Comparative Study of Causal Discovery Methods for Cyclic Models with Hidden Confounders

Boris Lorbeer
Technische Universität Berlin
Berlin, Germany
lorbeer@tu-berlin.de

Mustafa Mohsen
Technische Universität Berlin
Berlin, Germany
m.mohsen@campus.tu-berlin.de

***Abstract*—Nowadays, the need for causal discovery is ubiquitous. A better understanding of not just the stochastic dependencies between parts of a system, but also the actual cause-effect relations, is essential for all parts of science. Thus, the need for reliable methods to detect causal directions is growing constantly. In the last 50 years, many causal discovery algorithms have emerged, but most of them are applicable only under the assumption that the systems have no feedback loops and that they are causally sufficient, i.e. that there are no unmeasured subsystems that can affect multiple measured variables. This is unfortunate since those restrictions can often not be presumed in practice. Feedback is an integral feature of many processes, and real-world systems are rarely completely isolated and fully measured. Fortunately, in recent years, several techniques, that can cope with cyclic, causally insufficient systems, have been developed. And with multiple methods available, a practical application of those algorithms now requires knowledge of the respective strengths and weaknesses. Here, we focus on the problem of causal discovery for sparse linear models which are allowed to have cycles and hidden confounders. We have prepared a comprehensive and thorough comparative study of four causal discovery techniques: two versions of the *LLC* method [10] and two variants of the *ASP*-based algorithm [11]. The evaluation investigates the performance of those techniques for various experiments with multiple interventional setups and different dataset sizes.**

I. Introduction

Causal analysis [15, 16] of complex systems is nowadays an integral part of many sciences. It is heavily used in fields as diverse as medicine, biology, cognitive science, economics, predictive maintenance, root cause analysis, physics, and machine learning. Conventional data analysis investigates the probabilistic properties of the data to gain insight into the involved probability distributions which can then be used to e.g. predict new data. Causality on the other hand serves to not just learn the data but to learn about the system that generates the data. For instance, consider two random variables, one is binary and indicates the administration of a drug that is allegedly reducing blood pressure in patients, and the other is the patient's blood pressure itself. Then, ordinary data analysis can study the existence and size of a stochastic dependency between those two random variables. But if we are interested in the efficiency of the drug in changing blood pressure, this purely stochastic information is insufficient, since there can be a correlation between the two random variables without causation, i.e. without the drug having any effect on blood pressure. Causal analysis on the other hand provides techniques to answer questions about actual causation. And in this example, it answers the question of whether the drug is actually the reason for the change in the patient's blood pressure and also measures the size of this causal effect. I.e. causal analysis is concerned with the discovery and measurement of actual *mechanisms* in the underlying system, which in this case would be the patient's body.

A subfield of causality is *causal discovery*, which studies the existence of causal relations and is not concerned with the estimation of the size of the causal effects. Often, causal discovery is the first step and its results serve as input for the estimation of the effect size. This paper focuses on causal discovery.

Most of the research in causal analysis focuses on the acyclic situation, meaning that the considered system is presumed to have no cycles in its causal relations, i.e. there are no feedback loops. This renders the analysis less complex but excludes many realistic scenarios. An example from econometrics would be the study of supply, price, and demand: the demand is influencing the price but the price is also influencing the demand.

Another assumption in standard causal analysis is *causal sufficiency*: It is presumed that there are no unobserved variables, so-called *hidden confounders*, that causally influence multiple observed variables. Again, this simplifies the analysis but excludes many relevant use cases. E.g., in the example above of a drug for blood pressure, imagine that the drug is only given to younger people, which have lower blood pressure anyways, thus causing bias to the results. In this case, age is a hidden confounder.

While there are many causal discovery algorithms for the acyclic, causally sufficient situation, very few exist that are also applicable in the more demanding case of cyclic systems with hidden confounders. In this paper, we compare the properties of four of such methods, namely two techniques, *ASP-d* [11] and *ASP-s* [4], which are variants of a constraint-based technique using answer set programming, and two variants, *LLC-NF* and *LLC-F* [10], of a method of moments type estimator.

Those four approaches are evaluated on synthetic data from linear systems, which means the causal relationships between

the variables are linear.

II. Related Work

For a complete overview of the history of causality see the treatises [15] and [16]. Both references focus mainly on the acyclic case but do cover, to a certain extent, the situation with hidden confounders.

An early technique for cyclic systems, called *CCD*, is described in [19], but it presumes the absence of latent confounders. Amongst the few algorithms that allow for both cycles and hidden confounders are *LLC* [10], the method described in [11] which we refer to as *ASP-d*, *sigmasep* [4] which we refer to as *ASP-s*, *BACKSHIFT* [20], *CCI* [23], and *bcause* [17, 18].

Note that the methods above presume interventional data, that is, not only data from the system itself but also from other systems that are obtained by changing the original system in a certain way, i.e. *intervening* on it. If this interventional data is not available, causal inference becomes harder. There are, however, several approaches for this scenario, too, like the family of *Additive Noise Models* (*ANM*), see e.g. [8], ICA-based methods for linear systems like *LiNGAM* [22], *LiNG* [12], and the *Two-Step* algorithm [21], or *Information Geometric Causal Inference* (*IGCI*), see e.g. [14]. Some of those methods can also deal with latent confounders or cyclicity.

In recent years, the mathematical foundations of the theory of cyclic systems with hidden confounders have advanced considerably. A comprehensive exposition can be found in [5] and [2].

III. Description of the Methods

This section will present the main features of the *LLC* and *ASP* algorithms. Note, that this is a high-level overview, presenting only as much as is necessary to explain the evaluation below. For the details, the reader should consult the pertinent papers, see [3, 9, 10] for the *LLC* and [11, 4] for the *ASP* variants.

A. General Concepts in Causality

The main entity in causality is the *Structural Causal Model* (SCM) [2], which consists of *structural equations* of the form (note that we use $1:n$ as an abbreviation for the set $\{1, 2, \ldots, n\}$):

$$X_i = f_i(\mathbf{X}, \mathbf{E}), \qquad i \in 1{:}n, \tag{1}$$

where $\mathbf{X} = (X_1, \ldots, X_n)$ is an n-dimensional random vector containing the observed variables, $\mathbf{E}$ is an m-dimensional random vector containing the unobserved noise, and the functions $f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are the *causal mechanisms*. See [2] for the mathematical details. Many aspects of an SCM can be described by associating a graph that contains as nodes both the observed variables $X_i, i \in 1:n$ and the unobserved variables $E_k, k \in 1:m$. It is a *directed graph* (DG), i.e. all edges have exactly one arrowhead. Here, a directed edge only goes into observed nodes X_i, no edges are pointing to noise variables E_k, and there is an edge $N \to X_i$ from a node N to X_i iff f_i depends on N. This graph is called the *augmented graph of the SCM*, see [2].

