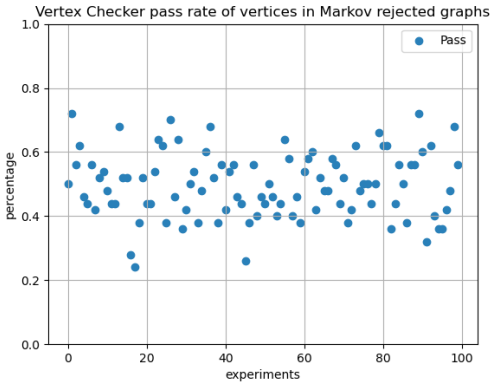

Figure 4: V_{pr} on M-rejected graphs

immune system cells with 853 observational samples, making it a suitable candidate for testing CDAs.

We applied four CDAs: PC, FGES, BOSS, and GRaSP. After generating the estimated graphs for each algorithm, we evaluated them using both the Markov Checker and the Vertex Checker, presented in Table 10 and Figure 5, respectively. Since the data are predominantly non-Gaussian and the relationships are nonlinear (see Appendix H), we use KCI (Zhang et al., 2011) for conditional independence tests. In Figure 5, variables encircled in red indicate rejected by the Vertex Checker, while those in blue signify accepted. Detailed parameter settings for the algorithms are in Table 5.

As indicated by the significant p-values from the Markov Checker, all graphs fail the Markov Checker. However, the Vertex Checker reveals that some local structures are still reliable. For instance, in the BOSS graph, the variable *erk* passes the Vertex Checker, suggesting a well-reconstructed local neighborhood around this node. There are ongoing disputes about the ground truth of the data, but certain causal relationships are generally accepted. In particular, *erk* pointing to *akt* is confirmed by interventional experiment in (Sachs et al., 2005). Also, the KEGG database (Kanehisa, 2002) suggests *erk* pointing to several other proteins upstream, which is also reflected in the BOSS and GRASP outputs. Additionally, it is known that *pip2* and *pip3* form a cycle, which aligns with the fact that they fail the Vertex Checker in all graphs. Given the limited information available, the results should be interpreted with caution. But generally it is demonstrated that the Vertex Checker is empirically valuable as it can provide valuable insights in the local reliability, even when the global structure is incorrect.

	Adjacency			Parents		
	PC	FGES	BOSS	PC	FGES	BOSS
d = 20, n = 500	(0.6419, 5.26E-71)	(0.4274, 7.96E-50)	(0.3699, 6.74E-21)	(0.2934, 2.20E-13)	(0.1157, 1.24E-04)	(0.0987, 1.55E-02)
d = 50, n = 500	(0.6341, 8.36E-69)	(0.4805, 5.95E-13)	(0.7650, 6.47E-59)	(0.2835, 1.46E-12)	(0.3768, 3.78E-08)	(0.2176, 1.45E-04)
d = 20, n = 1500	(0.6994, 2.59E-89)	(0.2858, 6.80E-06)	(0.2562, 1.40E-13)	(0.4690, 3.74E-34)	(0.1735, 7.02E-03)	(0.0173, 6.22E-01)
d = 50, n = 1500	(0.8200, 5.74E-85)	(0.2449, 2.98E-04)	(0.6647, 1.26E-39)	(0.5860, 6.52E-71)	(0.2574, 1.40E-04)	(0.1290, 2.55E-02)

Table 2: Correlation between F1 score and p-values on Adjacency and Parents: Linear Gaussian

	linear (exp)	nonlinear (mlp)	discrete
PC	(0.8687, 2.41E-62)	(0.7092, 5.40E-38)	(0.6716, 1.04E-40)
FGES	(0.8547, 1.94E-55)	(0.5876, 5.85E-20)	(0.1904, 6.13E-04)
BOSS	(0.4084, 1.93E-12)	(0.7210, 2.19E-33)	(0.2488, 1.74E-06)

Table 3: FAIL: Correlation between F1 Score on parent sets and Vertex Checker p-values

6. Limitations of the Vertex Checker and Future Work

In addition to the limitations of the Markov Checker already noted, it is also possible that there are not enough conditional independence relations in $\mathcal{P}$ in line 2 of the Algorithm 1 to make the Vertex Check a powerful enough test of whether the p-values are drawn from a $U(0,1)$ distribution to be useful. To deal with this we have drawn multiple subsamples from the data, to increase the number of p-values checked for uniformity. The tradeoff of doing this multiple subsampling is that it increases the dependence between the p-values, which is required for the uniformity to hold. However, simulations on the Markov Checker indicate that the dependence from subsampling has very little effect on the uniformity of the p-values (Ramsey et al., 2024). Another alternative is to use approaches other than the Uniformity test for the null hypothesis. We emphasize that this paper aims to provide a framework for testing the local structure (i.e., Markov Blanket), which is independent of the specific consistent test used for the null hypothesis. In addition to the vanilla Uniformity test, various multiple testing correction methods can be applied, including Bonferroni Correction, Benjamini-Hochberg (BH) Correction, Benjamini-Yekutieli (BY) Correction, and Holm-Bonferroni Correction, which theoretically enhance the model's stability. In addition, Fisher's method can be used as an alternative to the Uniformity test, addressing issues related to p-value dependency and the potential insufficiency of p-values for achieving a powerful test. (Tillman, 2009). In Appendix

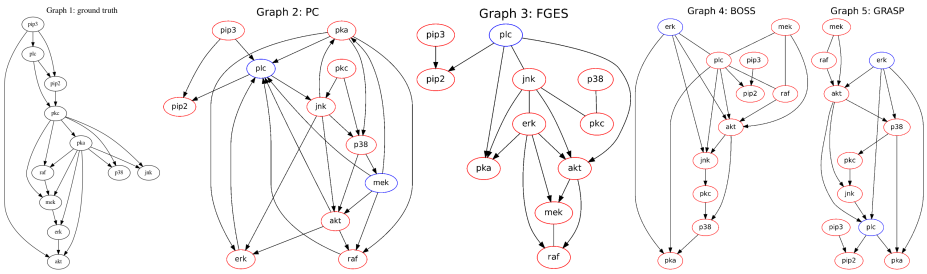

Figure 5: Vertex Checker on Sachs Dataset

	linear (exp)	nonlinear (mlp)	discrete
PC	(0.2359, 7.09E-05)	(0.3557, 1.43E-08)	(0.1838, 2.93E-05)
FGES	(0.3049, 1.20E-06)	(0.1730, 3.68E-03)	(0.2380, 3.04E-04)
BOSS	(0.1790, 1.55E-03)	(0.2189, 9.67E-05)	(0.1322, 3.24E-03)

Table 4: PASS: Correlation between F1 Score on parent sets and Vertex Checker p-values

G.4, we compare the empirical performance of the model under different tests on simulated data, suggesting that the choice of test does not significantly affect the Vertex Checker performance.

In addition, the Vertex Checker has a disadvantage shared by all valid statistical tests. At very large sample sizes, even tiny deviations from the true DAG (or any deviations from the distributional assumptions made by the conditional independence tests providing the p-values) will lead to rejection of a model that is very close to the truth and perfectly usable (Spirtes et al., 2000; Haughton, 1988). The simulation tests that we have done at common sample sizes (in the thousands) does not suffer from this problem, but at large enough sample sizes, this will become a significant problem. Since almost all real data sets will violate some assumptions to at least a small degree, this will lead to all models being rejected if there are enough measured variables and a large enough sample size. However, simulations indicate that the Vertex Checker is still useful at sample sizes where all plausible models are being globally rejected, because the local subgraphs connected to some individual vertices are not being rejected. One drawback of the Vertex Checker is that it may not be able to reject some vertices with unnecessary edges. Similarly to the Markov Checker, here we could allow the Vertex Checker impose a simplicity constraint, in addition to the simplicity constraints imposed by the CDAs used to generate the input. The full algorithm is in the Appendix D.1.

One potential direction for future work is to relax the assumptions underlying the Vertex Checker. For instance, while it is theoretically valid for any type of graph, we have not yet developed the corresponding algorithm to empirically validate this claim. Additionally, the current assumptions only consider graphs that consist solely of observed variables, whereas extending the approach to handle latent variables is crucial for real-world applications. Also, in biological datasets such as the Sachs dataset, new algorithms may be necessary to accommodate cycles in the graphs, expanding the applicability of the method to more complex causal structures.

7. Conclusion

In this paper, we introduced the Vertex Checker, the only statistical test that we are aware of that takes as input a causal graphical model $\mathcal{G}$, a vertex X, an alpha level, sample data, and a conditional independence test, and provides a non-parametric, asymptotically correct, statistical test of a local subgraph of X, is computationally feasible for dozens of variables, and is extendable to other kinds of causal graphical models. Through extensive simulations, we demonstrated the robustness of the Vertex Checker across various data types, causal graphs, and distributions both in terms of accuracy of graphical structure and of quantitative estimates of causal effects. Furthermore, we applied the Vertex Checker to the real-world Sachs dataset, showcasing its practical applicability in uncovering accurate substructures within causal graphs, even when the overall causal graphical model is rejected.

References

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

Paul Barrett. Structural equation modelling: Adjudging model fit. *Personality and Individual Differences*, 42(5):815–824, 2007.

P. M. Bentler and D. G. Bonett. Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin*, 88(3):588–606, 1980.

Peter M. Bentler. Comparative fit indexes in structural models. *Psychological bulletin*, 107(2): 238–246, 1990.

Wicher Pieter Bergsma. Testing conditional independence for continuous random variables. 2004.

Peter J. Bickel. Some theory for fisher's z-transformation and weighted chi-squared methods. *The Annals of Statistics*, 32(3):871–890, 2004.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Kenneth A Bollen. *Structural Equations with Latent Variables*. John Wiley & Sons, 1989.

Michael W Browne and Robert Cudeck. Alternative ways of assessing model fit. *Sociological methods & research*, 21(2):230–258, 1992.

Peter Bühlmann, Jonas Peters, and Jacob Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2553, 2014.

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

David Maxwell Chickering and David Heckerman. Efficiently approximating the most probable explanation in bayesian networks. *Artificial Intelligence*, 92(1-2):283–302, 1997.

David Edwards. *Introduction to graphical modelling*. Springer Science & Business Media, 2000.

Yingying Fan and Tien D. Pham. Conditional independence testing for mixed-type variables with extensions to causal discovery. *Journal of Machine Learning Research*, 19(17):1–47, 2018.

David A Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2005.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.

Dominique M.A. Haughton. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16(1):342–355, 1988.

Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5(1):371–391, 2018.

Patrik O Hoyer, Shohei Shimizu, Ari Kerminen, and Marko Palviainen. Estimation of causal effects using linear non-gaussian causal models with hidden variables. *International Journal of Neural Systems*, 18(03):145–155, 2008.

Patrik O. Hoyer, Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

Rick H Hoyle. *Structural equation modeling: Concepts, issues, and applications*. Sage, 1995.

Li-tze Hu and Peter M Bentler. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural equation modeling: a multidisciplinary journal*, 6(1):1–55, 1999.

Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncovering causal relationships in python. *Journal of Machine Learning Research*, 21(37):1–5, 2019.

Minoru Kanehisa. The kegg database. In *'In silico'simulation of biological processes: Novartis Foundation Symposium 247*, volume 247, pages 91–103. Wiley Online Library, 2002.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

A. N. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91, 1933.

Steffen L Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

Marloes H Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

Robert C MacCallum, Michael W Browne, and Hazuki M Sugawara. Power analysis and determination of sample size for covariance structure modeling. *Psychological methods*, 1(2):130, 1996.

Mary L. McHugh. The chi-square test of independence. *Biochemia medica*, 23(2):143–149, 2013.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Alessio Moneta, Nadine Chlaß, Dirk Entner, and Patrik O. Hoyer. Causal search in structural vector autoregressive models. *Econometric Reviews*, 32(5-6):697–733, 2013.

Ricardo P Monti, Kun Zhang, and Aapo Hyvärinen. Causal discovery with general non-linear relationships using non-linear ica. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

Joris M. Mooij, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf. Cyclic causal discovery from continuous equilibrium data. *Journal of Machine Learning Research*, 15:1337–1383, 2013.

Matthew D. Moran. Arguments for rejecting the sequential bonferroni in ecological studies. *Oikos*, 100(2):403–405, 2003. doi: 10.1034/j.1600-0706.2003.12010.x.