This directed graph can be reduced to a graph that contains as nodes only the observed variables and connects any two nodes that have a noise node as common parent by a bidirected edge. Graphs with both directed and bidirected edges are called *directed mixed graphs* (DMG). This reduced graph is simply called the *graph of the SCM*.

A *path* (in a DG or DMG) is a tuple $(\epsilon_1, \ldots, \epsilon_m)$ of edges where two consecutive edges have a common node. In particular, an edge can appear multiple times in a path. A path is a *directed path* if none of the edges are bidirected and all edges point in the same direction.

The directed edges in the graph of an SCM then depict the direct causal connections between variables, symbolizing *direct causal effects*, while bidirected edges describe confounding (see below). Directed paths consisting of more than one edge describe indirect causal connections, generating *indirect causal effects*. Note, however, that in cyclic SCMs we can have causal effects even between nodes that are not connected by a directed graph, see [2] section 7.1.

If this graph has *cycles*, i.e. directed paths that start and end at the same node and contain at least one edge, the SCM is called *cyclic*. Causal cycles describe systems with feedback loops, which are quite common in realistic scenarios.

If an unobserved node E_k influences two different observed nodes X_i, X_j, i.e. the augmented graph of the SCM contains the subgraph $X_i \leftarrow E_k \rightarrow X_j$ and the graph of the SCM contains a bidirected edge $X_i \leftrightarrow X_j$, then E_k is called a *hidden* (or *latent*) *confounder*. A system without hidden confounders is called *causally sufficient*.

A central notion in causality is that of an *intervention*. We consider only *surgical interventions* [10], which can be described as follows: The original SCM is changed by selecting a subset $\mathbf{X}_I, I \subset 1:n$, of the observed variables, and forcing a new distribution on $\mathbf{X}_I$, that is independent of all the other random variables $\mathbf{X}_{1:n\setminus I}$ and $\mathbf{E}$. An example would be randomized drug trials, which ensure that the people who do and do not get the drug are selected completely at random. In this case, we have an intervention on the assignment of the drug. The graph of the intervened SCM differs from the graph of the original SCM in that all the edges that point into intervened nodes are removed.

Data that is observed from a non-intervened system is called *(purely) observational data*.

A common approach to causal discovery consists in using *constraint-based* methods, which exploit conditional (in)dependences between random variables to infer causal connections. Two random variables X and Y are said to be independent conditioned on a set of random variables S with $X, Y \notin S$, denoted by $(X \perp\!\!\!\perp Y|S)$, if they are independent w.r.t. their conditional probabilities, i.e.:

$$(X \perp\!\!\!\perp Y|S) \quad \Leftrightarrow \quad p(X, Y|S) = p(X|S)p(Y|S), \tag{2}$$

presuming those conditional probabilities exist. The notion of "conditional independence" has a pendant in graphs which is

called *d-separation*. To properly explain it, we first need to define colliders: Given a path in a DMG, a node X_i on the path is a *collider on this path* if both neighboring edges on the path point into X_i, i.e. $X_{i-1} \to X_i \leftarrow X_{i+1}$, $X_{i-1} \leftrightarrow X_i \leftarrow X_{i+1}$, $X_{i-1} \to X_i \leftrightarrow X_{i+1}$, or $X_{i-1} \leftrightarrow X_i \leftrightarrow X_{i+1}$. Then two nodes X_i and X_j are said to be *d-connected* w.r.t. a conditioning set C if there is a path from X_i to X_j such that all the colliders on the path are in C. If two nodes are not d-connected they are called *d-separated*. The notation for X being d-separated from Y given C in the graph G is $(X \perp_G Y|C)$.

The idea of constraint-based methods is based on two assumptions (see [15, 16] for details):

1) The probability distribution of an SCM is *Markovian* w.r.t. the graph G of the SCM, i.e.:
$$(X \perp_G Y|C) \qquad \Rightarrow \qquad (X \perp\!\!\!\perp Y|C). \tag{3}$$
For linear, possibly cyclic SCMs, which is the case we are considering, this assumption holds under mild conditions. See Theorem A.7 in [5] for a precise statement of those conditions.
2) The probability distribution of an SCM is *faithful* w.r.t. the causal graph G of the SCM, i.e.:
$$(X \perp\!\!\!\perp Y|C) \qquad \Rightarrow \qquad (X \perp_G Y|C). \tag{4}$$
There are situations when this implication does not hold. E.g., imagine for an edge $X \to Y$ a second path $X \to Z \to Y$ which creates an effect from X on Y that is exactly the opposite of that of the edge $X \to Y$. Thus, the two effects cancel out, and, even though the system has a proper mechanism that links X and Y, it is invisible in the data.

Now, constraint-based methods usually presume those two assumptions and use them to obtain information about the causal graph structure from stochastic (in)dependence properties of the observational and interventional data. The advantage of the constraint-based approach is that it is nonparametric, i.e. we don't have to presume a specific model for the SCM. Note, however, that it only provides the structure of the graph of the SCM. For the quantitative estimation of the causal effects, one has to use further techniques on top of the constraint-based methods.

Note, that with constraint-based methods one tries to obtain the non-symmetric property "X causes Y" from symmetric stochastic (in)dependence properties which is usually not possible. Thus, one has to apply interventions: If X causes Y, i.e. $X \to Y$, then both Y is stochastically dependent on X and X is stochastically dependent on Y. But with intervention on X, the dependence still holds, while intervening on Y removes the dependence, thus breaking the symmetry.

B. The LLC algorithm

This section gives a high-level overview of the *LLC* algorithm; for more details, see [10].

LLC is a method of estimating the parameters of a *linear* causal system of the form:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e} \tag{5}$$

where $\mathbf{x}, \mathbf{e} \in \mathbb{R}^n, \mathbf{B} \in \mathbb{R}^{n \times n}$. Here, $\mathbf{x}$ contains the measurements of the observed variables $\mathbf{X}$ and $\mathbf{e}$ the hidden values of the unobserved variables $\mathbf{E}$. For $\mathbf{E}$, we only presume the expectation to be zero, $\mathbb{E}[\mathbf{E}] = 0$, and the covariance of $\mathbf{E}$ is abbreviated with $\mathbf{\Sigma_e}$. Note, that we do not require any particular distribution for $\mathbf{E}$, the only restriction is $\mathbb{E}[\mathbf{E}] = 0$, and even that can be lifted, see [10]. However, since this change has no bearing on our evaluation, we stick to the simple case $\mathbb{E}[\mathbf{E}] = 0$. The (i, j)-coefficient of the matrix $\mathbf{B}$ is denoted by b_{ij} and can be identified as the direct causal effect of variable X_j on variable X_i. $\mathbf{B}$ is allowed to describe cyclic paths in the causal graph. The covariance $\mathbf{\Sigma_e}$ is allowed to have off-diagonal elements, which can be interpreted as confounding. $\mathbf{B}$ is required to have a zero diagonal:

$$b_{ii} = 0, \qquad i = 1, \ldots, n, \tag{6}$$

which translates to the system not having self-loops.

The measurement of an intervened system, possibly with an empty intervention, is called an *experiment*. Experiments are denoted by $\mathcal{E}_k := (\mathcal{J}_k, \mathcal{U}_k), k \in 1\!:\!K$, where K is the number of considered experiments and each $(\mathcal{J}_k, \mathcal{U}_k)$ is a partition of $1 : n$, i.e. $\mathcal{J}_k \cap \mathcal{U}_k = \emptyset$ and $\mathcal{J}_k \cup \mathcal{U}_k = 1 : n$. Here, $\mathcal{J}_k$ is the set of indices of the nodes which are intervened on and $\mathcal{U}_k$ of those that are not. To each experiment $\mathcal{E}_k = (\mathcal{J}_k, \mathcal{U}_k)$ a pair of diagonal matrices $(\mathbf{J}_k, \mathbf{U}_k)$ is assigned: using the indicator function $I(\cdot)$, which equals one if its argument is true and zero otherwise, the diagonals of $\mathbf{J}_k$ and $\mathbf{U}_k$ are given by $\mathbf{J}_{k,ii} = I(i \in \mathcal{J}_k)$ and $\mathbf{U}_{k,ii} = I(i \in \mathcal{U}_k)$, resp. This allows us to write the structural equations for the experiment $\mathcal{E}_k$ in a very compact form. Let $\mathbf{c}$ be the vector containing at the indices $\mathcal{J}_k$ the intervention values (the values the pertaining variables are forced to), and zeros elsewhere. Then the structural equations for the experiment $\mathcal{E}_k$ are given by:

$$\mathbf{x} = \mathbf{U}_k\mathbf{B}\mathbf{x} + \mathbf{U}_k\mathbf{e} + \mathbf{c}. \tag{7}$$

For *LLC* to work, it is required that the SCM be *weakly stable*, i.e. for every experiment $\mathcal{E}_k = (\mathcal{J}_k, \mathcal{U}_k)$, the matrix $\mathbf{I} - \mathbf{U}_k\mathbf{B}$ must be invertible.

Next, we consider the covariance matrix $\mathbf{C}^k_\mathbf{x}$ of the measurements $\mathbf{X}$ in experiment $\mathcal{E}_k$. A covariance c_{ui} in $\mathbf{C}^k_\mathbf{x}$ is called the *total causal effect* of x_i on x_u with intervention set $\mathcal{J}_k$, and is abbreviated as $t(x_i \rightsquigarrow x_u||\mathcal{J}_k)$. A central identity of *LLC* provides the connection between those total causal effects and the matrix $\mathbf{B}$, i.e. the direct causal effects. This identity reads:

$$t(x_i \rightsquigarrow x_u||\mathcal{J}_k) = b_{ui} + \sum_{j \in \mathcal{U}_k \setminus \{u\}} t(x_i \rightsquigarrow x_j||\mathcal{J}_k) b_{uj}, \tag{8}$$

which is a linear equation in the b_{ij}. Gathering all those equations for all conducted experiments results in a system of linear equations:

$$\mathbf{t} = \mathbf{T}\mathbf{b}, \tag{9}$$

where $\mathbf{t}$ contains all the total effects on the left-hand side of (8), $\mathbf{b}$ is the concatenation of all the rows of $\mathbf{B}$ without the diagonal, i.e. $\mathbf{b} \in \mathbb{R}^{n^2-n}$, and $\mathbf{T}$ contains the total effects

from the right-hand side of (8). In [10] it is shown, that for $\mathbf{T}$ in (9) to have zero null space, the experiments $\{\mathcal{E}_k\}_{k=1}^K$ need to satisfy the *pair condition*, which requires that for each ordered pair of indices $(i,j), i,j \in 1\!:\!n$, there is an experiment $\mathcal{E} = (\mathcal{J},\mathcal{U})$ such that $i \in \mathcal{J}, j \in \mathcal{U}$.

Thus, the method of *LLC* becomes clear: First, the covariances $\mathbf{C}_\mathbf{x}^k$ are estimated from the data, those total effects are used to build the linear system (9), and finally, this linear system is solved for $\mathbf{b}$.

But *LLC* also provides an estimate for the covariance matrix $\mathbf{\Sigma_e}$, which describes the confounding in the system. If purely observational (i.e. non-intervened) data is available, equation (5) applies and since we obtained an estimate for $\mathbf{B}$ from (9), we can simply compute $\mathbf{\Sigma_e}$ as:

$$\mathbf{\Sigma_e} = (\mathbf{I}-\mathbf{B})\mathbf{C}_\mathbf{x}^0(\mathbf{I}-\mathbf{B})^T, \tag{10}$$

where $\mathbf{C}_\mathbf{x}^0$ is the covariance matrix of $\mathbf{x}$ for no intervention. If there is no purely observational data, we can still obtain $\mathbf{\Sigma_e}$ from the experiments: from (7) it follows for any experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ that:

$$(\mathbf{\Sigma_e})_{\mathcal{U}_k,\mathcal{U}_k} = \left[(\mathbf{I}-\mathbf{U}_k\mathbf{B})\mathbf{C}_\mathbf{x}^k(\mathbf{I}-\mathbf{U}_k\mathbf{B})^T\right]_{\mathcal{U}_k,\mathcal{U}_k}. \tag{11}$$

Depending on the experiments, for a given pair (i,j) there can be multiple experiments $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ with $i,j \in \mathcal{U}_k$, so it makes sense to compute the average of the covariances over all such experiments, i.e.:

$$\mathbf{\Sigma}_{\mathbf{e},ij} = \operatorname{avg}\left\{\left[(\mathbf{I}-\mathbf{U}_k\mathbf{B})\mathbf{C}_\mathbf{x}^k(\mathbf{I}-\mathbf{U}_k\mathbf{B})^T\right]_{i,j} | i,j \in \mathcal{U}_k\right\}. \tag{12}$$

Thus, to obtain the complete covariance $\mathbf{\Sigma_e}$, we need for each pair (i,j) at least one experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ such that $i,j \in \mathcal{U}_k$, which is called the *covariance condition*.