P Nandy, A Hauser, and Marloes H Maathuis. Estimating intervention effects in high-dimensional systems with latent variables. *The Annals of Statistics*, 45(2):647–674, 2017.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

David Peterson and Joseph Y Halpern. Causal learning with deep neural networks. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

Joseph D Ramsey, Stephen J Hanson, and Clark Glymour. Six problems for causal inference from fmri. *Neuroimage*, 49(2):1545–1558, 2010.

Joseph D Ramsey, Clark Glymour, and Ruben Sanchez-Romero. Million variables and beyond: Ges search for large-scale continuous and discrete causal inference. In *Proceedings of the 8th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

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

Joseph D. Ramsey, Bryan Andrews, and Peter Spirtes. Choosing dag models using markov and minimal edge count in the absence of ground truth. *arXiv preprint arXiv:2409.20187*, 2024.

Thomas S. Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002. doi: 10.1214/aos/1031689015.

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

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers. *PloS one*, 10(3):e0118432, 2015.

Richard Scheines. D-separation tutorial, n.d. URL https://www.andrew.cmu.edu/user/scheines/tutor/d-sep.html. Accessed: 2024-10-30.

Felix Schlussel. Causal graphs for neural networks. *Proceedings of the Workshop on Artificial Intelligence Safety (AISafety)*, 2020.

William R Shadish, Thomas D Cook, and Donald T Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Wadsworth Cengage Learning, 2002.

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

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael I Jordan. Estimation of a linear non-gaussian acyclic model using information geometry. *Journal of Machine Learning Research*, 7:1225–1248, 2009.

Shohei Shimizu, Kenneth A Bollen, Zhenzhen Chen, Patrik O Hoyer, and Aapo Hyvärinen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011.

Bill Shipley. Cause and correlation in biology: A user's guide to path analysis, structural equations and causal inference. *Cambridge University Press*, 2000. doi: 10.1017/CBO9781139017594.

N. Smirnov. Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, 19(2):279–281, 1948. doi: 10.1214/aoms/1177730256.

Peter Spirtes. Introduction to causal inference. *Journal of Machine Learning Research*, 11:1643–1662, 2010.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT press, 2000.

Robert E Tillman. Structure learning with independent non-identically distributed data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1041–1048, 2009.

C. J. van Rijsbergen. *Information retrieval*. Butterworth-Heinemann, 1979.

Lilian Weng and Denny Lee. Representation learning for causality. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*, 2019.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounding and selection bias. *Artificial Intelligence*, 172(16–17):1873–1896, 2008. doi: 10.1016/j.artint.2008.08.001.

Kun Zhang and Aapo Hyvärinen. Causal inference and graphical models with local structure. *Journal of Machine Learning Research*, 13:2409–2430, 2012.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. On the identifiability and consistency of nonlinear additive noise models for causal discovery. *Advances in Neural Information Processing Systems*, pages 619–626, 2009.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 804–813, 2011.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal discovery in the presence of missing data. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3584–3590. AAAI Press, 2017.

Appendix A. Background Information

A.1. Graphical Definitions

A *directed graph* contains a set of vertices **V** and a set of *directed edges* $X \rightarrow Y$ between distinct members of **V**. If $X \rightarrow Y$ in a directed graph, then X is a *parent* of Y and Y is a *child* of X. A path *P* from *X* to *Y* is a sequence of consecutive edges (independent of their direction). If all of the edges in a path *P* point in the same direction, then *p* is a *directed path*. Given a *path P* that contains $X \rightarrow Y \leftarrow Z$ we refer to *Y* as a *collider* on *P*. *X* is an *ancestor* of *Y* (and *Y* is a *descendant* of *X*), if there exists a directed path from *X* to *Y*. A directed graph is *acyclic* (a DAG) if there is no directed path from a vertex to itself.

d-separation is a graphical relation used in DAGs used to determine whether a DAG entails that two sets of variables are conditionally independent given a third set. It provides a formal rule for reading conditional independencies from the structure of a graph. Given a DAG, if **X**, **Y**, and **Z** are disjoint sets of vertices, then **X** and **Y** are *d-connected* by **Z** in G if and only if there exists a path U between some vertex in **X** and some vertex in **Y** such that for every collider C on U, either C or a descendant of C is in **Z**, and no non-collider on U is in **Z**. **X** and **Y** are *d-separated* by **Z** in G if and only if they are not d-connected by **X** in G (Pearl, 2009; Scheines, n.d.).

A.2. Causal Discovery Algorithms

In this section, we give brief introductions to different CDAs. They represent a diverse set of approaches within the causal discovery framework:

- **Peter-Clark (PC)**: The PC algorithm is a *constraint-based* method for learning the causal structure of a set of variables from data. It assumes the absence of latent variables, selection bias, cycles, etc. It takes data and operates by testing conditional independencies between variables and first progressively removing edges from a fully connected undirected graph, and then orienting the undirected edges as much as possible. The output is a CPDAG. It has been shown to produce the correct CPDAG with probability 1 in the large sample limit. (Spirtes et al., 2000).

- **Fast Greedy Equivalence Search (FGES)**: GES is a *score-based* method that searches over Markov equivalence classes of DAGs. It takes data and starts from an empty graph and greedily adds or removes edges to maximize a score, such as the Bayesian Information Criterion (BIC), over the space of CPDAGs. It outputs a CPDAG (Ramsey et al., 2017).

- **BOSS (Best Order Score Search)**: BOSS is a bayesian search-based algorithm, efficiently searching over the space of permutations of the order of the variables for causal graphs. Its increased accuracy as compared to PC and FGES makes it an ideal candidate for comparison against more traditional methods like PC and FGES. It takes data as input and outputs a DAG. (Andrews et al., 2023).

- **GRASP (Greedy Sparsest Permutation)**: GRASP aims to efficiently learn the underlying causal graph by finding the sparsest permutation that satisfies conditional independence relationships among variables. It is particularly useful when the number of variables is large relative to the number of samples. It takes data and outputs a fully oriented DAG Lam et al. (2022).

By including these algorithms (only three of which were used on the simulated data), we ensure that we are testing a range of methodologies—constraint-based (e.g. PC), score-based (e.g. FGES), and a permutation approach (e.g. BOSS)—which provides a comprehensive and reliable evaluation of causal discovery performance.

A.3. Evaluation Tools

- **Chi-squared test**: a method for evaluating model fit in structual equation modeling, where the test statistic measures the discrepancy between the maximum likelihood estimated covariance matrix and the observed sample covariance matrix, which has the Chi-squared distribution in the large sample limit (Bollen, 1989).
- **Comparative Fit Index (CFI)**: a measure used to evaluate the fit of a causal model by comparing it to a baseline model, with values close to 1 indicating a good fit (Bentler, 1990).
- **Normed Fit Index (NFI)**: a measure use to evaluate the goodness-of-fit of a hypothesized causal model to a null model, where values closer to 1 indicate a better fit (Bollen, 1989).

Appendix B. F1 score

B.1. Definition

The F1 score is a performance metric used to evaluate a model's accuracy, defined as the harmonic mean of precision and recall. An F1 score of 1 represents perfect precision and recall, and a score of 0 indicates the worst performance. In the following equations we denote the number of *True Positives* as TP, the number of *False Positives* as FP, and the number of *False Negatives* as NP.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We calculate the F1 score for different local structures around a vertex X as follows. Here the estimated $MB(X)$ and the true $MB(X)$ represent the Markov blankets of X from the estimated CPDAG and the true CPDAG, respectively.

- **F1 Score on Adjacency**:
  - **True Positives (TP)**: The number of edges adjacent to X in the estimated $MB(X)$ that are also adjacent in the true $MB(X)$.
  - **False Positives (FP)**: The number of edges adjacent to X in the estimated $MB(X)$ that are not adjacent in the true $MB(X)$.
  - **False Negatives (FN)**: The number of edges adjacent to X in the true $MB(X)$ that are not adjacent in the estimated $MB(X)$.
- **F1 Score on Orientation**:

  - **True Positives (TP)**: The number of directed edges in the estimated $MB(X)$ that are correctly oriented according to the true $MB(X)$.
  - **False Positives (FP)**: The number of directed edges in the estimated $MB(X)$ that are either undirected or incorrectly oriented (reversed) in the true $MB(X)$.
  - **False Negatives (FN)**: The number of undirected edges (adjacent in both graphs) in the estimated $MB(X)$ that are directed in the true $MB(X)$.

- **F1 Score on Parents**:
  - **True Positives (TP)**: The number of parents of X in the estimated graph that are parents in the true graph.
  - **False Positives (FP)**: The number of parents of X in the estimated graph that are not parents in the true graph.
  - **False Negatives (FN)**: The number of missed parents of X in the estimated graph.

- **F1 Score on Children**:
  - **True Positives (TP)**: The number of children of X in the estimated graph that are children in the true graph.
  - **False Positives (FP)**: The number of children of X in the estimated graph that are not children in the true graph.
  - **False Negatives (FN)**: The number of missed children of X in the estimated graph.

We also include the algorithm on calculating F1 score here in Algorithm 2.

Algorithm 2 Precision, Recall, and F1 Score Calculation with P-values from the Vertex Checker

Input: P-values from the Vertex Checker for $\{X : i = 1, \ldots, d\}$, threshold th
Output: Precision, Recall, F1 scores, and their correlations with p-values
Initialize: set `Pass` = $\{\}$, set `Fail` = $\{\}$ **foreach** X **do**
  if *P-value of* $X > th$ **then**
    Add X to `Pass`
  end
  else
    Add X to `Fail`
  end
end
Calculate precisions, recalls, and F1 scores for all X in the estimated CPDAG with respect to the true CPDAG
Compute correlations between precisions, recalls, F1 scores, and the p-values from the uniformity test; record the correlation values and their corresponding p-values

Appendix C. IDA score

In this paper, we use the widely-known IDA (Intervention Calculus when the DAG is Absent) method (Maathuis et al., 2009; Nandy et al., 2017) to evaluate a vertex's parent set in CPDAGs.

One major challenge when assessing the performance of a vertex in comparison to the ground truth model is that edges in CPDAGs can be undirected. As a result, we cannot directly compare the estimated structural coefficients to the true structural coefficients. We use IDA to estimate the bounds of a linear coefficient. Then we employ the following IDA score. If the true coefficient $Coef[X, Y]$ lies within the interval $(\min(Coef_{es}[X, Y]), \max(Coef_{es}[X, Y]))$, we assign a difference of 0, indicating no discrepancy between the true and estimated coefficients. However, if $Coef[X, Y]$ falls outside the interval, we assign the difference as the minimum absolute distance between the true coefficient $Coef[X, Y]$ and the boundary of the interval, i.e., the closest of $(\min(Coef_{es}[X, Y]), \max(Coef_{es}[X, Y]))$. The details are shown below in Algorithm 3.

Algorithm 3 GetDistances

Input: Estimated DAG or CPDAG $\mathcal{G}_{es}$, true DAG $\mathcal{G}$, true edge-strengths $Coef$, data
Output: dist
Initialize: estimated edge-strength $Coef_{es} \leftarrow \{\}$;
foreach X *in* G_{es} **do**
  PA$_{ls}(X)$ = list of possible parent sets of X
  foreach *parent set* p *in* ***PA***$_{ls}(X)$ **do**
    Set regcoeffs $\leftarrow$ regress(X, p)
    foreach *vertex* Y *in* p **do**
      Update $Coef_{es}[X, Y] \leftarrow Coef_{es}[X, Y]$ + regcoeffs[Y]
    end
    foreach *vertex* Y *not in* p **do**
      Update $Coef_{es}[X, Y] \leftarrow Coef_{es}[X, Y]$ + $\{0\}$
    end
  end
  foreach Y *in* $\mathcal{G}_{es}$ **do**
    if $\min(Coef_{es}[X, Y]) \leq Coef[X, Y] \leq \max(Coef_{es}[X, Y])$ **then**
      Set dist$[X, Y] \leftarrow 0$
    else if $Coef[X, Y] \leq \min(Coef_{es}[X, Y])$ **then**
      Set dist$[X, Y] \leftarrow$ abs($Coef[X, Y]$ - $\min(Coef_{es}[X, Y])$)
    end
    else
      Set dist$[X, Y] \leftarrow$ abs($Coef_{es}[X, Y]$ - $\max(Coef[X, Y])$
    end
  end
end
Return dist

Appendix D. Examine the Effectiveness of the Vertex Checker with Pearson Correlation

The detailed procedure for testing the effectiveness of the Vertex Checker is provided in Algorithm 4. We begin by applying the Vertex Checker to each vertex, dividing them into two groups: those that pass the test and those that fail. For each group, we calculate the Pearson correlation between the average of various metrics (see B) and the average p-values. A significantly positive correlation

(or negative in the IDA score case) indicates that higher p-values (vertices passing the test) are associated with better metric performance. This demonstrates the Vertex Checker's effectiveness in identifying vertices with strong performance.