Finally, if all conditions above are satisfied, *LLC* returns the pair $(\mathbf{B},\mathbf{\Sigma_e})$. Note, that in causal discovery, we are only interested in the causal graph, i.e. we only need to know which b_{ij} and $\mathbf{\Sigma}_{\mathbf{e},ij}$ are zero.

C. *The* LLC-F *algorithm*

Note, that *LLC* is not a constraint-based method, and does not presume faithfulness. However, the question is whether combining *LLC* with constraint-based methods could improve the accuracy of *LLC*. This has been investigated in [9] and the pertinent algorithm is called *LLC-F,* with the additional letter "F" indicating that now, faithfulness is presumed. We use the abbreviation *LLC-NF* to refer to the *LLC* variant that does not use constraint-based methods and does not presume faithfulness.

The idea is to add to the linear system (9) more linear equations obtained from conditional independences in the purely observed and intervention data. The following four methods are applied:

1) If, for some experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$, for two variables X_i, X_j with $i,j \in \mathcal{U}_k$ there exists a set S of variables with $X_i, X_j \notin S$ such that $(X_i \perp\!\!\!\perp X_j | S)$, then we have, by faithfulness, $b_{ij} = b_{ji} = \mathbf{\Sigma}_{\mathbf{e},ij} = 0$.
2) If, for some experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ and two variables $X_i, i \in \mathcal{J}_k$ and $X_u, u \in \mathcal{U}_k$, there exists a set S of variables with $X_i, X_u \notin S$ such that $(X_i \perp\!\!\!\perp X_u | S)$, then we have, by faithfulness, $b_{ui} = 0$.
3) If, for some experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ and three indices $i \in \mathcal{J}_k$ and $u,v \in \mathcal{U}_k$, we have $t(x_i \rightsquigarrow x_u || \mathcal{J}_k) \neq 0$ and $t(x_i \rightsquigarrow x_v || \mathcal{J}_k) = 0$, then it follows that $b_{vu} = 0$ by faithfulness.
4) If, for some experiment $\mathcal{E}_k = (\mathcal{J}_k,\mathcal{U}_k)$ and three indices $i \in \mathcal{J}_k$ and $u,v \in \mathcal{U}_k$, we have $t(x_i \rightsquigarrow x_u || \mathcal{J}_k) \neq 0$ and $(x_i \perp\!\!\!\perp x_v | \{x_u\})$ in $\mathcal{E}_k$, then it follows that $b_{uv} = 0$ and $\mathbf{\Sigma}_{\mathbf{e},uv} = 0$.

Apart from extending the linear system (9) with those equations, the algorithm does not differ from *LLC-NF*.

D. ASP-d

Next, we give an overview of the *ASP-d* algorithm, following [11]. *ASP* is an abbreviation of "Answer Set Programming", which is a declarative programming language, see [7].

The *ASP-d* algorithm belongs to the class of constraint-based methods and thus infers the causal graph from conditional independences as has been described above. It differs from most other constraint-based techniques in that it allows for cyclic causal graphs with hidden confounders and that it can deal with data from multiple experiments with different interventions. The conditional independences are obtained from independence hypothesis tests which have a certain probability of error. That means they can contradict each other. The innovation of *ASP-d* is to handle this issue by formulating this causal discovery problem as an optimization problem: let $\mathbf{K}$ be a set of conditional (in)dependence relations that are obtained from a given dataset, and let $w(k) \in \mathbb{R}_{\geq 0}$ be a nonnegative weight assigned to each $k \in \mathbf{K}$, describing how confident we are that k is indeed true. Then the task is to find a graph G^* which minimizes the following loss function:

$$L(G) := \sum_{k \in \mathbf{K}, G \not\models k} w(k), \tag{13}$$

where $G \not\models k$ means that the graph does not entail, via d-separation, the (in)dependence k. Since the loss function is using d-separation, *ASP-d* is applicable to *acyclic* SCMs that are linear or nonlinear, and to *cyclic* SCMs that are linear; see the description of *ASP-s* below for more details.

Several possibilities for how to determine the weights $w(k)$ have been proposed, see e.g. [11, 4, 18]. We will presume that the constraints $k \in \mathbf{K}$ have been obtained using conditional independence hypothesis tests with significance level α and then use weights as in [4]:

$$w(k) = |\log p_k - \log \alpha|, \tag{14}$$

where p_k is the p-value of the hypothesis test of k.

The optimization of (13) is done by encoding the problem as an *ASP* program which can then be solved by some *ASP* solver like e.g. `clingo` [6].

E. ASP-s

In constraint-based methods, the (in)dependences obtained from the measurements must be matched with the graph of the SCM to discover its causal structure. This matching is done, as explained above, via d-separation. However, it should be noted that for *nonlinear* cyclic SCMs, d-separation in general fails to be Markovian, see [5].

I.e., for nonlinear cyclic models, d-separation must be replaced with another separation property. This new separation property *σ-separation* has been introduced in [5]. The basic idea here is roughly that nodes in loops are so strongly connected that conditioning cannot separate them, so they behave as if they would be fully connected. That means, if loops are replaced by fully connected subgraphs, an operation which is called an *acyclification*, see [5, 2] for details, the resulting acyclic graph exposes d-separation properties that again correspond to the conditional (in)dependences of the original cyclic SCM. In [5] the authors then formulated a separation property for cyclic graphs, *σ-separation*, which is the "pull back" of d-separation via the acyclification operation.

Then, the authors of [4] took the *ASP-d* algorithm and changed the *ASP* encoding slightly to now use σ-separation instead of d-separation. In [18] this algorithm got further improved to the *ASP-s* algorithm we use in this paper.

Thus, *ASP-s* differs from *ASP-d* only in the type of separation that is used. In particular, *ASP-s,* too, is a constraint-based, nonparametric, optimization algorithm that minimizes (13), except that for *ASP-s* the notation $G \not\models k$ under the sum now refers to σ-separation.