Algorithm 4 Evaluate Effectiveness of Vertex Checker

Input: Set of vertices V from estimated CPDAG $\mathcal{G}_{es}$, Evaluation metrics M, p-values P
Output: Correlations ρ, p_ρ
Initialize: set $\mathbf{V}_{pass} = \{\}$, $\mathbf{V}_{fail} = \{\}$
foreach *vertex* $v \in V$ **do**
  if *VertexChecker(v) returns* `True` **then**
    Add v to $\mathbf{V}_{pass}$
  end
  else
    Add v to $\mathbf{V}_{fail}$
  end
end
Compute average p-values $\overline{P}_{pass}$ and metrics $\overline{M}_{pass}$ for $\mathbf{V}_{pass}$
Compute average p-values $\overline{P}_{fail}$ and metrics $\overline{M}_{fail}$ for $\mathbf{V}_{fail}$
Concatenate the p-values $P = [\overline{P}_{pass}, \overline{P}_{fail}]$ and evaluation metrics $M = [\overline{M}_{pass}, \overline{M}_{fail}]$
Calculate correlation and corresponding p-value ρ, p_ρ between $\overline{P}$ and $\overline{M}$

D.1. Algorithm on imposing simplicity on Vertex Checker

Here we present the detailed algorithm with the simplicity constraints on the Vertex Checker to avoid unnecessary edges in Algorithm 5.

Algorithm 5 Select Vertex with Fewest Adjacent Edges After Vertex Checker

Input: Set of vertices V from different causal graphs $\{G_1, G_2, \ldots, G_k\}$, Vertex Checker function
Output: Vertex with the fewest adjacent edges: SelectedVertex
Initialize: SelectedVertex $= \emptyset$, minEdges $= \infty$
foreach *vertex* $v \in V$ **do**
  Set adjacentEdges$_v = 0$ **foreach** *graph* G_i **do**
    if *VertexChecker(v, G_i) returns* `True` **then**
      Count the adjacent edges for v in G_i
      adjacentEdges$_v$ $+=$ edge count of v in G_i
    end
  end
  if *adjacentEdges$_v$ $<$ minEdges* **then**
    minEdges $=$ adjacentEdges$_v$
    SelectedVertex $= v$
  end
end
Output: Vertex with the fewest adjacent edges: SelectedVertex

Appendix E. Additional Information on the Consistency of CI tests in Various Cases

We include the additional asymptotic consistency of CI tests under more complex data types, demonstrating the consistency of the Vertex Checker in dealing with different kinds of data.

E.1. Linear continuous cases

When the data are generated by a linear Gaussian causal DAG, the Fisher's Z-test is guaranteed to be asymptotically consistent in estimating CI relations.

Theorem 6 *(Edwards, 2000)[Asymptotic Distribution of Fisher's z-test for Conditional Independence] Let $\rho_{XY|\mathbf{W}}$ be the partial correlation coefficient between two variables X and Y, conditioned on a set of variables $\mathbf{W}$. Under the null hypothesis of conditional independence, the test statistic:*

$$z = \frac{1}{2}\log\left(\frac{1 + r_{XY|\mathbf{W}}}{1 - r_{XY|\mathbf{W}}}\right)\sqrt{n - |\mathbf{W}| - 3},$$

asymptotically follows a standard normal distribution $N(0,1)$ as the sample size n goes to ∞, where $|\mathbf{W}|$ denotes the number of variables in the conditioning set $\mathbf{W}$ and $r_{XY|\mathbf{W}}$ is the sample estimate of the partial correlation.

E.2. Nonlinear continuous cases

When the causal relations between variables are nonlinear, we use KCI test to better capture the conditional independence relations between variables.

Under the following assumptions, we have Theorem 7.

Assumption 2

- *The observations are independent and identically distributed (i.i.d.).*
- *The kernels used for X, Y, and $\mathbf{Z}$ are characteristic, ensuring that the kernel embeddings capture all dependencies.*
- *The kernels are bounded and continuous*
- *Sufficient sample size n*

Theorem 7 (Asymptotic Distribution of KCI Test for Conditional Independence) *(Zhang et al., 2011) Under the null hypothesis $H_0 : X \perp Y \mid \mathbf{Z}$ and assuming that the kernel functions used for X, Y, and $\mathbf{Z}$ are characteristic, bounded, and continuous, the test statistic T_n satisfies:*

$$\sqrt{n}\,(T_n - \mathbb{E}[T_n \mid H_0]) \xrightarrow{d} N(0, \sigma^2),$$

as n to ∞, where $\xrightarrow{d}$ denotes convergence in distribution, and σ^2 is the asymptotic variance of the test statistic under the null hypothesis.

E.3. Discrete cases

When the variables are discrete, we use a Chi-squared test to test the conditional independence relations between variables.

Under the following assumptions, we have Theorem 8.

Assumption 3

- *The observations are independent and identically distributed (i.i.d.).*
- *The sample size n is sufficiently large*
- *The variables are discrete with a finite number of levels.*

Theorem 8 (Asymptotic Distribution of Chi-squared Test for Conditional Independence) *(McHugh, 2013)*

Define the observed frequency O_{ijk} as the count of occurrences where $X = i$, $Y = j$, and $\mathbf{Z} = k$. $\mathbf{Z}$ here denotes a potential set of variables and k represents a joint state of these variables. We let the expected frequency under the null hypothesis be $E_{ijk} = \frac{O_{i\cdot k}O_{\cdot jk}}{O_{\cdot\cdot k}}$, where the dot notation $\cdot$ indicates summation over the corresponding index.