It is important to note, however, that *ASP-d* and *ASP-s* have different application fields. While *ASP-d* can be used for acyclic and *linear* cyclic SCMs with hidden confounders, see above, *ASP-s* applies to acyclic and *nonlinear* cyclic SCMs with hidden confounders. In particular, linear cyclic SCMs are not faithful w.r.t. σ-separation, see [5, 4]. Thus, those two application fields have only the acyclic SCMs in common. And since this paper only examines linear SCMs, *ASP-s* is not strictly applicable here. However, we included *ASP-s* in the set of algorithms to compare. Since linear models are "of measure zero" inside the set of all SCMs, one might be tempted to always use σ-separation. This could be problematic, because nonlinear SCMs that are nearly linear could provide data that is only bearly distinguishable from that of linear SCMs. Thus, it would be instructive to see how much worse *ASP-s* performs compared to *ASP-d* on linear cyclic models.

IV. Evaluation

Here we describe our evaluation of the four methods *LLC-NF*, *LLC-F*, *ASP-d*, and *ASP-s*. The source code of our implementation in *Python* and *R* is available online[1], and uses publicly available code for *LLC* from Antti Hyttinen's homepage[2] and for *ASP* from the GitHub repository of [4][3].

[1]The code will be made available upon publication.

[2]https://www.cs.helsinki.fi/u/ajhyttin/

[3]https://github.com/caus-am/sigmasep

For various types of data, we measure the methods' capabilities of detecting features of the underlying SCM. Those features are edges and bidirected edges (confounders), which can be either absent or present. The evaluation thus consists of measuring the performance of binary predictions for each single feature. Recall that there are requirements on the experimental setups for the SCMs to be theoretically identifiable. Below, we consider the performances of the methods for both cases, where the experimental setups do and do not satisfy those requirements.

As described in Section III-B, the *LLC* variants are based on solving the system (9). For real data this system must be expected to contain contradicting equations. This could be handled by solving it as a least squares problem. However, as stated above, the system could also have a non-zero null space if there are not sufficiently diverse experiments to satisfy the pair condition. This would result in the indeterminacy of some or all of the coefficients in $\mathbf{b}$. To avoid this, we chose to use the version of *LLC* that applies a penalty term (L_1 or L_2) to (9), see the code base of [10], which also has the useful effect of regularization and promoting sparsity.

The covariance matrix is fully determined if the covariance condition is satisfied, see Section III-B. This is ensured by adding the "null-experiment", i.e. the experiment without any intervention, to each of our experimental setups.

The *ASP*-based methods, being constraint-based methods, might not be able to determine the presence of some features if there are not sufficiently many interventions available. One possible approach in this scenario is to fix the presence of undetermined features according to domain knowledge. For example, if it is known that the system under consideration corresponds to a sparse causal graph, a straightforward practice is declaring those undetermined features as being absent. This can be considered as an ensemble of two methods, the *ASP* algorithm and a weak classifier which classifies everything as absent. The ensemble consists in applying first *ASP* and returning its result unless it is undetermined in which case the weak classifier is applied. Most real-world causal graphs are sparse, so it is reasonable to confine our evaluations to data from sparse causal graphs and to always use the above ensemble. Thus, below, whenever we refer to *ASP-s* or *ASP-d,* we actually refer to this ensemble.

Since there are almost no real-world datasets with known causal ground truth available, let alone in sufficient quantities and satisfying the particular constraints required by our evaluation, we restrict our study to synthetic data. Because of the high computational complexity of the methods, in particular of the *ASP* variants, we simulate only graphs with five nodes and two confounders. The edges are randomly distributed with the constraint that the in-degree of any node is not larger than two. As a result, the average number of edges in our simulated graphs is 6.1 and that of bidirected edges is two, i.e. the simulated graphs are sparse.

The coefficients of the linear equations in the SCM are sampled from the uniform distribution over the set $[-1.1, -0.1] \cup [-0.1, 1.1]$. The effect sizes are thus bounded away from zero,

assuring detectability. Furthermore, It is ascertained that each simulated SCM has at least one cycle (which is not a self-loop). The study in this paper is based on data generated by 150 such randomly sampled SCMs.

For the evaluation of the methods, we need a metric. Both the *LLC* and the *ASP* algorithms return for each feature a score determining how strongly the algorithm "believes" this feature to be present. The code from [10] also provides a bootstrapped version of *LLC* which means we obtain for each coefficient b_{ij} and $\mathbf{\Sigma_e}$ a collection of estimates of which we can compute the z-score. This is the score we use for both *LLC* algorithms. For the *ASP* variants, we utilize a score proposed in [13]: the *ASP* algorithm is run twice for every single feature, once with the additional constraint that the feature is present and once with the additional constraint that it is absent. The score is then the difference between the loss under the constraint of absence and the loss under the constraint of presence. We call this the *ASP confidence score* and use this as our scoring function for both ASP variants.

Building on those two score definitions, we can now define the two metrics that we will base our evaluation on. The first is the *area under the ROC curve*, AUC ROC, given by those scores, and the second is the *accuracy* of the binary classification obtained by defining a threshold for those scores: if the score of a feature is below this threshold, the feature is considered absent, otherwise, present. While we compute the accuracy for each SCM separately, we compute the AUC ROC over the combined data of all the 150 SCMs.

The full evaluation procedure works as follows: We randomly choose 150 SCMs as described above and use them to create datasets of observations that are then used as input for the four models. The edges and bidirected edges estimated by the models are then compared with those of the original SCMs. Those datasets are created in different ways, varying in the size of the dataset and the structure of the applied interventions.

More precisely, we consider 21 different experimental setups that are applied to the random SCMs. Those 21 setups consist of five groups: The first row of Table I shows the first group, consisting of a single setup containing only the purely observational experiment, i.e. no interventions are applied. The second row contains setups with an intervention size equal to one. For instance, the setup with ID 11 contains two datasets from each random SCM, the purely observational dataset (denoted by "[]") and the dataset obtained from intervening on the first node (denoted by "[1]"). As another example, the setup with ID 15 consists of six datasets per random SCM, the purely observed one [] and the five datasets obtained from single node interventions on all the available nodes: intervention only on node 1 denoted by [1], intervention only on node 2 denoted by [2], and similarly for the other nodes. In this setup 15 the input for the four causal discovery algorithms consists of the union of those six datasets. The setups of size two have a similar structure, except that the intervention sets now have size two. For example, the setup with ID 23 creates as input to the four algorithms the union of four datasets, which consists of the measurements of the purely observational