Suppose the number of variables in $\mathbf{Z}$ is m. The Chi-squared test statistic is given by:

$$\chi^2 = \sum_{i,j,k} \frac{(O_{ijk} - E_{ijk})^2}{E_{ijk}}.$$

Under the null hypothesis H_0, the test statistic χ^2 asymptotically follows a Chi-squared distribution with degrees of freedom given by:

$$df = (|X| - 1)(|Y| - 1)\prod_{t=1}^{m} |Z_t|,$$

where $|X|$, $|Y|$, and $|Z_t|$ are the number of levels of X, Y, and Z_t, respectively. That is:

$$\chi^2 \xrightarrow{d} \chi^2_{df},$$

as $n \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution.

E.4. Mixed-type cases

In real world, it is often the case that the dataset is mixed type. That is, we have both the continuous and discrete variables in the dataset. We also take this into consideration and use Degenerate Gaussian to test the CI relations.

Assumption 4

- *The data are generated from a Gaussian copula model*
- *The sample size n is sufficiently large*

- *The residual correlation $r_{XY|Z}$ is well-defined and estimated consistently from the data.*

Theorem 9 (Asymptotic Distribution of Degenerate Gaussian Test for Conditional Independence) *(Fan and Pham, 2018; Bergsma, 2004)*

Let X, Y, and $\mathbf{Z}$ be mixed-type random variables (continuous or discrete) with a joint distribution modeled by a Gaussian copula. The test statistic T_n is constructed as follows:

$$T_n = n \cdot r^2_{XY|\mathbf{Z}},$$

the test statistic T_n asymptotically follows a Chi-squared distribution with one degree of freedom:

$$T_n \xrightarrow{d} \chi^2_1,$$

as $n \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution.

Appendix F. Experimental Setting

F.1. Data Generation Process on Simulated Data

The data for the simulations are generated either using Tetrad (Ramsey et al., 2018) or Python.

Linear Cases (Pearl, 2009) In the linear case, the relationships between variables can be expressed as $\mathbf{X} = \mathbf{W}\mathbf{X} + \mathbf{E}$, where $\mathbf{E}$ is a vector of independent noise terms, following distributions such as Gaussian, Exponential, etc. The matrix $\mathbf{W}$ contains the structural coefficients representing the causal relationships between variables. The data generation process begins by first generating $\mathbf{E}$ and $\mathbf{W}$, and then calculating $\mathbf{X}$ from these components.

In this paper, we generate the structural coefficients $\mathbf{W}$ uniformly between 0 and 1, and the noise term E_X for arbitrary variable according to Gaussian, Exponential, and Uniform distributions as follows:

- **Gaussian Distribution**:

  $$E_X \sim \mathcal{N}(0, 1)$$

  The noise z is drawn from a standard normal distribution with mean 0 and variance 1.

- **Exponential Distribution**:

  $$E_X \sim \text{Exp}(1)$$

  The noise z is drawn from an exponential distribution with rate parameter 1.

- **Uniform Distribution**:

  $$E_X \sim \mathcal{U}(-1, 1)$$

  The noise E_X is drawn from a uniform distribution between -1 and 1.

Nonlinear Cases (Kalainathan and Goudet, 2019) In the nonlinear case, linear structural relationships cannot be applied. Instead, for each variable X, the relationship is defined as:

$$X = f(\text{PA}(X)) + E_X,$$

where $f(\cdot)$ is a nonlinear function of the parent set $\text{PA}(X)$, and E_X is the noise term for X. The noise term for each variable is simulated according to a specified distribution, and the parent set $\text{PA}(X)$ is determined from the adjacency matrix. After determining the parent set, the nonlinear function $f(\text{PA}(X))$ is applied to calculate the value of X.

For each vertex X, we generate the corresponding noise term E_X from $\mathcal{N}(0, 1)$. Let d_X denote the number of parents of X, and h_X represent the hidden size, which is a hypterparameter we specify in the models. We use two types of nonlinear functions: **mlp** and **mim**, which are defined as follows:

MLP: Multilayer Perceptron (Goodfellow et al., 2016)

$$X = \sigma(\text{PA}(X)W_1)W_2 + E_X$$

- $W_1 \sim \mathcal{U}(0.5, 2.0)$, with $W_1 \in \mathbb{R}^{d_X \times h_X}$ being the weight matrix between the input and the hidden layer.
- $W_2 \sim \mathcal{U}(0.5, 2.0)$, with $W_2 \in \mathbb{R}^{h_X \times 1}$ being the weight matrix between the hidden layer and the output.
- Both W_1 and W_2 are centered by multiplying each element by -1 with a 50% probability.
- $\sigma(\cdot)$ denotes the sigmoid activation function.

MIM: Mixed Interaction Model (Zhang et al., 2017)

$$X = \tanh(\text{PA}(X)W_1) + \cos(\text{PA}(X)W_2) + \sin(\text{PA}(X)W_3) + E_X$$

- $W_1, W_2, W_3 \sim \mathcal{U}(0.5, 2.0)$, with $W_1, W_2, W_3 \in \mathbb{R}^{d_X \times 1}$ being the weight vectors.
- $\tanh(\cdot)$ is the hyperbolic tangent function.
- $\cos(\cdot)$ is the cosine function.
- $\sin(\cdot)$ is the sine function.
- Similar to MLP, we center W_1, W_2, and W_3 by multiplying them by -1 with a 50% probability.

F.2. Parameter Settings for Sachs Dataset

Here we list the parameter settings when testing on the Sachs data for different models.

	PC	FGES	BOSS	GRASP
Significance level (α) / Penalty discount	0.05	14	14	14
Score/Test	Fisher's z	Fisher's z	BIC	Fisher's z & BIC

Table 5: Parameter settings for the PC, FGES, BOSS, and GRASP algorithms.

Appendix G. Additional Simulation Results

The experiments were conducted on a system running AlmaLinux with the following specifications: an AMD EPYC 7272 12-Core CPU (24 threads, 2.9 GHz), 128 GB of RAM, and 6.7 TB of total disk space.

We present additional simulation results here to demonstrate the effectiveness of the Vertex Checker.

Table 6 shows the correlations (and corresponding p-values) between the F1 scores for the correctness of adjacency and parent sets, and the Vertex Checker p-values when the data follow linear exponential distribution. These results are evaluated across different variable sizes, sample sizes, and algorithms. The results indicate that the correlations are almost always significantly positive.

Table 7 shows that correlation between the IDA score (defined in C) and Vertex Checker p-values. The results strongly indicate the effectiveness of Vertex Checker in the estimation of coefficients in the linear case.