TABLE I: The experimental structure of the evaluation

Int. Size	Setup ID	Intervention Sets
0	0	[]
1	11	[], [0]
	12	[], [0], [1]
	13	[], [0], [1], [2]
	14	[], [0], [1], [2], [3]
	15	[], [0], [1], [2], [3], [4]
2	21	[], [0,1]
	22	[], [0,1], [1,2]
	23	[], [0,1], [1,2], [2,3]
	24	[], [0,1], [1,2], [2,3], [3,4]
	25	[], [0,1], [1,2], [2,3], [3,4], [4,0]
3	31	[], [0,1,2]
	32	[], [0,1,2], [1,2,3]
	33	[], [0,1,2], [1,2,3], [2,3,4]
	34	[], [0,1,2], [1,2,3], [2,3,4], [3,4,0]
	35	[], [0,1,2], [1,2,3], [2,3,4], [3,4,0], [4,0,1]
4	41	[], [0,1,2,3]
	42	[], [0,1,2,3], [1,2,3,4]
	43	[], [0,1,2,3], [1,2,3,4], [2,3,4,0]
	44	[], [0,1,2,3], [1,2,3,4], [2,3,4,0], [3,4,0,1]
	45	[], [0,1,2,3], [1,2,3,4], [2,3,4,0], [3,4,0,1], [4,0,1,2]

experiment and those of three (overlapping) interventions each of size two. And finally, setups in the last row of Table I contain the experiments of size four. Thus, each experimental setup can be described by the combination of the size of each applied intervention and the number of such interventions used. This enters the setup ID, where the first digit is the intervention size and the second digit is the intervention count.

Note, that we use the same size n of the dataset per intervention setting. Thus, since we use in each experimental setup the *union* of the datasets of each intervention, we get different data sizes for different setups. If the size n of a dataset is e.g. 1000, the setup with ID 0 will create a dataset of size 1000, while 15 will create one of size 6000. In summary, we construct 16 datasets from 150 randomly generated SCMs each, i.e. there are in total 2400 datasets for a given n. Each of our four models in the study will be evaluated on those datasets.

We also consider different sizes n of datasets. We investigate sizes 1000, 10,000, 100,000, and infinite. Here, the word "infinite" does not refer to actually infinite amounts of data, but rather a version of the experiments that correspond to its asymptotic behavior when $n \to \infty$. More precisely, this means the following (see also [10]): When creating random SCMs, the matrices $\mathbf{B}$ and $\mathbf{\Sigma_e}$ are sampled as described above. Then, we can obtain the covariance matrix $\mathbf{C_x}$ of the observations as follows, similar to (10):

$$\mathbf{C}^0_{\mathbf{x}} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Sigma_e}(\mathbf{I} - \mathbf{B})^{-T}. \tag{15}$$

To obtain datasets with finite size n, we then sample from a Gaussian distribution with mean $\mathbf{0}$ and covariance $\mathbf{C}^0_{\mathbf{x}}$, similarly for non-zero interventions. *LLC* then estimates the covariance matrix from the data and uses the estimated covariances as total causal effects. But if we instead use the covariances in (15) as the total causal effects directly, without the detour of sampling and covariance estimation, we obtain

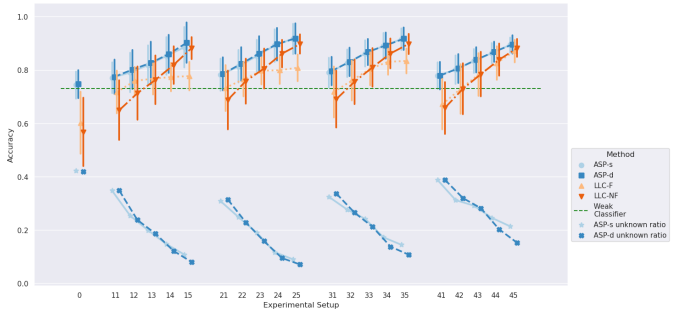

Fig. 1: Accuracies sample size 1000

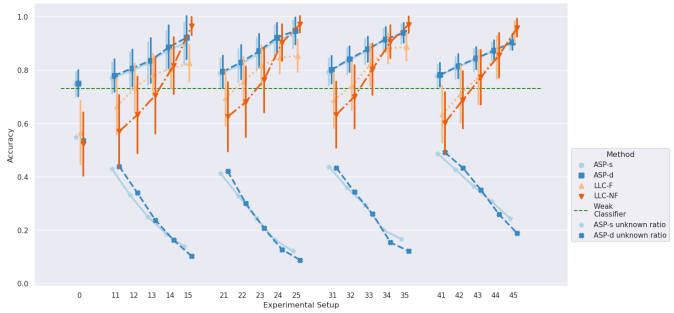

Fig. 2: Accuracies sample size 10,000

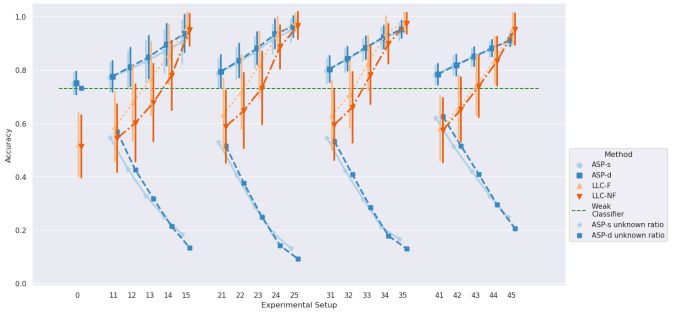

Fig. 3: Accuracies infinite sample size

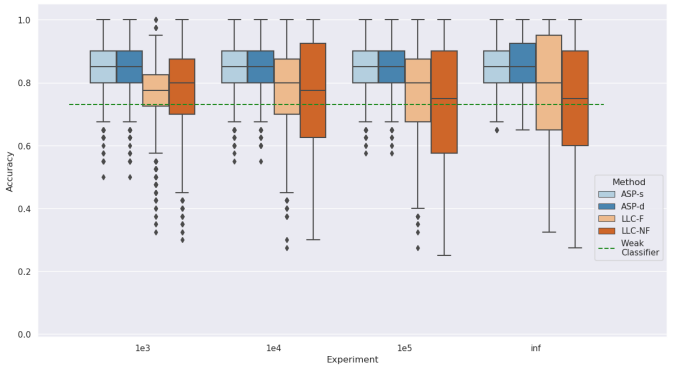

Fig. 4: Average accuracies by dataset size

the total effects that we would get for an infinite amount of data. That is meant by "infinite" data size for the *LLC* models. For the *ASP* methods the approach is different: These models use the data to obtain conditional independences. This is done by classical hypothesis tests, see Section III-D. The weights in (13) can become infinite for independences but stay finite for dependences for $n \to \infty$. To avoid those asymmetries, for the infinite scenario we skip the independence tests and collect a complete system of correct conditional dependence and independence relations from the ground truth (we know the SCM that generated the data) into the set $\mathbf{K}$, which is thus free of contradictions, and assign to each $k \in \mathbf{K}$ the weight one, $w(k) = 1$. So this construction does not exactly behave like *ASP* for $n \to \infty$, but it corresponds to it in so far as the set $\mathbf{K}$ is free of contradictions, which is why we will use the word "infinite" for brevity.

The methods we study have certain parameters that need to be fixed. For *LLC-NF* it is the penalty type, L_1 or L_2, and the belonging regularization parameter λ. For *LLC-F* we also have to fix the significance level α_{LLC} of the conditional independence tests. Furthermore, when using accuracy as metric, we need to choose a threshold t_{LLC} for the scores. For *ASP* we have the significance level α_{ASP} for the conditional independence tests and, in the case of using accuracy as a metric, again a threshold t_{ASP} for the score. As with all unsupervised learning tasks, the proper choice of those hyperparameters is tricky in practical applications. Here, we don't consider the problem of finding optimal hyperparameters but are rather interested in the performance of the studied techniques if the hyperparameters are chosen roughly appropriately, however, this is achieved. Since in our simulations we have access to the ground truth, we can optimize the hyperparameters. For this, we conducted rather comprehensive hyperparameter optimization studies for the type of SCMs that get generated by our adopted sampling scheme which produces sparse, small, cyclic models with hidden confounders as described above. This was done using the optimization tool *Optuna* [1]. As usual, we used one set of models for hyperparameter optimization and a different one for evaluation. Based on these optimization studies we chose the following hyperparameters for the *LLC* methods: penalty type L_1, $\lambda = 0.05$, $\alpha_{LLC} = 0.05$, and $t_{LLC} = 5$. For *ASP* we used $\alpha_{ASP} = 0.05$ and $t_{ASP} = 0$.

With those hyperparameters fixed, we ran the evaluations on the randomly sampled SCMs with different interventional setups and multiple dataset sizes as described above. Thus, for each combination of experimental setup, dataset size, metric, and causal discovery method, we get 150 values for the pertaining metric. To examine those results, we first collect the studies with accuracy as metric and a dataset size of $n = 1000$ in Figure 1. Here, for each experimental setup, we plot the accuracy mean together with error bars with length equal to the standard deviation. The plots are grouped according to the groups of the experimental setups and the four causal discovery methods are coded by color as given in the legend. The horizontal green line is the average accuracy of the weak classifier used in the *ASP*-ensembles, which just classifies every edge as absent.

Several things can be noticed: The worst performance is

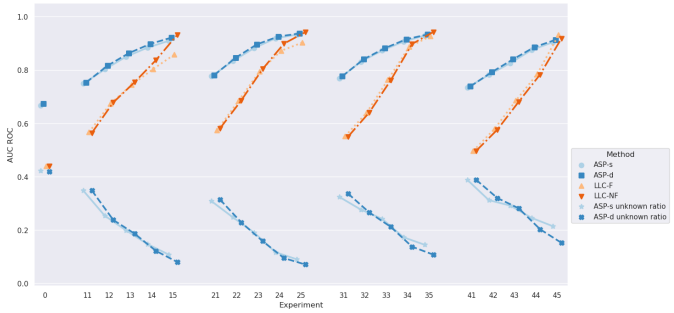

Fig. 5: AUC ROC sample size 1000

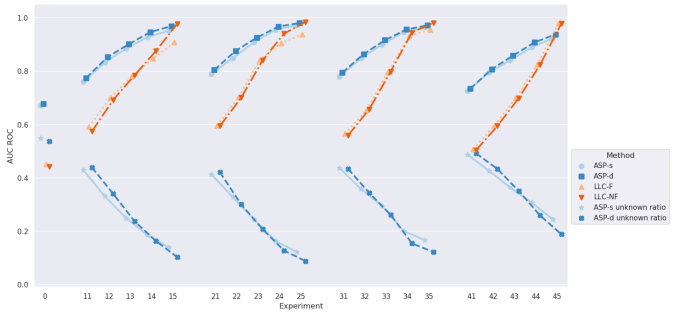

Fig. 6: AUC ROC sample size 10,000

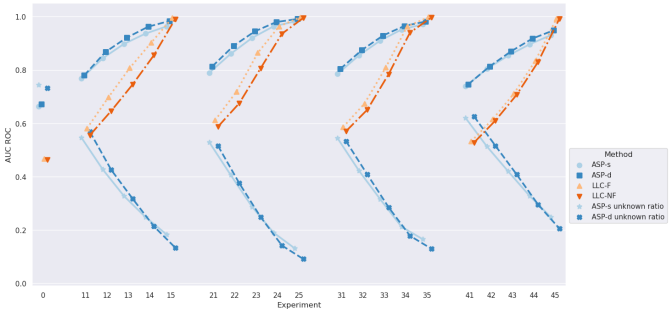

Fig. 7: AUC ROC infinite sample size

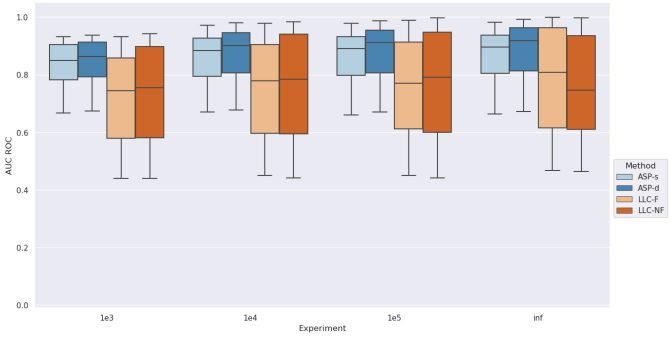

Fig. 8: Average AUC ROC by dataset size

measured for the purely observational experiments, showing the relevance of interventions. Next, it is clearly visible that combining the data of several different interventions is beneficial in all cases, as the plots are increasing in each group. Also, *LLC* requires a considerable amount of intervention data to beat the weak classifier. Furthermore, there seem to be no large deviations between the different sizes of the intervenion sets, since the differences between the four large groups are small compared to the size of the error bars. There is a slight decline in the last groups, though, which can be explained by the fact that intervening on all except one node removes all confounding effects, thus confounders cannot be detected.