G.1. Experimental Results on Chi-square Rejected Graphs

We evaluated the Vertex Checker on graphs that were rejected by the Chi-square test (Pearl, 2009; Bollen, 1989), demonstrating that it provides more informative insights than the classic Chi-square approach (Bollen, 1989). The results presented here focus on the scenario with 50 variables and 1500 samples. As shown in Figure 6, out of 100 simulated graphs that were all rejected by the Chi-square test, those variables that passed the Vertex Checker displayed strong performance in key metrics, including Adjacency and Parents. This highlights the Vertex Checker's effectiveness in identifying well-performing variables even within rejected graphs.

G.2. Experimental Results on Train-Test Data

It may be questioned whether it is appropriate to use the same dataset both to estimate the causal graph and to evaluate its performance. While this is not necessarily a critical issue, we also conducted experiments using a train-test split to address this concern. Specifically, we divided the data into training (70%) and test (30%) sets. The training set was used to estimate the causal graph, while the test set was used to apply the Vertex Checker to the resulting graph. Tables 8 and 9 present the F1 scores of the parent sets under nonlinear causal relationships, using the MLP and MIM models as described in Section F.1.

The same as the previous experiments, we first applied the Markov Checker to separate the vertices into *PASS* and *FAIL* groups, followed by the application of the Vertex Checker. The results show a significantly positive correlation between the F1 scores of the parent sets and the Vertex Checker p-values across different causal models, demonstrating the effectiveness of the Vertex Checker.

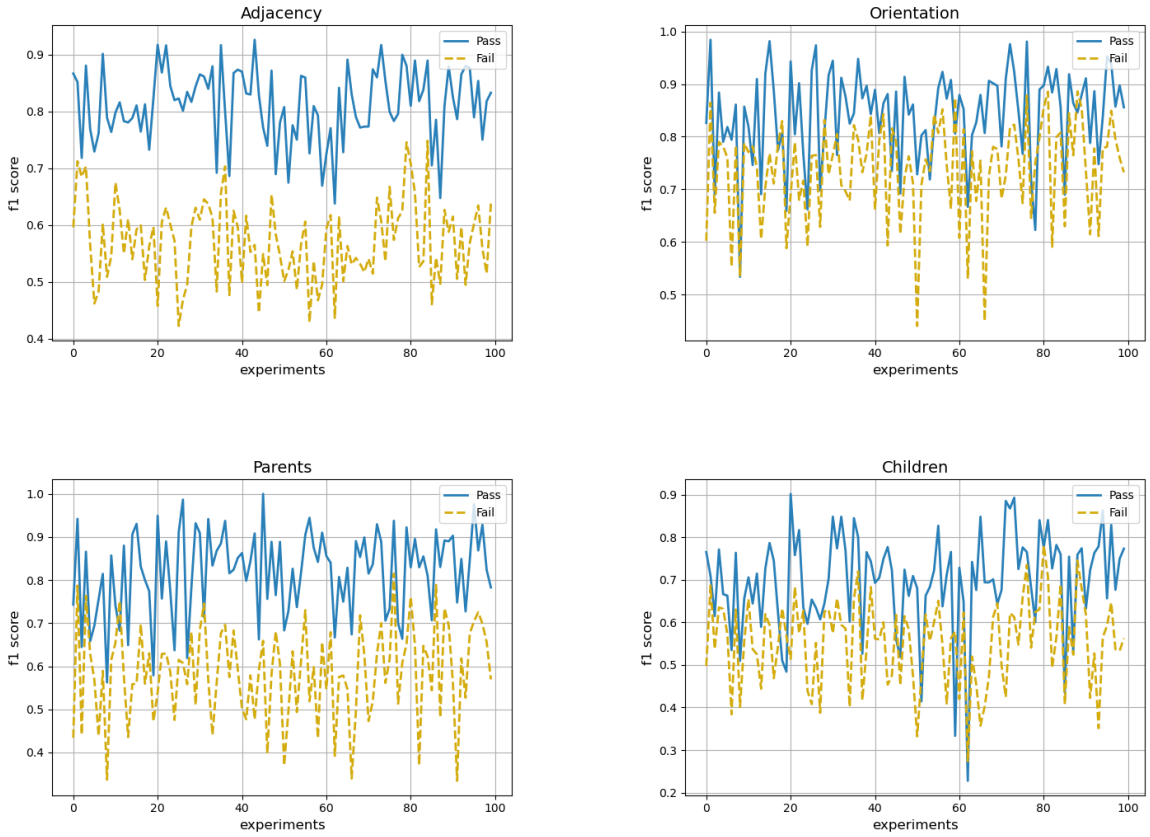

Figure 6: F1 score results for PC algorithm with linear exponential distribution on Chi-square rejected graphs (1500 Samples, 50 Variables)

	Adjacency			Parents		
	PC	FGES	BOSS	PC	FGES	BOSS
d = 20, n = 500	(0.4156, 2.79E-45)	(0.5702, 4.37E-17)	(0.2948, 2.26E-05)	(0.5009, 1.84E-20)	(0.2868, 8.62E-05)	(0.3004, 1.55E-05)
d = 50, n = 500	(0.7053, 1.54E-39)	(0.7292, 1.82E-34)	(0.4845, 1.56E-15)	(0.5203, 5.02E-19)	(0.5691, 1.45E-18)	(0.1638, 1.10E-02)
d = 20, n = 1500	(0.7621, 7.65E-58)	(0.1767, 1.23E-02)	(0.1911, 8.00E-04)	(0.3994, 7.61E-13)	(0.1632, 2.09E-02)	(0.1986, 5.39E-04)
d = 50, n = 1500	(0.6495, 9.48E-22)	(0.4400, 1.08E-10)	(0.2090, 1.57E-03)	(0.5035, 2.56E-12)	(0.3659, 1.33E-07)	(0.6990, 1.80E-34)

Table 6: Correlation between F1 score and p-values on Adjacency and Parents: Linear Exponential

	PC	FGES	BOSS
d = 20, n = 500	(-0.6720, 1.23E-27)	(-0.1580, 2.54E-02)	(-0.4311, 1.85E-10)
d = 50, n = 500	(-0.6223, 1.46E-36)	(-0.2357, 1.08E-08)	(-0.8053, 2.08E-92)
d = 20, n = 1500	(-0.6017, 4.33E-21)	(-0.1372, 5.26E-02)	(-0.2883, 3.81E-05)
d = 50, n = 1500	(-0.5857, 3.78E-13)	(-0.1305, 7.73E-04)	(-0.8236, 7.67E-31)

Table 7: Correlation between IDA score and Vertex Checker p-values: Linear Exponential

G.3. Computational Complexity

We empirically measured the time complexity of the Vertex Checker when the causal relations follow mlp (defined in F.1), with the results presented in Table 11. The results demonstrate that the Vertex Checker is highly efficient for conducting local tests on individual vertices.