Maybe most importantly, the performance of the *ASP* methods (recall that we use here the ensemble with the weak, sparsity-presuming classifier) seems better than that of the *LLC* methods, but for the setups with five experiments, this difference is well within the error margin. Note, that the five-experiment setups all provide sufficient interventions to satisfy the pair condition. It is interesting that there is almost no difference between *ASP-d* and *ASP-s*, even though we made sure that there is at least one cycle in each SCM. This encourages the universal use of *ASP-s* even if it is not clear whether the underlying SCM is linear. The difference between *LLC-F* and *LLC-NF* seems to switch signs within each group. This effect is quite consistent, but its origin is not entirely clear to us, so we leave this to future research.

In Figure 2 for dataset size $n = 10,000$, we see roughly the same behavior, except that, compared to $n = 1000$, *LLC* seems to be worse for smaller experiment counts and better for larger ones, beating *ASP* for setups that satisfy the pair condition. The larger dataset size seems to have no effect on *ASP* performance.

There is not much difference when changing to $n = \infty$, except that the switch between *LLC-F* and *LLC-NF* within a group is not visible anymore, and *LLC-F* improves considerably compared to *LLC-NF*. This suggests that this switching effect for finite n is due to incorrect conditional independence tests for finite data. As the last plot for the accuracy metric, we have averaged the performance over all experimental setups and compared it as a function of the dataset size in Figure 4. Here, we see that, overall, the *ASP* methods have a larger median accuracy and smaller interquartile range.

In figures 5 to 8 we visualize the performance with respect to the total AUC ROC metric. Since it is the AUC ROC of the combination of all the 150 SCMs, we have in each situation only a single value which is why there is no error bar as with the accuracy metric. The conclusion from those plots is similar to that for the accuracy metric. There is not much difference between *ASP-d* and *ASP-s*, despite the presence of cycles. Also, again *ASP* is better than *LLC* for fewer experiments but when the pair condition is satisfied, *LLC* can beat *ASP*. The switching between *LLC-F* and *LLC-NF* for finite n is not very prominent, although *LLC-F* gets worse for higher intervention counts. Furthermore, a larger dataset size improves the metric. Finally, the plot in Figure 8 again shows higher medians and smaller interquartile range for *ASP*.

V. Conclusion

We have evaluated four causal discovery methods that allow for cycles and hidden confounders: two *LLC*-based variants

LLC-NF and *LLC-F*, and two simple ensembles based on *ASP-d* and *ASP-s*. The study focused on sparse linear SCMs with only five nodes and two confounders since the *LLC* variants presume linearity and the *ASP* variants do not scale well with the number of nodes. We considered the dependence of the model's performance on various interventional setups and the size of the dataset and measured them with the metrics accuracy and AUC ROC. All models show very good discovery capabilities when applied to datasets with a sufficient amount of interventions. There is not much difference between *ASP-d* and *ASP-s* even though each sampled SCM contains at least one cycle. *LLC-F* is better than *LLC-NF* for datasets with fewer interventions and often worse for datasets with more interventions. For datasets with an insufficient amount of interventions, mainly thanks to the weak classifier in the ensemble, *ASP* is better than *LLC*.

References

[1] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[2] Stephan Bongers et al. "Foundations of structural causal models with cycles and latent variables". In: *The Annals of Statistics* 49.5 (2021), pp. 2885–2915.

[3] Frederick Eberhardt, Patrik Hoyer, and Richard Scheines. "Combining experiments to discover linear cyclic models with latent variables". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 185–192.

[4] Patrick Forré and Joris M Mooij. "Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders". In: *arXiv preprint arXiv:1807.03024* (2018).

[5] Patrick Forré and Joris M Mooij. "Markov properties for graphical models with cycles and latent variables". In: *arXiv preprint arXiv:1710.08775* (2017).

[6] Martin Gebser et al. "Potassco: The Potsdam answer set solving collection". In: *Ai Communications* 24.2 (2011), pp. 107–124.

[7] Michael Gelfond and Vladimir Lifschitz. "The stable model semantics for logic programming." In: *ICLP/SLP*. Vol. 88. Cambridge, MA. 1988, pp. 1070–1080.

[8] Patrik Hoyer et al. "Nonlinear causal discovery with additive noise models". In: *Advances in neural information processing systems* 21 (2008).

[9] Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. "Causal discovery for linear cyclic models with latent variables". In: *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models (PGM 2010)*. Citeseer. 2010, pp. 153–160.

[10] Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. "Learning linear cyclic causal models with latent variables". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3387–3439.

[11] Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. "Constraint-based Causal Discovery: Conflict Resolution with Answer Set Programming." In: *UAI*. 2014, pp. 340–349.

[12] Gustavo Lacerda et al. "Discovering cyclic causal models by independent components analysis". In: *arXiv preprint arXiv:1206.3273* (2012).

[13] Sara Magliacane, Tom Claassen, and Joris M Mooij. "Ancestral causal inference". In: *Advances in Neural Information Processing Systems* 29 (2016).

[14] Joris M Mooij et al. "Distinguishing cause from effect using observational data: methods and benchmarks". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1103–1204.

[15] Judea Pearl. *Causality*. Cambridge university press, 2009.

[16] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[17] Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. "Discovering causal graphs with cycles and latent confounders: An exact branch-and-bound approach". In: *International Journal of Approximate Reasoning* 117 (2020), pp. 29–49.

[18] Kari Rantanen, Antti Hyttinen, and Matti Järvisalo. "Learning optimal cyclic causal graphs from interventional data". In: *International Conference on Probabilistic Graphical Models*. PMLR. 2020, pp. 365–376.

[19] Thomas S Richardson. "A discovery algorithm for directed cyclic graphs". In: *arXiv preprint arXiv:1302.3599* (2013).

[20] Dominik Rothenhäusler et al. "BACKSHIFT: Learning causal cyclic graphs from unknown shift interventions". In: *Advances in Neural Information Processing Systems* 28 (2015).

[21] Ruben Sanchez-Romero et al. "Causal discovery of feedback networks with functional magnetic resonance imaging". In: *bioRxiv* (2018), p. 245936.

[22] Shohei Shimizu et al. "A linear non-Gaussian acyclic model for causal discovery." In: *Journal of Machine Learning Research* 7.10 (2006).

[23] Eric V Strobl. "A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias". In: *International Journal of Data Science and Analytics* 8 (2019), pp. 33–56.