G.4. Alternatives to Uniformity test for the Null Hypothesis

A potential concern regarding the validity of the Vertex Checker is the multiple testing problem, which raises the question of whether applying multiple test corrections could improve the results. To address this, we conducted simulations using several multiple testing correction methods, including Bonferroni Correction, Benjamini-Hochberg (BH) Correction, Benjamini-Yekutieli (BY) Correction, and Holm-Bonferroni Correction.

Beyond the issue of multiple comparisons, other potential concerns, as discussed in Section 6, include the dependency among p-values and the limited number of p-values available. While traditional multiple test corrections assume independence or a large number of tests, these assumptions may not always hold in practice. In such cases, approaches like Fisher's method in Shipley's C-Test offer an alternative model evaluation framework by aggregating individual d-separation tests, mitigating the limitations posed by p-value dependencies and finite sample sizes (Shipley, 2000; Tillman, 2009). However, we would also like to point out that relying on Fisher's method may introduce new challenges. For example, it can become unstable when p-values are close to zero, leading to numerical issues. The trade-off in choosing an appropriate test depends on the specific data structure in practice.

In this section, we compare the empirical performance of different testing strategies in evaluating the local structure. We present IDA score results for each method (C-test in the graph denotes Fisher's method) and compare them with the results from the Uniformity Test (which applies no correction). We assess the outcomes for vertices that pass and fail the Vertex Checker. As shown in Figure 7 and 8, the results before and after applying the corrections do not differ significantly, suggesting that multiple test corrections may have limited impact in this context.

	mlp	mim
PC	(0.1622, 5.45E-03)	(0.1550, 7.56E-03)
FGES	(0.2962, 4.46E-07)	(0.1723, 5.75E-03)
BOSS	(0.3182, 5.60E-04)	(0.3435, 2.22E-08)

Table 8: PASS: Correlation between F1 Score on parent sets and Vertex Checker p-values on *Test* set

	mlp	mim
PC	(0.3765, 7.36E-47)	(0.4079, 1.82E-07)
FGES	(0.3697, 1.39E-34)	(0.5562, 2.98E-50)
BOSS	(0.3848, 3.27E-21)	(0.7414, 9.43E-58)

Table 9: FAIL: Correlation between F1 Score on parent sets and Vertex Checker p-values on *Test* set

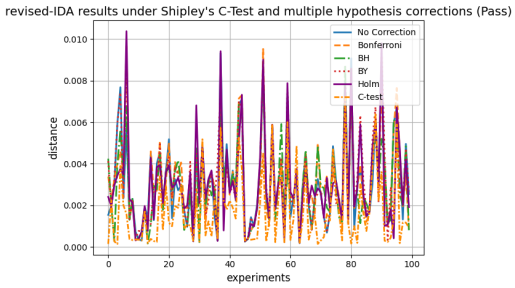

Figure 7: IDA score results for vertices that pass the Vertex Checker

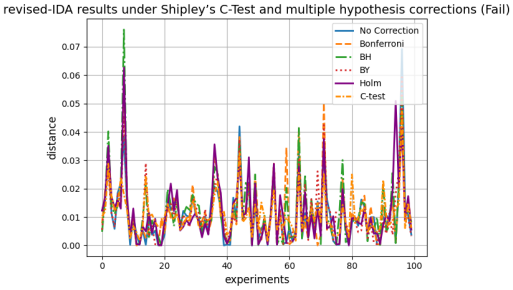

Figure 8: IDA score results for vertices that fail the Vertex Checker

Appendix H. Scatterplot on Sachs data

We attach the scatterplot on Sachs data (Sachs et al., 2005) here, which provides more information on the distributions and causal relations of the variables.

	PC	FGES	BOSS	GRASP
p-values	5.32E-11	1.07E-25	5.48E-18	9.53E-28

Table 10: P-values from Markov Checker on Sachs dataset

	(20,500)	(20,1500)	(50,500)	(50,1500)	(100,500)	(150,1500)
Seconds	0.0071	0.0106	0.0097	0.0227	0.0191	0.0318

Table 11: Average runtime in seconds for the Vertex Checker on each vertex

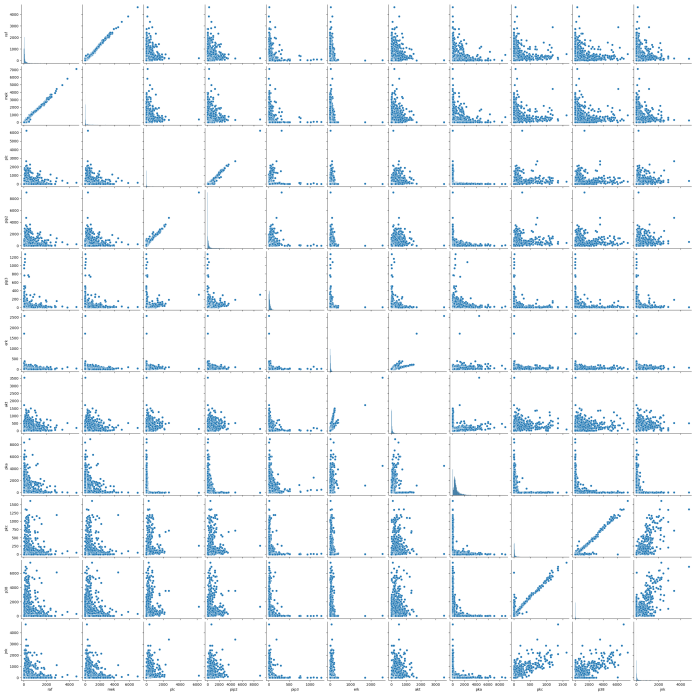

Figure 9: Scatterplot of Sachs